Category-level object detection and image classification

Chia, Alex Yong Sang

2010

Chia, A. Y. S. (2010). Category-level object detection and image classification. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/42225>

<https://doi.org/10.32657/10356/42225>

NANYANG
TECHNOLOGICAL
UNIVERSITY

**Category-Level
Object Detection and Image Classification**

Alex Yong-Sang Chia

SCHOOL OF COMPUTER ENGINEERING

A THESIS SUBMITTED TO THE NANYANG TECHNOLOGICAL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

2010

for my late dad

Abstract

Automatic recognition of object categories from complex real-world images is an exciting problem in computer vision. While humans can perform recognition tasks effortlessly and proficiently, replicating this recognition ability of humans in machines is still an incredibly difficult problem. On the other hand, successful automatic recognition technology will have significant and mostly positive impact in a plethora of important application domains like image retrieval, visual surveillance and automotive safety systems. This dissertation addresses two main goals of recognition: image classification and object detection. Image classification seeks to separate images which contain an object category from other images, where the focus is on identifying the presence or absence of an object category in an image. Object detection concerns the identification and localization of object instances of a category in an image, where the goal is to localize all instances of that category from the image.

Our main contributions toward this end are threefold. Firstly, we develop a novel and powerful method to detect ellipses from edge images. Our method specifically addresses the structural issues related to broken edge maps, background clutter and partial occlusion. Additionally, we incorporate a self-correcting mechanism into the ellipse detector which empowers it with an ability to identify weak ellipses and to regenerate new ellipses that better represent edge information. Experimental evaluation on complex synthetic and real images shows our ellipse detection method to have systematic and substantial improvements over previous methods. We are unaware of any other works that can detect ellipses from such difficult images. Consequently, the proposed method advances the state-of-the-art in ellipse detection.

ABSTRACT

Secondly, we demonstrate that very simple and generic shape primitives of line segments and ellipses can classify images and detect objects in complex real-world scenes. Here, we harness only image edge information. We pair primitives to form shape tokens, and learn discriminative combinations of shape tokens from noisy and unsegmented training data using just bounding boxes. A novelty of our approach is that we do not restrict a combination to have a fixed number of tokens, but allow each combination to flexibly adapt to an object category. This, coupled with the generic nature of the shape primitives, empowers a combination with much flexibility to represent an object category. Detailed evaluation over a diverse collection of object categories shows excellent recognition performances, where we achieve comparable or better recognition results than recent state-of-the-art contour based methods and other related methods.

Finally, we explore the combination of contour and region features in a unified framework for image classification and object detection. Each discriminative combination that is learned by the proposed hybrid method can contain either contour features, region features, or both contour and region features. Both the number and types of features in a discriminative combination are learned automatically from training data with only bounding boxes provided. Contour and region features are largely orthogonal and we demonstrate that our hybrid method which exploits the synergy between these complementary feature types attains improved recognition results, above what is achieved by either feature type individually. Additionally, comparisons against other recent hybrid methods also show very favorable recognition performances of our method.

Acknowledgements

First, I must extend my deepest and most sincere gratitude to my advisors, Professor Maylor Karhang Leung and Professor Susanto Rahardja, for their help and encouragements. I am also immensely grateful for the time and kind advice from Professor Deepu Rajan. Their pleasant way of guidance is very much appreciated, even though I have not explicitly mentioned this. In particular, the meticulousness of Maylor, the vision of Susanto and the enthusiasm of Deepu are a source of great inspiration and have left lasting impressions on me. Heartfelt thanks also go to Professor Wooi Boon Goh and Professor Vitali Zagorodnov who got me interested in research, and to my academic mentor Professor Ee Luang Ang who provided sound advice during my undergraduate studies and made my transition back to university easier. Shell Ying Huang is an inspirational professor, and I learned much from her during my undergraduate years. I would also like to thank the Agency for Science, Technology and Research (A*STAR) for their generous financial support, without which my studies and attendances at conferences would not have been possible. Additionally, I would like to thank the Ph.D. thesis committee members and examiners for finding the time to read and examine this thesis.

This thesis would not have been possible without the superb computing facilities in the Forensic and Security Laboratory. Particularly, I am thankful to the skilled administrator, Christina, who tirelessly maintains the laboratory's computers and infrastructure. Her helpfulness in allowing me access to the unutilized computers greatly sped up the completion of this thesis, for which I am especially grateful. Thanks also to Andy, Atiqur and Raj, who very kindly allowed me the use of their computers to run experiments after office hours, and during school and national holidays.

ACKNOWLEDGEMENTS

I also thank all my friends, near and far, for their great source of support and distraction. In particular, simulating discussions with Eng Yeow and Raj on research and other random topics had been an enjoyable source of entertainment. Delightful conversions and evenings out with April, Charlene, Jonathan, Ryan (brother-in-law), Wilson and Xue Dong had kept me sane through the years. They are a fun group of people to be around, and I look forward to spending more quality time with them after the completion of this thesis.

My family has been a pillar of immense strength to me. Several members deserve special mention. For her unending and tireless love and support for me, accepting my occasional research tantrums with unfathomable forgiveness and understanding, I am thankful to my Mum. My childhood was made all the more interesting with companion from my sister. Uncles Frank and Peter, auntie Irene, and cousins Jessie and Willie (who is like a brother I never have) have been an extraordinary source of comfort during difficult periods. Their unwavering support for me has made this journey possible. One cannot possibly ask for a better or a more loving family.

I am indebted to my wife Cheri, who was on by side throughout these four years. From learning to sleep with the lights on so that I could do late night works, to reorganizing her schedule to fit mine, to bearing with me when I brought along technical papers to read on our occasional weekend getaways, she had been extremely supportive during my Ph.D. adventure. Indeed, her unfailing love, second only to her amazing faith and patience of me which often go above and beyond the call of duty, had greatly strengthened my resolve to complete this research.

I reserve the deepest respect and thanks to my late Dad. He had been supporting my aspirations all the way through and had shown me in his subtle and gentle manners the qualities of a good husband, and an even better father. This thesis is dedicated to him.

Contents

Abstract	ii
Acknowledgements	iv
List of Figures	x
List of Tables	xiv
List of Algorithms	xv
1 Introduction	1
1.1 Objective	1
1.2 Motivations	3
1.3 Challenges	7
1.4 Contributions	13
1.5 Outline	14
2 Related Works	15
2.1 Image Features and their Descriptions	15
2.1.1 Region Based Features	16
2.1.2 Contour Based Features	24
2.2 Object Recognition	36
2.2.1 Region Based Object Recognition	36

CONTENTS

2.2.2	Contour Based Object Recognition	47
2.3	Learning Methods	60
3	Ellipse Detection	62
3.1	Introduction	62
3.2	Extraction of Elliptical-arcs from Connected Edge Pixels	68
3.2.1	Contour Fragments Extraction	68
3.2.2	Elliptical-arcs Candidates Identification	70
3.2.3	Fragments Merging	75
3.2.4	Elliptical-arcs Refinement	77
3.3	Ellipses Extraction by Grouping Elliptical-arcs	80
3.3.1	Pair-wise Cost Assignment	82
3.3.2	Optimal Pairing Identification	83
3.3.3	Weak Hypotheses Identification	84
3.3.4	Hypotheses Verification	88
3.4	Computational Complexity	91
3.5	Experimental Evaluation	93
3.5.1	Synthetic Images	94
3.5.2	Real Images	100
3.6	Summary	105
4	Contour Based Object Recognition	106
4.1	Introduction	106
4.1.1	Related Works	110
4.2	Shape Tokens	111

CONTENTS

4.2.1	Constructing Shape Tokens	111
4.2.2	Describing Appearances of Shape Tokens	114
4.2.3	Matching Tokens at a Single and across Multiple Scales . . .	115
4.3	Category-specific Codebook of Shape Tokens	117
4.3.1	Clustering by Appearance	119
4.3.2	Clustering by Relative Positions	122
4.3.3	Selecting Candidate Codewords into Codebook	126
4.4	Discriminative Codeword Combinations	131
4.4.1	Matching a Codeword Combination in an Image	132
4.4.2	Finding Combinations Matched in Training Images	134
4.4.3	Learning an Ensemble of Discriminative Combinations . . .	136
4.5	Object Detection and Image Classification	141
4.6	Experimental Evaluation	142
4.6.1	Evaluation Procedure	143
4.6.2	Datasets	145
4.6.3	Results	147
4.7	Summary	162
5	Fusion of Contour and Region Based Features	163
5.1	Introduction	164
5.1.1	Related Works	165
5.2	Category-specific Codebook of SIFT Descriptors	167
5.3	Discriminative Shape Token and SIFT Codeword Combinations . .	169
5.4	Object Detection and Image Classification	173

CONTENTS

5.5	Experimental Evaluation	173
5.5.1	Image Classification	174
5.5.2	Object Detection	176
5.5.3	Computational time performance	180
5.6	Summary	181
6	Conclusion	182
6.1	Summary of Main Contributions in this Thesis	182
6.2	Limitations and Future Works	183
6.3	Concluding Remark	187
	Bibliography	188

List of Figures

1.1	Illustrations of image classification and object detection	2
1.2	Recognition challenges from scale variations	7
1.3	Recognition challenges from illumination changes	8
1.4	Recognition challenges from viewpoint changes	9
1.5	Recognition challenges from occlusion and background clutter . . .	9
1.6	Recognition challenges from pose variations	10
1.7	Recognition challenges from intra-class variability	11
1.8	Recognition challenges from inter-class similarity	11
1.9	Extent of supervision from training data	12
2.1	Salient regions detected by different region detectors	18
2.2	Construction on spin image	21
2.3	Partition scheme for SIFT descriptor	23
2.4	Edge pixels detected by different edge detectors	27
2.5	Lines extraction by the dynamic two strip algorithm	34
2.6	Line segments extracted from example images	34
3.1	Flowchart to extract elliptical-arcs	69
3.2	Examples of extracted elliptical-arcs	69

LIST OF FIGURES

3.3	Identifying candidate elliptical-arcs on toy example fragments. . . .	72
3.4	Illustration of the Elliptical-arcs Candidates Identification process. .	74
3.5	Illustration of the Fragments Merging process	77
3.6	Illustration of the Fragments Refinement process	80
3.7	Flowchart to group disconnected elliptical-arcs	81
3.8	Parameters used for identifying weak hypotheses	85
3.9	Generation and verification of ellipse hypotheses	88
3.10	Overlap error between two ellipses	91
3.11	Ellipse Detection accuracy at different τ values	91
3.12	Detection results on images with overlapping ellipses	96
3.13	Example of detections on an image with overlapping ellipses	96
3.14	Detection results on images corrupted by noise	97
3.15	Example of detections on an image corrupted by noise	97
3.16	Detection results on images with occluded ellipses	99
3.17	Example of detections on an image with occluded ellipses	99
3.18	Detection results as error threshold is relaxed	100
3.19	Ellipse detection on a simple real-world image	101
3.20	Ellipse detection on complex real-world image	102
4.1	Outline of our contour based recognition method	109
4.2	Finding the neighbors of a reference line segment	113
4.3	Finding the neighbors of a reference ellipse	114
4.4	Geometrical attributes to describe tokens' appearances	115
4.5	Example tokens extracted from Weizmann horse training images . .	117

LIST OF FIGURES

4.6	Example tokens extracted from Graz-17 motorbike training images .	118
4.7	Clustering of tokens by their appearances	123
4.8	Candidate codewords of the Weizmann horse dataset	124
4.9	Candidate codewords of the Graz-17 motorbike dataset	125
4.10	Codewords selection for Weizmann horse dataset	127
4.11	Codewords selection for Graz-17 motorbike dataset	128
4.12	Outline of codeword selection method	128
4.13	Applying radial method on Weizmann horse dataset	129
4.14	Applying radial method on Graz-17 motorbike dataset	130
4.15	Shape token codewords for other Graz-17 object categories	131
4.16	Matching a codeword combination in images	133
4.17	Finding all matched codeword combinations.	133
4.18	Learning discriminative xCC for the Weizmann horse dataset	141
4.19	Learning discriminative xCC for the Graz-17 motorbike dataset . .	142
4.20	Performance evaluation curves	144
4.21	Performance on the Weizmann horse test set	149
4.22	Example detections on the Weizmann horse test set	151
4.23	Detection results as viewpoint is varied - Rigid object class	152
4.24	Detection results as viewpoint is varied - Articulated object class . .	153
4.25	Average object instance at selected viewpoints	155
4.26	Example of correct detections on similar shaped object classes . . .	157
4.27	Example of erroneous detections on similar shaped object classes . .	158
4.28	Example detections on all object classes of the Graz-17 test set . . .	161

LIST OF FIGURES

5.1	Outline of our hybrid recognition method	166
5.2	Category-specific codebooks of SIFT and shape token features . . .	169
5.3	Learning discriminative $xMCC$ for the Weizmann horse dataset . .	172
5.4	Learning discriminative $xMCC$ for the Graz-17 motorbike dataset .	173
5.5	Example detections by our hybrid based method	177
5.6	Example incorrect detections by our hybrid based method	179

List of Tables

3.1	Computational time comparison of ellipse detection	101
4.1	Detection and discrimination results on similar shaped categories .	156
4.2	Recognition results by our contour based method	158
4.3	Classification results on four Graz-17 categories	159
4.4	Runtime performance of our contour based method	161
5.1	Image classification results by our hybrid based method	175
5.2	Object detection results by our hybrid based method	178
5.3	Runtime performance of our hybrid based method	181

List of Algorithms

3.1	Identifying elliptical-arcs candidates	71
3.2	Merging contour fragments	76
3.3	Refining elliptical-arcs	79
3.4	Optimal pairing elliptical-arcs	83
3.5	Generating ellipse hypotheses	89
3.6	Verifying ellipse hypotheses	90
4.1	Adapting the bisecting k -medoid clustering method	121
4.2	Boosting an ensemble of discriminative codeword combinations . . .	139

Chapter 1

Introduction

This chapter introduces this research, beginning by detailing our objective in this work. Additionally, we discuss the motivations and important challenges related to this research. This should give insights to the importance and difficulty of this research. We highlight our main contributions, and close this chapter by giving a brief overview of the other chapters in this thesis.

1.1 Objective

This thesis presents new techniques to recognize object categories automatically from images. In particular, we address two goals of recognition, *image classification* and *object detection*. These goals are defined as follows,

Image classification seeks to separate images that contain at least one instance of an object category from other images. Here, the focus is to assign a label to an image to indicate if the image contains an object category.

Category-level object detection concerns the identification and localization of object instances belonging to a category across scale and space. Images may contain zero, one or multiple object instances, and the aim is to localize all instances of a category from an image.

Figure 1.1 illustrates these two recognition goals. It is clear that the additional task of localizing object instances increases the problem complexity of object detection. However, once a robust object detector is developed, classification of an image becomes almost trivial. Correspondingly, we first learn a detector for an object cat-

Figure 1.1: Illustrations of image classification (top) and object detection (bottom). Examples shown are actual test results by our vision system.

egory, and then adapt this detector to classify images. Note that while humans typically form categories based on functional information of objects (*i.e.* what an object does) [1,2], for this work, we concentrate only on visual appearances to categorize objects (*i.e.* objects are grouped into a same category if they have similar appearances). For example, instances of a side view and frontal view of a bike is separated into two categories, since they have pronounced appearance differences, even though they belong to the same “bike” family. We will use the term *category* synonymously with *class* throughout this thesis. Also, we will use the term *object recognition* to refer to the goals of image classification and category-level object

detection.

1.2 Motivations

Humans classify and recognize a vast number of object categories with remarkable ease and consistency [3, 4]. We can recognize familiar object categories (car, bike and horse) by vision alone, without focal attention [5] or relying on other perceptual cues such as sound, touch, and motion. This ability is evident even in early vision where neuroscience studies [6, 7] show infants as young as three months of age can organize objects into perceptual meaningful categories such as cat, dog and chair. This ability to group objects into categories, and to generalize to new unseen instances of an object category, is a fundamental cognitive process in humans. It reduces the amount of information to be processed, learned and remembered, and enables us to react equivalently to perceptually similar stimuli [8]. Consequently, it empowers humans to understand and interpret real-world scenes. While this recognition ability is honed to perfection across an adult life, recognizing objects across large datasets and the internet, or real-time monitoring of image/video streams demand colossal human effort and is impractical for any single human.

Replicating the recognition abilities (image classification and object detection) of humans in machines that surrounds us is an appealing alternative. It frees us from the mundane task of sieving through large number of images (in which recent study [9] estimates more than three billion photos in Flickr [10] and an even larger number in Google Image Search database [11]), and focuses our attention only on desired or meaningful images. Our motivation, other than pure scientific curiosity, is driven by the profound impacts a computer visual system has in important application domains as diverse as,

Image retrieval. Image retrieval refers to the searching and retrieving of relevant images from a large collection of digital images. Till date, personal computers or internet image search engines (*e.g.* Google [11], Picsearch [12], and Yahoo [13]) rely on image filenames, metadata, watermarks, or surrounding text near the image on a webpage to find relevant images of a queried image. While this is efficient in quickly gathering millions of images, the use of textual cues often returns images

unrelated to the queried image. Instead, a more natural way of retrieving images is to analyze the appearance of an image and then find other images with similar image contents. Such a method has been exploited by humans to find images, and is likely to improve retrieval results of image search engines. In the same way, the current generation of video search engines and video sharing websites such as Blinkx [14], Truveo [15], Videosurf [16] and YouTube [17] index videos by its textual information (*e.g.* movie title). Techniques developed to analyze the image contents can be extended to analyze video contents and to return a larger set of videos that are more relevant to a queried video.

Medicine. Recent developments in medical non-invasive visualization techniques have led to the creation of many medical images (*e.g.* radiographic and magnetic resonance imaging) on a daily basis. These images are used not only to provide views of anatomic structures for inspection, but are also necessary for pathology detection and tracking, and brain function analysis. While medical diagnostic is an intensive and highly skilled field, the development of a computer vision medical system can help doctors spot tumors and other abnormalities automatically in medical images. This has the potential to not only speed up medical diagnostic, but to also reduce human errors. Some success in this area are the works by Sluimer *et al.* [18] who developed a visual system to detect and classify abnormalities in the lung parenchyma, and Nayak *et al.* [19] who proposed a computer vision approach to recognize different stages of diabetic retinopathy for treatment.

Visual surveillance. Visual surveillance finds important uses in a wide spectrum of areas including nursing (*e.g.* video-based intelligent system for monitoring in a health care environment) and security. Here, it is required that a visual system be able to recognize specific individuals and to detect unusual events such as fighting or falling robustly. Additionally, it is also important that the system possesses an ability to recognize object categories. For example, by differentiating humans from other categories (*e.g.* animal), a surveillance system can more accurately detect meaningful occurrences of illegal entry into a secured location, which reduces false alarm rates.

Automotive. Smart vehicles which are able to automatically respond to dan-

gerous situations, *e.g.* by warning the driver or by applying gentle brake pressure to appropriate wheels, can improve safety standards and driving experience. This requires accurate and real-time recognition of objects such as pedestrians, lane markings, trees and other vehicles. Recent deployment of visual systems in vehicles includes the driver alertness monitoring system in Lexus LS 600h sedan (which uses a face detector to recognize if a driver's face is directed towards the road), the traffic sign recognition function in Saab 9-5 (which recognizes speed limits and no overtaking signs), the lane departure warning system in BMW 5 and 6 Series (which uses a camera to monitor the lines on the road), the blind spot indicator system in Volvo S80 (which mounts cameras on wing mirrors to detect objects in peripheral view of the driver) and the pedestrian detection system in the new Mercedes-Benz E300 and E350 series.

Robotics. Substantial current effort is being directed towards the development of a new generation of autonomous robots capable of complex tasks. In particular, recent advancement in control mechanisms has allowed a robot to demonstrate a diverse range of motions such as running [20], climbing [21], swimming [22], snake-like [23] and legless locomotion [24]. However, the usefulness of a robot is heavily dependent on its ability to understand and interpret real-world scenes. For example, to navigate its way through an environment, it must be able to recognize its location anywhere within a large map. This is an instance of scene recognition problem. Similarly, manipulation and grasping tasks require a robot to exploit visual cues for object recognition and pose estimation. In this aspect, both navigation and visual systems are fundamental for a robot to interact freely within a dynamic environment, and also form the basis to learn intelligent human-like behavior.

Human-computer interaction. Most computers receive inputs from classical devices such as mouse, keyboard and drawing tablet. While cost-effective and accurate, these devices do not support natural and expressive interaction with the computers. Research into new methods of information transfer between user and computer has led to interface devices like data glove and body suite which exploit natural human motion to interact with computers. However, such devices wrap body parts with a vast array of sensors and wirings which constrain movement

and reduce user experience. Instead visual information can be harnessed to track hand and head movements, and to recognize hand and body gestures. This allows effective communication with computers at a distance without the need for physical contact with interface devices. Additionally, an ability to recognize facial expression and human attention allows a computer to sense and respond appropriately to user feedback, and is fundamental for intelligent human-computer interaction.

Defense. Object recognition technologies enable a wide spectrum of military applications such as intelligence gathering, interpretation of reconnaissance imagery, target (*e.g.* boats, aircrafts and road vehicles) recognition, vision guided weapons, enemy detection and urban target tracking. Additionally, modern military concepts like “battlefield awareness” seek to relay a rich set of information including images (*e.g.* 3D maps) and real time videos about combat scenes to soldiers to enable better strategic decisions. In this case, automatic recognition and discrimination of important categories such as enemy troops and friendly units can reduce information overload and is fundamental for practical deployment.

Environment monitoring. Publicly available aerial and satellite images *e.g.* from Google maps [25] and Flash Earth [26] allow in-depth monitoring of environmental changes by environmentalist and scientists alike. In particular, by comparing the latest images with previous images, one can detect changes to land formation, water depth and sea bed, and even estimate the rate of these changes. For such purposes, manual comparison between images is laborious and prone to errors. Instead, recognition techniques can be harnessed to detect changes in landscapes across large temporal and spatial extents without human intervention. Additionally, one can also exploit image classification techniques to automatically and quickly map areas of severe flooding, forest fires and droughts [27,28] for time-critical humanitarian relief purposes.

Photo editing. An ability to recognize the category each object in a photo belongs to allows us to extract valuable category-level image data. This data can then be used to support exciting photo editing functions such as automatic cropping, color correction, blurring of background to concentrate attention on foreground objects, depth of field estimation for object manipulation or new object insertion, alpha matte extraction for object recoloring or new texture synthesis, and object

Figure 1.2: Scale variations. (a) Facial elements become less obvious with decreasing scales. (b) Individual strands of hairs become more pronounce at larger object scales.

replacement. For example, Brand and Pletscher [29] developed a system to remove acne and blemishes for automatic touch-up in face images. Additionally, category-level data can also be harnessed to improve user interface in existing photo editing softwares. For example, editing functions such as red-eye correction can be presented only when a user selects human or animal, but is not available when building or sky are selected (since red-eye correction is not appropriate for objects of these categories).

1.3 Challenges

Despite great research effort in the computer vision community, unconstrained classification of images and detection of many different kinds of object categories remain generally unsolved (though progress has been made on restricted versions of these two goals). This problem is difficult since a computer based recognition system has to distinguish a particular object category from all others categories, and to recognize previously unseen instances of the object category. Here, we discuss the primary challenges of recognizing object categories that are present in images. All images shown in this section are from publicly available datasets that we used for evaluation in this thesis.

Scaling. It is well known that objects in real-world scenes appear differently depending on the scale of observation. In particular, object details that are visible and topologically significant at larger object scales, may not be available at smaller scales. This can be observed in Figure 1.2(a) in which facial elements (*e.g.* eyes, nose and mouth) from frontal views of faces which are valuable for recognition and are visible at large object scales become greatly diminished at lower scales. In a similar fashion, when objects are seen too near, fine object details (*e.g.* individual

Figure 1.3: Illumination changes. Observe that object details are corrupted and removed to a certain extent. Specularities from reflectance surface also distort image content, *e.g.* hallucination of a building on the bonnet of car in the rightmost image.

strands of hairs of cows in Figure 1.2(b)) which are irrelevant for category-level detection can show up to confuse a recognition system. Consequently, for robust recognition, it is crucial that a recognition system be capable of extracting stable physical descriptions of objects across multiple scales. Beyond this problem which is common to both image classification and object detection tasks, a more specific problem related to object detection is the difficulty of accurate object localization at variable scales. Specifically, even when a camera remains fixed, objects of interest may appear at varying scales due to varying distances from the camera. Given that an object scale is unknown apriori in real-world images, the difficulty in scale estimation can prevent accurate delineation of an object (*e.g.* by bounding box or figure-ground segmentation) from an image.

Illumination. Illumination changes as a result of linear photometric transformation often arise in outdoor images (*e.g.* varying appearances of road scenes due to different extent of illumination from the sun at different times and weather conditions). While this can be compensated to a certain extent by contrast normalization or homomorphic filtering, illumination changes due to camera saturation, multiple lighting sources or illumination changes on 3D surfaces are more difficult to be rectified. Such changes not only alter the visual appearances of objects, but also introduce new distortions (*e.g.* cast shadows and specularities of reflectance surfaces) that further corrupt image contents. Additionally, for extreme lighting conditions (*e.g.* when the object is between the camera and a lighting source), object details necessary for recognition can be eliminated in an image. Examples of illumination changes are shown in Figure 1.3.

Viewpoint. Real-world objects are inherently three dimensional and project different appearances depending on its viewing angle. While a handful of object categories demonstrate greater degree of viewing invariance (*e.g.* bottles when viewed

Figure 1.4: Viewpoint changes. Note the change in visual appearances and forms of cars (top) and bicycles (bottom) as seen from the front, the side and the rear. Background clutter can also be observed in these images, which further complicates object localization.

Figure 1.5: Occlusion and background clutter. Note that occlusion where object are blocked by unrelated objects (first three images) or outside the field-of-view of the camera (last three images) remove considerable amount of object information. Background clutter present in these images further makes recognition difficult.

around the vertical axis), no object (other than a sphere) has a same appearance across unconstrained viewpoints. For example, as shown in Figure 1.4, under substantial viewpoint changes, new parts of a car appear (*e.g.* car rims) and previous parts disappear (*e.g.* car headlights). In addition, viewpoint variations also alter the aspect ratio of an object and increase the difficulty in localizing objects from images. Achieving good invariance to viewpoint is difficult for exactly known instance of a rigid object and this difficulty is greatly compounded when reliable generalization is needed across unseen instances of articulated object categories.

Occlusion and background clutter. Distortions from occlusion (when object are blocked by other unrelated objects, or are outside the field of view of a camera) lead to a degradation of visual appearance of object parts in an image. Correspondingly, as salient and meaningful parts of an object are occluded, it can lead to a total degradation in the performance of a recognition system. On the other hand, background clutter introduces noise and unrelated details which are irrelevant for the recognition of an object. While one can see these two challenges as complementary in nature, their effects are intrinsically related in that both reduce the signal-to-noise ratio of the object in an image. We point out that background

Figure 1.6: Pose variations. Unlike rigid objects (*e.g.* bicycle and car shown in Figure 1.4), articulated objects such as human in Figure 1.6(a) and horse in Figure 1.6(b) can assume many different poses. For example, Figure 1.6(a) shows instances of a person sitting, standing with arms at the side, and standing with arms partially raised respectively. These differences in object pose increase the difficulties in recognizing and detecting objects.

clutter poses greater difficulties for detecting objects composing of thin skeletal structures *e.g.* bicycles and ladders in which the background pixels are located a few pixels away from object pixels. Consequently, one needs to sieve out true object pixels from nearby background pixels to form accurate descriptors. This increases recognition difficulty. Figure 1.5 show examples of occlusion where strong background clutter is also evident in these images (as also in Figure 1.4).

Pose. While rigid objects *e.g.* motorbike and bicycle have a relatively fixed object pose, articulated objects *e.g.* human and horse do not have a fixed geometrical configurations or positions for its object parts. For instance, Figure 1.6(a) shows instances of a human sitting and standing upright (with arms lowered and raised). Similarly, images in Figure 1.6(b) show horses which are standing upright or have its head lowered. The complexity of recognizing such object categories is directly related to the number of joints involved, where a greater number results in a larger set of possible object poses. Even though there are a set of favorable object poses for which other occurrences are quite rare and unnatural, the possible variations are still very large. Correspondingly, the uncertainty of object poses in unseen test images makes it difficult for a visual system to robustly recognize objects. Recognition of readily deformable and viscoelastic objects *e.g.* tissue paper, dough and cloth poses an even more daunting challenge.

Intra-class variability and inter-class similarity. Teaching a computer to classify image or to detect objects in images is a particularly formidable challenge due in part to the variations of object instances belonging to a same category (intra-class variability). As an example, each of the first three images in Figure

Figure 1.7: Intra-class variability. Note the substantial differences in visual appearances of objects despite belonging to the same category. For example, the left-most cow appears with a pair of horns while other cow instances do not.

Figure 1.8: Inter-class similarity. Note the appearance similarities between instances belonging to different categories which reflect high inter-class similarities. For example, both horse and cow instances have similar legs, head and belly. Similarly, instances from bicycle and motorbike categories have similar side-wheels and handlebars.

1.7 contains the frontal view of a cow whose visual appearances and forms are different. For robust recognition, we therefore require an approach that is able to generalize across all possible (and previously unseen) instances of a category. At the same time, instances from different categories can have very similar visual appearance (inter-class similarity) and it is required that a recognition system be able to successfully discriminate between them (*e.g.* horse versus cow for articulated categories and bicycle versus motorbike for rigid categories as shown in Figure 1.8). These seemingly paradoxical goals of learning an object model that is highly specific to the category yet sufficient flexible to discard variations within a same category make recognition difficult.

Training data. There is a broad consensus of the fact that both the number of training images and the extent of supervision can impact recognition performance. In particular, for many categories (*e.g.* horse, human) which have high intra-class variability, a recognition system generally requires a larger number and a more diverse variety of instances of the category to learn a representative model. However, the number of images for a category is often limited by time and human

Figure 1.9: Extent of supervision from training data. (a) Bicycle images. (b) Random patches cropped from images with knowledge of image label, bounding box and segmentation mask. Observe that bicycle-specific information can be more easily collected when there is increase delineation of object or with reduced image complexity.

effort. Correspondingly, a recognition system should also cope when presented with a smaller training set. Similarly, the extent of supervision a system requires to learn a category is a challenge in this domain. Here, we use the term supervision to refer to the object information from training images that is presented to a system. For example, Figure 1.9(a) shows two training bicycle images. A set of random patches (from the two images) that are cropped out from within the whole image, bounding box and figure-ground segmentation are shown in Figure 1.9(b). It is obvious that with increasing delineation of the object, on the one hand, the collection of category-specific information becomes simpler, and on the other, the effort to pre-process the training data increases. This example also reveals the subtle issue that the real degree of supervision depends on the complexity of training images. For training images with object prominently displayed with little background clutter (*e.g.* bottom bicycle image in Figure 1.9(a) and first face image in Figure 1.2(a)), category-specific information can be easily obtained by relying solely on the knowledge of image labels. In contrast, it is much more difficult to collect useful information when given training images whose objects are embedded in substantial background clutter (*e.g.* top bicycle image in Figure 1.9(a) and third face image in Figure 1.2(a)), even though such images are more easily collected in real-world situations. Going towards recognition of tens of thousands of real-

world categories [3] therefore requires a system to learn object models with weak supervision from training images of varying complexity.

1.4 Contributions

We take a modern approach to image classification and category-level object detection. Specifically, we exploit popular and proven machine learning algorithms to extract category-specific information of objects from 2D images (using no 3D models) for recognizing previously unseen objects. Our main contributions in this dissertation are threefold,

- We develop a novel and powerful method to detect ellipses which actively counter the problems related to outliers (image clutter), broken contours, noise and partial occlusion. Our method addresses the structural issues when ellipses are broken over multiple disconnected elliptical-arcs, and is complemented by a self-correcting mechanism capable of identifying weak ellipses and regenerating new ellipses that better represent edge information. Detail evaluation on challenging images demonstrates our approach to systematically and substantially outperform existing methods.
- We present a contour based method for image classification and object detection in complex real-world scenes. Here, very simple and generic shape primitives of line segments and ellipses are harnessed as the underlying recognition features and is a critical departure from the more commonly used region based (image patches) features. Object models are learned automatically from noisy and unsegmented training data, with only bounding boxes provided. We evaluate this system on publicly available image datasets over 17 challenging object categories, for a total of more than 2400 test images, using standard experimental protocols. Excellent image classification and object detection results are obtained and demonstrate the effectiveness of our method. Additionally, comparisons with recent state-of-the-art contour based recognition systems and related methods yield very competitive recognition results of our method.

- We explore the fusion of contour features with region features in a unified framework to learn object models. These two features are largely orthogonal in nature in which contour features correspond to image edges and region features to image patches. We demonstrate that our hybrid method which exploits the synergy between these complementary feature types yields improved image classification and object detection results, above what is achieved by either feature type alone. We also compare against other recent hybrid based methods, where very favorable and strong recognition performance of our method is demonstrated.

1.5 Outline

This thesis is divided into six chapters, the first of which is this introduction. In Chapter 2, we discuss related works in the field of object recognition, focusing on different image feature detection and representation approaches, recognition methods, and learning techniques. The methods for detecting edges (by the Canny edge detector [30]) and extracting line segments (by the polygonal line fitting algorithm [31]) which we used extensively in this thesis are also reviewed in Chapter 2. Then, in Chapter 3, we present our novel method for detecting ellipses, with detailed evaluation of our ellipse detector against existing methods. Chapter 4 describes our contour based image classification and object detection method. This uses edge information as the sole recognition cue, and we explore the fusion of edge information with appearance information (image patches) in Chapter 5. Finally, we discuss limitations, suggest future extensions and give concluding remarks in Chapter 6.

Chapter 2

Related Works

In this chapter, we review the influential and exciting works that are related to this thesis. As far as possible, we present these works in a hierarchical order. Specifically, approaches to object recognition generally first extract salient (information rich) features from images and describe these features with some form of representations, before capitalizing on these representations to recognize object categories in an image. Correspondingly, we begin with a discussion of the related works on features extraction and representation methods. Following that, we review methods to recognize object categories. A brief discussion of generally used learning techniques for object recognition concludes this chapter.

2.1 Image Features and their Descriptions

Broadly speaking, image features can be divided into two main types: (i) region based features which make direct use of the pixel brightness or color levels, and (ii) contour based features which exploit the spatial configurations or statistics of edge pixels. In the following, we first discuss the methods for detecting and representing region based features, before discussing that for contour based features.

2.1.1 Region Based Features

2.1.1.1 Detecting Interest Regions

Region based features are generally detected by applying an interest region detector to an image, in which the detected regions represent salient and information rich portions of an image. Interest region detectors can be divided into scale invariant region detectors¹, and affine invariant region detectors (which can be seen as a generalization of scale invariant detectors to non-uniform scaling). We first review related works on scale invariant region detectors, before discussing that for affine invariant region detectors.

Scale invariant region detectors. One of the more popular detectors is the scale invariant Harris-Laplace detector [33, 34] developed by Mikolajczyk and Schmid. This detector combines the Harris corner detector [35] with scale space theory of Lindeberg [36] to identify scale stable Harris corners and to select characteristic scales at identified Harris corners. Specifically, they first exploit the Harris corners to identify keypoints at several scale levels of an image, and compute the Laplacian measure [36] at each Harris corner $\mathbf{x}$ as

$$|\text{LoG}(\mathbf{x}, \sigma_n)| = \sigma_n^2 |L_{xx}(\mathbf{x}, \sigma_n) + L_{yy}(\mathbf{x}, \sigma_n)|, \quad (2.1)$$

where $L_{xx}(\mathbf{x}, \sigma_n)$ and $L_{yy}(\mathbf{x}, \sigma_n)$ are the results of applying the second order partial derivative in the x and y directions, respectively, to the image that is smoothed by a Gaussian kernel with scale σ_n . Harris corners whose Laplacian measures attain local maxima in scale space, and are also above a threshold, are retained together with the scale level (characteristic scale) where its Laplacian measure is locally maximum. Each retained Harris corner gives the center coordinates for a circular salient region in the image, in which the radius of the circular region is given by its characteristic scale. These regions represent the scale invariant Harris-Laplace regions of the image. Kadir and Brady [37] proposed another scale invariant salient

¹Note that scale invariant region detectors should be more correctly termed as scale *covariant* region detectors since the detected regions respond covariantly with scales, as pointed out in [32]. However, in this thesis, we will term them as scale invariant region detectors to retain conventional naming terminology. Similarly, while affine invariant region detectors should be more correctly named as affine covariant region detectors, we will use their traditional naming convention of affine invariant region detectors.

region detector based on entropy measure. In their work, they defined a circular region (where the radius of the region corresponds to its scale) at each image pixel and computed an intensity histogram at each pixel using the pixels within the circular region. The local entropy of a histogram at every pixel is computed across a narrow range of scales, and the characteristic scale is selected at each pixel for which the entropy is local maximum. By searching across the spatial dimensions of an image, circular regions whose histograms entropy is sufficiently high are detected as salient regions.

Another popular detector is the Difference-of-Gaussian (DoG) detector, which has gained prominence in recent years due in part to its close relationship with the powerful SIFT local descriptor developed by Lowe [38, 39]. In these works, Lowe presented an efficient method to find DoG keypoints which are stable across scales, and to assign each keypoint with its characteristic scale. Here, he built a scale space image pyramid by first smoothing the original image with a Gaussian filter of a pre-defined scale. The filtered image undergoes a second smoothing operation with the same Gaussian filter. By subtracting these two smoothed image, he obtained the first level of the scale space representation. Next, the twice smoothed image is down-sampled with the scale corresponding to scale of the filter, and is used to build the subsequent DoG scale level. Each sample point in the current DoG image is compared to its neighbors in scale space and is chosen to be a keypoint (with its corresponding scale) if its DoG value is an extremum. This identifies DoG keypoints and its characteristic scales. Examples of DoG regions detected from a real image are shown in Figure 2.1(b). The DoG detector is a very close approximation to the Laplacian-of-Gaussian (LoG) detector [36], which is known to be robust to image transformations. Importantly, it can also be very efficiently computed, where a few images per second can be processed by Lowe’s method.

The above detectors which we have discussed belong to the class of detectors commonly referred to as scale invariant region detectors. Such detectors are able to robustly detect the same image patch (*i.e.* interest region) from an image under different image scaling. However, a key weakness of these detectors is their assumption that the scale changes uniformly in all directions. Specifically, in the case of affine transformations, scaling often differs in each direction. Such non-uniform scaling

Figure 2.1: Examples of salient regions detected using different region detectors. We show a gray scale image from the Weizmann horse dataset [40] in (a). Salient regions extracted by the DoG detector are shown in (b), Harris-affine detector in (c), Hessian-affine detector in (d) and MSER detector in (e). Each salient region is represented by a circle in (b) and an ellipse in (c), (d) and (e). For clarity, only the top 50 salient regions from each detector are shown; in practice, each detector typically detects salient regions in the order of hundreds. DoG, Harris-affine and Hessian-affine regions are detected by the implementation of Zhao, available at [41]. MSER regions are detected using the implementation at [42].

influences the locations, scales and shapes of identified salient regions. Hence, these detectors often fail to detect corresponding regions from images seen under affine transformations.

Affine invariant region detectors. Extensive works have been devoted to the design of affine invariant salient region detectors that are robust to affine transformations. Given that a smooth surface can be locally approximated to be piecewise planar, and that perspective transformations of planar surface due to viewpoint changes can be locally approximated by affine transformations, such detectors which respond covariantly to affine transformations are therefore robust to viewpoint changes [32]. Examples of such works included that by Baumberg [43] who extracted the Harris corners at several scales and used an iterative process to adapt the shape of the detected salient region to a local image patch. However, this method does not detect locations and scales of the regions in an affine invariant way, and hence is affine invariant only for fixed locations and scales. Mikolajczyk and Schmid [34, 44] extracted Harris corners, and exploited the second moment matrix and the automatic scale selection of Lindeberg [36] to construct the Harris-affine detector. In the same paper, they developed another affine region detector by replacing the Harris corner detector with the Hessian keypoint detector, in which Hessian keypoints are found as local maxima of the determinant of the Hessian matrix. They termed this affine region detector based on Hessian matrix as the Hessian-affine detector. Given that their method adapts the locations, scales and

shapes of a detected region to a local structure in a unified framework, they found their detected regions to be more stable than the regions detected by Baumberg in the presence of large affine transformations. Figures 2.1(c) and 2.1(d) show examples of salient regions detected by the Harris-affine and Hessian-affine detectors respectively.

Inspired by information theory, Kadir *et al.* [45] proposed an affine invariant region detector which improves on their earlier scale invariant region detector [37]. In their new work, they evaluated the entropy of the intensity probability density function at each pixel. The entropy is evaluated over three parameters (axes ratio, scale and orientation) of an ellipse that is centered on the pixel. An ellipse whose entropy is maximum over scale is recorded at each pixel, and the recorded ellipses define salient regions of the image. Matas *et al.* [46, 47] proposed the Maximally Stable Extremal Region (MSER) detector which is stable to a variety of imaging conditions. This detector first searches the image for extremal regions in which all pixels within a region have either higher or lower intensities than all pixels outside the region. Given that the detection scheme for MSER is dependant only on the order and not the value of pixel intensities, this detector is thus invariant to monotonic illumination variations. In addition, since affine transformations does not affect the order of pixel intensities, a set of pixels modeling a connected part of an object will be mapped to a same single connected part following affine transformations. Correspondingly, the MSER detector is robust against affine transformations. We show MSER regions that are detected from an example image in Figure 2.1(e). A weakness of the MSER detector is that it is highly sensitive to image blur, where mild corruption to pixel intensities readily undermines its stability criterion. Recent extensions by Perdoch [48] partially overcame this weakness. A detail comparison of the performance of commonly used interest region detectors on 2D planar (flat) scenes can be found in [32] and on 3D objects in [49].

2.1.1.2 Describing Interest Regions

Having detected interest regions from images, the next step is to describe these regions so that efficient comparison between regions is possible. Towards this end, a common approach is to represent each region by a numerical descriptor. Com-

parison between salient regions is then achieved by evaluating the distance (*e.g.* using the Euclidean or χ^2 distance measures) between its descriptors. A descriptor should be compact to support efficient comparison. More importantly, a descriptor should be distinctive and remain invariant under different image transformations like viewpoint variations and illumination changes. The distinctiveness and invariance properties of the descriptors substantially influence its robustness, and are reciprocally intertwined. In general, the more invariant a descriptor is, the less information it expresses. Correspondingly, the less information content that is conveyed by the descriptor makes the descriptor less distinctive. There are a plethora of descriptors which have been proposed to represent salient regions. Here, we limit this review to the more representative works.

Distribution based descriptors. The simplest description of a region is by a normalized intensity histogram of the pixel values within the region. Histogram intersection can then be used to evaluate the similarities of different regions. For example, Swain and Ballard [50] employed color histograms to represent objects for object identification. While this representation is very efficiently computed, however given that an intensity histogram ignores the spatial distribution of pixels, this representation is unlikely to be distinctive enough for reliable matching. This can be easily understood by considering two synthetically generated patches whose pixels values are sampled with a uniform distribution. While visual appearances of the patches will be vastly different, their intensity histograms are likely to be very similar (since pixels values are sampled uniformly). Instead, a better alternative to describe a region is by exploiting the distributions of both the spatial positions and intensities of the pixels within the region. An example of such methods is the work by Lazebnik *et al.* [51] who extended on the previous surface patch representation of Johnson and Hebert [52]. Specifically, Lazebnik *et al.* first normalized a detected salient region to a unit circle (see Figure 2.2). Following that, they defined a histogram on the circle, where the histogram is parameterized by the distance of each neighboring point (within the normalized region) to the circle center, and the intensity value of each neighboring point. They term this histogram as a spin-image. To ensure a spin-image is invariant to monotonic changes in illumination and is robust to noise, they normalized the intensity function range of the pixels

Figure 2.2: Construction of spin image. A spin image is constructed by first normalizing detected salient region (shown on the left) into a unit circle (center). The intensity of each pixel and its distance from the circle center are then used to compose a 2D histogram (termed spin image) as shown on the right.

within each region and slightly blurred each region using a Gaussian kernel before computing its representative spin-image. Cross correlation measure can then be used for comparing spin-images.

Frequency based descriptors. Other techniques describe a region based on the frequency contents within the region. For example, Fourier transform can be exploited to decompose image content within a region into a set of basis frequencies. A weakness of this method is that Fourier transform is localized only in frequency and not in space. Consequently, representation with the Fourier transform (*e.g.* by its Fourier coefficients) discards useful information on the spatial relationships between image pixels. Research findings from cognitive psychology studies [53, 54] convincingly demonstrated that Gabor filters are similar to the receptive field profile of primate cortical cells and therefore suggested it to be biologically relevant in human visual system. Consequently, Gabor filters have been widely used to represent detected salient regions, where using a Gaussian window, image contents within a detected region is decomposed into a set of basis frequencies by Gabor transform. This provides a description which is localized in space and frequency and hence improved on the Fourier Transform representation. A key weakness of a descriptor constructed by the Gabor transform is its dimensionality, in which many Gabor filters are required in order to describe small changes in frequency and orientation [55].

Derivative based descriptors. Local image derivatives within a region can be used to give an expressive and compact representation of a region. For example, Koenderink and Doorn [56] employed a truncated Taylor expansion to locally

approximate a differential function. Each term in the expansion corresponds to a partial derivative of the intensity values within the salient region, where the derivative of order n can be computed by convoluting the detected salient region with the n^{th} derivative of a Gaussian kernel. The concatenation of the Gaussian derivatives forms a compact descriptor and represents the geometric and photometric properties of the salient region. This descriptor is named local jet. Freeman and Adelson [57] provided a framework to synthesize new filters of arbitrary orientations as a linear combination of simple basis filters. Given the components of local jets, these filters steer the local derivatives in the orientation of the local image gradient. The steering of the derivatives in the direction of the gradient allows the derivatives to be rotation invariant. The filter responses that are computed (termed steerable filters) are subsequently used to describe a region.

SIFT based descriptors. Lowe [39] developed the very popular and powerful Scale Invariant Feature Transform (SIFT) descriptor which is invariant to translation, rotation, scale changes and is robust to illumination and viewpoint changes. While this descriptor also uses histograms to represent different characteristic of image regions in the same way as distribution based descriptors discussed earlier, the discriminative ability of SIFT descriptors have led to many other extensions and hence we consider SIFT descriptors and their extensions as a class on its own. We first review the SIFT descriptor, before highlighting some of its extensions. Given a detected interest region extracted by the DoG detector, Lowe defined a 4×4 grid of localized histograms of gradient orientations, where each histogram has 8 orientation bins (see Figure 2.3). Gradient orientation of each sample point within the grid is defined with respect to the dominant gradient orientation of the region, thereby achieving rotation invariance. The contribution of each sample point to location and orientation bins are weighted by the gradient magnitude and a Gaussian window overlaid on the region. To be robust against boundary effects in which a sampled point shift from being in one histogram to the other (or from one orientation bin to the other), trilinear interpolation is used to smoothly distribute the contribution of every sample point to its adjacent bins. A 128 dimensional vector is then constructed by the values of all histogram bins. Given that this descriptor is based on gradient orientation, it is therefore invariant to monotonic

Figure 2.3: Partition scheme for SIFT descriptor. (a) SIFT descriptor for a DoG region centered on a keypoint (red dot) is computed from a 4×4 grid of localized histograms of gradient orientations. Each histogram has eight histogram bins, in which a bin is visualized by an oriented arrow in this figure with the length of the arrow representing the sum of gradient magnitude near that direction in a grid location. (b) Each histogram is computed from a 4×4 image subregion, where arrows in this figure depict the gradient magnitude and orientation at an image sample point.

changes to illumination. More importantly, the loose grid that is defined on the detected region affords tolerance towards translation and scale offsets that may arise in the extraction of interest regions. As shown in a detailed evaluation by Mikolajczyk and Schmid [55], the SIFT descriptors outperform the steerable filters, the local jets and a host of other descriptors in terms of matching results under affine transformations.

Owing to the strengths of the SIFT descriptors, there have been a number of extensions to the basic SIFT implementation. For example, Bosch *et al.* [58] exploited the color information of an image and combined the color information in a SIFT framework to derive the color-SIFT descriptors. Another approach to enhance SIFT descriptors with color information is proposed by Goedeme *et al.* [59]. Ke and Sukthankar [60] defined a 39×39 grid on an interest region, and computed the gradient maps for both the x and y directions within the salient region. This yields a 3042 dimensions descriptor, which they reduced to 36 with PCA. They termed this descriptor as PCA-SIFT. While the low dimensions of PCA-SIFT features allow for more efficient matching than the original SIFT features, in an inde-

pendent experimental evaluation of the descriptors by Mikolajczyk *et al.* [55], they found the PCA-SIFT descriptors to have a weaker matching capability than the original SIFT descriptors (due possibly to the loss of discriminative information as a result of substantial dimension reduction where the initial dimension size of 3042 is reduced by more than 80 times to 36). In the same paper, Mikolajczyk *et al.* proposed another extension to the SIFT descriptors to increase its robustness and distinctiveness. Specifically, they computed the SIFT descriptors in the log-polar space. This log-polar space corresponds to three radial rings and eight angular directions resulting in seventeen sectors (the central bin is not divided in the angular directions). They quantized the gradient orientation into sixteen bins. This gives the descriptor an initial dimension of 17×16 . PCA was used to reduce the dimension of the descriptor to 128. They termed this descriptor as the Gradient Location Orientation Histogram (GLOH). In their experiments, they found the GLOH descriptors to be more distinctive than the SIFT descriptors. Unfortunately, the construction of the GLOH descriptors is computationally more expensive than that for the SIFT descriptors. Other extensions to SIFT descriptors include the works by Stein and Hebert [61] who incorporated local boundary information with SIFT descriptors to develop BSIFT which is robust against background changes, and by Morel and Yu [62] who very recently developed the ASIFT descriptor which is an affine invariant extension of the SIFT descriptors.

2.1.2 Contour Based Features

Almost all methods to extract contour based features start from an edge map. As such, we first discuss popular techniques to detect edge pixels, before reviewing contour based features that are employed to represent object shape structures.

2.1.2.1 Detecting Edge Map

Traditional edge detectors. In monochrome images, an edge is a pixel location where there is strong intensity contrast *i.e.* an abrupt discontinuity in pixel intensities between the pixel and its neighbors. The goal of edge detection is to find the locations of these intensity transitions. Edge detection is one of the earliest

studied field in image processing with many well established methods, such as the Robert cross gradient operator [63], Prewitt gradient operator [64], Sobel gradient operator [65] which exploit local extrema of the first order derivatives of pixel intensities, and the Difference-of-Gaussian and Laplacian-of-Gaussian operators [66] which harness the zero crossings of the second order derivatives of pixel intensities to locate edge pixels.

Among these traditional edge detectors, one of the most well known and frequently used edge detectors is the Canny detector [30]. Even though, this edge detector was developed in 1980s, nevertheless it is still quite competitive and is generally considered a state-of-the-art. In this thesis, we have also used this detector extensively to detect edge pixels. As such, we devote here a detail review of this edge detector. The Canny edge detector was designed specifically to optimize three criteria: (i) good detection *i.e.* a high probability of locating real edges and a low probability finding false edges, (ii) good localization *i.e.* minimal separation between the detected and real edges, and (iii) single response to an edge *i.e.* exactly one response for a single edge. The first criteria of good detection is directly related to the signal-to-noise ratio of the detector output, where a higher ratio implies that most real edges are detected and very few false edges are erroneously detected. Assuming the optimal detector has a finite impulse response bounded by $[-w, w]$ at an edge, Canny modeled the output signal-to-noise ratio as

$$\text{SNR} = \frac{\left| \int_{-w}^{+w} E(-x)f(x)dx \right|}{n_0 \sqrt{\int_{-w}^{+w} f^2(x)dx}}, \quad (2.2)$$

where $f(x)$ denote the response of the optimal detector, $E(x)$ is a form of the edge under consideration (*e.g.* step edge), and n_0 denote the root-mean-square value of the noise. For the localization criterion, Canny used the reciprocal of the root-mean-square distance of the detected edge from the center of the true edge as an

expression to be maximized,

$$\text{Localization} = \frac{\left| \int_{-w}^{+w} E'(-x) f'(x) dx \right|}{n_0 \sqrt{\int_{-w}^{+w} f'^2(x) dx}}, \quad (2.3)$$

where $f'(x)$ and $E'(-x)$ denote the first order derivatives of $f(x)$ and $E(-x)$ respectively. For the third criterion on only one response to a single edge, he derived an expression for the distance between adjacent maxima in the noise response (assuming additive Gaussian noise) of the detector as,

$$x_{\max} = 2\pi \times \left(\frac{\int_{-\infty}^{+\infty} f'^2(x) dx}{\int_{-\infty}^{+\infty} f''^2(x) dx} \right)^{\frac{1}{2}}, \quad (2.4)$$

which is constrained to be some positive fraction k of the detector width w , *i.e.*

$$x_{\max} = kw.$$

The aim now is to find an edge detector which maximizes the product of equations (2.2) and (2.3), subjected to the single response constraint. Canny found that for 1D step edges, the optimal detector can be closely approximated by the first derivative of a Gaussian function, *i.e.* $f(x) \approx -\frac{x}{\sigma^2} \exp(-\frac{x^2}{\sigma^2})$. For 2D edge detection, his edge detector employed two filters representing derivatives of the Gaussian function in the x and y directions. Importantly, Canny noted that all earlier proposed detectors encounter the problem of streaking (where edge contours are broken into disjointed contours due to the filter outputs fluctuating above and below a single threshold value), and hence improved his detector with adaptive thresholding using two hysteresis. Specifically, after filtering by Gaussian derivatives in the x and y directions, non-maximal suppression algorithm is applied to the filtered image to localize a set of ridges. Considering only pixels belonging to the set of ridges, all pixels whose response values are above the higher threshold value are labeled as edges, while all pixels whose response values are below the lower threshold are as-

Figure 2.4: Edge pixels detected by traditional, multi-scale and modern edge detectors. (a) Example image shown previously in Figure 2.1(a). (b) Edges detected by the traditional Canny operator [30] using the Matlab implementation. (c) Soft edge map (left) extracted by the multi-scale edge detector [67] using the implementation in [68]. Binary edges (right) obtained by thresholding using the Otsu’s method [69] followed by a thinning operation. (d) Soft edge map (left) obtained by the modern edge detector of [70], and binary edges (right) obtained by applying the Otsu’s method [69] and a thinning operation.

signed as non-edges. Unlabeled pixels (*i.e.* those with response values in-between the higher and lower thresholds) are labeled as edges if they are connected to a ridge containing edge pixels. The likelihood of streaking is reduced drastically since the ridge pixels must fluctuate above the upper threshold and below the lower threshold for streaking to occur. We show edges detected by the Canny detector for an example image in Figure 2.4(b). As observed, edge pixels corresponding to the horse are well detected, though many detected edges belong to the cluttered background.

Multi-scale edge detector. The above edge detectors find edge pixels at a single scale, in which a Gaussian kernel of a single fixed width is applied to pre-smooth an image before detecting edge pixels. An inherent issue that arises in these detectors is the amount of smoothing that is to be applied by a detector. In particular, a Gaussian smoothing kernel of a large width can be used as a noise suppression mechanism to avoid detecting noisy edge pixels. However, this eliminates sharp edges which are visible at fine scales. Importantly, in many real-world images, edge pixels are present at varying scale levels. Lindeberg [67] argued that to detect edge pixels in real images, an edge detector needs to apply different scale levels (*i.e.* smoothing with Gaussian kernels of different widths) in different parts of an image, where sharp edges are better detected at fine scales, while diffused edges at coarse scales. Continuing on his previous work scale space theory [36], he proposed a scale invariant edge detector [67] which automatically selects local appropriate scales for different local structures in an image. Specifically, he

developed a scale adapted differential operator to compute a normalized measure for the edge strength of pixels at different scales. A scale space edge is defined as one whose gradient magnitude is a local maximum in the gradient direction, and whose normalized measure is locally maximal over scales. Edge pixels are chained in scale space by assigning each edge with an orientation, and linking adjacent edges if they have similar orientations and are sufficiently close together. We show the soft edge map detected by [67] for an example image in the left of Figure 2.4(c), where a pixel is shown in darker gray if it has a higher edge strength. We threshold this soft edge map using the Otsu’s method [69] and apply a thinning operation to obtain the binary edge map shown in the right of Figure 2.4(c). As observed, both sharp edges (*e.g.* from the belly of the horse) and diffused edges (*e.g.* from the mane and tail of the horse) are well detected by their method. It has to be mentioned that this scale invariant edge detector requires the computation of high order derivatives (up to the seventh order), which contribute to the computational difficulties. Other techniques which also employ multi-scale analysis to detect edges include [71–73].

Modern edge detector. By defining edges as locations of abrupt intensity transitions, the above edge detectors increase the difficulty for an object recognition system to construct meaningful contour based features. In particular, due to textures, illumination phenomenon or background clutter, edges pixels which do not contribute meaningful shape information of objects are detected (often in large quantity), and a recognition system has to be substantially robust to sieve out relevant edge pixels to build useful representations. Recent advancements in edge detection techniques mitigate this issue substantially by taking a more modern approach to edge detection. In particular, rather than modeling edges as positions where pixel intensities change abruptly, they identify an edge as a pixel location where there is a change in pixel membership from one object (or surface) to another, and learn the detector directly from training data. Such detectors avoid picking edges which do not correspond to an object shape, and instead detect edges that correspond to the boundary of objects. The powerful Berkeley edge detector by Martin *et al.* [70] belongs to this family of modern detectors. In their work, they carefully choose nine different local appearance cues (texture gradient,

color gradient and brightness gradient at three scales each) to be extracted from a pixel of a training image. Each training image is paired with a hand-labeled segmentation mask which indicates the boundary pixels in the image, and they considered a pixel of a training image to be “on-boundary” if it is sufficiently close to a hand-labeled boundary pixel, and “off-boundary” otherwise. Supervised learning algorithms such as decision tree [74], or Support Vector Machines (SVM) [75] can then be employed to learn the optimal cue combinations for detecting boundary edge pixels. Example edge detection by the Berkeley edge detector is shown in Figure 2.4(d), where we show the detected soft edge map on the left, and the extracted binary edges on the right. As observed, this detector picks edge pixels that correspond to meaningful object boundaries. Another detector based on a similar framework of cue combinations was proposed by Dollar *et al.* [76]. In that work, they learned a detector from tens of thousands of region based features that are extracted from both large and small image patches. Correspondingly, with a much larger pool of features, they gained an advantage over the Berkeley edge detector in that their approach is sufficiently flexible and versatile to learn a more complex edge detector from the training data. Additionally, by using features that cover both large and small image patches, they can exploit low, mid and high level image cues simultaneously to learn a detector. Note that a drawback of using these modern detectors is that since they require a learning step, the quality of edge detection is therefore strongly dependent on the training data. In this aspect, good edge detection performance can be achieved when the detector is specifically trained on images which contain the object category whose edges are desired. This however demands an additional time consuming training procedure (*e.g.* in the order of days in [76]) for each object category, and also increases human effort, where figure-ground segmentations of training objects are required.

2.1.2.2 Extracting Features from Edge Map

The edge maps which are obtained from edge detectors discussed above provide the basis to extract contour based features that represent local object shape structures. Here, we discuss popular types of such features that are extracted from edge maps and are used for object recognition. We order this discussion by the granularity

of these features, where we first present features which are of coarser granularity (such as contour fragments) and then present features which are of slightly finer granularity (like line segments). There are also a number of recognition methods which build rich local descriptors based on the statistics (*e.g.* density) of edge pixels, and provide a tailored approach to use these local descriptors for recognition. For easier comprehension, we postpone discussion of such edge-based local descriptors until Section 2.2.2.3 where we first discuss the construction of these descriptors and then immediately review the tailored approaches which make use of these descriptors for recognition.

Contours and skeletons. The simplest type of features that can be extracted from an edge map is the contour itself. Measurements which account for the differences in shape structures between a pair of contours can then be used to evaluate the distance between the two contours. For example, the Hausdorff distance [77] computes the difference between two contours M and I as

$$d_{haus}(M, I) = \max(h(M, I), h(I, M)), \quad (2.5)$$

where

$$h(M, I) = \max_{m \in M} \left(\min_{i \in I} (dist(m, i)) \right), \quad (2.6)$$

m and i are edge pixels from contours M and I respectively, and $dist(\cdot)$ measures the Euclidean distance between the spatial positions of the two edge pixels. $h(M, I)$, which measures the asymmetric distance from contour M to contour I , is computed by taking every pixel in contour M , finding the distance between that pixel and the nearest pixel of contour I , and recording the maximum distance. $h(I, M)$ is computed in a similar way, and $d_{haus}(M, I)$ takes the maximum of $h(M, I)$ and $h(I, M)$. Note that the Hausdorff distance is sensitive to noise, in which a single noisy edge pixel in contour M or I can sharply affect the distance computation. To mitigate errors from noisy edge pixels, one can replace equation (2.6) with

$$h_f(M, I) = f^{th} \left(\min_{i \in I} (dist(m, i)) \right), \quad (2.7)$$

where $h_f(M, I)$ finds the distance between every edge pixel in M to its nearest

edge pixel in I , orders these distances, and takes the f^{th} quantile value of the set of ordered distances. For example, with $f = \frac{|M|}{2}$, where $|M|$ is the number of edge pixels in contour M , $h_f(M, I)$ finds the median distance from edge pixels in contour M to their nearest edge pixels in contour I . This modified distance measure is often known as the partial Hausdorff distance measure.

A distance measure which is closely related to the Hausdorff distance is the Chamfer distance [78],

$$d_{cham}(M, I) = c(M, I) + c(I, M), \quad (2.8)$$

where

$$c(M, I) = \frac{1}{|M|} \sum_{m \in M} \min_{i \in I} (dist(m, i)). \quad (2.9)$$

As observed, $c(M, I)$ gives the mean distance of edge pixels in contour M to their nearest edge pixels in contour I . Like the Hausdorff distance, the Chamfer distance measure is sensitive to noise, where a single outlier in contours M or I can affect the mean distance computation in equation (2.9). One can reduce the influence of outliers by either taking a quantile value of the distance (similar to equation (2.7) above) rather than using the mean value in equation (2.9), or by using a truncated distance measure

$$c_\tau(M, I) = \frac{1}{|M|} \sum_{m \in M} \min_{i \in I} (dist(m, i), \tau), \quad (2.10)$$

where threshold τ is application dependent. We mention here that a disadvantage of the Hausdorff and Chamfer distance measure is that they require comparisons between individual edge pixels from both contours and hence the speed of computation is strongly dependent on the size of the contours, where comparison between larger size contours is more expensive.

Skeleton is another type of feature that can be used to describe shape structures of an edge map. In particular, unlike contour fragment which is sensitive to articulations and non-rigid deformations [79], skeleton representation copes well with non-rigid deformation [80] since it models global shape structure of an object. Skeletons are extracted from a contour based on the media axis transform [81], which we summarized as follows. Given a contour M which encompasses a region

R , we collect a set of pixels from R in which these pixels have more than one closest neighbor in M . The collected set of pixels corresponds to the skeleton of the contour. Such a method typically collects skeletons that have many redundant branches, and thus a skeleton pruning method *e.g.* [82, 83] is often used to post-process the extracted skeletons.

Shape based primitives. Rather than using contours/skeletons of edge map directly for object recognition, one can instead represent contours by generic shape primitives. The geometrical and structural relationships between these primitives can then be used to describe local shape structures.

One popular type of shape primitive that is frequently used to represent contours is line segments. Lowe [84] was among the first to extract line segments from edge image for object recognition, where line segments are found with a split-and-merge approach. There, a line segment is fitted to the end points of a contour. A contour is recursively split into two smaller fragments, each approximated by a line segment, until all fragments are sufficiently small. A merging step is then applied to merge two fragments together if the significance of a line fitted to the merged fragment is greater than that of the two fragments, where the significance of a line is evaluated by the ratio of its length divided by the maximum normal distance from an edge pixel of the fragment to the line. Ren *et al.* [85] extracted line segments from contours by recursively partitioning contours at high curvature points until every contour fragment is approximately straight. Specifically, given a contour, they find three edge pixels on the contour a , b , c , where edge pixel b is between edge pixels a and c , such that the smaller internal angle $\angle abc$ is minimized. If the angle is below a pre-defined threshold, it signifies that edge pixel b is an acute turning point (*i.e.* high curvature point), and the contour is partitioned at edge pixel b into two smaller fragments. A similar approach to represent contours by line segments is proposed by Kokkinos [86].

Leung and Yang [31] proposed a line fitting algorithm based on polygonal approximation. Given that we have used this line fitting method extensively in our work, we therefore provide a detail review of this method here. In their work, they approximate a contour by a strip, where a strip is defined by a *critical line* and two *boundary lines* parallel and a distance w from the critical line (see Figure 2.5(a)),

where w is the half-width of the strip. The critical line is defined by two consecutive edge pixels of the contour, which we denote as points P_i and a in Figure 2.5(a). Two key observations made by Leung and Yang is that (i) many consecutive edge pixels of a contour can be included within the strip by rotating the strip using P_i as the pivot point, and (ii) the critical line of the strip can approximate the set of edge pixels that lies within the strip. These observations are illustrated in Figure 2.5(b). Based on these observations, they developed a two-step algorithm, which they named as Dynamic Two Strip Algorithm (D2SA), to extract line segments from an edge map. In the initial step, two strips of minimum width $w_{initial}$ are fitted to the left and right sides of the current edge pixel P_i . These strips are then rotated using P_i as the pivot point to include as many edge pixels as possible. The qualities of fit are then calculated for the left and right strips as $E = \frac{\text{Length of strip}}{\text{Width of strip}}$. Intuitively, the quality of fit E can be interpreted as the measure for the extent of elongation of the strip. Longer and narrower strips are preferred. By varying the value of w from $w_{initial}$ to w_{max} , the best fitted left and right strips can be derived. The merit of edge pixel P_i is then computed as

$$Merit(P_i) = E_i^{left} \times A_i^\theta \times E_i^{right}, \quad (2.11)$$

where E_i^{left} and E_i^{right} are respectively the elongation measure of the best fitted left and right strips, and A_i^θ is the measure of the acuteness of the angle at point P_i , $A_i^\theta = |180^\circ - \theta|$ with θ as the angle subtended by these two strips. In the second step, a subset of the most representational edge pixels is selected based on the local maxima of the merit measurements. Specifically, they first search for locally non-maxima edge pixels and eliminate these pixels temporarily. Following that, a check is performed to determine if the over elimination has occurred. Correspondingly, some edge pixels which have been previously eliminated are added back. Finally, strips of narrow widths are fitted to the filtered set of edge pixels to eliminate those consecutive edge pixels that align to straight lines. Critical lines of these narrow strips are then extracted as line segments from the edge map. We show line segments that are extracted by this method on two example images in Figure 2.6, where each line segment is shown as a separate color. Such representations of

Figure 2.5: Fitting of edges to a strip by the Dynamic Two Strip Algorithm of Leung and Yang [31]. (a) Original strip. (b) Rotated strip.

Figure 2.6: Line segments extracted by [31] on two example images. First column shows gray scale images. Second column shows edge maps that are extracted by the Canny edge detector [30]. Final column shows the line segments that are extracted from the edge maps. Each line segment is shown as a separate color. These representations of edge maps by line segments are often referred to as line edge maps (LEM).

edge maps by line segments are commonly referred to as line edge maps (LEM) .

A circle is another type of shape primitive that have been used to represent (curved) contours. Well known methods to extract circles from an edge map include the Hough Transform methods [87] and robust regression techniques [88]. Chi [89] proposed a novel method to detect circles from an edge map based on curvature of the contours. In that work, Chi first represents contours of an edge map by line segments. Partition points (such as corner and inflexion points) are identified based on the turning angles of the line segments and are used to break the edge map into several sets of connected line segments. For each set, a measure *Arcooid*

is computed as,

$$Arcoid = \frac{SPAN_{ArcLength}}{1 - \frac{SPAN_{ArcAngle}}{2\pi}}, \quad (2.12)$$

where

$$SPAN_{ArcAngle} = \sum_{i=1}^{m-1} \theta_i, \quad (2.13)$$

is the sum of the turning angles θ of the m line segments, and

$$SPAN_{ArcLength} = \sum_{i=1}^m l_i, \quad (2.14)$$

is the sum of the lengths of the line segments. Chi and Leung considered a set to represent a circle if its *Arcoid* value is greater than the diagonal length of the smallest bounding box enclosing the line segments.

Although a circle provides a meaningful shape representation for curved segments, however it offers only 3 degrees of freedom (spatial position and radius) and thus represents a limited class of object shapes. Additionally, shape structures remain circular only for specific viewpoint in which it easily deforms into an ellipse when its viewpoint is slightly changed. Consequently, under moderate viewpoint changes, a circle will no longer be sufficient to model a curved contour. An attractive alternative to a circle primitive is the ellipse primitive, where an ellipse has 5 degrees of freedom that are its spatial position, half lengths of its major and minor axes and its orientation. Unlike a circle, an ellipse remains an ellipse over a wide range of viewpoints (though with different parameter values). This, coupled with the observation that 2D images of man-made scenes often contain elliptical shape instances [90], make ellipse an attractive shape primitive to represent contours. Here, contours which can be modeled by ellipses are found from the edge map by applying an ellipse detector. Note that while the additional two degrees of freedom provided by an ellipse affords it greater flexibility to model curved segments as compared to a circle primitive, it also increases the difficulty in accurate detection of ellipses from an edge map, where due to fragmentation of edge contours and background clutter, most approaches often either detect false ellipses or miss true ellipses that are present in edge map. In this thesis, we have developed a robust ellipse detection method (Chapter 3) which compares favorably to previous ellipse

detection methods. We postpone discussing previous works on ellipse detection methods until the presentation of our method in Chapter 3.

2.2 Object Recognition

Modern object recognition methods extract features from an image (by the methods introduced in Section 2.1), and represent an object by a collection of features, or *parts*, where each part has a distinctive appearance. This strategy, which we also exploit in this thesis, offers several attractive advantages. Firstly, it can cope with clutter and partial occlusion since it explicitly accounts for the presence or absence of individual parts. Secondly, it can be made tolerant to affine transformations by adopting affine invariant features (*e.g.* [44]). Additionally, by modeling individual parts of an object as opposed to the entire object, it allows for a reduced training set even when modeling categories with large intra-class variability or with considerable articulation [91]. This improves training efficiency. For our interest, we divide the wide plethora of recognition approaches into two main categories: (i) those that are based on region based features, and (ii) those that are based on contour based features, and discuss them in Sections 2.2.1 and 2.2.2 respectively. In addition, methods of each category are further divided into sub-categories for a more meaningful discussion. Of course, there is generally no clear division that can be made, which make all discussed recognition methods relevant for the whole thesis. Note also that there are a limited number of recent methods which combined region features with contour features, and we include their discussions in Section 2.2.2.

2.2.1 Region Based Object Recognition

Majority of object recognition methods use region based features as the underlying features to represent an object category. Such region based recognition methods can be further divided into different sub-categories for a clearer discussion. For example, one can differentiate these methods by the type of interest region detectors and descriptors employed, as introduced in Section 2.1.1. However, interest

region detectors/descriptors can often be readily substituted for one other, and does not constitute a significant factor to differentiate between these recognition methods. For example, while SIFT descriptors was originally designed to describe interest regions detected by the DoG detector, they can just as easily be used to describe regions detected by the Harris-affine detector. Additionally, region based recognition methods often utilized a collection of several detectors/descriptors combinations, and thus no clear separation between these methods is possible based on the type of detectors/descriptors used. Instead, we believe a better classification of these recognition methods is by how strictly spatial relationships (*i.e.* geometry) within region based features, or between features and hidden variables (*e.g.* object centroid), are constrained. In this case, we further split region based recognition methods into three sub-categories of geometry-free, geometry-moderate and geometry-dense recognition methods, and discussed related methods in each sub-category next.

2.2.1.1 Geometry-free recognition methods

One of the well known region based approach is the work by Csurka *et al.* [92], who introduced the bags-of-features recognition method. In that work, they first collect a set of training images comprising category-specific images (*i.e.* images that contain instances of the target object category) and background images (*i.e.* images that do not contain any target category instances). Interest regions are extracted from these images using the Harris-affine region detector [34], and are described with the SIFT descriptors [39]. k -means clustering [93] is then applied to these SIFT features to obtain a generic bag (*i.e.* codebook) of SIFT codewords. SIFT features of a training image are quantized using this bag, and a histogram which describes the distribution of codewords in the image is used to represent the image. This histogram only considers the frequency for the occurrences of codewords in an image, and completely ignores their spatial configurations. Finally, histograms from all training images are collected together with their image labels to learn a SVM [75] classifier to distinguish between histograms of different object categories.

Nowak *et al.* [94] also adopted a bags-of-features method to object recognition. In that study, they found that a single interest region detector often extracts very

sparse regions from an image and cannot provide dense enough regions to characterize the image. Instead, they randomly extracted a dense collection of image regions using an image pyramid with regular grids in positions and densely sampled scales. They found that this approach offers a good tradeoff between collecting sufficient informative regions, while not picking up too many featureless (and hence uninformative) regions. Opelt *et al.* [95] improved on the bags-of-feature methods by using a diverse set of interest region detectors (such as scale invariant DoG and Harris detector) and descriptors (like SIFT and moment invariants [96]). Additionally, unlike Csurka *et al.* and Nowak *et al.* that are discussed above, they adopt boosting as a learning technique where each boosting round selects a descriptor type and a descriptor vector which best distinguishes object categories as a weak classifier. Correspondingly, by iterating through a pre-defined number of boosting rounds, very diverse descriptor types/vectors are learned. The effectiveness of their improvement is demonstrated in which complex images are successfully classified by their method. An in-depth study by Zhang *et al.* [97] revealed that the recognition performance of such bags-of-features methods is strongly dependent on the type of region based features used. Specifically, they found that using interest regions that possess a high level of invariance may not lead to the best possible recognition results (due possibly to the loss of discriminative information when regions are normalized to obtain invariance, and to the instability of detectors which possess high degree of invariance). Instead, they showed on complex image datasets that complementary region detectors (Harris-Laplace detector which responds to corner-like regions and LoG detector which responds to blob-like regions), and descriptors (SIFT and spin images) together with a SVM classifier can achieve good recognition performances.

The above bags-of-features methods require image labels (indicating whether a training image contains an object category) as input during training. Sivic *et al.* [98] proposed a bags-of-features method which does not require image labels during training, but only requires the number of object categories that are present in the training images. Specifically, following Csurka *et al.*, they represent an image by a histogram of codewords occurrences (though they used more diverse regions such as the MSER [46] and Harris-affine [34, 43] regions). The probabilistic

Latent Semantic Analysis (pLSA) [99] from text analysis is then applied to the histograms extracted from all training images to discover visually coherent topics (*i.e.* object categories). Correspondingly, by finding the topics embedded in each test image, they can discover the object category present in the image. While their work has the remarkable advantage of requiring only very weak supervision and can also give object relevant regions (thereby yielding object localization), a weakness of their approach is that the number of parameters in pLSA grows linearly with the number of training images. Consequently, the category models which are learned are therefore prone to overfitting [100]. Recently, Jegou *et al.* [101] investigated the memory requirements of the bags-of-features method and proposed to reduce its memory requirement by an efficient approximation to the histogram representation of images. Specifically, they projected an image histogram representation onto a pre-assigned set of sparse projection functions to decompose the histogram representation into several smaller sized vectors. Each vector encodes partial and independent information of the original histogram representation, and a fusion strategy is employed to merge information during run time. They apply their extension in the image retrieval domain where they retrieve images with similar scene categories, and demonstrate their compact representation to achieve excellent recognition accuracy.

Geometry-free methods which are discussed above recognize object categories in images by modeling only the appearances of the features constituting an object category, but ignore the geometry (spatial relationships) between features. While they achieved excellent image classification results, a common weakness of these methods is that unless there is a clean representation of training objects (*e.g.* by figure-ground segmentation), object localization (detection) by these methods is likely to be poor [102]. This is because codebook (and consequently the learned classifier) typically contains features arising from both the object category and from the background clutter. Consequently, features from test images can be matched to training features that arises from background clutter. Similarly, features from background clutter of a test image can also be matched to the codewords. Without information on the relative geometry and spatial layout of features, these methods have severely limited discriminative and descriptive ability, which render object

localization in cluttered images difficult² [102, 103]. Additionally, since no information regarding object location is encoded in these methods, it is unlikely that they are able to detect (and localize) multiple instances of an object category in an image.

2.2.1.2 Geometry-dense recognition methods

The geometry-free methods reviewed above discard all spatial information of features. On the other extreme, geometry-dense recognition methods, which we now discussed, learn a tight geometric configuration between features to produce a rich representation of an object category. Such methods are commonly referred to as constellation methods, and have been gradually developed through a number of influential publications [104–107]. Burl *et al.* [104] introduced the initial idea for the constellation method. In their work, they modeled explicit spatial relationships between all pairs of parts (*i.e.* derivative based features) in a probabilistic setting. Specifically, they first select parts that will be used in the constellation (*e.g.* eyes and nose for a face category), where parts’ positions are assumed to be distributed according to a multivariate Gaussian distribution whose mean and covariance are estimated from manually labeled parts. Given a test image, they calculated the likelihood ratio of detected parts based on the joint probability for the appearance of parts and its spatial configurations according to the distribution, and used the MAP approach³ [93] to decide if an image contains an object instance. Sets of parts with high likelihood ratio can also be used to localize object instances in the test image. Their method requires examination of all possible configurations of detected parts to localize test object instances. While Burl *et al.* proposed a method for searching among the more plausible constellations, their proposal is based on a heuristic whose validity is difficult to control. Additionally, their approach demands substantial supervision during training in which hand-labeled parts are required.

²Intuitively, the strong performance of these methods in image classification and its weak performance in object detection can be understood by considering that the detection of an “eye” in an image gives strong cues for the presence of a person in the image (image classification), but does not provide reliable information to localize the person (object detection).

³MAP approach refers to the maximum a posteriori probability approach and is a basic concept of the Bayes decision theory. Specifically, a classification decision is framed in a probabilistic framework as: Select H_i if $P(H_i|X) > P(H_k|X) \forall k$, where $P(H_i|X)$ denotes the posterior probability of a class being H_i given observed data X .

These limitations reduce its applicability.

Weber *et al.* [105,108] made a significant contribution to the original constellation method in which they proposed a weakly supervised technique to learn object models. In their work, a set of codewords are extracted from training images in a similar way to that extracted by geometry-free methods described earlier. However, unlike those methods, they used the Förstner detector [109] to find keypoints and defined a fixed size (11×11) patch around each keypoint as an interest region. Expectation-Maximization (EM) algorithm [93] is then exploited to learn subsets of codewords together with their geometric configurations that achieve best classification performance on a validation image set. This forms the statistical model of the object category. Image classification is achieved by a similar MAP approach. Note that their approach, even though require only weak supervision (where only image labels are required), is not scale invariant, since fixed size regions are used as parts. Fergus *et al.* [107,110] further extended the work of Weber *et al.* to be robust to scale variations, in which salient regions are detected from an image with the scale invariant interest region detector of Kadir and Brady [37], and represented by a vector of PCA coefficients. Additionally, they explicitly model appearance variability where both appearance and spatial relationships between parts are jointly learned in an unified framework. Here, a model consists of P parts and is specified by parameters θ_{fg} . Given a test image with N salient features of locations $\mathbf{X}$, scales $\mathbf{S}$ and appearances $\mathbf{A}$, recognition of an object is assumed if the ratio of the category posterior densities R is above a threshold,

$$\begin{aligned} R &= \frac{p(\text{Object}|\mathbf{X}, \mathbf{S}, \mathbf{A})}{p(\text{No object}|\mathbf{X}, \mathbf{S}, \mathbf{A})} \\ &\approx \frac{p(\mathbf{X}, \mathbf{S}, \mathbf{A}|\theta_{fg}) p(\text{Object})}{p(\mathbf{X}, \mathbf{S}, \mathbf{A}|\theta_{bg}) p(\text{No object})}, \end{aligned} \quad (2.15)$$

where θ_{bg} corresponds to the parameters of the background. A factorization of the

likelihood functions is obtained as

$$\begin{aligned}
 p(\mathbf{X}, \mathbf{S}, \mathbf{A} | \theta_{fg}) &= \sum_{\mathbf{h} \in H} p(\mathbf{X}, \mathbf{S}, \mathbf{A}, \mathbf{h} | \theta_{fg}) \\
 &= \sum_{\mathbf{h} \in H} p(\mathbf{A} | \mathbf{X}, \mathbf{S}, \mathbf{h}, \theta_{fg}) p(\mathbf{X} | \mathbf{S}, \mathbf{h}, \theta_{fg}) p(\mathbf{S} | \mathbf{h}, \theta_{fg}) \\
 &\quad \times p(\mathbf{h} | \theta_{fg})
 \end{aligned} \tag{2.16}$$

and

$$\begin{aligned}
 p(\mathbf{X}, \mathbf{S}, \mathbf{A} | \theta_{bg}) &= p(\mathbf{A} | \mathbf{X}, \mathbf{S}, \mathbf{h}_0, \theta_{bg}) p(\mathbf{X} | \mathbf{S}, \mathbf{h}_0, \theta_{bg}) p(\mathbf{S} | \mathbf{h}_0, \theta_{bg}) \\
 &\quad \times p(\mathbf{h}_0 | \theta_{bg}),
 \end{aligned} \tag{2.17}$$

where the vector $\mathbf{h}$ describes a model hypothesis, $\mathbf{h}_0$ the null hypothesis and H the set of all valid hypotheses. $p(\mathbf{A} | \mathbf{X}, \mathbf{S}, \mathbf{h}, \theta)$ models the appearance term of the object likelihood, $p(\mathbf{X} | \mathbf{S}, \mathbf{h}, \theta)$ the shape term of the object likelihood, $p(\mathbf{S} | \mathbf{h}, \theta)$ the relative scale term of the object likelihood and $p(\mathbf{h} | \theta)$ the occlusions of model parts and the statistics of the feature detector. Learning of parameter θ_{fg} for the object model is obtained by the EM algorithm, where the goal is to find parameters that best account for the data $\mathbf{X}$, $\mathbf{S}$, and $\mathbf{A}$ that are observed from all training images. While very good image classification results were reported by the authors on several object categories, a weakness of their approach is that a learned object model has many parameters (order of 100) to be estimated. A common knowledge in statistic is that learning values of a given number of parameters requires many fold larger number of training examples. Consequently, without a large number of training images, their method does not learn object models which generalize well to unseen test instances. Hence, in their work, object models are learned with some hundreds of training images for each category.

Fei-Fei *et al.* [106, 111] proposed a Bayesian extension to the constellation method of Fergus *et al.*, in which they investigated the possibility to learn object categories with only a few training images. Specifically, using a similar constellation method, they first learn object models for a number of object categories. Bayesian priors on parts appearance and their spatial relationships from these object cate-

gories are then incorporated into the learning process for a new and unrelated object category, in which only a small number of between one to five training images of the new category is used. Object detection and image classification are then achieved with the MAP approach. Holub *et al.* [112] also used the constellation method of Fergus *et al.* for image classification and object detection. Specifically, they first learn an object model using Fergus’ *et al.* constellation method. Following that, derivative of the log likelihood of the model’s parameters are extracted from the model (as Fisher scores) and are exploited to learn a SVM discriminative classifier for object recognition. Their experimental results demonstrate convincingly that their method of using a discriminative classifier achieved higher recognition rates as compared to that by Fergus *et al.*, which relies on the MAP approach. Recently, Kamiya *et al.* [113] extended Fergus’ *et al.* constellation method by modeling the distribution of parts’ appearance and position with a multimodal Gaussian distribution (as opposed to unimodal in Fergus *et al.*). This allows it to learn object model which exhibits greater variations in appearances. Additionally, they also adapt the approach of [114] to speed up the learning of the object model in their work. We note that the use of multimodal Gaussian distribution greatly increases the number of model parameters to be learned. As such, to ensure that an object model generalizes well to unseen test instances, their method therefore requires many fold increase in the number of training images as compared to Fergus *et al.* [107, 110].

The above constellation methods harness the joint spatial layout distribution of parts, together with its appearance, to describe an object category. Such methods afford rich and expressive descriptions of an object category and have led to some success in the tasks of image classification and object detection on several difficult object categories. Additionally, the methods of Weber *et al.*, Fergus *et al.*, Fei-Fei *et al.* and Kamiya *et al.* offer another appealing property in that they require only weak supervision⁴, where only image labels are used for training (and posi-

⁴Note that while these methods used image labels as the only supervision during training, the training images used show the category object instances prominently (covering a large part of the image) in the center of the image. In this aspect, the low complexity of training images inadvertently increases the real extent of supervision in their approaches (see also our early discussion in Section 1.3).

tion information on training objects are not utilized), and Holub *et al.* supports concatenating labeled images with unlabeled images for semi-supervised learning of object models. This low degree of supervision reduces training effort. However, these constellation methods share a common and serious Achilles' heel. Specifically, the joint spatial distribution between the parts of an object model results in complex probabilistic dependencies between the parts. As such, learning an object model requires an exhaustive enumeration of parts' configurations, which is exponential in the number of parts. This formidable computation complexity forces the model to be learned from very sparse representation of training images (typically 20-30 detected features per image) in which the number of parts in a model is substantially constrained (*e.g.* up to 7 parts per object model in Fergus *et al.*, and 4 parts per model in Fei-Fei *et al.*). Such restrictions imply that a significant amount of available object information is ignored during the learning phase, and hence these methods lack tolerance to slight mismatched, occlusions and missed feature detections [115, 116]. Additionally, given that the parts correspond to region based features which are limited in size, and in number (in object models), detected parts are unlikely to cover entire object instances. Hence, such methods cannot be expected to segment object instances from test images.

2.2.1.3 Geometry-moderate recognition methods

The spatial relationships between parts are clearly valuable and investigations by [117, 118] show good image classification and object detection performances are possible even with loose pair-wise relationships between parts. Correspondingly, recognition methods which adopt a middle ground between the two extremes of geometry-free and geometry-dense methods (discussed above) have been intensively studied. The aim in these methods is to reduce spatial dependencies between parts to form computationally simpler object model, while ensuring that the model is still sufficiently informative to classify complex test images and to localize test object instances in cluttered images.

Towards this end, Fergus *et al.* [119] learned object models in which spatial information of parts in the model are defined with respect to the centroid location in a scale and translation manner. Their method first obtains a set of training images

based on a keyword of the target object category (*e.g.* airplane) from the Google image search engine. Salient regions are detected from these images with region detectors such as the scale invariant Harris detector, and are described with SIFT descriptors. Location information of interest regions are then incorporated into the pLSA to find visually coherent components. Finally, a set of validation images is obtained from the top few images returned from Google image search and is used to find valid components of the object category. Note that their method uses pLSA to learn category models, and hence suffers the same weakness as [98] discussed earlier. Bar-Hillel *et al.* [116] extended their previous bags-of-features approach [120] for image classification and object detection. In that extension, location and scales of parts in an object model are defined with respect to two central (hidden) nodes that describe an object location and scale, as captured by a star-like Bayesian network. Model parameters of the object category are discriminatively optimized with an extended boosting process. Good object detection results are demonstrated, in which matched parts often correspond to relevant object regions in test images. Carneiro and Lowe [121] proposed a sparse flexible model in which spatial relationships between a part is only modeled to its k -nearest neighbors. In so doing, geometry of parts is tightly constrained locally and loosely constrained global. This allows it to represent flexible object categories. In their work, SIFT features are extracted from images. An unsupervised learning algorithm is then applied to find cluster of SIFT features that share similar appearances, and also estimate the parameters of the object model. Their work allows the learning of an object model that is represented by hundreds of parts, and hence provides robustness to missing feature detections.

Leibe *et al.* [122–124] greatly advanced the field of object detection, in which they show compelling detection performance on cluttered and complex images. In that work, a codebook of visual words is built from category-specific training images by agglomerative clustering on interest region descriptors. Each codeword is parameterized with a set of offsets relative to centroid positions of all training objects. Both appearance and location information of codewords are then exploited in a Generalized Hough Transform framework [125] to vote for object presences in test images. Their method requires strong supervision where object instances are pre-

segmented from training images to learn object models. Gall and Lempitsky [126] proposed a similar Hough Transform based method for object detection. However, unlike Leibe *et al.*, they require only bounding box annotations in training images. In their work, training SIFT features are assigned a class label (indicating if it is detected from within bounding box of training objects) and an offset vector (relative to an object centroid). Random forest [127] is then employed to learn a category-specific SIFT codebook by optimizing the purity of class labels and offset vectors at each non-leaf nodes. At run time, test SIFT features are passed down the random forest which cast probabilistic votes for object presence in test images. While their method reduces the extent of supervision that is required as compared to Leibe *et al.*, their training method is not scale invariant. Hence, training objects are required to be scale-normalized.

Karlinsky *et al.* [128] proposed an unsupervised approach to object detection and image classification by a star-like geometry model. There, a collection of generic SIFT codewords are first learned from training images using the method of [129]. The unsupervised consistency amplification algorithm is then applied to training images to learn object models from these codewords, where the algorithm alternates between a model learning phase and a features partitioning phase. The feature partitioning phase identifies features that are very likely to belong to the object category, and are used to improve the appearance and geometric aspect of the object model (that is learned in the model learning phase). Object detection is achieved by a voting scheme (similar to the method in [124]). A weakness of this method is that it is limited to single object instance detection and test object instances are required to be scale-normalized prior to detection. Quack *et al.* [130] proposed a method which finds spatial configurations of features that occur frequently on the object instances and rarely on the background for object detection. In their work, a visual codebook of SIFT descriptors are first obtained by agglomerative clustering. SIFT descriptors from each training image are quantized by these codewords, and a scale invariant tiled neighborhood is defined on each SIFT descriptor to describe the spatial configurations of other codewords in its vicinity. The frequent itemset mining algorithm [131] is then exploited to mine for recurring neighborhoods that are distinctive against background images. Correspondingly,

this finds discriminative spatial configurations of codewords that efficiently describe the object model. Compared to Karlinsky *et al.* that is discussed above, their work uses stronger supervision, where training images with bounding box annotation are required. Lee and Grauman [132] also used a similar method as Quack *et al.* [130] to describe the spatial configurations of features around each detected SIFT feature. However, unlike Quack *et al.*, bounding box annotations of training object instances are not required. Instead, they proposed an unsupervised and iterative method, which finds discriminative object features based on the distribution of local region features of training images. Object localization by their method is achieved by matching features of test images to the learned object features.

The above geometry-free, geometry-moderate and geometry-dense recognition methods relies on using region based features for object recognition. However, even though texture or pixel brightness/color values may be discriminative for some object categories (*e.g.* motorbike and spotted cat), other categories (*e.g.* horse and cup) have much weaker texture cues and are instead better represented by contour features. Additionally, for complex categories of objects that are characterized by holes and thin stick-like structures, like bicycle and ladder, background pixels are often located very near object pixels. Consequently, region based features that are extracted from an image can be easily corrupted by the background pixels, leading to difficulty in learning discriminative features of the object category. Instead, such object categories are better defined by their shape, and hence are better characterized by contour features.

2.2.2 Contour Based Object Recognition

The realization for the importance of shape in visual recognition can be traced at least as far back as 1897 to the seminal work by Mach [133] who wrote, “In examining two figures which are alike but *differently colored* we recognize their *sameness of form* at the first glance, in spite of the differences of color-sensation.” (emphasis added). Neuroscience research [134–137] brings further evidence to advocate the important role of shape information in human vision system, in which detail findings reveal that shape cues are used by humans to recognize object cate-

gories even in early vision. Motivated by these findings, there have been a number of computer vision systems that exploit shape information (in the form of contour based features) for recognition. We review these contour based recognition methods in this section. Additionally, to make this discussion more meaningful, we classify these methods into several sub-categories based on the granularity of the contour features used (according to our earlier discussion on contour based features in Section. 2.1.2.2). In this case, we split these recognition methods into three sub-categories of contour-fragments, shape-primitives and edge-statistics methods, and discuss related methods of each sub-category. Note that some recent methods complement contour based features with region based features to recognize objects, and we also include their discussion in the relevant sub-category.

2.2.2.1 Contour-fragments recognition methods

Kumar *et al.* [138] used the pictorial structure method [139] to detect objects, but extended it by employing both contour fragments (extracted from edge images of video sequences) and texture enclosed within the fragment as features for detecting articulated objects in still images. Another extension in their method is that rather than an exhaustive search across the image to detect object instance, loopy belief propagation [140] was used. This achieves sufficient computational savings. As a limitation, their method requires tracked video sequences or manually labeled parts for training. Sun and Super [141] adopted a probabilistic approach to model the appearance variations of contour fragments. Specifically, they harnessed the distribution of contour fragments of known object classes and learned a Bayesian classifier for recognizing contour shapes. In their method, contour fragments are transformed to an invariant reference frame, and compared by the Mahalanobis distance in the eigenspace [93]. Contour shape recognition is performed within a three-level statistical framework which consists of models for contour fragments, shape classes and the entire training dataset. Their approach was further generalized by Bai *et al.* [142] who added skeletons to the model, with the inspiration that skeletons can improve stability against non-rigid transformations. Both [141, 142] demand segmentation of training objects from the background (*e.g.* by a modern boundary detector [76]) and are mainly focused on classifying synthetic images

which contain test shapes that have been cleanly segmented from the background, rather than classifying cluttered real-world images.

Felzenszwalb and Schwartz [143] also used contour fragments for object detection. In their work, contour fragment are extracted from a single hand-drawn object model. A hierarchical representation is then used to describe the contour, in which they recursively partition a contour at its midpoint to describe the geometrical properties of a contour at multiple level of resolutions; coarser levels describe the global shape aspects while finer levels the local curvatures. For object detection, every image contour of a test image is first matched to each sub-contour of the model using dynamic programming to find the matching that has the least deformation between the image and model contours. A second dynamic programming step is then applied to find an optimal matching between the model contours and a subset of image contours. This matching scheme demands excessive computational cost, and hence their algorithm requires the powerful boundary detector [70] to remove cluttered contours from a test image prior to detection. Lu *et al.* [144] proposed another method which used a single training example for object detection. Given an anchoring edge point on a contour, they defined a contour descriptor which is parameterized by the distance between two points on the contour, and the angle subtended between the two points (with the anchoring point as pivot). Dissimilarity between a pair of contours is evaluated with the histogram intersection of their corresponding descriptors. A particle filtering framework is then employed to match test contours to model contours, thereby yielding object detection. Their method requires the modern boundary detector [70] to extract clean edge maps that have minimal cluttered edges prior to object detection. Schindler and Suterb [145] proposed another method for detecting objects which is based on elastic matching of contours. Like the above methods, an object model is obtained from one hand-drawn sketch. At runtime, they extract closed contour chains from an edge map of a test image. A probabilistic measure, which involves the finding for the correspondences between relevant points of the contours using dynamic programming, is developed to evaluate the distance between contours. Test image contours which have high similarity with that of the object model are found using a greedy approach to yield object detection. Note that since the above discussed

methods [143–145] derive object model from a single training example and do not learn class variability. Hence, these methods cannot be expected to perform well on categories with large intra-class variations (*e.g.* horse) or on instances with moderate viewpoint variations.

Recently, Heitz *et al.* [146] employed a diverse set of features such as contour fragments and SIFT features for object detection. In that work, they developed a probabilistic method to learn a deformable shape model and appearance model of an object category. Landmark points are automatically selected from the object model, and are used to locate objects in test images by a greedy approach. Their method achieves not only accurate object detection performance, but also allows descriptive characterization of an object’s pose. As one shortcoming, their method demands manual delineation of the outlines of training object instances to learn the object model. Bai *et al.* [80] proposed a method which harnesses both skeletons and contour fragments to learn object models of articulated object categories in a unified framework. Specifically, training objects are used to learn representative skeletons of a target object category, where each skeleton branch is associated with its corresponding contour fragments. Object detection is then achieved by combining local responses of the matches (evaluated with the oriented Chamfer distance [91]) between test edge map and the training contour fragments, where the matching scheme is guided by the structure of the skeleton. Trinh and Kimia [147] also used skeletons in their object model, where dynamic programming was used to search for the optimal shape hypothesis in test images. While good detection results on difficult test objects are shown in [80, 147], their methods require figure ground segmentation of training objects. Moreover, the method of [147] is not robust to noise, in which image contours are required to be detected by the modern boundary detector [70] prior to object detection. This further limits its applicability.

Shotton *et al.* [91] proposed a method to learn a category-specific codebook of contour fragments, and demonstrated how these codewords can be employed for object detection. Specifically, they defined random rectangles within bounding boxes of training objects, and considered contours within rectangles as training examples. The oriented Chamfer distance measure, which is an extension of the Chamfer distance where orientation information of contours is incorporated, is

developed to measure the dissimilarity of two contours, a template T and an edge map E , as follows,

$$d_{\text{orient}}^{\text{cham}}(T, E) = (1 - \lambda) \cdot c_\tau(T, E) + \lambda \cdot o(T, E), \quad (2.18)$$

where

$$o(T, E) = \frac{2}{\pi \times n(T)} \sum_{x_t \in T} |\phi(x_t) - \phi(f(x_t, E))| \quad (2.19)$$

and

$$f(x_t, E) = \arg \min_{x_e \in E} \text{dist}(x_t, x_e). \quad (2.20)$$

Here, λ is the orientation specificity term which interpolates between the asymmetric distance term $c_\tau(T, E)$ from equation (2.10), and the orientation term $o(T, E)$. The function $\phi(x)$ gives the orientation of the edge pixel x modulo π , $|\phi(x_1) - \phi(x_2)|$ gives the smallest circular difference between $\phi(x_1)$ and $\phi(x_2)$, and $n(T)$ denote the number of edge pixels in template T . Finally, the function $f(x_t, E)$ gives the edge pixel in E that is closest to edge pixel x_t in the Euclidean distance sense. A codebook of category-specific contour fragments is then learned by clustering for the appearance and relative positions of the fragments. Boosting is employed to find codewords which reliably predict the presence of objects in scale space. This work was further extended in [148] where Shotton *et al.* complemented contour fragments with dense texton features (that are obtained from the responses of training images to a filter bank) for object detection. Opelt *et al.* [149] also learned a category-specific codebook of contour fragments. However, unlike [91, 148], they learned discriminative contour fragments by starting at random points on image contours and growing the fragments along the contours. The quality of a fragment is evaluated on validation images, where they seek fragments which match on positive images and not in negative images, and also give reliable prediction to object centroid. High quality fragments are then selected into a codebook. They exhaustively pair a fixed number (two) of contour fragment code-

words together, and boosting is employed to find discriminative fragment pairings as weak detectors. Given a test image, these weak detectors vote probabilistically in a Hough voting space for the presence of object instances. Mean-shift algorithm [150] is applied on this voting space to find the final set of detections. In the same paper [149], Opelt *et al.* extended their contour based method and proposed a hybrid method which harnessed both region and contour based features together for object detection. Specifically, they reuse their method of contour fragment codebook learning to construct a codebook of SIFT descriptors. They pair every two contour fragment codewords as a single contour feature, and every two SIFT codewords as a single region feature. Boosting is then employed, in the same way as their contour based method, to find weak detectors from the pool of contour and region based features. While very compelling image classification and object detection results are achieved by these methods [91, 148, 149] on complex test images, they (as most other methods which directly used contour fragments) suffer several shortcomings. Firstly, matching between fragments requires comparison between individual edge pixels where the efficiency in comparison is directly dependent on the fragment size. Additionally, unlike shape primitives which can be efficiently represented (*e.g.* a line segment requires only four parameters to record its end points), each fragment codes for specific positions of edge pixels and hence have higher storage demands. Finally, since contour fragments are not scale invariant, for multi-scale matching, one is forced either to rescale fragments which introduces aliasing effect (*e.g.* when edge pixels are squeezed together or pulled apart), or to resize images before extracting fragments which is time consuming and reduces image resolution.

2.2.2.2 Shape-primitives recognition methods

The use of shape primitives for object recognition has been explored by a number of authors. Lowe [84] was among the first to use line segments to represent contours. In that work, he exploited perceptual meaningful cues such as the proximity, parallelism and co-linearity of line segments to find groupings of line segments that are likely to arise from the same object. The potential of such line segments groupings was demonstrated where razors in a jumbled collection were successfully localized.

Line segments were also used by David and DeMenthon [151] to localize planar and polyhedral objects in real images. In that work, approximate correspondences between model and image line segments are first found to induce hypotheses of object instances in a test image. These hypotheses are then ranked based on the similarity of the local neighborhoods of the corresponding model and image lines. Finally, a robust refinement and verification step is applied to retain valid hypotheses. A key assumption in their work is that at least one model line segment can be perfectly detected without fragmentation from the test object instance. This rigid assumption may not hold for complex objects seen in cluttered background, where line segments may not be fully detected at their intersections with other line segments.

Kokkinos *et al.* [86] also used line segments in their object detection system, but empowered their system with greater descriptive ability by also including LoG image patches. In their work, edge and ridges chains are detected in a scale invariant way using [67], which are then broken into straight line segments. A codebook of category-specific contour and region based features is found by clustering on the poses of these features. Features from a test image that are matched to code-words identify possible locations of parts. Finally, top-down image information is exploited to evaluate the likelihood of hypothesized part location. Their learning method requires training object instances to be of a fixed scale and location in images. This restriction increases training effort. Ren *et al.* [152] used line segments for detecting humans in a wide variety of complex poses. In that work, they used the modern boundary detector [70] to extract edge maps, from which line segments are extracted. Location and orientation of line segments from training objects are used to solve for correspondences between sets of line segments (across multiple training objects), and k -means is used to find a small number of model poses, which they termed as *line aspects*. At runtime, matching of line aspects to a test image is based on the pair-wise cues of relative locations, orientations, adjacency and parallelism of line segments. While good detection results are shown, their method requires segmented object instances for training.

Line segments were also harnessed by Gao and Leung [153] as contour based features to recognize faces. There, they modified the Hausdorff distance measure to

compute the distance between two line sets. This modified Hausdorff measure considers the parallel, perpendicular and orientation distances between line segments of each set to evaluate the dissimilarity between different line sets, and does not require searching for the correspondences between line segments of each set. Nearest neighbor matching scheme was used for recognition. Their method was shown to recognize specific instances of faces relatively well, but they did not evaluate their technique on the tasks of category-level object detection or image classification. As one weakness, their method assumes that faces in test images are shown prominently with minimal occlusion, and that background clutter are eliminated prior to recognition. In a related work by the same group, Li and Leung [154] used line segments to recognize specific instances of palmprint images. In their work, a coarse to fine matching scheme based on generalized Hough transform and the modified Hausdorff distance measure [153] were used. Good recognition performance was shown, though they suffer similar weakness as [153].

Ferrari *et al.* [155] proposed a method which harnesses line segments for object detection. Specifically, line segments of test images are first extracted from edge maps to build a segment network. Object detection is then achieved by finding paths through this network that resemble the model outlines. Their work does not include any object model learning procedure, but instead use one hand-drawn example as object model. Consequently, their work suffers similar weakness to other methods that use a single training template shape (*i.e.* non-robustness on object categories with wide intra-class variations or with moderate viewpoint variations). An extension to this work was provided in [156] to account for class variability by learning object model from multiple training examples. There, groupings of adjacently connected line segments are first extracted from the segment network [155] as local contour features, which they termed as k Adjacent Segments (k AS). Each k AS is represented with a low dimensional descriptor that characterizes the geometrical properties between its line segments. A generic codebook of k AS is learned from category-neutral outdoor images, and k AS detected from training images are quantized according to these codewords. A descriptor is used to represent the spatial configurations of codewords within a window (in a similar framework as the geometry-moderate region based recognition method of Quack *et al.* [130]), and

SVM is trained to distinguish between window descriptors of positive and negative examples. This work was further extended by Ferrari *et al.* [157] who learned a category-specific codebook for the object model by clustering k AS extracted from within bounding boxes of training objects. By explicitly modeling the intra-class deformations of the object category, a statistical model for the object category is learned. A Hough voting scheme that is closely coupled with a non-rigid point matching algorithm is used to localize object instances in test images. A weakness of this method is that they learn object models only from positive images (or from a single hand-drawn example), and not negative images, which possibly reduces its potential to discriminate between object and background instances. Note that the problem of cluttered edges due to background clutter and object texture are significantly mitigated in the methods [155–157] by their use of the powerful Berkeley boundary detector [70] which automatically finds meaningful object boundaries.

The above methods which exploit line segments as features have demonstrated good recognition performance. However, certain shape structures are defined by curved contours, and are therefore better represented by curved segments (as opposed to line segments⁵). Curved segments have been explored by several authors for object recognition. Fergus *et al.* [159] used the constellation method [110] for object detection, where they adopt curved contour segments and PCA coefficients that describe salient interest regions [37] as basic features to learn a model for the target object category. There, they focused on the learning of object model from contaminated data, and demonstrated the effectiveness of their approach in re-ranking images returned by Google search engine. Note that since they adopted the same constellation method as [110], they therefore suffer the same drawback of learning only sparse object models. This work was further extended in [160] where rather than representing the relative locations of model parts by a joint Gaussian density, they model part’s location relative to a landmark part. This leads to a simplified model which has much reduced learning complexity as compared to [159].

⁵While one can always use several short connected line segments to represent a single curved segment, as in the k AS of [156, 157], such representations run contrary to how humans perceive curved shapes. In particular, the Gestalt psychology [158] proposes that the operational principle of the brain is holistic with self-organizing tendencies. In this aspect, a human will mentally organize the series of connecting line segments as a *single unit* to visualize the curved segment.

Another difference from [159] is that instead of explicitly modeling for the shape of the curved segments as done in [159], here they used texture regions that are around detected curves. As one weakness, edge information is not fully exploited in both works [159, 160] as they used only clean planar curved fragments that are in between bitangent edge points.

Chi and Leung [161] used line segments and circle primitives as contour features. In that work, they exploited the Gestalt psychology [158] to identify local groupings of primitives that are likely to describe the same object part. Geometrical relationships between primitives are used to form the local descriptors of objects. Their method suffers from a number of drawbacks. Firstly, they do not learn representative features of training objects, but instead store all features (extracted from training images) into a potentially large hypercube to vote on the presence of objects in test images. Consequently, their approach demands figure-ground segmentation of training objects. More importantly, a circle primitive is not an affine-invariant shape in which it readily deforms to an ellipse under slight view-point changes. Consequently, under viewpoint changes, circles may either not be detected in test/training images or the descriptors characterizing object parts may be adversely corrupted. Hence, their work is unlikely to be robust to viewpoint variations. Note that their work considers only the categorization of similar shaped synthetic logo images, and does not address the more challenging task of object localization in complex real-world images.

2.2.2.3 Edge-statistics recognition methods

The use of rich local descriptors based on the statistics of edge pixels for object recognition has been popularized by several important works. One of the most influential works in this field is that by Belongie *et al.* [162] who developed the semi-local shape descriptor *shape context*. In that work, they recorded a histogram (*i.e.* shape context) at an edge point, in which each bin of the histogram counts the number of edge pixels in a neighborhood near the edge point. To ensure that a shape context is more sensitive to nearby edge pixels than far-away edge pixels, the histogram bins are partitioned uniformly in the log-polar space. These bins correspond to 12 angular sectors and 5 radial rings resulting in a 60-dimensional descrip-

tor. To perform matching between two shapes, they first computed shape contexts at sampled points along the contours of the shapes. Correspondences of shape contexts between the two shapes are found with bipartite graph matching [163]. Based on these correspondences, a shape distance measure which exploited the degree of shape deformation is used to evaluate the dissimilarity between the two shapes. For object categorization, prototype examples of an object category (akin to object models) are first found by k -medoid clustering on the shape distance, and k -Nearest Neighbor (k NN) classifier is used to identify the category of a test object. Although shape contexts have very strong descriptive potential, where Belongie *et al.* demonstrated the effectiveness of shape contexts in classifying silhouettes, they are extremely sensitive to background clutter. Hence, their method requires careful segmentation of the object from the image prior to extracting shape contexts.

Recently, Zhu *et al.* [164] extended the shape contexts for object detection in cluttered images, where they used a single hand-drawn example for object detection. In that work, they formulated object detection as the matching of the model contours to image contours of a test edge map, in a similar framework as [143]. However, rather than matching contours directly as in [143], they harness contextual information around control points in a shape context framework, with contours as the basic unit, and perform contour matching based on the shape contexts. Linear programming is employed to speed up the matching procedure. Their method is dependent on [165] to remove clutter edges and to retain long salient contours before object detection. Note that since they used a single hand-drawn example as model, their method is unlikely to perform well on categories with large intra-class variations. As another weakness, we highlight that their method requires strong supervision in that a manual step is required to break the model outline into separate contours to represent distinctive parts of the training object prior to object detection in test images. Jurie and Schmid [166] developed an edge based descriptor (which is similar to shape contexts) for object detection. They first searched for circles across all scales and positions in an edge map, and retained salient circles which are well supported by edge pixels falling in a thin annular neighborhood of the circle boundary. A histogram is then employed to describe the density/spatial organization of edge pixels in a neighborhood around the circle (by partitioning

the neighborhood with 4 radial rings and 32 angular sectors). Chi-square distance measure is used to find dissimilarity between histograms. For object detections, they extracted the edge descriptors from test images and matched them to descriptors extracted from segmented training examples. Each match votes on the scale and position of hypothesized object instances, and test instances are found by a generalized Hough transform framework. While good object detection results on cluttered images are demonstrated, their method suffers several shortcomings. Firstly, they require pre-segmented training object instances. Secondly, their work is directly dependent on the detection of salient circles, which represent a very limited class of object shapes. For categories which are not characterized by circles, their method is therefore unlikely to give rich description of the object model. Finally, their method suffers the same viewpoint variations weakness as [161] (where it is likely that they are more adversely affected since only circles are used as features, unlike [161] which also used line segments).

Carmichale and Hebert [167] also used the spatial distribution of edge pixels in an aperture around each edge point as contour features. A cascade of decision tree classifiers is trained with increasing aperture size to distinguish between object edges from clutter edges. While they successfully detect specific instances of wiry objects from clutter scenes, their method presents two key weaknesses. Firstly, their method requires many training images to learn the discriminative edge features, where edge pixels in training images are required to be labeled as “object” or “clutter”. Additionally, the method only distinguish between object and background edge pixels, but does not differentiate between edge pixels that belong to multiple object instances. Hence, their method cannot localize multiple instances of an object from an image. Berg and Malik [168] extended their previous work on geometric-blur [169] to develop the geometric blur descriptor. There, they extracted a number of channels of oriented edge response from an image to obtain a sparse signal, and identified points which have high edge energy as feature points. A small region centered at each feature point is blurred with a spatially-varying Gaussian kernel to obtain the geometric-blur [169]. The geometric-blur is subsampled across all channels at a small fixed number of distinct locations in a circular grid. Correspondingly, by concatenating the samples across all locations, they derived the

geometric blur descriptor. Approximate correspondences between descriptors are solved by linear programming where a cost function based on similarity between corresponding geometric descriptors and the geometric distortion between pairs of descriptors is used. Object recognition proceeds in a nearest neighbor matching framework, and good image classification and object detection performance under considerable object deformation, partial occlusion and cluttered background is demonstrated. Their work uses substantially fewer training examples per object category than [167], though cluttered edges arising from texture or background are mitigated by using their choice of the boundary detector [70] to extract edges.

Mikolajczyk *et al.* [170] adapted SIFT descriptors on edge pixels to develop a scale and orientation invariant local descriptor. Specifically, given an image, they extract edges at multiple scales and select a scale invariant neighborhood for an edge point using the Laplacian method of [36, 67]. A histogram, parameterized by neighboring edge pixels coordinates and their gradient orientations, is then used to describe the neighborhood of an edge point. In their work, model edge descriptors are extracted from a single segmented training example. These descriptors are then matched to those extracted from a test image at runtime, where a coarse-to-fine scheme to ensure geometrically consistent matching is incorporated to reject outlier matches and to yield object detection. SIFT region based features was also the key inspiration behind the edge based descriptor developed by Ma and Grimson [114]. Specifically, rather than computing SIFT features on interest regions, they center a neighborhood on an edge point, and compute a modified SIFT descriptor to characterize the neighborhood. These descriptors are then used to learn an object model in a similar way as [110]. Impressive classification results were achieved, in which they successfully distinguished between object categories with very similar shapes like (car versus minivan, and sedan versus taxi), though they require segmentation of object instances from the background prior to training or testing (by background modeling and subtraction of video frames).

In the above, we have separately highlighted the weakness of each individual edge-statistic recognition method. However, a weakness that is common among these methods is due to the form of descriptors used. Particularly, these descriptors consider each edge pixel individually and exploit the statistic (*e.g.* density) of

edge pixels in a small neighborhood for description. No attempt (other than [164]) has been made to analyze the connectivity between edge pixels or the local shape modeled by a set of connected edge pixels. In this regard, methods which harness these descriptors cannot be expected to provide meaningful structural cues (*e.g.* parallelism and adjacency) of matched local object parts, unlike the contour-fragments and shape-primitives recognition methods discussed earlier.

2.3 Learning Methods

In this section, we discuss a few popular and well grounded learning methods that have been used in the field of object recognition.

Decision Tree [74]. This simple but powerful method learns a decision boundary based on the distribution of the training examples. Specifically, given a collection of training features, the algorithm searches through all attributes of the features and for each attribute across all possible threshold values, to learn an attribute and a threshold value that optimize a pre-selected criterion on the current set of training features. Popular criteria include misclassification rates, Gini ratio, and information gain. The learned attribute and threshold value correspond to a node in the tree, and collectively define a hyperplane which split training features into two mutually exclusive subsets. Correspondingly, by recursively splitting the training feature vectors in each subset, a decision tree can then be learned⁶. Each leaf node at the base level of the decision tree is assigned a predicted label based on the majority class of the training features that transverse to the leaf node. A pruning step can also be incorporated to remove sections of the tree that provide little discriminative power to classify features. This allows the tree to generalize better to unseen test features and support better classification. Some popular examples of decision trees includes the CART [74], ID3 [173], C4.5 [174], SSV [175] and random [127] decision trees.

Support Vector Machine (SVM) [75]. The key idea in SVM is that features

⁶Here, we describe the learning of a binary univariate decision tree. This scheme can be easily extended, though with a higher computational cost, to learn an oblique or multivariate decision tree where training features are split based on two or more attributes. Details for learning such trees can be found in [171, 172]

that cannot be linearly separable in the feature space can be mapped onto higher dimensional space where they can always be separated by a hyperplane [93]. This mapping can be efficiently performed by exploiting the “kernel trick” [176] which reduces the mapping to a dot product. Training a SVM involve learning the optimal hyperplane such that the training features of different class labels lies on opposite sides of the hyperplane with the hyperplane having maximum margin/distance from the nearest training features. During testing, a test feature is mapped into the same high dimension space (at which the SVM was trained on), and is assigned a predicted class label based on which side of the hyperplane it falls on.

k Nearest Neighbors (k NN) [177]. In a strict sense, this method does not learn a decision boundary from the training examples (unlike the above discussed decision tree and SVM), but instead directly exploits training examples to define decision regions (voronoi tessellation). In particular, an appropriate distance measure (*e.g.* Euclidean distance measure) is first defined to compute the dissimilarity between examples. Given a test example, k NN assigns a class label to the test example based on majority class labels of the k nearest training examples. To avoid ties, one can choose k to be odd (in binary class problem), or use prior knowledge of the distribution of the training examples (in multi-class problem).

AdaBoost [178]. This is an iterative method which learns a strong classifier based on a combination of several weak learners, and is an extension of a previous work by Freund [179]. A weak learner is required to having an accuracy of slightly more than 50% (*i.e.* perform only better than random guessing), an example of which is one which compares an attribute of a feature to a scalar value. Each iteration (commonly referred to as boosting round) in AdaBoost adapts the weights of the training features so that weak learners will home in on features that are harder to classify at subsequent boosting rounds. Additionally, each boosting round also learns a weight for the weak learner based on its performance on the training features. By iterating through a fixed number of boosting rounds (or until a terminating condition is reached), a set of weak learners are learned. These weak learners are then combined into a strong classifier. Compelling success achieved by AdaBoost has inspired a host of other boosting variations such as the GentleBoost [180], NadaBoost [181], LPBoost [182] and SmoothBoost [183].

Chapter 3

Ellipse Detection

In this chapter, we present a novel method to detect ellipses from an edge image. Ellipses that are detected from an image will be exploited to construct contour based features for object recognition in the subsequent chapters. Our proposed detector proceeds with a two-stage hierarchical approach. In the initial stage, we exploit edge connectivity to construct a set of elliptical-arcs. Disconnected elliptical-arcs which describe a same ellipse are identified and grouped together by incrementally finding optimal pairings of elliptical-arcs in the second stage. A self-correcting mechanism is incorporated into the detector, empowering it with an ability to identify weak ellipses, and to regenerate new ellipses that better represent edge information. Detail evaluations on complex synthetic and real images yield excellent detection performances, where it is shown that the proposed ellipse detector systematically and substantially outperform existing methods. We are not aware of any other works that can detect ellipses from such difficult images. Therefore this chapter presents a significant contribution towards ellipse detection.

3.1 Introduction

Accurate detection of ellipses from an edge image is an important and classical problem in computer vision. In particular, many manmade scenes often exhibit instances of ellipses [90], in which circular and spherical objects (which are frequently seen) are projected as ellipses on an image plane. Consequently, one can exploit elliptical shapes that are observed from a single 2D image to evaluate the

3D pose of a camera or the 3D orientation of circular shaped objects [184–187]. Additionally, a number of object categories have curved contours (like molecular cells, face of a human and rear of a horse for articulated object categories, and traffic sign and mouth and handle of a cup for rigid object categories) whose shapes can be flexibly modeled by ellipses. Accurate modeling of such contours supports abstract and perceptually meaningful reasoning of object shape structures (such as parallelism and adjacency) and can be exploited as shape based features for object recognition [161, 188] or as domain specific shape cues for top-down/bottom-up segmentation [189, 190]. In this chapter, we consider *ellipse* synonymously with *elliptical shaped contour*, as those image edges which traces a partial or complete ellipse.

Several difficult challenges impede the development of a robust ellipse detector. In particular, detecting ellipses is difficult due to the presence of cluttered edges, in which outliers often substantially overwhelm edge pixels of true ellipses. Ellipse detection is further complicated by image noise and occlusion, in which an ellipse can be broken into multiple disconnected fragments, with some parts of the ellipse missing. In this aspect, a robust detector is required to recognize ellipses from a reduced set of data points while ignoring outliers. Due to the difficulty in collecting true edge pixels of unique ellipses in the presence of substantial outliers, accurate ellipse detection is still a significantly challenging task.

There are three general classes of ellipse detection approaches: Hough Transform based methods, optimizations algorithms and edge following techniques. Hough Transform (HT) [87, 191] is one of the best known ellipse detection approaches. In its original form, each edge pixel votes independently on the parameters of an ellipse in a five dimensional parametric space. Ellipses can then be extracted by identifying clusters in the parametric space. A key advantage of HT is that it does not require perfect connectivity of all edge pixels belonging to an ellipse, and thus, in principle, can detect fragmented ellipses. This advantage motivated a number of variants to HT. For example, to improve computation complexity of HT, Yuen *et al.* [192], Ho and Chen [193], and Chia *et al.* [194] exploited geometrical properties of an ellipse (*e.g.* symmetry) to decompose the five dimensional parametric space into lower dimensional subspaces. While these methods accelerate the ellipse de-

tection process, they will fail on images that do not have the required geometrical properties. Directional properties of edge pixels [195, 196] have also been used to decompose the parametric space. These methods require accurate calculation of gradients and tangents information and hence are affected by image noise.

Another commonly used approach to detect ellipses with HT is to subsample edge pixels of an image and to use only the sampled pixels to vote on the parameters of valid ellipses. For example, the Probabilistic Hough Transform [197] chooses a proportion of edge pixels from an image and applies the HT on these data points. Xu *et al.* [198] proposed the Randomized Hough Transform which randomly chooses n -tuples of data points to vote on the parameters of circles. This method was adapted by McLaughlin [199] who detected ellipses by randomly selecting three non collinear edge pixels to vote on the parameters of an ellipse. Further extensions to the Randomized Hough Transform include the work in [200, 201]. The accuracy of such algorithms is dependent on the sampled edge pixels. We point out that despite many variations of HT, these HT-based ellipse detectors share a common Achilles' heel: HT-based methods are sensitive to the choice of the quantization of the parametric space in which incorrect quantization leads to the detection of false ellipses (false votes accumulating in a cell due to coarse quantization) or the missing of true ellipses (spreading of votes around the true ellipse parameters due to fine quantization). Additionally, the accuracy of HT in detecting ellipses rapidly deteriorates as the number of ellipses in an image increases [202]. This can be attributed to the difficulty in identifying true peaks in the parametric space since edge pixels of other ellipses contribute towards the outliers of each ellipse. Cheng [203] developed a curve distinctiveness measure and used the measure to select valid ellipses from among a set of detections. However, given that their proposed measure is grounded on HT, they suffer similar weakness as other HT-based ellipse detection methods.

The second class of ellipse detection approaches formalizes ellipse detection as an optimization problem. Methods in this class include genetic algorithms (GA), least-squares and robust regression methods. GA-based approaches [204–206] encode a potential solution as a chromosome and evolve a population of chromosomes iteratively until a termination condition is satisfied. For example, Yao *et al.* [205]

recorded the coordinates of five edge pixels with a chromosome. Valid chromosomes are those whose five pixels correspond to a unique ellipse. In [204], Yin encoded the parameters of an ellipse into a chromosome. The fitness of each chromosome is evaluated to identify a set of candidate chromosomes. A local search scheme is then used to identify solutions from the set of candidates. A weakness of GA-based techniques is that they are sensitive to the choices of fitness evaluation criteria and the thresholds used in the fitness evaluation in which incorrect choices lead to false and missing detections [207]. These choices are often application dependent. Kasemir and Betzler [208] recently proposed a hybrid approach which combines GA with a modified HT for ellipse detection. However, the weakness of HT (as highlighted previously) cannot be avoided by their approach.

Least squares [209–211] and robust regression [88, 212] methods generally extract ellipses by fitting edge pixels to a general conic, where an objective function (*e.g.* distance between data points to the fitted conic) is optimized, and then rejecting non-elliptical fits. A major weakness of such an approach is that even if the edge pixels are sampled from an ellipse, the fitted equation may define a non-elliptical curve in the presence of outliers. For such cases, the parameters of the ellipse cannot be extracted by such methods. To alleviate this problem, Fitzgibbon *et al.* [213] developed a mathematical framework to fit an ellipse *directly* to a set of edge pixels. The main advantage of their approach is that it is “ellipse-specific”, providing a least squares elliptical fit to any set of edge pixels. Despite this, there is still an inherent weakness in [213]: like all other robust regression and least squares techniques, this method is unable to ignore outliers that may be present in a set of edge pixels. Instead, all edge pixels are used collectively to derive the best fit ellipse. In light of this, robust regression and least squares methods, while achieving optimal results when a set of edge pixels corresponds to a whole or partial ellipse, will yield false or missing detections when the set contains more than 50% outliers [211].

Detection of false ellipses is a frequently encountered problem among the above two classes of approaches. The key reason is that these approaches do not explicitly consider the connectivity of edge pixels that define an ellipse. Consequently, unrelated edge pixels which are randomly distributed in an image are identified by

the ellipse detector as defining the same ellipse. This results in false detections. To circumvent this problem, edge following techniques have been developed for ellipse detection. The Upwrite method [202] belongs to this class of ellipse detection technique. This method computes the local models of pixels within a small resolution radius in which those models that are positioned along the same geometric feature are grouped together. Each geometric feature is then classified into either ellipse, circle or line based upon its Zernike moments. Although this method has good computation efficiency, missing edge pixels or the presence of extraneous pixels corrupt the local models and hence it is unable to detect partially occluded ellipses or ellipses corrupted by image noise. Kim *et al.* [214] connected edge pixels by line fragments and constructed arc fragments from the line fragments through a series of pair-wise grouping. Ellipse detection is achieved by grouping the arc fragments together. A similar approach was developed by Mai *et al.* [215] for detecting ellipses. A weakness of these approaches is that the grouping of line and arc fragments relies solely on the local properties between the pair of primitives currently considered for grouping; more global shape information present collectively by the other primitives exerts no influence on the merging between the two primitives. This may result in incorrect grouping decisions. Given that incorrect grouping decisions made in the earlier phase cannot be corrected but are propagated to the ellipse detection phase, these algorithms therefore run the risk of false and missing ellipse detections. In addition, the method presented in [214] relies on accurate calculation of the tangents information of line and arc fragments and is therefore adversely influenced by image noise. This further limits its applicability.

Recently, Qiao and Ong [207] proposed a method which employs edge connectivity to detect ellipses. They first segment edge pixels into sets of connected edge pixels. Elliptical-arcs are extracted from each set of connected edges separately by removing outliers. An ellipse [213] is then fitted to the elliptical-arcs to detect ellipses. Their algorithm assumes the presence of at least one elliptic-arc whose edge pixels are connected and is sufficiently long in order to recover the original ellipse. This assumption may not hold in cluttered real-world scenes, in which due to the well known brittleness of edge detection, elliptical-arcs belonging to the same ellipse are often broken into multiple disconnected fragments. As another weak-

ness, their algorithm also suffers from the similar problem of error propagation as in [214, 215].

In this chapter, we present a novel and robust ellipse detector that does not suffer the shortcomings of the ellipse detectors among the three classes presented above. It is based on the idea of edge following, in which we model edge connectivity by line segments. These line segments are used to construct elliptical-arcs. However, unlike existing methods, we pool lower level information of these primitives to derive a better global shape understanding of elliptical contours present in an edge image. Armed with more complete shape information, the elliptical-arcs are refined to better describe the elliptical contours. The elliptical-arcs are separated into groups in which each group contains elliptical-arcs that are identified as belonging to the same ellipse. Hypothetical ellipses that may be present in the image are extracted by fitting an ellipse to the elliptical-arcs of each group. Finally, a feedback loop is developed to sieve out low confidence ellipses and to regenerate another set of better hypothetical ellipses by combining elliptical-arc from different groups. In this aspect, the proposed algorithm performs self-correction and focuses in on the detection of “difficult” ellipses. A comprehensive evaluation of the proposed ellipse detector is performed on a set of synthetic images under background clutter, salt-and-pepper noise and partial occlusion. It is shown that the proposed ellipse detector has superior performance in terms of recall, precision and F -measure as compared to existing methods. In addition, we apply the proposed detector on a set of challenging real images from the Caltech-256 database [216]. Detection results demonstrate that the proposed method can better detect visually salient ellipses than existing approaches.

This chapter is structured as follows. In the next section, we describe our method to extract elliptical-arcs from an edge image. Our method to group disconnected elliptical-arcs of unique ellipses together is presented in Section 3.3. Computational complexity analysis of the proposed algorithm is presented in Section 3.4. In Section 3.5, we evaluate the performance of the proposed ellipse detector and compare against existing methods. Section 3.6 summarizes this chapter.

3.2 Extraction of Elliptical-arcs from Connected Edge Pixels

An edge pixel when considered independently and as unrelated to other edge pixels does not contribute significant information for ellipse detection. Instead, a valid ellipse is characterized in most applications by one or more sets of connected edge pixels. Given this, by collectively analyzing sets of connected edge pixels, we can identify elliptical-arcs that provide valuable clues to the detection and localization of valid ellipses. In our work, we model the connectivity of edge pixels by line segments. These line segments are then exploited to search for meaningful elliptical-arcs. Figure 3.1 shows the block diagram to extract elliptical-arcs, which consists of four separate processes with references to sections in this chapter. We first apply a polygonal line fitting step [31] to an edge image to obtain its line edge map (LEM). The *Contour Fragments Extraction* process analyzes the geometrical properties of line segments within a LEM and breaks the LEM into several sets of connected line segments such that each set represents an arc that is smooth and convex. We term the edge pixels that have been fitted to the i^{th} set of connected line segments as contour fragment $f_i^{(1)}$, where the superscript denotes the output of the first process. Each fragment $f_i^{(1)}$ is analyzed by the *Elliptical-arcs Candidates Identification* process to determine how well its shape corresponds to an ellipse. Finally, the *Fragments Merging* and *Elliptical-arcs Refinement* processes are employed to extract more global shape information of the fragments and to identify the elliptical-arcs of an edge image. We use the simple real image in Figure 3.2(a), whose edge image and LEM are given respectively in Figures 3.2(b) and 3.2(c), as a running example to illustrate intermediate steps of extracting elliptical-arcs.

3.2.1 Contour Fragments Extraction

The *Contour Fragments Extraction* process breaks the LEM into a set of contour fragments such that each fragment corresponds to the arc of at most one ellipse. In this case, since an elliptical model is composed by a smooth and convex curve, a fragment therefore cannot contain any corner or inflexion points. Given this, we

CHAPTER 3. ELLIPSE DETECTION

Figure 3.1: Flowchart for extracting elliptical-arcs, with references to sections in this chapter.

Figure 3.2: Examples of extracted elliptical-arcs. Top row: Results from intermediate steps of extracting elliptical-arcs. (a) Test image. (b) Edge image. (c) Line edge map (LEM) representation, where line segments are shown by separate colors. (d) Contour fragments of the edge image. Each fragment $f_i^{(1)}$ is shown by a different color. (e) Candidate elliptical-arcs $f_i^{(2)}$ that are identified are shown by the thick contours. (f) Fragments that are identified by the fragments merging step as $f_i^{(3)}$ are represented by thick contours. Arrows show portions which have been merged. (g) Elliptical-arcs $f_i^{(4)}$, shown by thick contours, which are retained after the refinement step. Refined portions are highlighted with arrows. Bottom Row: Synthetic images (left) and the elliptical-arcs $f_i^{(4)}$ (right) extracted after the refinement process. (h) Overlapping ellipses. (i) Corruption by 4% salt-and-pepper noise. (j) Partially occluded ellipses.

analyze the geometrical relationships of line segments within the LEM to identify corner and inflexion points and break the LEM at these points. This results in a set of contour fragments $\{f_i^{(1)}\}$ in which each fragment $f_i^{(1)}$ traces a smooth

curve that turns either in the clockwise or anti-clockwise direction. In this work, we adopt the method of Chi and Leung [161] to detect corner and inflexion points from a LEM. Figure 3.2(d) shows the initial set of contour fragments $\{f_i^{(1)}\}$ that are extracted from the edge image of Figure 3.2(b) based on the line segments in the LEM of Figure 3.2(c). Here, each fragment is denoted by a separate color, and as observed represents a convex shape with no extreme turning point.

3.2.2 Elliptical-arcs Candidates Identification

The *Elliptical-arcs Candidates Identification* process analyzes each fragment $f_i^{(1)}$ to identify candidates of elliptical-arcs. For this purpose, we fit a least squares ellipse to each fragment $f_i^{(1)}$ and compute the alignment of $f_i^{(1)}$ to its fitted ellipse. Owing to the ellipse-specific advantage of [213], we adopt this ellipse fitting method and represent the ellipse fitted to $f_i^{(1)}$ as $E_i^{(1)}$. Due to image discretization and digital rounding off errors, it is uncommon for every edge pixel of $f_i^{(1)}$ to coincide with a boundary pixel of its fitted ellipse, even if $f_i^{(1)}$ corresponds to an ideal ellipse. Given this, we distinguish the pixels of $f_i^{(1)}$ that are near $E_i^{(1)}$ from those that are away from $E_i^{(1)}$ and compute the alignment of $f_i^{(1)}$ to $E_i^{(1)}$ as

$$\text{Align}(E_i^{(1)}, f_i^{(1)}) = \frac{\sum_{\forall x \in f_i^{(1)}} s(x, E_i^{(1)})}{N_i^{(1)}} \quad (3.1)$$

where $N_i^{(1)}$ is the number of edge pixels in contour fragment $f_i^{(1)}$, and

$$s(x, E_i^{(1)}) = \begin{cases} 1 & \text{if } \min_{y \in \mathbb{B}(E_i^{(1)})} \|x - y\|_2 \leq 1, \\ 0.5 & \text{if } \min_{y \in \mathbb{B}(E_i^{(1)})} \|x - y\|_2 = 2, \\ 0 & \text{otherwise,} \end{cases} \quad (3.2)$$

with $\mathbb{B}(E_i^{(1)})$ as a set containing the boundary pixels of $E_i^{(1)}$ and $\|\cdot\|_2$ as the L_2 norm. $s(\cdot)$ assigns a score to an edge pixel of $f_i^{(1)}$ based on its closest pixel in $\mathbb{B}(E_i^{(1)})$. While one can directly use the ellipse function for estimating the shortest distance between an edge pixel to an ellipse boundary, this requires solving for the

Algorithm 3.1: Elliptical-arcs candidates identification procedure as described in Section 3.2.2.

Input : A set of fragments $\{f_i^{(1)}\}$.

Output : A set of candidate elliptical-arcs $\{f_i^{(2)}\}$ and a set of non-candidate elliptical-arcs $\{\bar{f}_i^{(2)}\}$.

- 1 Initialize set $X = \emptyset$, set $Y = \emptyset$ and $T = \text{Number of fragments within } \{f_i^{(1)}\}$;
- 2 **for** $i=1$ **to** T **do**
- 3 Let $f_i^{(1)}$ be the current fragment under consideration;
- 4 Fit an ellipse to the edge pixels of $f_i^{(1)}$ by [213], where the fitted ellipse is denoted as $E_i^{(1)}$;
- 5 Compute $\text{Align}(E_i^{(1)}, f_i^{(1)})$ by equation (3.1);
- 6 **if** $\text{Align}(E_i^{(1)}, f_i^{(1)}) > \gamma$ **then**
- 7 Update set X as follows: $X = X \cup f_i^{(1)}$;
- 8 **else**
- 9 Update set Y as follows: $Y = Y \cup f_i^{(1)}$;
- 10 **end**
- 11 **end**
- 12 Output set X as $\{f_i^{(2)}\}$, set Y as $\{\bar{f}_i^{(2)}\}$ and terminate elliptical-arcs candidates identification procedure;

roots of a high order quartic function *e.g.* by the Newton method as in [217] (which needs an initial guess for a root of the function). Here, we instead compute the distance efficiently by the distance transform [218] which gives the distances of all edge pixels $x \in f_i^{(1)}$ to its closest boundary pixel of the ellipse, and hence the $\min(\cdot)$ operation in equation (3.2) reduces to a simple look up. Note that distance transform is done only once per ellipse $E_i^{(1)}$. We identify a fragment $f_i^{(1)}$ to be a candidate elliptical-arc if $\text{Align}(E_i^{(1)}, f_i^{(1)})$ is greater than a threshold γ or if $f_i^{(1)}$ belongs to a single line segment of the LEM. In order to ensure that very few elliptical-arcs present in an edge image are missed, we assign γ a low value of 0.2. We declare $f_i^{(1)}$ as $f_i^{(2)}$ if it is identified to be an elliptical-arc candidate and $\bar{f}_i^{(2)}$ otherwise. We summarize the method to identify elliptical-arcs candidates in Algorithm 3.1.

Figure 3.3 gives an illustration of the process to identify candidate elliptical-arcs for three toy example fragments $f_a^{(1)}$ (first row), $f_b^{(1)}$ (second and third rows) and $f_c^{(1)}$ (fourth row). Specifically, we show edge images and its corresponding LEM in the first and second columns of Figure 3.3 respectively. $f_a^{(1)}$, $f_b^{(1)}$ and

CHAPTER 3. ELLIPSE DETECTION

Figure 3.3: Identifying candidate elliptical-arcs on three toy example fragments $f_a^{(1)}$ (first row), $f_b^{(1)}$ (second and third rows) and $f_c^{(1)}$ (fourth row). First column: Edge images. Second column: LEM. Third column: Fragments $f_i^{(1)}$ shown by thick blue contours. Fourth column: Computation of $\text{Align}(\cdot)$ score for each $f_i^{(1)}$, where correspondences between edge pixel of $f_i^{(1)}$ and the closest pixel from $E_i^{(1)}$ are shown. Fifth column: Zoomed in portion of the figure in the fourth column. Green correspondences indicate that the distance between an edge pixel of $f_i^{(1)}$ and its closest pixel from $E_i^{(1)}$ is within 1 pixel away, yellow within 2 pixels away and red otherwise. Sixth column: Candidate elliptical-arcs are shown as thick blue contours, and non-candidate elliptical-arcs as thin contours. Note that candidate elliptical-arcs in the first, second and fourth rows are identified using $\gamma = 0.2$, while that in the third row using $\gamma = 0.001$. Seventh column: Elliptical-arcs detected by the later processes are shown as thick blue contours, and non-elliptical-arcs as thin contours.

$f_c^{(1)}$ that are obtained by the *Contour Fragments Extraction* process are shown by the thick blue contours in the third column of Figure 3.3. We fit an ellipse to each fragment $f_i^{(1)}$, and depict the fitted ellipse $E_i^{(1)}$ as the cyan outline in the fourth column of Figure 3.3. The $\text{Align}(\cdot)$ score, computed with equation (3.1), for each fragment is also given in the figures of the fourth column. Additionally,

each figure also shows the correspondences between the edge pixels of a fragment and its closest pixel from its fitted ellipse. For a clearer illustration, zoomed in portions for these correspondences are presented in the fifth column of Figure 3.3, where green correspondences indicate that the distance between an edge pixel of the fragment and its closest pixel from the ellipse is within 1 pixel away, yellow within 2 pixels away and red otherwise. Consider first the fragment $f_a^{(1)}$ (Figure 3.3(1c)). Given that its $Align(\cdot)$ score is 0.7321 which is greater than γ (fixed as 0.2 in this thesis), this fragment is thus considered as a candidate elliptical-arc $f_a^{(2)}$. We depict $f_a^{(2)}$ as the thick blue contour in Figure 3.3(1f). Figure 3.3(1g) shows the elliptical-arc (thick blue contour) that is obtained after applying the *Fragments Merging* and *Elliptical-arcs Refinement* processes (see ahead to Sections 3.2.3 and 3.2.4). In contrast to $f_a^{(1)}$, $f_b^{(1)}$ (Figure 3.3(2c)) has an $Align(\cdot)$ score of 0.0812 which is less than γ . This is because edge pixels of $f_b^{(1)}$ does not align well to its fitted ellipse as shown in Figures 3.3(2d) and 3.3(2e). As such, $f_b^{(1)}$ is identified to be a non-candidate elliptical-arc $\overline{f_b^{(2)}}$ and we depict $\overline{f_b^{(2)}}$ by the thin contour in Figure 3.3(2f). It is worth noting that since the identification of candidate elliptical-arcs is dependent on threshold value γ , a lowering of this threshold value identifies more candidate elliptical-arcs. For example, by reducing γ to 0.001, $f_b^{(1)}$ is identified to be a candidate elliptical-arc $f_b^{(2)}$ (thick blue contour in Figure 3.3(3f)). Despite this, no elliptical-arc was identified for *both* $\overline{f_b^{(2)}}$ and $f_b^{(2)}$ as shown in Figures 3.3(2g) and 3.3(3g) after applying the *Fragments Merging* and *Elliptical-arcs Refinement* processes since $f_b^{(1)}$ does not resemble the arc of an ellipse, though more computational effort is spent on analyzing $f_b^{(2)}$ than $\overline{f_b^{(2)}}$. In our experience, we found that a low value of 0.2 for γ affords a good trade-off between computational efficiency, while ensuring very few elliptical-arcs are missed. Finally, fragment $f_c^{(1)}$ has an $Align(\cdot)$ which is greater than γ (0.2) and is identified to be a candidate elliptical-arc (thick contour in Figure 3.3(4f)). However, since it does not have an elliptical shape, it is therefore not identified to be an elliptical-arc by the later processes, as indicated by the thin contours in Figure 3.3(4g).

Figure 3.4 gives an example for the identification of candidate elliptical-arcs for the real image in Figure 3.2(a). Specifically, we show its edge image in Figure 3.4(a), and fragment $\{f_i^{(1)}\}$ as thick contours in Figure 3.4(b) where each $f_i^{(1)}$ is shown

Figure 3.4: Illustration of the *Elliptical-arcs Candidates Identification* process. (a) Edge image. (b) Contour fragments $\{f_i^{(1)}\}$, where each $f_i^{(1)}$ is shown by a separate color. (c) A portion of (b) depicting $f_a^{(1)}$ and its fitted ellipse (cyan outline). This figure also show the correspondences between edge pixel of the fragment and its closest pixel of fitted ellipse, where green correspondences indicate that the distance between edge pixels and its closest pixel from the ellipse is within 1 pixel away (see zoomed in portions within gray outline boxes for clearer illustration). Alignment score for $f_a^{(1)}$ is given in the bottom left corner of (c).

by a separate color. A portion of Figure 3.4(b) depicting $f_a^{(1)}$ (and its fitted ellipse $E_a^{(1)}$) is shown in Figure 3.4(c). This figure also shows correspondences between edge pixel of $f_a^{(1)}$ to its closest pixel from $E_a^{(1)}$, where green correspondences indicate that the distance between them is within 1 pixel (see zoomed in portions within gray outlined boxes for better illustration). We show $Align(\cdot)$ score computed by equation (3.1) for $f_a^{(1)}$ in the bottom left corner of Figure 3.4(c). The perfect $Align(\cdot)$ score obtained by $f_a^{(1)}$ is due to close proximity of its edge pixels to the boundary of its fitted ellipse. Consequently, it is identified to be a candidate elliptical-arc $f_a^{(2)}$. Finally, Figure 3.2(e) shows the set of all candidate elliptical-arcs that are identified from the real image of Figure 3.2(a), where fragments corresponding to $f_i^{(2)}$ are shown by thick contours while those corresponding to $\overline{f_i}^{(2)}$ are shown by thin contours. As observed, most contour fragments which have an elliptical shape are retained as $f_i^{(2)}$. These served as seed fragments to initiate the merging procedure, described next.

3.2.3 Fragments Merging

In a cluttered environment or due to image noise, a LEM can include a large number of inflexion and corner points. This may results in true elliptical-arcs to be excessively broken by the *Contour Fragments Extraction* process in Section 3.2.1. Consequently, a valid elliptical-arc can be better represented by intra-merging the fragments within $\{f_i^{(2)}\}$ or inter-merging the fragments from $\{\bar{f}_i^{(2)}\}$ to $\{f_i^{(2)}\}$.

To this end, we exploit structural cues to guide the merging procedure, where we merge fragments with a two-step procedure. In the first step, we consider only fragments from set $\{f_i^{(2)}\}$, and begin merging from the longer (and hence more informative) fragments. Two fragments $f_j^{(2)}$ and $f_k^{(2)}$ are merged to form $f_{j,k}^{(2)}$ if (i) $f_j^{(2)}$ and $f_k^{(2)}$ are connected, and (ii) the alignment of the merged fragment $f_{j,k}^{(2)}$ to its fitted ellipse $E_{j,k}^{(2)}$ is no worse than that of either $f_j^{(2)}$ to $E_j^{(2)}$ or $f_k^{(2)}$ to $E_k^{(2)}$ *i.e.* they satisfy the following condition,

$$Pair(f_j^{(2)}, f_k^{(2)}) : Align(E_{j,k}^{(2)}, f_{j,k}^{(2)}) \geq \max \left(Align(E_j^{(2)}, f_j^{(2)}), Align(E_k^{(2)}, f_k^{(2)}) \right). \quad (3.3)$$

The first condition for connectedness avoids merging a valid elliptical-arc to clutter edges which very often lie in the vicinity, while the second condition in equation (3.3) ensures that the qualities for fit of fragments to ellipses is at least as good as that before merging. Finally, after considering all possible merges between fragments within set $\{f_i^{(2)}\}$, in the second step, we merge a fragment of set $\{\bar{f}_i^{(2)}\}$ to a fragment of set $\{f_i^{(2)}\}$. The decision to merge two fragments is evaluated in the same way as that during the first step merging procedure. We summarize the merging process in Algorithm 3.2 and denote the resulting set of fragments as $\{f_i^{(3)}\}$.

We illustrate the fragments merging process to form a fragment $f_e^{(3)}$ in Figure 3.5. Specifically, we show the test image of Figure 3.2(a) and a zoomed in portion of this image in Figure 3.5(a). Corresponding zoomed in portions from its edge image and LEM are shown in Figures 3.5(b) and 3.5(c) respectively. Additionally, we show candidate and non-candidate elliptical-arcs from the zoomed in portion as thick and thin contours in Figure 3.5(d) respectively. Consider candidate elliptical-arc $f_a^{(2)}$

Algorithm 3.2: Fragments merging procedure as described in Section 3.2.3

Input : Sets of candidate elliptical-arcs $\{f_i^{(2)}\}$ and non-candidate elliptical-arcs $\{\bar{f}_i^{(2)}\}$.

Output : A set of fragments $\{f_i^{(3)}\}$.

• **Fragments Merging Procedure** $\left(\{f_i^{(2)}\}, \{\bar{f}_i^{(2)}\}\right)$

- 1 Order fragments within sets $\{f_i^{(2)}\}$ and $\{\bar{f}_i^{(2)}\}$ by its lengths;
- 2 $\{t_i\} = \text{Perform Fragments Merging}\left(\{f_i^{(2)}\}, \{\bar{f}_i^{(2)}\}\right)$;
- 3 $\{f_i^{(3)}\} = \text{Perform Fragments Merging}\left(\{t_i\}, \{\bar{f}_i^{(2)}\}\right)$;
- 4 Output $\{f_i^{(3)}\}$ and terminate fragments merging procedure;

• **Perform Fragments Merging** (set X , set Y)

- 1 Initialize $j = 1$ and $T = \text{Number of fragments within set } X$;
- 2 **while** $j < T$ **do**
- 3 Let $f_j \in X$ be the fragment under consideration;
- 4 Identify the longest fragment f_k that is connected to f_j where $f_k \in Y$ and $f_j \neq f_k$;
- 5 **if** f_j and f_k satisfy Pair $(\cdot)$ of equation (3.3) and can be merged **then**
- 6 Update sets X and Y as follows: $X = X - \{f_j, f_k\}$, $X = X \cup f_{j,k}$ and
 $Y = Y - \{f_j, f_k\}$;
- 7 Order fragments within set X by its lengths;
- 8 Reset $j = 1$ and $T = \text{Number of fragments within updated set } X$;
- 9 **else**
- 10 $j = j + 1$;
- 11 **end**
- 12 **end**
- 13 return set X ;

shown in blue in Figure 3.5(d). Since it is directly connected to non-candidate elliptical-arc $\bar{f}_x^{(2)}$ (magenta fragment in Figure 3.5(e)) and the alignment score for an ellipse fitted to the collective edge pixels of $f_a^{(2)}$ and $\bar{f}_x^{(2)}$ is not worse than that of $f_a^{(2)}$, $f_a^{(2)}$ is thus merged with $\bar{f}_x^{(2)}$ to form $f_d^{(2)}$ as shown by the blue fragment in Figure 3.5(f). By the same token, $f_d^{(2)}$ is next merged with non-candidate elliptical-arc $\bar{f}_y^{(2)}$ (red fragment in Figure 3.5(f)) to form $f_e^{(3)}$ as shown in the Figure 3.5(g). Note that in Figure 3.5(d), $\bar{f}_w^{(2)}$ is not merged with $f_a^{(2)}$ but instead merged with $f_b^{(2)}$ to form $f_c^{(2)}$ (brown fragment in Figure 3.5(e)) since $f_b^{(2)}$ is longer and hence has more priority for merging than $f_a^{(2)}$. Also, $f_c^{(2)}$ is not merged with $f_a^{(2)}$ since such merging does not satisfy the condition in equation (3.3). Finally, Figure 3.2(f) shows the results of applying the merging process, where thick contours represent

Figure 3.5: Illustration of the *Fragments Merging* process. (a) Test image (left) and zoomed in portion (right). (b) Zoomed in portion of edge image. (c) Zoomed in portion of LEM. (d) Zoomed in portion showing candidate and non-candidate elliptical-arcs extracted after the *Elliptical-arcs Candidates Identification* process. Candidate elliptical-arcs are represented as thick contours and non-candidate elliptical-arcs as thin contours. (e)-(g) Sequences of fragments merging steps to form $f_e^{(3)}$.

set $\{f_i^{(3)}\}$. Observe that the merging process merges only those fragments whose shapes correspond to an ellipse, and we highlight these merged portions with arrows in Figure 3.2(f).

3.2.4 Elliptical-arcs Refinement

Having identified the set of fragments $\{f_i^{(3)}\}$ whose shapes are representative of an ellipse, the *Elliptical-arcs Refinement* process further isolates the elliptical portions of these fragments. Specifically, it considers all edge pixels within each $f_i^{(3)}$ collectively to sieve out those sets of connected edge pixels that provide strong support to the boundary of a partial or complete ellipse. This yields the elliptical-arcs of an edge image.

Let $f_i^{(3)}$ denote the current fragment to be refined. We represent as $\{l_m\}$ the

set of line segments within the LEM that are fitted to edge pixels of $f_i^{(3)}$. Let $l_m(a)$ denote the a^{th} edge pixel that belongs to the m^{th} line segment l_m . We define the ellipse that is fitted to $f_i^{(3)}$ as $E_i^{(3)}$ and the j^{th} boundary pixel of $E_i^{(3)}$ as $E_i^{(3)}(j)$. The *Elliptical-arcs Refinement* process calculates for each line segment l_m the percentage of its edge pixels that are within φ pixels away from the boundary of ellipse $E_i^{(3)}$ as,

$$g(E_i^{(3)}, l_m) = \frac{\sum_{a=1}^{K_m} h(l_m(a), E_i^{(3)})}{K_m} \quad (3.4)$$

where K_m denotes the number of edge pixels that belongs to l_m and

$$h(l_m(a), E_i^{(3)}) = \begin{cases} 1 & \text{if } \exists j, E_i^{(3)}(j) \in \mathbb{B}(E_i^{(3)}) \text{ and } \|E_i^{(3)}(j) - l_m(a)\|_2 \leq \varphi, \\ 0 & \text{otherwise.} \end{cases} \quad (3.5)$$

Let $f_i^{(4)}$ denote the set of edge pixels that belongs to line segments whose $g(\cdot)$ values are above ϕ . We assigned ϕ and φ the values 0.5 and 2 respectively. As such, each line segment in $f_i^{(4)}$ has more than 50% of its fitted edge pixels lying within 2 pixels (small enough) away from the boundary of the fitted ellipse. Consequently, edge pixels within fragment $f_i^{(4)}$ provide strong support to the boundary of an ellipse and localize an elliptical-arc. We define the set of elliptical-arcs extracted after this refinement process to be $\{f_i^{(4)}\}$. Algorithm 3.3 summarizes the refinement process.

We give examples for the refinement process in Figure 3.6. Specifically, Figure 3.6(a) shows an edge image from Figure 3.2(b). Line segments extracted from this edge image are shown in the LEM in Figure 3.6(b). Additionally, we color code edge pixels that are used to obtain these line segments in Figure 3.6(c). Consider line segment l_a that belongs to fragment $f_a^{(3)}$. A zoomed in portion of the edge pixels belonging to l_a (i.e. $\{l_a(i)\}$) is depicted by the purple contour in Figure 3.6(e). Additionally, we represent a part of an ellipse that is fitted to edge pixels of $f_a^{(3)}$ by the cyan outline in the same figure. Let this ellipse be denoted as $E_a^{(3)}$. Correspondences between $\{l_a(i)\}$ and its closest pixels from $E_a^{(3)}$ are shown in this figure, where green correspondences indicate that distance between $l_a(i)$ and its closest pixel from $E_a^{(3)}$ is within φ and red correspondences otherwise. The $g(\cdot)$

Algorithm 3.3: Elliptical-arcs refinement procedure as described in Section 3.2.4.

Input : A set of fragments $\{f_i^{(3)}\}$.
Output : A set of refined elliptical-arcs $\{f_i^{(4)}\}$.

- 1 Initialize set $X = \emptyset$ and $T = \text{Number of fragments within } \{f_i^{(3)}\}$;
- 2 **for** $i=1$ **to** T **do**
- 3 Let $f_i^{(3)}$ be the current fragment under consideration;
- 4 Let $\{l_m\}$ be the set of line segments within the LEM that are fitted to the edge pixels of $f_i^{(3)}$;
- 5 Fit an ellipse to the edge pixels of $f_i^{(3)}$ by [213], where the fitted ellipse is denoted as $E_i^{(3)}$;
- 6 Initialize set $Y = \emptyset$ and $M = \text{Number of line segments within } \{l_m\}$;
- 7 **for** $m=1$ **to** M **do**
- 8 Let l_m be the current line segment under consideration;
- 9 Compute $g(E_i^{(3)}, l_m)$ by equation (3.4);
- 10 **if** $g(E_i^{(3)}, l_m) > \phi$ **then**
- 11 Update set Y as follows: $Y = Y \cup l_m$;
- 12 **end**
- 13 **end**
- 14 Let $f_i^{(4)}$ denote the set of edge pixels that belongs to line segments of set Y ;
- 15 Update set X as follows: $X = X \cup f_i^{(4)}$;
- 16 **end**
- 17 Output set X as $\{f_i^{(4)}\}$ and terminate elliptical-arcs refinement procedure;

score for line segment l_a is computed to be 0.2619 with φ fixed as 2 (based on those $l_a(i)$ having green correspondences with a pixel of $E_a^{(3)}$). Similarly, for line segment l_b , we show correspondences between $\{l_b(i)\}$ and its closest pixels from $E_a^{(3)}$ in Figure 3.6(f), where $g(\cdot)$ score for l_b is computed as 0.7500. The higher $g(\cdot)$ score for line segment l_b as compared to that of line segment l_a is due to better alignment of $\{l_b(i)\}$ to $E_a^{(3)}$. Using ϕ to be 0.5, edge pixels belonging to l_a are removed from $f_a^{(3)}$ during the refinement process, while that belonging to l_b are retained. Finally, Figure 3.2(g) shows the result of applying the refinement process to each fragment $f_i^{(3)}$ of Figure 3.2(f). We highlight refined portions of the fragments with arrows and show the extracted set of elliptical-arcs $\{f_i^{(4)}\}$ as the thick contours in Figure 3.2(g). Further examples of elliptical-arcs extracted from synthetic test images are shown in the bottom row of Figure 3.2. Each pair of images in Figures 3.2(h), 3.2(i) and 3.2(j) shows the original edge image on the left and the extracted elliptical-arcs as thick contours on the right. As observed, due to corruption caused by image

Figure 3.6: Illustration of the *Fragments Refinement* process. (a) Edge image. (b) Corresponding LEM in which line segments are shown by a separate color. (c) Edge pixels that belong to the line segments of the LEM. Here, line segments and edge pixels are shown color coded. (d) Fragments identified to belong to $\{f_i^{(3)}\}$ after applying the *Fragments Merging* process are shown as thick contours, where each $f_i^{(3)}$ is shown by a separate color. (e) Correspondences between edge pixels belonging to line segment l_a (i.e. $\{l_a(i)\}$) and its closest pixel from ellipse $E_a^{(3)}$. (f) Correspondences between $\{l_b(i)\}$ and its closest pixel from $E_a^{(3)}$. Note that correspondences in (e) and (f) are shown color coded, where green indicates that distance between edge pixel and its closest pixel from $E_a^{(3)}$ is less than φ and red otherwise. $g(\cdot)$ scores obtained by l_a and l_b are given in the bottom left corner of (e) and (f) respectively.

clutter (Figure 3.2(h)) and noise (Figure 3.2(i)), some parts of valid ellipses have not been identified as elliptical-arcs. Nevertheless, despite these corruptions, each extracted elliptical-arc corresponds correctly to a single ellipse. We demonstrate in the following section that our approach is robust towards missing elliptical-arcs, such as those shown Figures 3.2(h) and 3.2(i), in which the original ellipses can still be accurately detected. Consequently, our approach is able to detect partially occluded ellipses (Figure 3.2(j)).

3.3 Ellipses Extraction by Grouping Elliptical-arcs

The previous section has extracted a set of elliptical-arcs from an edge image by exploiting edge connectivity to ignore clutter edges. Ellipses in simple images whose contours are neither broken nor occluded can be detected by fitting an

Figure 3.7: Flowchart for grouping disconnected elliptical-arcs for ellipse detection, with references to sections in this chapter.

ellipse to each elliptical-arc. However in complicated scenes, due to corruption from image clutter, noise or occlusion, an ellipse is often broken into many disconnected elliptical-arcs (see Figure 3.2). Such disconnected elliptical-arcs are insufficient to recover the original ellipse present in an image. Therefore, in order to detect broken ellipses, elliptical-arcs that belong to the same ellipse must be identified and grouped together. While one can adopt an exhaustive search approach to determine which of the elliptical-arcs should be grouped together, such an approach has exponential computation complexity and quickly becomes intractable for small number of elliptical-arcs. To detect ellipses robustly and efficiently, we develop a polynomial order approach which incrementally finds optimal pairings of elliptical-arcs.

We show the flowchart for detecting ellipses based on the grouping of elliptical-arcs in Figure 3.7. For clarity, we will omit the superscript (4) and represent an elliptical-arc detected by Section 3.2 as f_i where possible. In the initial step, the hypotheses generation system (which consists of the *Pair-wise Cost Assignment* and the *Optimal Pairings Identification* processes) exploits the local information of all extracted elliptical-arcs collectively to generate a set of hypotheses regarding the groupings of elliptical-arcs. By pooling together the lower level information of the elliptical-arcs in each group, the *Weak Hypotheses Identification* process identifies weak hypotheses and returns elliptical-arcs belonging to weak hypotheses to the hypotheses generation system to produce a refined set of hypotheses. Finally, the *Hypotheses Verification* process analyzes all generated hypotheses to sieve out

the best set of hypotheses. Consequently, by fitting an ellipse to the elliptical-arcs corresponding to these hypotheses, ellipses present in an image can be detected.

3.3.1 Pair-wise Cost Assignment

Given two elliptical-arcs f_j and f_k , $j \neq k$, we propose to assign a low cost to their pairing if they belong to the same ellipse and a high cost otherwise without resorting to the intractable analysis of other elliptical-arcs. It is expected that when f_j and f_k belong to the same ellipse, $f_{j,k}$ will be better aligned to its fitted ellipse $E_{j,k}$ as compared to when f_j and f_k belong to different ellipses. In light of this, we compute the cost of pairing elliptical-arcs f_j to f_k as

$$C_{j,k} = \begin{cases} 1 - \text{Align}(E_{j,k}, f_{j,k}) & \text{if } \text{Pair}(f_j, f_k) \text{ is satisfied,} \\ 1 & \text{otherwise} \end{cases} \quad (3.6)$$

where a cost $C_{j,k}$ of 1 represents that the pairing of f_j to f_k will not be considered. In order to have robust handling of outliers, we can modify the cost function in equation (3.6) as

$$\hat{C}_{j,k} = \begin{cases} 1 & \text{if } C_{j,k} > e_d, \\ C_{j,k} & \text{otherwise.} \end{cases} \quad (3.7)$$

In this case, an elliptical-arc f_j will not be paired with other elliptical-arcs whenever there is no real match available at a cost lower than e_d , which can be seen as a threshold parameter for detecting outliers. In our work, we set e_d to be 0.5. To avoid the pairing of an elliptical-arc to itself, we set the cost $\hat{C}_{j,j}$ to be 1.

The above cost $\hat{C}_{j,k}$ considers only the shape information of elliptical-arcs. We can include additional terms based on the local appearance differences between local neighborhoods of f_j and f_k . For example, we can add a cost based on the histogram intersection of the gray level patches within ellipses E_j and E_k to $\hat{C}_{j,k}$. This is especially useful as false pairings between elliptical-arcs can then be quickly identified and rejected. However, inclusion of such an appearance-based term is application dependant and is motivated by the robustness and invariance needs *e.g.* illumination variations alter pixels values and reduce the reliability of such

Algorithm 3.4: Elliptical-arcs optimal pairing identification procedure as described in Section 3.3.2.

Input : A set of elliptical-arcs $\{f_i^{(4)}\}$ and a cost matrix of $\hat{C}_{j,k}$.

Output : A set of enriched elliptical-arcs $\{f_i^{(5)}\}$.

- 1 Initialize set $X = \emptyset$;
- 2 **while** $\exists j, k$ such that $\hat{C}_{j,k} < 1$ **do**
- 3 Apply [219] to find the optimal pairing of elliptical-arcs π_* ;
- 4 Let set Y be the current set of enriched elliptical-arcs obtained by pairing based on π_* ;
- 5 Update set X as follows: $X = X \cup Y$;
- 6 Compute a new cost matrix of $\hat{C}_{j,k}$ based on Y by equation (3.7);
- 7 **end**
- 8 Output set X as $\{f_i^{(5)}\}$ and terminate optimal pairing identification procedure;

terms. We do not employ such terms in this thesis.

3.3.2 Optimal Pairing Identification

Given a cost matrix of $\hat{C}_{j,k}$ between every pair of elliptical-arcs from $\{f_i^{(4)}\}$, we propose to minimize the total cost of pairing subject to the constraint that the pairing is one-to-one. Let π denote a particular pairing of all elliptical-arcs and $\pi(j)$ denote the index of the elliptical-arc that is paired to f_j . We define the total cost of pairing based on π as $Cost(\pi) = \sum_j \hat{C}_{j, \pi(j)}$. Given this, an optimal pairing π_* is identified as

$$\pi_* = \arg \min_{\pi_x} Cost(\pi_x). \quad (3.8)$$

This is an assignment problem (minimum weighted matching) which can be solved in polynomial time $O(n^2 \log n)$ [219], where n denote the number of elliptical-arcs. We pair the elliptical-arcs based on π_* to derive a set of enriched elliptical-arcs. These enriched elliptical-arcs are fed back to the *Pair-wise Cost Assignment* process to compute the cost of pairing between them. The cost assignment and pairing operations continue until all entries of the cost matrix are 1. Let $\{f_i^{(5)}\}$ denote the set containing all enriched elliptical-arcs that are created by the pairing operation. In this case, each $f_i^{(5)}$ corresponds to a hypothesis that it contains a grouping of elliptical-arcs from $\{f_i^{(4)}\}$ that belong to the same ellipse. We summarize the *Optimal Pairing Identification* process in Algorithm 3.4.

3.3.3 Weak Hypotheses Identification

The above approach exploits local information of elliptical-arcs $\{f_i^{(4)}\}$ to hypothesize its groupings $\{f_i^{(5)}\}$. Although ellipses present in an image may be detected by fitting an ellipse to every $f_i^{(5)}$ and then rejecting poor elliptical fits, for robust ellipse detection, it is insufficient to rely solely on the current set $\{f_i^{(5)}\}$. In the presence of severe clutter or noise, pairings of elliptical-arcs of different ellipses may be assigned a lower cost than pairings of elliptical-arcs of the same ellipse and be incorrectly paired. Consequently a true ellipse may have its elliptical-arcs fragmented over different groupings and hence may not be detected. In order to have robust handling of such false groupings, we evaluate the overall goodness of each $f_i^{(5)}$ with a more global perspective by exploiting the edge image. Weak hypotheses are identified based on its goodness measure, and are harnessed to regenerate another set of better hypotheses. Consequently, by performing this self-correction, the proposed method homes in to detect difficult ellipses.

Let $f_i^{(5)}$ denote the current hypothesis whose goodness is to be evaluated and $E_i^{(5)}$ represent the ellipse that is fitted to $f_i^{(5)}$. We define q_k to be the k^{th} connected set of edge pixels from the edge image in which all points in q_k are within two pixels away from the boundary of $E_i^{(5)}$. While the goodness of $f_i^{(5)}$ can be immediately computed by the alignment of q_k to $E_i^{(5)}$, the robustness of the goodness computation can be enhanced by considering the *minimal* set of edge pixels within q_k that is required to model the part of $E_i^{(5)}$ to which q_k is aligned. Let e_j denote a j^{th} edge pixel from q_k and $p(e_j)$ denote the boundary pixel of $E_i^{(5)}$ that is closest to e_j . We identify from among q_k the set of edge pixels $\hat{q}_k$ such that for every $e_j \in \hat{q}_k$, its distance to $p(e_j)$ is the minimum among all other edge pixels of q_k whose nearest point on the boundary of $E_i^{(5)}$ is the same $p(e_j)$, *i.e.*

$$\hat{q}_k = \bigcup_{e_j \in q_k} e_j \quad (3.9)$$

subject to the constraint

$$\|e_j - p(e_j)\|_2 = \min \left(\|e_m - p(e_j)\|_2 \right), \quad \forall e_m \in q_k \text{ and } p(e_m) = p(e_j). \quad (3.10)$$

Figure 3.8: Illustrations of parameters used by the *Weak Hypotheses Identification* process. (a) Ellipse $E_i^{(5)}$ superimposed on an edge image, where grid represents edge image and black grid cell represents edge pixels. (b) Collection of black cells denote q_k . Each cell is a e_j that is within two pixels away from the boundary of the ellipse, and $p(e_j)$ is the boundary pixel of the ellipse that is closest to e_j (see zoomed in portion on the right). Green lines in the figure pair e_j with its corresponding $p(e_j)$. (c) Collection of black cells denote $\hat{q}_k$. Notice that $\hat{q}_k$ is the minimal set of edge pixels within q_k that is required to model the ellipse.

We depict $\hat{q}_k$ extracted from a toy edge image in Figure 3.8. Specifically, Figure 3.8(a) shows an ellipse superimposed on an edge image, where black cells denote edge pixels. We show the set of edge pixels that are within two pixels away from the boundary of the ellipse (*i.e.* q_k) in Figure 3.8(b). Each black cell denote a e_j , where green lines represent the correspondences between e_j and $p(e_j)$. We show $\hat{q}_k$ in Figure 3.8(c). As observed, $\hat{q}_k$ represents the minimum set of edge pixels within q_k that is required to model the part of the ellipse to which q_k is aligned.

We extract $\hat{q}_k$ efficiently with the distance transform and compute its support to $f_i^{(5)}$ as

$$\text{Support}(\hat{q}_k) = \sum_{\forall x \in \hat{q}_k} s(x, E_i^{(5)}) \times w(x) \quad (3.11)$$

where $s(x, E_i^{(5)})$ is computed from equation (3.2) and $w(x)$ denotes the weight of the x^{th} edge pixel and is equal to one for all edge pixels (these weights will be decayed progressively during the *Hypotheses Verification* process as described in Section 3.3.4). Let Ψ_i represent an ordered set of $\{\hat{q}_1, \dots, \hat{q}_n\}$ corresponding to hypothesis $f_i^{(5)}$ in which $\text{Support}(\hat{q}_k) \geq \text{Support}(\hat{q}_{k+1})$.

We measure the goodness of a hypothesis by an *ellipse visibility* term and a *contour continuity* term. The ellipse visibility term evaluates the extent to which the boundary of a detected ellipse comprises well-aligned edge pixels of the image,

$$T(\Psi_i) = \frac{\sum_{k=1}^n \text{Support}(\hat{q}_k)}{\left| \mathbb{B}(E_i^{(5)}) \right|} \quad (3.12)$$

and is large when the detected ellipse is well supported by edge pixels that are closely aligned to its boundary. This term considers *all* $\hat{q}_k$ that are aligned to the ellipse. In the presence of image clutter, there are likely to be noisy edge pixels that lie very close to an ellipse $E_i^{(5)}$, and which falsely increase the ellipse visibility score. We observe that since each $\hat{q}_k$ is a minimal set of connected edges that model a part of $E_i^{(5)}$, a $\hat{q}_k$ which arises from clutter edges will generally comprises a smaller set of edge pixels than those genuinely modeling a part of the ellipse, and therefore has a lower $\text{Support}(\cdot)$ value. Correspondingly, we harnesses the top few $\hat{q}_k$ which have stronger support to the ellipse and compute the contour continuity term as,

$$H(\Psi_i) = \frac{\sum_{k=1}^{\lambda} \text{Support}(\hat{q}_k)}{\left| \mathbb{B}(E_i^{(5)}) \right|}. \quad (3.13)$$

Note that $H(\cdot)$ is very similar to $T(\cdot)$ except for a partial summation over the top few $\hat{q}_k$. However, while $T(\cdot)$ evaluates the extent of ellipse visibility, $H(\cdot)$ measures the degree in which this visibility is due to longer chains of well aligned edge pixels, and penalizes those detections due to collection of many short $\hat{q}_k$. Thus, $H(\cdot)$ evaluates the continuity of contours modeled in Ψ_i . In this thesis, we fixed λ to be 3. Since computations for both $T(\cdot)$ and $H(\cdot)$ are within the range $[0, 1]$, we combine them to compute the goodness of hypothesis $f_i^{(5)}$ as

$$\text{Goodness}(f_i^{(5)}) = \sqrt{T(\Psi_i) \times H(\Psi_i)}. \quad (3.14)$$

We point out that this goodness function is a valuable choice to seek out weak hypothetical ellipses. It possesses the ability to identify those ellipses that are well

supported by edge pixels aligned to its boundary. At the same time, it is flexible to avoid identifying false hypothetical ellipses arising from image clutter as strong hypotheses. The goodness of a hypothetical ellipse is maximal when it corresponds to a complete ellipse of an edge image. We show examples of the goodness measures in Figure 3.9. Each row in Figure 3.9(a) shows a synthetic test image. We fit ellipses to exemplar hypotheses and depict these ellipses with an arrow in Figures 3.9(b), 3.9(c) and 3.9(d) in which the goodness of a hypothesis is given in the bottom right corner of each figure. As observed, equation (3.14) assigns a high goodness value to a hypothesis if the arcs of its fitted ellipse align well to the edge image. We identify those hypotheses whose goodness values are less than ω as weak hypotheses $\{f_i^{(6)}\}$. In our work, we assign a high value of 0.80 to ω to get good quality ellipses. Refined hypotheses are generated from $\{f_i^{(6)}\}$ by looping through processes *Pair-wise Cost Assignment* and the *Optimal Pairings Identification* of Sections 3.3.1 and 3.3.2, respectively, again with a modified condition $\hat{Pair}(f_j^{(4)}, f_k^{(4)})$,

$$\hat{Pair}(f_j^{(4)}, f_k^{(4)}) : \left[Pair(f_j^{(4)}, f_k^{(4)}) \right] \text{ AND } \left[f_j^{(4)} \in \{f_i^{(6)}\} \text{ OR } f_k^{(4)} \in \{f_i^{(6)}\} \right]. \quad (3.15)$$

In this way, the generation of a refined set of hypotheses focuses in only on the originally wrongly paired elliptical-arcs. To avoid re-computation of the cost matrix between pairs of elliptical-arcs, we record the cost matrices previously computed from process *Optimal Pairings Identification* in a hash table and refer to this hash table to generate the refined hypotheses. The iteration of the hypotheses generation and weak hypotheses identification procedures continues until no new pairing satisfy $\hat{Pair}(\cdot)$ of equation (3.15). In our work, we find that correct hypotheses can be very quickly identified within the initial few iterations. Given this, we can reduce computational time by restricting the number of iterations. As an example, we apply the above approach to generate hypotheses on the test images shown in Figure 3.9(a). Thirty-six, forty-seven and twenty-one hypotheses are generated respectively for the test images in the first, second and third rows of Figure 3.9(a) after two iterations. We show the ellipses corresponding to these hypotheses in Figure 3.9(e) by the cyan color outlines and represent the set of all hypotheses gen-

Figure 3.9: Hypotheses generation and verification. Each row is for one test image, where each test image contains eight ellipses with the test image in second row corrupted by 4% salt-and-pepper noise. (a) Test images. (b)-(d) Exemplar hypotheses with their corresponding goodness values in the bottom right corner. (e) Ellipses corresponding to all generated hypotheses after two iterations of the hypotheses generation and weak hypotheses identification processes. (f) All ellipses extracted by the *Hypotheses Verification* process.

erated as $\{f_i^{(7)}\}$. Figure 3.9(f) will be explained in the following section. We give a summary of the procedure to generate ellipse hypotheses by the identification of weak hypotheses in Algorithm 3.5.

3.3.4 Hypotheses Verification

The previous processes have extracted a set of hypotheses about the groupings of elliptical-arcs. The final step is to retain true hypotheses such that an ellipse fitted to the elliptical-arcs of each retained hypothesis agrees well with an edge image. Such ellipses represent the detected ellipses and model elliptical shaped contours of an image.

We adopt the goodness computation of equation (3.14) to assess the validity of a hypothesis. Our approach to identify true hypotheses is motivated by the observation that a valid ellipse of an image is unlikely to share majority of its boundary pixels with other ellipses. Conversely, false ellipses are often characterized by edge pixels belonging to multiple ellipses. Given this, we progressively reduce the

Algorithm 3.5: Generating ellipse hypotheses as described in Section 3.3.3.

Input : Sets of elliptical-arcs $\{f_i^{(4)}\}$ and enriched elliptical-arcs $\{f_i^{(5)}\}$, a weak hypotheses goodness threshold ω and an iteration threshold b .

Output : A set of ellipse hypotheses $\{f_i^{(7)}\}$.

```

1 Initialize  $j = 1$  and set  $X = \emptyset$ ;
2 while  $j \leq b$  do
3   Initialize  $\{f_i^{(6)}\} = \emptyset$  and  $T = \text{Number of enriched elliptical-arcs within } \{f_i^{(5)}\}$ ;
4   for  $i=1$  to  $T$  do
5     Let  $f_i^{(5)}$  be the current enriched elliptical-arc under consideration;
6     Compute  $\text{Goodness}(f_i^{(5)})$  by equation (3.14) with edge weights fixed at one;
7     if  $\text{Goodness}(f_i^{(5)}) < \omega$  then
8       Update set  $\{f_i^{(6)}\}$  as follows:  $\{f_i^{(6)}\} = \{f_i^{(6)}\} \cup f_i^{(5)}$ ;
9     end
10    Update set  $X$  as follows:  $X = X \cup f_i^{(5)}$ ;
11  end
12  Compute cost matrix of  $\hat{C}_{j,k}$  with equation (3.7) using modified condition of
    equation (3.15) and current set  $\{f_i^{(6)}\}$ ;
13   $\{f_i^{(5)}\} = \text{Apply Algorithm 3.4 using } \{f_i^{(4)}\} \text{ and cost matrix of } \hat{C}_{j,k}$ ;
14   $j = j + 1$ ;
15 end
16 Output set  $X$  as  $\{f_i^{(7)}\}$  and terminate hypotheses generation procedure;
```

contribution of an edge pixel towards the goodness measurement of hypotheses as the edge pixel aligns to more detected ellipses. False hypotheses that are composed by edge pixels of other ellipses are therefore assigned a lower goodness score and can be readily rejected by a threshold parameter τ . Let Φ represent the set containing the hypotheses that are retained. Algorithm 3.6 details the hypotheses verification procedure.

The value for τ is crucial to the overall performance (both in its recall and precision scores) of the detector since it determines the identities of false hypothesized ellipses that will not be reported by the detector. We experimentally derive this value of τ as follows. We generate a set of one hundred synthetic images each containing between five to twenty ellipses. Half of these images contain overlapping ellipses while the other half contain partially occluded ellipses. These ellipses are generated randomly. Using the procedure described above, we detect the ellipses that are present in an image. A detected ellipse E_d is considered to be correctly identified if there is a true ellipse E_t such that the overlap error between E_d and

Algorithm 3.6: Hypotheses verification procedure as described in Section 3.3.4.

Input : A set of ellipse hypotheses $\{f_i^{(7)}\}$ and an edge image I .
Output : A set Φ containing retained hypotheses.

- 1 Initialize $\Phi = \emptyset$ and $T = \text{Number of hypotheses within } \{f_i^{(7)}\}$;
- 2 Order the hypotheses in $\{f_i^{(7)}\}$ by its goodness measure in which the weights of all edge pixels are initialize to be one;
- 3 **for** $j=1$ **to** T **do**
- 4 Let $f_j^{(7)}$ be the current hypothesis under consideration and $E_j^{(7)}$ be an ellipse fitted to $f_j^{(7)}$;
- 5 Compute $\text{Goodness}(f_j^{(7)})$ with equation (3.14) based on updated weights assigned to edge pixels;
- 6 **if** $\text{Goodness}(f_j^{(7)}) \geq \tau$ **then**
- 7 Update Φ as: $\Phi = \Phi \cup f_j^{(7)}$;
- 8 Update the weights of edge pixels as: $\forall x, w(x) = w(x) - s(x, E_j^{(7)}) \times w(x)$;
- 9 **end**
- 10 **end**
- 11 Output Φ and terminate hypotheses verification procedure;

E_t is less than 0.05,

$$\text{OverlapError}(E_d, E_t) = 1 - \frac{\text{Area}(E_t) \cap \text{Area}(E_d)}{\text{Area}(E_t) \cup \text{Area}(E_d)} \quad (3.16)$$

where $\text{Area}(E_i)$ denotes the set of pixels within ellipse E_i . The low threshold of 0.05 for the overlap error requires that the boundaries of the true and detected ellipses be closely aligned before the true ellipse is considered as correctly detected (see Figure 3.10). For each value of τ , we compute the precision $P = \frac{\text{Number of correctly detected ellipses}}{\text{Total number of ellipses detected}}$ and recall $R = \frac{\text{Number of correctly detected ellipses}}{\text{Total number of ellipses present in test images}}$. The precision and recall values are then combined in the F -measure $F = \frac{2}{R^{-1} + P^{-1}}$, which is a common index used for detector evaluation. Figure 3.11 shows the F -measure for different values of τ . As observed, the highest F -measure is obtained when threshold τ is set to be 0.4, and this is the value used for all experiments throughout the thesis. Figure 3.9(f) shows the results after retaining only the valid hypotheses. Specifically, we fit an ellipse to the retained hypotheses and depict the ellipses by a differently colored outline. In addition, we score these ellipses by the goodness computation of equation (3.14) and show these scores in the legends of Figure 3.9(f).

Figure 3.10: Overlap error between two ellipses. Each image shows a pair of ellipses (shaded in green and blue), in which overlap error computed by equation (3.16) for each pair of ellipses is given in the bottom of each image. Cyan regions indicate intersect area of the ellipses, while colored regions its union area. Notice that close alignment of the boundaries of ellipses are required for low overlap errors (last two images).

Figure 3.11: F -measure for different values of τ . This curve is generated from a set of 100 synthetic images, in which each image contains between five to twenty ellipses.

3.4 Computational Complexity

The computational cost of the proposed algorithm is governed by the sum of the cost of the two key stages. The first stage of the algorithm constructs elliptical-arcs from line segments of an edge image. The complexity of extracting line segments is in the order of $O(p)$, where p denotes the number of edge pixels. Let ι and n represent respectively the total number of line segments and elliptical-arcs that are present in the edge image. Identifying contour fragments by the *Contour Fragments Extraction* process described in Section 3.2.1 requires the investigation of the connections between line segments and therefore the computation at this step is $O(\iota)$. Similarly, the *Elliptical-arcs Candidates Identification* process (Section 3.2.2) examines each contour fragment to identify candidate elliptical-arcs and is at worst

linear in the number of line segments. The *Fragments Merging* process (Section 3.2.3) examines the shape described collectively by a candidate elliptical-arc and its connected primitives and is upper bounded by $O(\iota^2)$. However, in practice, a candidate elliptical-arc is typically connected to at most one other primitive at each of its end points, so its complexity is on average linear in the number of line segments *i.e.* $O(\iota)$. The *Elliptical-arcs Refinement* process presented in Section 3.2.4 examines each elliptical-arc to sieve out poorly fitted line segments. In this case, its computational complexity is linear in the number of elliptical-arcs $O(n)$. Consequently, the worst case and average case computational complexity for the first stage of the proposed algorithm is upper bounded by $O(p + \iota^2 + n)$ and $O(p + \iota + n)$ respectively.

The second stage of the algorithm extracts a set of hypothetical ellipses by exploiting the elliptical-arcs generated from the first stage. The *Hypotheses generation system* (Sections 3.3.1 and 3.3.2) iteratively pairs elliptical-arcs together until no more elliptical-arcs can be paired. As pointed out previously, the pairings between elliptical-arcs in each iteration of the hypotheses generation can be effectively computed with a worst case runtime complexity of $O(n^2 \log n)$. Let $\frac{n}{2^k}$ denote the number of ellipses in the image. For the ideal case where the *Optimal Pairing Identification* process described in Section 3.3.2 pairs every elliptical-arc to another elliptical-arc in each iteration, the cost of finding the true groupings of elliptical-arcs can be derived to be

$$\begin{aligned}
& n^2 \log n + \left(\frac{n}{2^1}\right)^2 \log\left(\frac{n}{2^1}\right) + \left(\frac{n}{2^2}\right)^2 \log\left(\frac{n}{2^2}\right) + \cdots + \left(\frac{n}{2^k}\right)^2 \log\left(\frac{n}{2^k}\right) \\
& < n^3 + \left(\frac{n}{2^1}\right)^3 + \left(\frac{n}{2^2}\right)^3 + \cdots + \left(\frac{n}{2^k}\right)^3 \\
& = \frac{8n^3}{7} \left(1 - \frac{1}{2^{3(k+1)}}\right) \\
& = O(n^3).
\end{aligned}$$

Similarly, in the worst case where one optimal pairing is found in each iteration and

there are $n - k$ ellipses in the image, the cost required to find the true groupings is

$$\begin{aligned}
& n^2 \log n + (n-1)^2 \log(n-1) + (n-2)^2 \log(n-2) \\
& + \cdots + (n-k)^2 \log(n-k) \\
& < n^3 + (n-1)^3 + (n-2)^3 + \cdots + (n-k)^3 \\
& = n^3(k+1) - n^2 \frac{3k^2 + 3k}{2} + n \frac{2k^3 + 3k^2 + k}{2} - \frac{k^4 + 2k^3 + k^2}{4} \\
& = O(n^4)
\end{aligned}$$

since k is upper bounded by n . Both the *Weak Hypotheses Identification* process (Section 3.3.3) and the *Hypotheses Verification* process (Section 3.3.4) examine the set of hypothetical ellipses extracted and are at worst linear in the number of elliptical-arcs *i.e.* $O(n)$. Given this, the second stage of the algorithm is dominated by the *Hypotheses generation system* and has a worst case complexity of $O(n^4)$ and a best case complexity of $O(n^3)$. Consequently, the worst and best case complexity of the proposed algorithm is bounded by $O(p + \iota^2 + n^4)$ and $O(p + \iota + n^3)$ respectively.

3.5 Experimental Evaluation

We present detailed comparisons of our ellipse detector with existing methods on a set of challenging synthetic and real images. Given that our method belongs to the class of edge following techniques, we adopt the ellipse extraction methods proposed by Kim *et al.* [214] and Mai *et al.* [215] as reference methods for comparison. These ellipse extraction methods have been shown to possess good ellipse detection abilities. In addition, we also include the Standard Hough Transform [202] and the Randomized Hough Transform [199] as reference methods since these HT-based methods are widely cited in the literature and are often used as baseline methods for comparison. For all experiments, we use the same set of parameters for each reference method. These parameters are the default parameters supplied by the authors. The following abbreviations are used in this experimental evaluation,

- KIM: Kim *et al.* method [214];

- MAI: Mai *et al.* method [215];
- SHT: Standard Hough Transform [202]; and
- RHT: Randomized Hough Transform [199].

Ellipse detection by our method on all test images is achieved with the same set of parameter settings (details about the parameters can be found throughout this chapter). Edge pixels for all real images are extracted by the Canny edge detector [30] implemented in Matlab using its default values. We set threshold γ that is used in process *Elliptical-arcs Candidates Identification* as 0.2. Parameters ϕ and φ used in process *Elliptical-arcs Refinement* are assigned to be 0.5 and 2 respectively. The value of e_d used in process *Pair-wise Cost Assignment* is fixed as 0.5. The values of λ and ω that are used in process *Weak Hypotheses Identification* are assigned to be 3 and 0.8 respectively. Ellipse detection by our approach on all test images throughout the thesis is achieved by applying two iterations of the hypotheses generation and weak hypotheses identification procedures. These were manually chosen sensible values, but were not hand-optimized. The most important parameter used in this chapter is τ which we employed in process *Hypotheses Verification* to reject false ellipses. For this parameter, we experimentally derive the best value, giving τ to be 0.4.

3.5.1 Synthetic Images

Overlapping ellipses. We first assess the performance of our proposed ellipse detector in the presence of image clutter. Given an image containing η overlapping ellipses, the edge pixels of other ellipses contribute towards the outliers of each ellipse. In addition, for the case when the η overlapping ellipses are of equal circumference, the inliers-to-outliers ratio for each ellipse can be computed as $\frac{1}{\eta-1}$. In this case, the robustness of our algorithm towards outliers can be assessed by evaluating how detection accuracy varies as the number of overlapping ellipses in an image increases. For this purpose, we generate a set of 300×300 synthetic images in which each image contains η different ellipses, $4 \leq \eta \leq 24$. The parameters of the ellipses are generated randomly; we allow the center point of the ellipses to be arbitrarily located within the image and randomly assign a value between

$\left[10, \frac{\sqrt{300^2+300^2}}{2}\right]$ to the half-lengths of its major and minor axes. No restriction is placed on the orientation of the ellipses. The only constraint is that each ellipse must be completely contained in the image and overlap with at least one other ellipse. For each value of η , we evaluate the detection accuracy on fifty different images. In this work, we consider a true ellipse E_t to be correctly detected if there is a detected ellipse E_d such that $OverlapError(E_d, E_t)$ in equation (3.16) is less than 0.05. We show examples of the synthetic test images in Figure 3.12(a) and indicate the number of overlapping ellipses present in the bottom left corner of each test image. The effects on recall, precision and F -measure as η is increased are shown in Figures 3.12(b), 3.12(c) and 3.12(d) respectively. As observed, our detection results in terms of recall, precision and F -measure exceed that obtained by the reference methods for all tested η . We achieve perfect recall and precision when there are four overlapping ellipses in an image and more than 0.95 for both recall and precision when η is increased to twelve. This suggests that ellipses present in images can be robustly detected with little false positives even when there are around 90% outliers in an image. In addition, as η is increased to twenty-four, high values for recall (0.68) and precision (0.82) are still achieved by our approach. In contrast, the best performing reference method (KIM) obtains 0.32 for recall and 0.46 for precision at this value of η . As an example, Figure 3.13 shows detection results by the proposed and reference methods on a synthetic image containing sixteen overlapping ellipses. Correct detections are shown in green, false positives in red and false negatives in yellow.

Salt-and-pepper noise. We next investigate the performance of the proposed ellipse detector in the presence of salt-and-pepper noise. The addition of noise reduces the inliers-to-outliers ratio. More importantly, the noise corrupts the ellipse contour (either by breaking the ellipse contour or by introducing spurious edge pixels that are connected to the ellipse contour) and increases the difficulty in identifying elliptical-arcs belonging to the same ellipse. We follow the previous approach to generate six different sets of synthetic images, in which each image contains eight overlapping ellipses. Each set comprises fifty different images and is corrupted with a separate noise level. Figure 3.14(a) shows examples of the test

CHAPTER 3. ELLIPSE DETECTION

Figure 3.12: Testing on synthetic images that contain overlapping ellipses. (a) Examples of test images with the corresponding number of overlapping ellipses given in the bottom left corner. (b) Recall, (c) Precision and (d) F -measure are plotted against the number of overlapping ellipses. Each point of these curves is obtained from the detection results of fifty different synthetic images.

Figure 3.13: Ellipse detection on a synthetic image containing sixteen overlapping ellipses. (a) Test image. (b)-(f) Ellipses detected by our method, KIM, MAI, RHT and SHT respectively. Green represents correct detections, red false positives and yellow ground truth for false negatives.

images in which the bottom left corner of each image indicates the noise level that is used to corrupt the image. We plot the recall, precision and F -measure obtain by the proposed and reference methods at each noise level in Figures 3.14(b), 3.14(c) and 3.14(d) respectively. It can be observed, that unlike the reference methods, our proposed method is relatively invariant to moderate levels of salt-and-pepper noise. This is evident from the slope of the curves for which a nearly horizontal slope (for recall, precision and F -measure) is obtained by our approach at up to 8% salt-and-pepper noise. On the other hand, the reference methods demonstrate a degrading performance with increasing noise levels. At higher noise levels, due to a lack of candidate elliptical-arcs identified by our approach, ellipse detection

CHAPTER 3. ELLIPSE DETECTION

Figure 3.14: Testing on synthetic images each containing eight overlapping ellipses after corruption by salt-and-pepper noise. (a) Examples of test images with the noise density shown in the bottom left corner. (b) Recall, (c) Precision and (d) F -measure are plotted against noise density. Each point of these curves is obtained from the detection results of fifty different synthetic images.

Figure 3.15: Ellipse detection on a synthetic image containing eight overlapping ellipses that is corrupted by 16% salt-and-pepper noise. (a) Test image. (b)-(f) Ellipses detected by our method, KIM, MAI, RHT and SHT respectively. Green represents correct detections, red false positives and yellow ground truth for false negatives.

deteriorates. Despite this, our approach obtains the best detection accuracy at all tested noise levels. For example, at 12% salt-and-pepper noise, high values for recall (0.92) and precision (0.94) are attained by our method. On the other hand, the best performing reference method (KIM) obtains 0.28 for recall and 0.54 for precision. Similarly, superior performance is obtained by our method at adverse salt-and-pepper noise of 24%, where we achieve 0.60 for recall and 0.66 for precision. In contrast, recall and precision achieved by KIM deteriorates respectively to 0.07 and 0.39 at this noise level. We show in Figure 3.15 the ellipses that are detected by the proposed and reference methods on a synthetic image that is corrupted by 16% salt-and-pepper noise.

Occluded ellipses. In the third test, we investigate the robustness of our algorithm to detect occluded ellipses. We follow the procedure given above to generate a set of synthetic images each containing η occluded ellipses, $4 \leq \eta \leq 24$. We also include an additional constraint that each ellipse has at least 50% of its boundary pixels visible in the test image. Fifty different synthetic images are generated for each value of η . Examples of these test images are shown in Figure 3.16(a) in which the bottom left corner of each image denotes the number of occluded ellipses present. Figures 3.16(b), 3.16(c) and 3.16(d) show the detection results. As compared to the previous detection results on the images containing overlapping ellipses (Figure 3.12), there is a deterioration in the performance in terms of recall, precision and F -measure at each value of η for all detectors. This can be explained by the lack of inliers, on which ellipse detection by the proposed and reference methods are based. Despite this, our proposed method outperforms the reference methods in a systematic and significant way; for simple images where η is four, we obtain 0.90 for both recall and precision, while the best performing reference method (MAI) obtains 0.46 for recall and 0.74 for precision. The difference in performance is more apparent for larger η : at η equal to twenty-four, we obtain 0.55 and 0.60 for recall and precision respectively. In contrast, at this value of η , the best reference method obtains 0.02 for recall and 0.18 for precision. Figure 3.17 shows ellipse detection on a synthetic image containing sixteen occluded ellipses.

Choice of overlap error to identify matching ellipses. In the previous experiments, we consider a true ellipse to be detected if the overlap error between the true and detected ellipses is less than 0.05. We now test the effect of varying this threshold value. Specifically, a more detailed analysis is performed on the image set containing sixteen overlapping ellipses. Figures 3.18(a), 3.18(b) and 3.18(c) plot respectively the recall, precision and F -measure obtained by different detectors as a function of the overlap error. It is expected that as the overlap error threshold is relaxed, more ellipses will be quantified as matching. Correspondingly, the recall, precision and F -measures achieved by a detector should increase. While this observation can be made for all reference methods (except SHT which detects no ellipse from the test images), a nearly horizontal slope for the recall, precision and F -measure curves is obtained by our proposed ellipse detector after overlap

CHAPTER 3. ELLIPSE DETECTION

Figure 3.16: Testing on synthetic images that contain occluded ellipses. (a) Examples of test images with the corresponding number of occluded ellipses given in the bottom left corner. (b) Recall, (c) Precision and (d) F -measure are plotted against the number of occluded ellipses. Each point of these curves is obtained from the detection results of fifty different synthetic images.

Figure 3.17: Ellipse detection on a synthetic image containing sixteen occluded ellipses. (a) Test image. (b)-(f) Ellipses detected by our method, KIM, MAI, RHT and SHT respectively. Green represents correct detections, red false positives and yellow ground truth for false negatives.

error of 0.15. This is due to the capability of our detector to provide accurate localization of true ellipses present in test images. Consequently, further relaxing of the overlap error threshold does not improve the performance scores. Finally, despite the improved performance of the reference methods that is induced by increasing the overlap error value, the rankings of the tested detectors remain unchanged, with our detector performing the best.

Computational time. We now report the runtime performance of our method and compare it with that of the reference methods. Table 3.1 reports the mean and standard deviation time (in seconds) for detecting ellipses on synthetic images containing overlapping ellipses. Timings are for a Pentium 2GHz PC. In

CHAPTER 3. ELLIPSE DETECTION

Figure 3.18: Testing on fifty synthetic images each containing sixteen overlapping ellipses as overlap error threshold is varied. (a) Recall, (b) Precision and (c) F -measure for different detectors are plotted as a function of the overlap error.

general, computational time for ellipse detection by the methods increases as the number of overlapping ellipses in an image increases. This is not surprising since these methods rely on edge contours for detecting ellipses and hence require more computational time when presented with images containing more edge contours. Overall, our method has weaker runtime performance as compared to the reference methods. For example, ellipse detection by our method takes an order of minutes in an image containing twenty-four ellipses. In contrast, reference methods like MAI and RHT take an order of seconds and KIM takes less than a second to process the image. Consequently, given that our method demonstrates substantially robust detection performance, it is therefore more suitable in applications which require accurate ellipse detection for which speed is a secondary consideration. For example, in image retrieval, object detection and image classification applications, we can detect ellipses from an image to build object models off-line so that query processing time is not affected. Note also that for such applications, ellipse detection only needs to be done once per image, and hence the cost of extracting ellipses is greatly amortized when detecting object instances and classifying images over several object classes.

3.5.2 Real Images

We apply the proposed ellipse detector to a set of real images. Given that there are unpredictable disruptions to the expected contours in real images, detection of ellipses in real images is considerably more difficult than that in synthetic images.

CHAPTER 3. ELLIPSE DETECTION

Table 3.1: Computational time for detecting ellipses on synthetic images containing overlapping ellipses by our method and other reference methods. Results shown are the mean and one standard deviation time (in seconds) for detecting ellipses on a single 300×300 synthetic image.

Method	Number of overlapping ellipses per image					
	4	8	12	16	20	24
Our method	15.64 ± 5.74	78.70 ± 24.94	209.90 ± 49.35	503.14 ± 124.86	571.48 ± 106.53	807.46 ± 143.17
KIM	0.16 ± 0.19	0.25 ± 0.23	0.21 ± 0.16	0.18 ± 0.05	0.19 ± 0.07	0.27 ± 0.23
MAI	0.41 ± 0.19	1.06 ± 0.22	1.48 ± 0.25	1.77 ± 0.25	2.05 ± 0.25	2.41 ± 0.35
RHT	0.74 ± 0.14	3.56 ± 1.18	5.55 ± 1.88	9.22 ± 3.23	7.63 ± 1.46	22.14 ± 9.72
SHT	6.99 ± 2.21	25.56 ± 3.05	59.53 ± 8.67	67.38 ± 14.53	82.99 ± 15.88	129.35 ± 9.39

Figure 3.19: Detection of ellipses in a no-smoking sign. (a) Test image from [207]. Copyright Elsevier (2007). Reprinted by permission. (b) Edge image. (c) Magnified portion of edge image in which elliptical shaped contours corresponding to the smoke, tip and the internal of the cigarette can be seen. (d)-(f) All ellipses detected by Qiao and Ong [207], KIM and MAI respectively. (g) All ellipses detected by our approach. Both our approach and the reference methods have correctly detected the two large concentric ellipses (though detection by MAI is poorer). However, unlike the reference methods, five other contours having an elliptical shape have also been identified and fitted to perceptually agreeable ellipses by our approach.

Figure 3.19(a) shows a test image of a no-smoking sign that was tested previously by Qiao and Ong in [207]. We show its edge image in Figure 3.19(b). There are two large concentric ellipses with one of them having missing data points while the other is broken into two disconnected contours and are linked to straight line segments. In addition, as observed from the close up portion of the edge image in Figure 3.19(c), there exist other contours having elliptical shapes. For example, visual inspection reveals that the contours corresponding to the internal of the cigarette and the smoke have elliptical shapes. We show all ellipses detected by Qiao and Ong [207], KIM and MAI in Figures 3.19(d), 3.19(e) and 3.19(f) respectively. Figure 3.19(g) shows all ellipses detected by our method. Similar to the three reference methods (where MAI has more inferior detection results), we have correctly detected the two large concentric ellipses that encircle the cigarette. More importantly, unlike the

CHAPTER 3. ELLIPSE DETECTION

Figure 3.20: Comparison of ellipse detection methods on test images from the Caltech-256 database [216]. (a) Test images. (b) Edges detected by Canny operator. (c) All ellipses detected by our approach. (d) Top twenty ellipses based on its goodness scores. Darker shades represent ellipses with higher goodness scores. (e) All ellipses detected by KIM. (f) All ellipses detected by MAI.

reference methods, our algorithm has modeled five other elliptical shaped contours corresponding to the cigarette and the smoke by perceptually agreeable ellipses. Given that there are cluttered edge pixels in the vicinity of these contours, their detection demonstrates the robustness of our approach.

We next evaluate our method on eight complex real images from the Caltech-256 database [216] (see the left most column of Figure 3.20). These images present greater challenges than the real image tested in Figure 3.19 due to adverse corruption to the contours of elliptical shaped objects by complex and varied backgrounds, illumination variations, partial occlusions, image noise, shadows and spectral reflections. Consider the test image in Figure 3.20(1a). This image consists of a fountain having a flat elliptical top. There is also a small dark colored elliptical patch in the interior of the fountain top. In addition, multiple elliptical contours can be seen along the stem of the fountain. As seen from its edge image in Figure 3.20(1b), these contours are broken into many disconnected fragments and have missing data points. Despite these challenges, accurate detection of these elliptical contours is achieved by our method as shown in Figure 3.20(1c). In addition, two ellipses corresponding to the base of the two fountains in the background have also been detected by our approach. On the other hand, given that the base of the fountain in the foreground is not elliptical shaped, therefore no ellipse is modeled to its base. Although some ellipses corresponding to the spurious contours of the edge image have also been detected, we point out that such detections can be readily filtered. For example, we rank the detected ellipses by their goodness scores computed from equation (3.14) and show the best twenty ellipses in Figure 3.20(1d). Ellipses that have higher goodness scores are shown by darker colors. As observed, these detections correspond to visually salient ellipses of the edge image. We show detection results by KIM in Figure 3.20(1e). Even though their algorithm detects an elliptical contour along the stem of the foreground fountain, the elliptical top of the fountain has not been detected accurately. In addition, neither base of the two background fountains are detected by their approach. Detection results by MAI show no significant improvement (see Figure 3.20(1f)); although elliptical top of the foreground fountain has been localized by their approach, no elliptical contours corresponding to its stem have been detected. In addition, as evident from both the test image in Figure 3.20(1a) and its edge image in Figure 3.20(1b), their detected ellipse that corresponds to the base of the left background fountain is poorly localized. In this aspect, ellipses detected by our method are considerably better than both reference methods.

Of particular interest are the test images shown in Figures 3.20(2a) and 3.20(3a) which contain natural objects (watermelon and tomatoes) having elliptical shapes. As observed from their edge images in Figures 3.20(2b) and 3.20(3b), there is substantial overlap of the elliptical contours belonging to these objects. In addition, due to illumination variations or partial occlusion, there are missing edge pixels along the contour of the objects. The presence of clutter edge pixels corresponding to the background (especially the highly textured background in Figure 3.20(3a)) further complicates detection. Despite these challenges, good detections of the elliptical shaped objects by our method is achieved as shown in Figures 3.20(2c) and 3.20(2d) and Figures 3.20(3c) and 3.20(3d), in which the detected ellipses closely correspond to the contours of the watermelon and tomatoes in the test images. In contrast, challenges encountered in these test images render KIM and MAI ineffective in detecting the ellipses (*c.f.* Figures 3.20(2e) and 3.20(3e), and Figures 3.20(2f) and 3.20(3f)).

Figure 3.20(4a) shows a test image of a jumbled collection of golf balls. We show its edge image in Figure 3.20(4b). It is observed from Figures 3.20(4c) and 3.20(4d) that dominant ellipses corresponding to the golf balls have been identified by our approach. In contrast, KIM identifies only one elliptical contour corresponding to a partially occluded golf ball (see Figure 3.20(4e)) while MAI fails to detect any elliptical contours from this test image (see Figure 3.20(4f)). We show a test image of a motorbike in Figure 3.20(5a) and display its edge image in Figure 3.20(5b). Figures 3.20(5c) and 3.20(5d) show the ellipses detected by our approach. Obvious ellipses corresponding to the wheels of the motorbike (as seen from its edge image) have been detected by our approach. On the other hand, both KIM and MAI fail to detect obvious ellipses from this test image (see Figures 3.20(5e) and 3.20(5f)). Ellipse detection of the bowling ball image (Figure 3.20(6a)), bicycle image (Figure 3.20(7a)) and pram image (Figure 3.20(8a)) bear similar results and clearly demonstrate the robustness of our approach to extract ellipses from difficult real-world images.

3.6 Summary

A new method for ellipse detection has been described. It is based on the concept of edge following where we model edge connectivity by line segments. There are two key stages in our algorithm. In the initial stage, connected line segments that collectively describe elliptical-arcs are identified and grouped together. We highlight that unlike current methods, we pool low level information described by the line segments and elliptical-arcs together to achieve a more global shape understanding of the elliptical contours present in an edge image. Consequently, by exploiting the more complete shape information, the constructed elliptical-arcs are refined to better describe the elliptical contours. In the second stage, disconnected elliptical-arcs that belong to the same ellipse are identified and grouped together. For this purpose, we propose an approach which incrementally finds optimal pairings of elliptical-arcs. Hypothetical ellipses that are present in an edge image are detected by fitting an ellipse to the elliptical-arcs of each group. Finally, a feedback loop is developed to sieve out low confidence ellipses and to regenerate another set of better hypothetical ellipses by combining elliptical-arcs from different groups. In this aspect, our algorithm performs automatic self-correction and homes in on “difficult” ellipses.

We perform detailed evaluation of the proposed approach on a set of challenging synthetic images. It is shown that the proposed method systematically and substantially outperforms existing methods in terms of recall, precision and F -measure under background clutter, salt-and-pepper noise and partial occlusion. In addition, we apply our method on a set of challenging real-world images from the Caltech-256 database. Successful detection of ellipses from these images is demonstrated. As far as we know, no other detector can extract ellipses from such difficult images. Consequently, this chapter advances the state-of-the-art in ellipse detection.

Chapter 4

Contour Based Object Recognition

This chapter demonstrates that image classification and object detection on complex real-world images can be achieved by very simple and generic shape primitives of line segments and ellipses. We pair primitives to form shape tokens, and learn category-specific combinations of tokens which reliably predict the presence of object instances across scale and space. We do not restrict combinations to have a fixed number of tokens, but allow them to flexibly adapt to an object category. This, together with the generic nature of line segments and ellipses, empowers a combination with much flexibility to represent discriminative shape structures of varying complexity. We harness noisy and unsegmented training images in our work to learn an object model, with only bounding boxes provided. We experimentally evaluate our approach against recent state-of-the-art contour based methods and other related methods. Very competitive results are attained.

4.1 Introduction

In this chapter, we harness discriminative combinations of very simple and generic shape primitives of line segments and ellipses for image classification and object detection in real-world images. The proposed contour based features differ from region based features that have been commonly used for object recognition. However,

unlike region based features, contour features offer a number of unique advantages: it is less affected by changes in color or texture; it is more robust to illumination variations; it corresponds naturally to object parts for both rigid and articulated object categories; it can be matched along object boundary even under changing background (in contrast, region features extracted at object boundary are easily corrupted by the nearby presence of background pixels). The proposed shape primitives of line segments and ellipses are largely complementary in nature; the former models straight edge contours while the latter models curved contours. We choose an ellipse since it is one of the simplest circular shapes yet offers sufficient flexibility to represent curved object structures [220]. These primitives have several attractive properties. First, unlike edge based descriptors (*e.g.* shape contexts [162]) that are used by other contour based recognition methods described in Section 2.2.2.3, these primitives support abstract and perceptually meaningful reasoning like parallelism and adjacency. Also, unlike contour fragments and skeletons, storage demands by these primitives are independent of object size and hence can be efficiently represented (with four parameters for a line segment, and five parameters for an ellipse). Additionally, distance computations between similar typed primitives can be quickly evaluated (*e.g.* by the differences in their geometrical properties) as compared to contour fragments and skeletons which often require comparison between individual edge pixels (*e.g.* by Chamfer [78] and Hausdorff [77] distances as described in Section 2.1.2.2). More importantly, these primitives simplify matching at multiple scales, where its geometrical properties are easily scale normalized. On the other hand, contour fragments and skeletons are not scale invariant and one is forced either to rescale the fragments which introduces aliasing effects (when edge pixels are compressed together or pulled apart) or to resize images before extracting fragments which degrades image resolution.

Three important issues which are particularly relevant to finding useful primitive combinations that characterize a target object category are: (i) which primitives are to be combined together, (ii) how many primitives are to be selected into a combination, and (iii) how can these combinations be found efficiently. The first two issues deal with finding informative and flexible primitive combinations in which different combinations can have different number of primitives. However,

the search for these combinations is constrained by practical realities (third issue). Indeed, given N primitives and with a maximum of m primitives per combination, the number of primitive combinations to be individually assessed is in the order of $O(N^m)$. This search space is exponential in m , and the search for useful combinations becomes intractable even for moderate values of N and m .

This chapter presents an efficient approach to learn category-specific primitive combinations. Specifically, each combination is a two-layer abstraction of primitives: pairs of primitives (termed *shape tokens* in this chapter) at the first layer, and a learned number of tokens at the second layer. We do not constrain combinations to have a fixed number of tokens, but allow them to automatically and flexibly adapt to an object category. The number of tokens in a combination influences the ability of a combination to represent object shapes (and structures), where simple shapes favor combinations with fewer tokens than complex ones. Consequently, discriminative primitive combinations of varying complexity can be harnessed to represent simple and complex shape structures of a category. We learn these combinations by exploiting distinguishing geometric, structural and appearance constraints of a category in a unified framework. Geometric constraints describe the spatial layout (configurations) of shape tokens, while structural constraints enforce possible poses/structures of an object by the relationships (*e.g.* XOR relationship) between tokens. Appearance constraints describe the visual aspect of tokens which we represent using generic shape primitives of line segments and ellipses.

Figure 4.1 outlines our proposed approach, with references to corresponding sections in this chapter which are structured as follows. In Section 4.2, we describe our method to construct shape tokens from the primitives of line segments and ellipses. We extract line segments by applying the dynamic two strip algorithm in [31], and ellipses by our ellipse detector in Chapter 3 on an edge map that is computed with the Canny edge detector [30]. The constructed shape tokens are local contour features, in which each token has a well-defined position, and the appearance of the token is described by the geometrical properties between the shape primitives that constructed the token. The matching of tokens at a single scale and across multiple scales is also described in this section.

Section 4.3 details our method to learn a category-specific codebook of shape

Figure 4.1: Outline of our contour based method to detect objects and classify images. The dataset (of object and background images) is split into training and testing sets. Shape primitives are extracted from training and testing images in the same way, and we harness these primitives to construct shape tokens. The training images are used to build a category-specific codebook of shape tokens, which is then exploited to learn a boosted ensemble of discriminative codeword combinations. This ensemble is used for object detection and image classification, and we evaluate recognition performance on the testing set.

tokens. Here, we harness shape tokens that are extracted from within the bounding boxes of training objects, and cluster similar tokens together by exploiting the appearance and positional cues of these tokens. A radial ranking method is developed to select representative tokens into a codebook, with the aim that the selected tokens cover a large spatial extent of the object model.

Our approach to learn discriminative combinations from the training images is presented in Section 4.4. Each combination has a variable number of codewords, and models discriminative appearance, geometric and structural constraints of the

target object category. A boosted ensemble of discriminative codeword combinations is also learned to distinguish object presence and absence at object hypotheses across scale space.

In Section 4.5, we describe the multi-scale sliding window technique which we employ to detect objects across space and scale in test images. The boosted ensemble analyzes each window individually and gives the confidence of an object present within the window. Extension of the boosted ensemble for image classification is also discussed in this section.

Section 4.6 provides an extensive evaluation involving 17 challenging object categories for a total of more than 2400 test images. Both tasks of image classification and object detection are considered. Specifically, we study how detection performance varies as a result of viewpoint changes. In addition, we investigate its ability to discriminate object categories that have high visual shape similarity. Finally, we evaluate our technique against recent state-of-the-art contour based recognition methods, and other related methods. Very competitive results are obtained in which robustness towards intra-class variation, articulation and scale changes, illumination variations, partial occlusion and background clutter are demonstrated.

Finally, this chapter closes with a summary of findings in Section 4.7.

4.1.1 Related Works

Here, we discuss previous works that are directly related to the contour based recognition technique proposed in this chapter. For a broader discussion on other object detection and image classification works, including those methods using region based features and other contour representations, we refer the reader to Chapter 2.

Chi and Leung [161] used line segments and circles as contour features. They do not learn representative features of training objects, but instead store all features (extracted from training images) into a potentially large hypercube to vote on the presence of objects in test images. Consequently, their approach demands figure-ground segmentation of training objects which limits their applicability; we require only bounding boxes. Additionally, their work considers only the retrieval

of synthetic logo images, and does not address the (more challenging) task of object localization in complex real-world images. Recently, Ferrari *et al.* [156] used connected straight contour fragments as features. This is similar to our work where line segments (and ellipses) are paired as shape tokens. However, their method requires the powerful Berkeley boundary detector [70] to find meaningful object boundaries and to filter noisy background contours as a preprocessing step (before either training or testing). In contrast, our method extracts contours by the traditional Canny edge detector and is robust to detected noisy edge pixels. Additionally, their method represents shapes by features extracted from a set of category-neutral images. We depart from this framework and instead explicitly tailor contour features to the target object category.

Our choice for constructing category-specific features is motivated by the substantial success in Shotton *et al.* [91] and Opelt *et al.* [149]. A difference between our work and theirs is that they represent local shapes by contour fragments while we employ generic and simple shape primitives. Consequently, they suffer the shortcoming presented earlier in Section 4.1. A more important difference from their work is that to keep the learning of discriminative features tractable, they limit each feature to contain a fixed number of fragments (single fragment in [91] and two fragments in [149]). Our approach imposes no such restriction, and instead learns category-specific contour features that have a variable number of shape primitives.

4.2 Shape Tokens

In this section, we present our contour based features of shape tokens, discussing how the appearances of tokens are described, and how tokens are matched at a single scale and across multiple scales. First however, we detail our method to construct shape tokens.

4.2.1 Constructing Shape Tokens

We construct a shape token by pairing a reference primitive to its neighboring primitive, where a primitive is either a line segment or an ellipse. Line segments

and ellipses are extracted by the line fitting method in [31] and by our ellipse detector presented in Chapter 3 respectively. Given two primitives of different types (*i.e.* one line segment and one ellipse), an ellipse will always be considered to be the reference primitive. For two primitives of the same type, we consider each primitive in turn as the reference primitive. This gives the following types of shape tokens: line-line, ellipse-line and ellipse-ellipse.

We first present our method to identify the neighboring line segments of a reference line for constructing line-line tokens. Here, our goal is to identify neighboring lines which are very likely to belong to the same boundary as the reference line. A key challenge that presents itself is that, locally, a line lacks specificity and lines belonging to a same boundary can be identified only when higher level knowledge of the boundary is given. In the absence of higher level information, connectivity is a natural and intuitive mechanism for selecting neighbors, where we identify lines that are adjacently connected at either endpoints of a reference line as its neighbors. A weakness of directly exploiting connectivity is that due to brittleness in the edge maps of complex real-world images, contours are often broken into multiple disconnected fragments. Consequently, a reference line may not have its neighbors connected to it. For robust collecting of neighboring lines, we adapt the approach in Ferrari *et al.* [156], and define a small isosceles trapezium (search area) at both endpoints of the reference line. The minor base of each trapezium is centered on an endpoint of the reference line, and its height is oriented in the direction of the reference line and pointing away from the reference line⁷ (see Figure 4.2). A line segment is considered to be connected to the reference line if any point within this line segment is within either of the two trapeziums. This bridges small breaks between line segments, thereby providing robustness to fragmented contours and to short missing fragments, and constrains a line-line token to be string-like shape structures. Note that our scheme to find neighboring lines of a reference line defines a trapezium at both ends of a reference line, and is a generalization of [156] where trapeziums are defined only at edge discontinuities. This generalization provides tolerance to some poorly fitted lines (since neighbors are found from a small search

⁷The lengths of the minor and major bases of an isosceles trapezium are fixed at 5 and 9 pixels respectively, and its height is 11 pixels. These are sensibly chosen values, but are not hand-optimized. For all experiments, we used the same fixed size trapezium.

Figure 4.2: Finding the neighbors of a reference line segment. (a) An example contour detected by an edge detector, where breaks along the contour are highlighted by two red-outlined circles. (b) Representation of the contour by a line edge map (LEM) that is obtained by [31], from which all shape primitives are extracted. (c) A reference line is shown in bold, and we define two trapeziums at its end points. Any line which has at least one point inside either trapezium is considered to be connected to the reference line, and constitutes its neighbor. We show these neighboring lines in brown. Note that this trapezium method is also employed to bridge breaks in the LEM, before finding the neighbors of a reference ellipse.

area and are not restricted to only adjacently connected lines) and affords a larger pool of line-line tokens modeling the same boundary to be constructed, at the cost of constructing some tokens from line segments of different boundaries.

Next, we describe the construction of an ellipse-ellipse and an ellipse-line token. Unlike a reference line, a reference ellipse (being a convex self-complete curve) does not have any endpoints, and is possibly detected from multiple disconnected elliptical fragments. While one can define a trapezium (by the above method) at the endpoints of these elliptical fragments, such trapeziums will be oriented along the ellipse, and hence likely to collect neighboring primitives that are already described by the ellipse. The pairings of such neighbors to the reference ellipse yield no additional shape/structure information. Instead, given a reference ellipse, we first center a circular search area on this ellipse. The radius of the circular search area is dependent on the size of the reference ellipse. In all experiments, we assigned the circular area a radius equal to three times the half-length of the major axis of the reference ellipse. Note that the circular search area is independent of the orientation of the reference ellipse (unlike *e.g.* an elliptical search area), and hence provides robustness against missing neighbors when pose of an object changes. For example, consider a motorbike that is seen from the side. A reference ellipse representing the front wheel of the motorbike will pick the *same* set of neighbors despite substantial in-plane rotation of the front wheel about its turning axis. We consider any other primitives within this search area that is “weakly” connected to the reference ellipse as neighbors. Our concept of weak connectivity is described as

Figure 4.3: Finding the neighbors of a reference ellipse. (a) An example LEM. (b) A reference ellipse is shown in bold with its circular search area (shown shaded) centered on it. Any line which is weakly connected to reference ellipse and is within its search area constitutes the neighboring line of the reference ellipse. We show these neighboring lines in brown. (c) Similarly, any ellipse which is weakly connected to reference ellipse and is within its search area forms the neighboring ellipse of the reference ellipse. We show such a neighboring ellipse in brown.

follows. First, we bridge breaks in the line edge map (LEM) by defining trapeziums at edge discontinuities. A line segment is considered to be weakly connected to a reference ellipse if a path can be traced from it to the reference ellipse by following along line segments and across bridges between line segments of the LEM. In the same spirit, we identify an ellipse to be weakly connected to a reference ellipse if a path can be traced between any of their underlying line segments. Line segments and ellipses that are weakly connected to the reference ellipse and are within the circular search area forms the neighboring line segments and neighboring ellipses of the reference ellipse (see Figure 4.3). Consequently, ellipse-line tokens and ellipse-ellipse tokens are then formed by pairing the reference ellipse to its neighboring line segments and neighboring ellipses respectively.

4.2.2 Describing Appearances of Shape Tokens

A low dimension numerical descriptor comprising simple geometrical attributes is used to describe the appearance of a shape token. Let θ denote the orientation of a primitive, $\theta \in [0, \pi]$. For an ellipse whose eccentricity is more than λ_ϵ (fixed at 0.8), θ is assigned to be the orientation of its major axis; orientation of a circle or an ellipse whose eccentricity is less than λ_ϵ is fixed as π . We define $\begin{bmatrix} v^x & v^y \end{bmatrix}^T$ as the unit vector from the center of the reference primitive to the center of its neighbor, and h as the distance between their centers. The midpoint between their centers

Figure 4.4: Illustration for the geometrical attributes used to describe the appearance of (a) line-line, (b) ellipse-line, and (c) ellipse-ellipse tokens. Reference primitives are shown in bold, and neighboring primitives in light-gray.

is defined to be the token centroid. We denote the length and width of a primitive as l and w respectively. For an ellipse, the length is given by its major axis. We define the width of an ellipse by its minor axis, and fix the width of a line segment to be one pixel thick. Given these notations, the appearance descriptor of a token is

$$A = [l^r \ w^r \ \theta^r \ l^n \ w^n \ \theta^n \ h \ v^x \ v^y]^T, \quad (4.1)$$

where the superscripts r and n differentiate attributes of a reference primitive from its neighbor. Figure 4.4 illustrates these geometrical attributes used to describe the appearances of line-line, ellipse-line and ellipse-ellipse tokens.

4.2.3 Matching Shape Tokens at a Single Scale and across Multiple Scales

Having defined the appearance descriptor of a shape token, we now present our approach to compare different tokens. As stated before, three different types of shape tokens are constructed. A token is compared only with similar typed tokens. We first describe the approach to compare tokens at a single scale. For this purpose, we define a symmetrical distance measure between two tokens with descriptors A_i and A_j as

$$D(A_i, A_j) = \sum_{p \in (r, n)} D_l(l_i^p, l_j^p) + \sum_{p \in (r, n)} D_l(w_i^p, w_j^p) + \sum_{p \in (r, n)} D_\theta(\theta_i^p, \theta_j^p) + D_l(h_i, h_j) + D_v(v_i^x, v_i^y, v_j^x, v_j^y), \quad (4.2)$$

where

$$\begin{aligned}
D_l(l_i, l_j) &= \min \left(\left| \ln \left(\frac{l_i}{l_j} \right) \right|, 1 \right), \\
D_\theta(\theta_i, \theta_j) &= \frac{\min \left(|\theta_i - \theta_j|, \pi - |\theta_i - \theta_j| \right)}{\frac{\pi}{2}}, \\
D_v(v_i^x, v_i^y, v_j^x, v_j^y) &= \sqrt{(v_i^x - v_j^x)^2 + (v_i^y - v_j^y)^2},
\end{aligned}$$

and $D_l \in [0, 1]$ measures the difference in lengths, $D_\theta \in [0, 1]$ the difference in orientations, and $D_v \in [0, 2]$ the difference in relative primitive positions. Notice that the distance range of D_v is twice that of both D_l and D_θ . This is equivalent to giving more emphasis to the contribution of D_v to the $D(\cdot)$ distance computation in equation (4.2), where the relative primitive position is often more reliably computed as compared to the primitive lengths and their orientation. Correspondingly, equation (4.2) combines the differences in geometrical attributes of the tokens together into a single useful dissimilarity measure (matching cost), in which the distance range for comparing line-line, ellipse-line and ellipse-ellipse tokens are $[0, 7]$, $[0, 8]$ and $[0, 9]$ respectively. This formulation in equation (4.2) is similar to that used by Ferrari *et al.* [156] for comparing between their features of connected straight contour segments (*k adjacent segments*), but has been generalized to evaluate geometrical differences between ellipse primitives, and to account for difference in spatial separation between primitives.

It is easy to extend the above matching to multiple scales. Specifically, the descriptor of a token can be normalized against an object scale b_s as

$$\begin{aligned}
\bar{A} &= f \left(A, \frac{1}{b_s} \right) \\
&= \left[\frac{l^r}{b_s} \quad \frac{w^r}{b_s} \quad \theta^r \quad \frac{l^n}{b_s} \quad \frac{w^n}{b_s} \quad \theta^n \quad \frac{h}{b_s} \quad v^x \quad v^y \right]^T.
\end{aligned} \tag{4.3}$$

Matching at scale s between a scale-normalized $\bar{A}_i$ and an unscaled A_j is then computed by using $f(\bar{A}_i, s)$ and evaluating $D(f(\bar{A}_i, s), A_j)$ with equation (4.2).

Figure 4.5: Example tokens extracted from within the bounding boxes of Weizmann multi-scale horse training images [40]. Green ‘×’ denote token centroids, and yellow ‘+’ depict object centroids. Line segments are shown in blue and ellipses in magenta. As observed from these bounding boxes, the generic nature of the shape primitives enables the tokens to represent very diverse shape structures. Additionally, tokens modeling same parts of different object instances (as highlighted in the first three columns) have similar appearances and positions, and so we apply a clustering procedure to identify tokens that are consistent with the object model.

4.3 Category-specific Codebook of Shape Tokens

In this section, we learn a codebook⁸ of representative shape tokens of the target object category, before selecting codewords into a discriminative combination (see ahead to Section 4.4). An advantage of codebook representation is that rather than comparing a test token to every training token, comparison with a subset of training tokens (*i.e.* codewords) suffices. This reduces computation and storage demands. To learn this codebook, we first extract an initial set of shape tokens from within the bounding boxes bb of all training objects for the target category. We normalize the appearance descriptors of each token by the object scale b_s (defined as the diagonal length of bb). Each token is also parameterized by a vector $\bar{x}$, that is directed from the object centroid (center of bb) to the token centroid and is normalized by b_s .

Figure 4.5 shows example shape tokens extracted from within the bounding boxes of the Weizmann horse training images [40]. As observed, the generic nature

⁸The category-specific codebook which we learned comprises characteristic features of an object category (*e.g.* legs and belly for a horse category, and wheels and engine for a motorbike category). This differs from generic/universal codebook, used in *e.g.* [92, 156, 221], which contains category-neutral features that are obtained from images of *all* categories under consideration.

Figure 4.6: Example tokens extracted from within the bounding boxes of Graz-17 motorbike training images [149]. First three columns show object tokens with similar appearances and positions from the same parts of different motorbikes, while the remaining columns show background tokens.

of the proposed shape primitives clearly affords great flexibility in modeling complex parts of objects (*e.g.* ‘belly’ and ‘leg’). While some tokens from the background (background tokens) have similar appearances to those extracted from an object (object tokens), object tokens from the same parts of different object instances have similar appearances and positions (with respect to the object centroid), as shown in the first three columns in Figure 4.5. Similar observations can also be made from the motorbike training images of the Graz-17 dataset [149] shown in Figure 4.6. Given this, a two-step clustering procedure is performed to sieve out object tokens that agree in both appearances and relative positions, considering that background tokens are unlikely to agree simultaneously in its appearances and positions with object tokens. Instead, background tokens are expected to either lie at the periphery of valid clusters or be members of weakly-populated clusters. On the other hand, a token that is the medoid of a compact cluster is representative of a characteristic part of the object model. In this aspect, the clustering procedure filters background tokens and finds characteristic object tokens by pooling evidence from all training tokens together. The identified medoids form a set of candidate codewords. We score each candidate by virtue of its ability to represent a set of stable and spatially coherent tokens having similar appearances. Finally, to reduce redundancy in the codebook, a radial ranking method is developed to select the best set of candidates which covers a large spatial extent of the object model as

codewords.

4.3.1 Clustering by Appearance

In the first step of the clustering procedure, we find clusters in which the shape tokens in each cluster have similar appearances. Many clustering techniques are available for this purpose. One of the more robust clustering techniques is the agglomerative clustering which was used in Liebe *et al.* [124] and Opelt *et al.* [149]. However, agglomerative clustering is computationally demanding, requiring distance computations between every pair of features. Given that there are on average more than 2000 tokens extracted per training image, agglomerative clustering is impractical for our work. While one can adapt the reciprocal nearest pair algorithm for efficient agglomerative clustering as in [124], this restricts the choices of clustering criteria and distance measures that can be used. Alternatively, as done in [149], one can examine and rank each feature separately, and consider only higher ranking features for agglomerative clustering. A disadvantage of such an approach is that it ignores the whole distribution of the features, and instead identifies codewords based on a restrictive and reduced set of features. Correspondingly, since low ranking features are eliminated from future consideration, the quality of agglomerative clustering is directly dependent on the ranking method. This intertwine of the ranking method and agglomerative clustering mitigates the initial motivation for using agglomerative clustering.

Another popular clustering technique is the k -medoid clustering used in [91, 92, 152, 222, 223]. Compared to agglomerative clustering, it has significantly lower computational complexity. In particular, it finds clusters by optimizing a square-error criterion with a greedy hill-climbing strategy. This allows it to deal effortlessly with many features even if each feature is represented by a high dimensional descriptor. However, k -medoid clustering requires the user to specify the number of clusters in advance. When this value is set too low, clusters formed often have high sum of square errors and hence its medoids do not efficiently represent its members. On the other hand, if this value is set too high, clusters identified are unnecessarily fragmented. This value is unknown apriori and varies between different datasets.

In our work, we adapt the bisecting k -medoid method [224] to cluster training tokens. In the original form, bisecting k -medoid splits the initial set of data points into two clusters, select one of the cluster to split and continue splitting clusters until the desired number of clusters (user-defined input) is produced. Our adaption of bisecting k -medoid method retains the computational simplicity of k -medoid clustering, but does not require the number of clusters to be pre-specified. Instead, its input is an intra-cluster dissimilarity threshold th (akin to agglomerative clustering threshold) which can be easily set. Specifically, we apply 2-medoid clustering to similar-typed tokens (*i.e.* line-line, ellipse-line and ellipse-ellipse tokens are separately clustered), where distances between tokens are computed using its scale-normalized appearance descriptors with equation (4.2). For each cluster, we evaluate its intra-cluster appearance dissimilarity value by the average appearance distance between the medoid and its members. This gives a rough indication of the compactness of a cluster. We repartition a cluster by 2-medoid clustering if its dissimilarity value is more than th (fixed at 20% of the maximum range of $D(\cdot)$ in equation (4.2) for all experiments), but retain those whose dissimilarity values are less than th . This first step clustering procedure terminates when the intra-cluster appearance dissimilarity value for every cluster is less than th . Algorithm 4.1 summarizes this adapted bisecting k -medoid clustering method.

We are aware of other possibilities in the data mining literature which modify k -medoid clustering to avoid specifying the cluster number in advance. For example, we can run k -medoid clustering multiple times with different initial settings of cluster numbers, and select the final clustering from the best run (as in Csurka *et al.* [92]). However, this approach demands substantial computation since k -medoid clustering is applied several times. Other modifications include [225, 226] which continually split clusters whose Bayesian information criterion is substantially high. However, this requires prior knowledge of the density or compactness of the clusters. Additionally, each run in [225] requires twice the computation of k -medoid clustering. These shortcomings reduce their applicability for our case. On the other hand, our adapted bisecting k -medoid clustering does not suffer any of these shortcomings, and visual inspection suggests that it works well enough to find clusters of tokens with similar appearances. We show example clusters that

Algorithm 4.1: Adapted bisecting k -medoid clustering method as described in Section 4.3.1

Input : Sets of line-line tokens Φ , ellipse-line tokens Ψ and ellipse-ellipse tokens Ω and their respective clustering threshold w , x and y .

Output : Sets of line-line medoids LL , ellipse-line medoids EL , and ellipse-ellipse medoids EE .

• **Appearance Based Clustering Procedure** (Φ, Ψ, Ω)

- 1 Initialize sets $LL = \emptyset$, $EL = \emptyset$ and $EE = \emptyset$;
- 2 $LL = \text{Perform Appearance Based Clustering} (LL, \Phi, w)$;
- 3 $EL = \text{Perform Appearance Based Clustering} (EL, \Psi, x)$;
- 4 $EE = \text{Perform Appearance Based Clustering} (EE, \Omega, y)$;
- 5 Output sets of medoids LL , EL and EE and terminate appearance based clustering procedure;

• **Perform Appearance Based Clustering** (set A , set X , th)

- 1 Apply 2-medoid clustering on appearance descriptors of tokens in set X to form two clusters X_1 and X_2 whose medoids are m_1 and m_2 respectively;
 - 2 $d_1 = \text{Average appearance distance between } m_1 \text{ and its members of cluster } X_1$;
 - 3 $d_2 = \text{Average appearance distance between } m_2 \text{ and its members of cluster } X_2$;
 - 4 **if** $d_1 > th$ **then**
 - 5 $A = \text{Perform Appearance Based Clustering} (A, X_1, th)$;
 - 6 **else**
 - 7 $A = A \cup m_1$;
 - 8 **end**
 - 9 **if** $d_2 > th$ **then**
 - 10 $A = \text{Perform Appearance Based Clustering} (A, X_2, th)$;
 - 11 **else**
 - 12 $A = A \cup m_2$;
 - 13 **end**
 - 14 **return** set A ;
-

are composed by the adapted bisecting k -medoid clustering method for the Weizmann horse and the Graz-17 motorbike datasets in Figures 4.7(a)-(c) and Figures 4.7(d)-(f) respectively. The middle of each of these figures shows a bounding box having the same aspect ratio as that of the average training bounding box of the object category. Votes on vectors $\bar{x}$ from all members of each cluster are shown as small green ‘ $\times$ ’ around the object centroid (yellow ‘+’) in the bounding box. In addition, we also show example member tokens of each cluster around each bounding box. As observed, even though this clustering step considers only the appearances of tokens, tokens from the same parts of different objects have clustered together

(*e.g.* ‘belly-leg’ of a horse in Figure 4.7(a), and ‘wheel-engine’ of a motorbike in Figure 4.7(d)). More importantly, as shown in each bounding box, votes for $\bar{x}$ by these tokens congregate in the image space forming sub-clusters, while background tokens (and object tokens from other unrelated parts of objects) have randomly scattered votes. For example, the cluster in Figure 4.7(a) has resulted in a single mode in the image space that is formed by tokens modeling the ‘belly-leg’ of a horse, as indicated by the solid bold lines. Additionally, a ‘leg’ cluster has resulted in two modes modeling the front and back legs of a horse (shown respectively by the solid and dashed lines in Figure 4.7(b)). Similarly, tokens which model the ‘tyre-rim’ of a motorbike concentrate in two distinct modes in the image space as shown in Figure 4.7(e), and represent the front and back wheels of a motorbike. Also, tokens modeling the ‘tyre-exhaust’ of a motorbike have resulted in a single mode as shown in Figure 4.7(f). On the other hand, as shown by the red dotted lines in Figures 4.7(a)-(f), background tokens, as well as tokens which are extracted from other unrelated parts of objects, vote randomly in the image space with no meaningful pattern.

4.3.2 Clustering by Relative Positions

A second clustering step is applied separately to each of the above clusters on vectors $\bar{x}$ of its members to find sub-clusters. Tokens in each sub-cluster therefore have similar appearances and are also located at similar positions relative to the object centroid. For this purpose, we employ the very robust mean shift mode estimation algorithm [150, 227]. Although, mean-shift has high computation requirement (*e.g.* for a seven-dimensional feature, a fine-to-coarse hierarchal approach has to be introduced for video sequence segmentation in [228]), for our purpose, feature $\bar{x}$ is two-dimensional and we apply mean-shift separately to each of the above clusters. Given that mean-shift is applied to a reduced set of low dimensional features, sub-clusters can thus be readily found.

We identify the medoid (based on the appearance distance measure) in each mean-shift sub-cluster as a candidate codeword φ , and associate it with an appearance distance allowance τ . This allowance indicates the range of appearance the

CHAPTER 4. CONTOUR BASED OBJECT RECOGNITION

Figure 4.7: Results of clustering by appearance. Each figure corresponds to a cluster, where Figures 4.7(a)-(c) are from the Weizmann horse dataset, and Figures 4.7(d)-(f) from the Graz-17 motorbike datasets. See texts for discussions.

candidate represents, and is computed as the mean appearance distance between the candidate and its sub-cluster members plus one standard deviation. Each candidate is also parameterized with a scale-normalized circular window specifying where the candidate is expected to be found relative to an object centroid. We

CHAPTER 4. CONTOUR BASED OBJECT RECOGNITION

Figure 4.8: Example mean-shift sub-clusters from Weizmann horse dataset. Each row is for one mean-shift sub-cluster. Candidate codewords are shown in the first column. Votes by its sub-cluster members for the candidate's position are shown as small green 'x' around an object centroid (yellow '+') in the second column. The second column also shows the circular window of the candidate codeword, where black '+' depicts window center. Example members of each sub-cluster are shown in the last three columns, where an image from these three columns may contribute a few members to the sub-cluster (*e.g.* last column of third sub-cluster shows two members). Note that members in each sub-cluster have similar appearances and are also located in similar positions relative to an object centroid.

compute the relative center of this window, $\bar{c}$, as the mean of vectors $\bar{x}$ of the sub-cluster members, and its radius, $\bar{r}$, as the mean Euclidean distance between $\bar{c}$ to $\bar{x}$ of each sub-cluster member plus one standard deviation.

Figures 4.8 and 4.9 show examples of the resulting mean-shift sub-clusters for

CHAPTER 4. CONTOUR BASED OBJECT RECOGNITION

Figure 4.9: Example mean-shift sub-clusters from the Graz-17 motorbike dataset. Layout in this figure follows that for the Weizmann horse dataset in Figure 4.8.

the Weizmann horse and Graz-17 motorbike datasets respectively. Each row in these figures is for one mean-shift sub-cluster, in which we show the candidate codeword of each sub-cluster in the first column. Votes by its sub-cluster members for the expected position of the candidate codeword are depicted in the second column as small green 'x' around an object centroid (yellow '+'). Additionally, we represent the circular window of each candidate codeword with a green circle in the second column, and depict the center of this window by a black '+'. Note that some small green 'x' exist outside the circular window, since the radius of the window is computed as the mean distance between the window center and the sub-cluster members plus one standard deviation, as discussed earlier. Example members of each sub-cluster are shown in the remaining three columns, where an image among these three columns may contribute a few members to the sub-cluster. For example, the last image in the third sub-cluster of Figure 4.8 contributes two

members to the sub-cluster. As observed, each sub-cluster represents a set of spatially coherent tokens having similar appearances. These candidates model not only the boundaries or interior silhouettes of objects, but also those discriminative contours of the background *e.g.* shadows cast by horse and motorbike shown in the bottom rows of Figure 4.8 and Figure 4.9 respectively.

We highlight that our choice for the order of the two-step clustering procedure, where we first cluster shape tokens by its appearance descriptors and then by its positional vectors, is motivated by efficiency considerations. Specifically, while mean-shift is highly robust, it demands the search for the closest neighbors of a feature in the space and is therefore computationally expensive. The number of operations in each application of mean-shift clustering can be sharply reduced by applying mean-shift on low dimension features, and on a small set of features⁹. We attain the former by evaluating mean-shift clustering on positional vector which is two-dimensional rather than on a nine-dimensional appearance descriptor. The latter is achieved by the order of our two-step clustering procedure, where we adapt the bisecting k -medoid technique to first partition the large initial set of tokens into a number of clusters, before applying mean-shift separately on each of the resulting clusters. Thus, our two-step clustering procedure which couples the inexpensive bisecting k -medoid method with the powerful but expensive mean-shift clustering technique yields computation efficiency.

4.3.3 Selecting Candidate Codewords into Codebook

We show in Figure 4.10(a) all (30K) candidate codewords that are obtained from an initial set of 330K shape tokens for the Weizmann horse dataset, in which a candidate codeword is shown darker if it is obtained from a more populated mean-shift sub-cluster. Similarly, Figure 4.11(a) shows all candidate codewords for the Graz-17 motorbike dataset. Note that there are many candidates corresponding to the background in these figures since candidates from every (even those very weakly

⁹For even greater efficiency during mean-shift clustering, we can utilize a kD -tree to find the nearest neighbors of a feature. The computation complexity for constructing a kD -tree is $O(n \log n)$, and for querying the tree is $O(n^{\frac{d-1}{d}})$, where n is the number of features to be searched and d the dimension of a feature. Given that mean-shift is applied to two-dimensional features, the search for closest neighbor therefore takes sub-linear time in the order of $O(n^{\frac{1}{2}})$. This can lead to substantial speedup during mean-shift clustering, though we leave this for future endeavor.

Figure 4.10: Selection of candidates into codebook for the Weizmann horse dataset. (a) Candidates codewords from all mean-shift sub-clusters, where candidates from more populated sub-clusters (*i.e.* sub-clusters with more members) are shown in darker gray. (b) Candidates from the most populated 350 sub-clusters. (c) 600 candidates collected by our radial ranking method, where candidates with higher scores are depicted in darker gray. (d) 350 candidates selected by our method into the codebook. Both this sub-figure and that in (b) are obtained using the same number of codewords. Observe that fewer codewords from the background are picked using our method in (d) as compared to that based on cluster size in (b). More importantly, selected codewords by our radial method corresponds to salient parts of the horse *e.g.* crest, front and back legs, and collectively represent a well-define horse model.

populated) mean-shift sub-cluster are visualized. While one can consider all candidates of each category as codewords, this results in an unnecessary large codebook where many entries may not correspond to the object model. To reduce redundancy in the codebook, our aim here is to select a subset of high quality candidates which are representative of the object category as codewords. A simple heuristic to select candidates based on cluster size can inadvertently pick non-salient candidates. For example, candidates from the 350 most populated mean-shift sub-clusters for the horse dataset are shown in Figure 4.10(b). It can be observed in the figure that many candidates are identified from the background, with little or no candidates representing the neck and the crest of a horse model. Similar observations can be made for that of the motorbike dataset in Figure 4.11(b), where the seat and rear of a motorbike model are weakly represented. In this work, rather than identifying codewords by the cluster size, we first compute a score for each candidate codeword based on its appearance and geometric qualities, and then employ a radial ranking method to select the best scoring candidates as codewords. Figure 4.12 gives a structural diagram which depicts the four steps to select codewords.

We first extract the medoids of all mean-shift sub-clusters that are obtained from Section 4.3.2 as candidate codewords. Next, we score each candidate as

Figure 4.11: Selecting candidates into codebook for the Graz-17 motorbike dataset, where the layout in this figure follows that for the Weizmann horse dataset in Figure 4.10. Note that codewords selected for the seat and the rear of a motorbike are better defined by our method shown in (d) as compared with that based on cluster size in (b). Additionally, discriminative contours from exhaust pipes and shadows cast by motorbike are better represented in (d), as compared to (b).

Figure 4.12: Structural diagram depicting the four steps employed to select codewords from the medoids of all mean-shift sub-clusters.

a product of (i) its intra-cluster appearance similarity value, (ii) the number of unique training bounding boxes the members of its sub-cluster are extracted from, (iii) the average strength of edge pixels that are represented by its members, and (iv) its value of $\frac{1}{r}$. We compute the intra-cluster appearance similarity value (first term in the product) as $d - \tau$, where d is the maximum range of appearance distance for the type of candidate currently considered, and τ is the appearance distance allowance (as explained before). Note that the first two terms in the product seek candidates that have distinctive appearance and are flexible enough to accommodate intra-class variations, while the third term identifies candidates whose members are well-defined and prominent in the training images. The last term seeks candidates that estimate a stable and precise location for its members, and measures the geometric quality of the candidates (unlike the first three terms which measure its appearance quality).

To ensure selected candidates represent a large spatial extent of the object model, as opposed to a localized portion, we select high scoring candidates by a

Figure 4.13: Radial method for collecting candidate codewords for the Weizmann horse dataset. Top row shows four example sectors that are defined between a pair of rays emanated from an object centroid (yellow ‘+’). Green ‘x’ within each sector depict the centroid positions of the top scoring t candidates collected from the sector. Candidate codewords collected in each sector are visualized in the bottom row, where they represent (from left to right) the head, crest, back-leg and front-leg of a horse model.

radial ranking scheme. For this purpose, we emanate a pair of rays from the object centroid to delineate a sector. Candidates within each sector are identified and we collect the top scoring t candidates of each sector. We illustrate this radial method for collecting candidates of the horse dataset in Figure 4.13, where the top row shows four example sectors with centroid positions of the t candidates in each sector represented as small green $\times$, and the bottom row visualizes the t candidates. Observe that spatial layout of each sector has selected salient shape structures corresponding to head, crest, belly, back and front legs of a horse. Similarly, Figure 4.14 shows example sectors for the motorbike dataset, where the collected candidates correspond to salient structures of a motorbike such as wheels, seat and engine. For all experiments, we divide the object into 30 non-overlapping sectors and fix t to be 20. Finally, instead of using all collected candidates as codewords, we retain the 350 highest scoring candidates as codewords. This provides robustness against collecting “poor” candidates when a sector has substantial overlap with the background.

Figures 4.10(c) and 4.11(c) show the collected 600 candidate codewords (by harnessing the top 20 scoring candidates from each of 30 sectors) for the horse and motorbike datasets respectively. Candidates which attain higher scores are shown darker. While there are some candidates from the background that have been collected, particularly for the horse dataset shown in Figure 4.10(c), most of

Figure 4.14: Radial method for collecting candidate codewords for the Graz-17 motorbike dataset. Layout in this figure follows that for the horse dataset in Figure 4.13. Four example sectors whose candidates correspond to wheels, seat and engine of a motorbike model are collected as shown.

the collected background candidates have been evaluated to have a lower score as compared to those from the object silhouette. We show the final 350 candidates selected into the codebook for the horse dataset in Figure 4.10(d). Note that both this visualization and that based on cluster size in Figure 4.10(b) are obtained by using 350 codewords. Compared to the Figure 4.10(b), Figure 4.10(d) shows substantially less codewords corresponding to the background. In addition, a large spatial extent of a horse model comprising salient shape structures *e.g.* head, neck, legs (both front and back), belly, back and rear-end are accounted for, with typically several codewords representing different poses of a same object part. This provides tolerance towards intra-class variations, small pose changes and partial occlusion. The better selection of codewords is also evident for the motorbike dataset where, as shown in Figure 4.11(d), our method has selected codewords that better define the seat and rear of a motorbike model. Additionally, discriminative contours corresponding to exhaust pipes and shadows cast by a motorbike are better represented in Figure 4.11(d), as opposed to that based on cluster size in Figure 4.11(b). We show shape token codewords selected for other categories of the Graz-17 dataset in Figure 4.15 and in the bottom row of Figure 5.2. As observed, most salient shape structures, *e.g.* the front wheel of a bicycle and the handle of a mug (shown respectively as the second and fifth images of Figure 4.15) are represented.

Our approach to construct a codebook shares some similarities to Shotton *et al.* [91]. Both cluster features based on its appearance and then identify mean-shift sub-clusters whose members agree in positions, but there are important differences. We cluster tokens formed by pairs of generic shape based primitives of line segments

Figure 4.15: Shape token codewords that are selected for five Graz-17 object categories. From left to right: car-front, bike-front, bottle, cow-side, and mug.

and ellipses, while they cluster contour fragments. They used k -medoid clustering to partition features based on its appearance where the input for the number of clusters is arbitrary fixed. In contrast, we adapt the bisecting k -medoid clustering, which only require the desired intra-cluster appearance dissimilarity value as input, to find good quality clusters. Finally, they select medoids from each mean-shift sub-cluster based on a simple heuristic of cluster size, where a medoid is a codeword if it is from a sub-cluster with more than five members. On the other hand, we couple the appearance and geometric qualities of a candidate codeword, with a radial ranking method to construct a category-specific codebook.

4.4 Discriminative Codeword Combinations

In the previous section, we have described a method to learn a category-specific codebook of shape tokens. Each codeword carries with it both appearance and geometric information, and can be used to detect objects across space and scale. There is now a choice to the extent to which codewords can be combined to improve the reliability of object detections. In the simplest form, one can use a *single* codeword that is matched to a token of a test image to predict the presence of object instances in the test image. Such a scheme was adopted by Seemann *et al.* [229], and Gall and Lempitsky [126] who used a codebook of image patches. However, unlike image patches, contour codewords (as used in this chapter) are not so discriminative and often match in the background of complex images. Instead, a combination of several codewords is more discriminative for the object class and therefore finds fewer false matches. This observation has also been made recently and independently by Opelt *et al.* [149], who restricted a combination to contain

two contour codewords. In contrast to their method, our approach discussed here places no such restriction and allows a combination to contain a variable number of codewords. Specifically, in this section, we exploit the codebook of shape tokens to learn discriminative primitive combinations, where each combination comprises a subset of codewords. We start by describing how a combination of different codewords (referred to as codeword combination) is matched in an image. Next, we present a linear complexity algorithm to find the exhaustive set of codeword combinations that are matched in training (both positive and negative) images. Finally, we capitalize on distinguishing appearance, geometric and structural constraints of a category to learn discriminative codeword combinations. Each discriminative combination has a variable number of x codewords, and is termed in the following as a x -codeword-combination or xCC .

4.4.1 Matching a Codeword Combination in an Image

A codeword combination is matched at scale $\hat{s}$ in an image I if (i) appearance distance at scale $\hat{s}$ between each codeword in that combination and a shape token of image I is within the appearance distance allowance τ of the codeword (appearance constraint), and (ii) centroid predictions by all codewords in the combination concur (geometric constraint). A codeword which satisfies the appearance constraint with a token that is located at position $\mathbf{x}$ in image I will predict an object centroid by a circular window with center $\hat{\mathbf{x}} = \mathbf{x} - \hat{s} \times \bar{\mathbf{c}}$ and radius $\hat{s} \times \bar{r}$, where $\bar{\mathbf{c}}$ and $\bar{r}$ are defined previously in Section 4.3.2. These centroid predictions concur if there is a common overlapping region among their windows.

Figure 4.16 illustrates the appearance and geometric constraints for matching a combination of four codewords, where codewords, matched tokens and centroid predictions are shown color-coded. A combination comprising four codewords is depicted in Figure 4.16(a). Using a same scale $\hat{s}$, tokens from two positive images and one negative image which satisfy the appearance constraints with the codewords are shown in Figure 4.16(b), where colored ‘ $\times$ ’ denote token centroids. Centroid windows predicted by these codewords are shown in Fig 4.16(c), and colored arrows denote scaled vectors $-\hat{s} \times \bar{\mathbf{c}}$. These windows share a common region

Figure 4.16: Matching a combination of four codewords in positive and negative images. (a) A combination of four codewords from the Weizmann horse dataset. (b) Shape tokens from positive (top two rows) and negative (bottom row) images which satisfy the appearance constraints with the codewords at a same hypothesized scale are shown in color, where color 'x' denote token centroids. (c) Centroid predictions by the four codewords as depicted by circular windows. Codewords, tokens and centroid predictions are shown color-coded. This codeword combination is matched in the two positive images since both appearance and geometric matching constraints are satisfied, but not in the negative image (as centroid predictions do not concur).

Figure 4.17: Illustration for finding matched codeword combinations. See text for details.

in both positive images but not in the negative image, and hence this codeword combination is matched only in the positive images.

4.4.2 Finding Codeword Combinations Matched in Training Images

We learn discriminative xCC for the target object category (described later in Section 4.4.3) by harnessing the set of codeword combinations that are matched in training (both positive and negative) images. Here, we propose an efficient method to find the set of matched codeword combinations, where the computational complexity of our method is not only independent in the number of codewords in a combination, but is also only linear in the codebook size. The following theorem states the basic idea of this method,

Theorem 1. *For a scale $\hat{s}$ and location $\hat{x}$ in image I , all codewords of a target object category which satisfy the appearance matching constraint with at least one token located within its estimated window of center $\hat{x} + \hat{s} \times \bar{c}$ and radius $\hat{s} \times \bar{r}$ in I also satisfy the geometric matching constraint.*

Proof. Consider Figure 4.17. Let $\hat{x} = (0, 0)$ be the origin in image I , where we depict this origin as a yellow ‘+’ in the figure. For scale $\hat{s}$ and location $\hat{x}$ in I , let the scaled vector $\hat{s} \times \bar{c}$ of a codeword be $[a \ b]^T$. Thus, this codeword estimates a window of center (a, b) and radius $\hat{s} \times \bar{r}$. Suppose a token in I satisfies the appearance matching constraint with this codeword, and is located at position $x = (a + dx, b + dy)$ in this window. We depict the position of this token as a small green ‘×’ in the figure. Then, it follows that

$$\sqrt{dx^2 + dy^2} \leq \hat{s} \times \bar{r}. \quad (4.4)$$

From Section 4.4.1, this codeword also predicts an object centroid in I by a window of center

$$\begin{aligned} x - \hat{s} \times \bar{c} &= (a + dx, b + dy) - \hat{s} \times \bar{c} \\ &= (a + dx, b + dy) - (a, b) \\ &= (dx, dy) \end{aligned}$$

and radius $\hat{s} \times \bar{r}$. From equation (4.4), the point $\hat{x} = (0, 0)$ is within this window.

Consequently, at a same scale $\hat{s}$ and location $\hat{x}$ in I , all codewords which satisfy the appearance matching constraints with a token located in its estimated windows will predict centroids that are guaranteed to contain a common point $\hat{x}$, and thus also satisfy the geometric matching constraint. $\square$

For a scale $\hat{s}$ and location $\hat{x}$ in image I , we use a numerical value $\mathfrak{R}_i(\hat{s}, \hat{x})$ to indicate if a codeword γ_i finds matching tokens which satisfy the appearance matching constraint and which are also located within its estimated window. Let A' be the appearance descriptor of a token t' located at position x' in I . The token which best matches codeword γ_i is defined as one whose appearance is most similar to γ_i (and within appearance distance allowance τ_i of the codeword), and whose position is closest to its expected position (and within the estimated window of the codeword),

$$t^* = \arg \min_{t'} (d_{app}(\gamma_i, t') + d_{geo}(\gamma_i, t')), \quad (4.5)$$

where

$$d_{app}(\gamma_i, t') = \begin{cases} \frac{D(f(\overline{A}_i, \hat{s}), A')}{\tau_i} & \text{if } D(f(\overline{A}_i, \hat{s}), A') \leq \tau_i \\ \infty & \text{otherwise} \end{cases}, \quad (4.6)$$

$$d_{geo}(\gamma_i, t') = \begin{cases} \frac{\|\hat{x} + \hat{s} \times \overline{c}_i - x'\|_2}{\hat{s} \times \overline{r}_i} & \text{if } \|\hat{x} + \hat{s} \times \overline{c}_i - x'\|_2 \leq \hat{s} \times \overline{r}_i \\ \infty & \text{otherwise} \end{cases} \quad (4.7)$$

and $\|\cdot\|_2$ is the L_2 norm. $\mathfrak{R}_i(\hat{s}, \hat{x})$ for codeword γ_i at scale $\hat{s}$ and location $\hat{x}$ in an image is then computed as

$$\mathfrak{R}_i(\hat{s}, \hat{x}) = d_{app}(\gamma_i, t^*) + d_{geo}(\gamma_i, t^*). \quad (4.8)$$

It is easily verified that a codeword which finds a token that satisfies its appearance matching constraint and that is in its estimated window has $\mathfrak{R}(\cdot)$ value in the range $[0, 2]$, where a lower value indicates better matching. In contrast, a non-matching codeword has infinity $\mathfrak{R}(\cdot)$ value. Then, from the above theorem, at $\hat{s}$ and $\hat{x}$ in image I , every combination of codewords whose $\mathfrak{R}(\hat{s}, \hat{x})$ values are less than or equal to 2 are matched in I . By iterating through all scales and locations

across all training images, the exhaustive set of matched codeword combinations can thus be found. The computational complexity of this search is $O(l\sigma N)$, where N is the codebook size, l the number of locations being searched (and is limited by the image size), and σ the number of scales being searched (where the scales to be searched is bounded between the minimum and maximum object scales of the target object category).

One could iterate through every single scale and location at each training image to find matched codeword combinations. This is however unnecessarily inefficient since (for reasonable scales) each codeword estimates a circular window which encompasses an image region and not just a pixel point. Correspondingly, windows which overlap in image space often share a set of common pixels (see example in Figure 4.16(c)). With this insight, improved efficiency of the search algorithm is possible by sampling locations and scales densely enough to discover these codeword combinations, but sparsely enough to avoid unnecessary computation overheads. This is similar in concept to the highly efficient multi-scale sliding-window mechanism. In our experiments, we sample locations at every 15 pixels in the horizontal and vertical directions, and use a number of scales that covers the range of object scales in training images. Note that comparison of tokens to codewords at each scale can be implemented simply by rescaling the appearance descriptors of codewords. It is unnecessary to rescale an image to extract tokens since they are constructed by shape primitives which automatically adapt to image shape structures at different scales.

4.4.3 Learning an Ensemble of Discriminative xCC

We seek a xCC which models discriminative appearance, geometric and structural constraints of a category to reliably predict object locations. The formulation for $\mathfrak{R}(\cdot)$ in equation (4.8) provides a mathematically convenient method to find such a xCC . We take the $\mathfrak{R}(\cdot)$ values of all codewords at every sampled scale and location $(\hat{s}, \hat{x})$ for all training (both positive and negative) images as input. Each $(\hat{s}, \hat{x})$ represents an object hypothesis, and we pair it with a binary label $z = +1$ if a bounding box (with aspect ratio equal to that of the average training bounding box)

centered on the hypothesis localizes an object, and $z = -1$ otherwise. Since there are more negative hypotheses than positive hypotheses, hypotheses are weighted to correct for this imbalance.

Consider first a xCC which comprises two codewords γ_i and γ_j . From Section 4.4.2, the matching of this xCC at $(\hat{s}, \hat{x})$ in an image I can be mathematically represented as

$$\mathfrak{R}_i(\hat{s}, \hat{x}) \leq \vartheta_i \quad \text{and} \quad \mathfrak{R}_j(\hat{s}, \hat{x}) \leq \vartheta_j, \quad (4.9)$$

where ϑ is a threshold in the range $[0, 2]$. Note that since this xCC is matched at the hypothesis, it therefore implicitly models the appearance and geometric configurations of shape tokens at this hypothesis in I . This representation can be further generalized to the following form,

$$p_i \mathfrak{R}_i(\hat{s}, \hat{x}) \leq p_i \vartheta_i \quad \text{and} \quad p_j \mathfrak{R}_j(\hat{s}, \hat{x}) \leq p_j \vartheta_j, \quad (4.10)$$

where p has value $+1$ or -1 to indicate the direction of the inequality, and ϑ is relaxed to take on any real number value. With this representation, structural configurations, *e.g.* XOR relationship between codewords at this hypothesis (matching of γ_i implies γ_j is weakly matched or unmatched¹⁰), can be readily integrated with the appearance and geometric configurations of shape tokens at the same hypothesis. This representation can be modeled by a xCC , with each of its codeword having a p and ϑ values. Additionally, by allowing a xCC to have a variable number of codewords, we can impart greater flexibility to this xCC to model discriminative appearance, geometric and structural constraints of varying complexity. Our aim here is to learn such a discriminative xCC (*i.e.* codewords in the xCC , and the p and ϑ values of each codeword in the xCC) which can reliably predict the presence/absence of an object instance at a hypothesis $(\hat{s}, \hat{x})$.

We exploit a binary decision tree [74] to find such a discriminative xCC . For our purpose, the input to learn a decision tree is a $N \times M$ training matrix, a $1 \times M$ binary label vector and a $1 \times M$ weight vector, where N is the number of codewords and M the number of object hypotheses. An entry (i, j) in the training matrix is

¹⁰Such a XOR structural constraint can be represented by the following form: $\mathfrak{R}_i(\hat{s}, \hat{x}) \leq \vartheta_i$ and $\mathfrak{R}_j(\hat{s}, \hat{x}) > \vartheta_j$. Here, p_i has value $+1$ and p_j -1 .

the $\Re(\cdot)$ value of the i^{th} codeword at the j^{th} object hypothesis, and each entry of the label vector has value +1 or -1 to indicate the presence or absence of an object at a hypothesis with the weight of the hypothesis given in its corresponding entry of the weight vector. Given these training inputs, a decision tree which maximizes accuracy on the weighted training hypotheses can be easily learned in a greedy fashion by [74] in which it is easily shown that a path from a root node to any leaf node in the learned decision tree encounters between 1 to k non-leaf nodes, where k is the number of splits in the tree. Each non-leaf node is a predicate of the form $p \Re(\hat{s}, \hat{x}) \leq p \vartheta$, and the predicates in a path can be combined into the mathematical representation of equation (4.10). The leaf node in each path gives the predicted label of a hypothesis which satisfies all predicates in the path. Each path (*i.e.* the number of non-leaf nodes and their predicates, and the predicted label at the leaf node) automatically adapts to an object category, and is discriminatively learned to predict the presence/absence of an object at an object hypothesis. Hence, by learning a binary tree with k splits, discriminative xCC can be found by simple path transversal from a root node to each leaf node, where each xCC has a variable number of between 1 to k codewords, and a hypothesis is assigned a predicted label by exactly one xCC of the tree. The prediction of a xCC at hypothesis $(\hat{s}, \hat{x})$ is then,

$$xCC[\hat{s}, \hat{x}] = \begin{cases} \ell & \text{if } p_i \Re_i(\hat{s}, \hat{x}) \leq p_i \vartheta_i \text{ and } p_j \Re_j(\hat{s}, \hat{x}) \leq p_j \vartheta_j \text{ and } \dots \\ 0 & \text{otherwise} \end{cases}, \quad (4.11)$$

where $\ell \in [-1, +1]$ is the predicted label given by the leaf node in the path, and prediction of 0 implies that the xCC does not vote for or against an object presence at hypothesis $(\hat{s}, \hat{x})$.

We learn an ensemble of discriminative xCC by AdaBoost, where every boosting round outputs a decision tree with k splits. This method is presented in Algorithm 4.2. At each boosting round $t = 1, \dots, T$, we increase the weights of training hypotheses which are wrongly predicted by the set of xCC extracted from the current tree. This encourages the learning (in subsequent boosting rounds) of discriminative xCC which correctly predicts these hypotheses. In this aspect, the

Algorithm 4.2: Boosting an ensemble of discriminative xCC .

Input : A set of training object hypotheses $\{(\hat{s}_i, \hat{x}_i)\}$, the $N \times M$ training matrix with its corresponding binary label vector $\{z_i\}$ and weight vector $\{w_i\}$, and pre-defined values for k (number of splits in a tree) and T (boosting rounds).

Output : An ensemble H of discriminative xCC each with its associated weight.

- 1 Initialize hypotheses weights as $w_i = \frac{1}{2r}$ and $\frac{1}{2s}$ for $z_i = +1$ and -1 respectively, where r is the number of positive hypotheses and s the number of negative hypotheses. Initialize ensemble H to be empty;
- 2 **for** $t = 1$ **to** T **do**
- 3 Normalize hypotheses weights as $w_i = \frac{w_i}{\sum w_i}$ so that $\{w_i\}$ is a probability distribution;
- 4 Learn a binary decision tree [74] with k splits which maximizes accuracy on the weighted hypotheses according to its binary labels;
- 5 Compute training error of the current decision tree on the training hypotheses as $e_t = \sum w_i b_i$, where $b_i = 0$ if hypothesis $(\hat{s}_i, \hat{x}_i)$ is correctly predicted and $b_i = 1$ otherwise;
- 6 **if** $e_t > 0.5$ **or** $e_t = 0$ **then**
- 7 Return the ensemble H of discriminative xCC each with its associated weight;
- 8 **end**
- 9 Compute α_t as $\alpha_t = \frac{1}{2} \ln \left(\frac{1-e_t}{e_t} \right)$;
- 10 Update training hypotheses weights as $w_i = w_i \exp(\alpha_t b_i)$;
- 11 **foreach** j^{th} path from root node to each leaf node of the current tree **do**
- 12 Collect the predicates in non-leaf nodes and the predicted label in leaf node as a discriminative xCC_j ;
- 13 Augment ensemble H with this learned xCC_j and associate it with a weight α_t ;
- 14 **end**
- 15 **end**
- 16 Return the ensemble H of discriminative xCC each with its associated weight;

boosting algorithm homes in on the difficult and informative hypotheses to learn an ensemble of xCC . Line 6 of the algorithm is necessary since a decision tree is required to perform better than chance, *i.e.* e_t must be less than 0.5, to ensure the training error converges¹¹. Additionally, when $e_t = 0$, it implies either the perfect prediction of all training hypotheses (*i.e.* b_i of all hypotheses are 0), or the negligible significance of wrongly predicted hypotheses (*i.e.* w_i of all wrongly predicted hypotheses are 0). In both cases, no further useful information can be learned from the training hypotheses, and hence we terminate the boosting

¹¹Note that for binary prediction between two classes (as done here for distinguishing positive from negative object hypotheses), e_t is at worst 0.5. This can be understood by considering that a decision tree whose error is e_t can be converted to one whose error is $1 - e_t$ by simply flipping the predicted labels at all its leaf nodes *e.g.* a leaf node whose predicted label is $+1$ is flipped to be -1 . Consequently, since a decision tree classifier outputs a tree which minimizes the error, e_t will therefore be always less than 0.5 in our work. In this aspect, condition $e_t > 0.5$ will never be satisfied and hence can be excluded from line 6 of Algorithm 4.2.

algorithm.

Figures 4.18 and 4.19 show the first few xCC that are learned automatically for the horse and motorbike datasets respectively. Each row in these figures is for one xCC . The first column visualizes xCC in which codewords in a xCC are shown in separate colors. We denote the estimated windows of each codeword (*i.e.* where it is expected to find matching shape tokens) by colored circles. A green-colored circle depicts a requirement that the codeword must find matching tokens and corresponds to $p = +1$ of predicate $p \Re(\hat{s}, \hat{x}) \leq p \vartheta$. Conversely, a red-colored circle illustrates the requirement for the codeword to be unmatched (or be weakly matched) and corresponds to $p = -1$. The remaining three columns show positive hypotheses $(\hat{s}, \hat{x})$ in training images which satisfy the appearance, geometric and structural constraints modeled by the xCC . We indicate the location $\hat{x}$ of a hypothesis by a blue ‘+’ and represent its scale $\hat{s}$ by the diameter of a blue-outlined circle in the training image. Codewords of the xCC and its matching tokens in training images are shown by the same color. All xCC learned by the boosting algorithm are then combined to form an ensemble, whose detection confidence on an object hypothesis is

$$H[\hat{s}, \hat{x}] = \sum_{j=1} \alpha_j \times xCC_j[\hat{s}, \hat{x}]. \quad (4.12)$$

For clarity of presentation, the xCC shown in Figures 4.18 and 4.19 are obtained with decision trees having 3 splits (*i.e.* $k = 3$), and hence each xCC has between 1 to 3 codewords. However, rather than fixing a same split number k for all object categories, in our experiments, we learn a k value separately for each category by m -fold cross validation over values 1 to 10 on the training hypotheses. Such an approach to find parameter value that is customized for an object category has been used previously by [91, 230–233], and render the ensemble to be less prone to overfitting. After finding the optimal k value, we learn an ensemble of discriminative xCC from the entire training hypotheses, and use this ensemble to detect objects in unseen test images. In all experiments, we fix m to be 3.

Figure 4.18: Learning discriminative xCC for the Weizmann horse dataset. First column visualizes the first few xCC that are learned automatically by boosting, and the remaining columns show hypotheses in training images which satisfy the appearance, geometric and structural constraints modeled by the xCC . For clarity of presentation, we show xCC that are learned by decision trees with 3 splits (*i.e.* $k = 3$); in practice we find the optimal split number for each object category by 3-fold cross validation over values 1 to 10 on the training hypotheses, and use this split number to learn the xCC .

4.5 Object Detection and Image Classification

We detect objects in a test image by a simple sliding window technique, which has been shown to be successful for object detection [91, 156]. Specifically, we slide a window (with aspect ratio equal to that of the average training bounding box of the targeted object category) across a test image at multiple scales. The sliding and scale steps for each object category are the same as that used in training (Section. 4.4.2). Each window corresponds to an object hypothesis at scale-space location $(\hat{s}, \hat{x})$, in which $\hat{s}$ is the diagonal length of the window and $\hat{x}$ its center. We evaluate the detection confidence of each window with the boosted ensemble of xCC by equation (4.12), which gives a 3D confidence map for the spatial location and scale of an object instance in the test image. We consider local maxima as candidate

Figure 4.19: Learning discriminative xCC for the Graz-17 motorbike dataset. Examples shown are for the first few xCC for $k = 3$. Layout in this figure follows that for the Weizmann horse dataset in Figure 4.18.

object detections and, to avoid multiple detections of a same object instance, apply a post-filtering step to remove candidate detections whose windows overlap with a stronger candidate. The retained detections yield the final set of detections of the test image. For image classification, we use the confidence of the strongest (*i.e.* most confident) detection as the classification confidence of the test image.

4.6 Experimental Evaluation

We present extensive evaluations on existing datasets, which comprise 17 diverse and challenging object categories, for a total of more than 2400 test images. Performance of our technique is evaluated by its object detection and image classification accuracy, and we compare against recent state-of-the-art contour based methods [91, 149], and other related methods. We describe the evaluation procedure in Section 4.6.1 and introduce the datasets in Section 4.6.2. Experimental results are reported in Section 4.6.3.

4.6.1 Evaluation Procedure

We partition the images of each object category into two disjoint sets: training and testing sets. Objects in both sets are annotated by ground truth bounding boxes. Bounding boxes of training images are used for learning a category-specific codebook and a boosted ensemble of xCC . On the other hand, bounding boxes of test images are used solely for performance evaluation.

Performance is evaluated in terms of the object detection and image classification accuracy. For object detection, we consider a detection to be correct if the intersection area between the predicted bounding box and the ground truth bounding box exceeds 50% of its union (from the VOC Challenge [234]). This adheres to the experimental protocols of other recognition systems that are compared here. In addition, to conform to the evaluation criteria of [234], we consider multiple detections of the same object instance as false positives. We measure detection performance by two closely related curves. The first curve plots recall¹² against 1-precision by varying a global confidence value, where

$$\text{Recall} = \frac{\text{Number of correctly detected positives}}{\text{Total number of positives in testing set}}, \quad (4.13)$$

and

$$1 - \text{Precision} = \frac{\text{Number of falsely detected positives}}{\text{Number of correctly detected positives} + \text{Number of falsely detected positives}}. \quad (4.14)$$

This curve (termed as RP curve) therefore gives the trade-off between correct detections and false detections over a range of confidence values. Quantitative comparisons between RP curves are given by the equal error rate (RP-EER) and the area under curve (RP-AUC). RP-EER corresponds to a single point in the curve

¹²*Recall* is also referred to as *recall rate* in some fields *e.g.* segmentation domain [235]. We have however used the term *recall* in this thesis as it is more commonly used in the object detection domain.

Figure 4.20: Performance evaluation curves of (a) Recall-precision, (b) Recall false positive per image, and (c) Receiver operating curves, with quantitative measures area-under-curve and equal-error-rate depicted in (a) and (c).

i.e. the point at which recall is equal to precision, and reports performance at a single precision value. In contrast, RP-AUC takes the entire curve into account and measures detection performance across all precision levels. Consequently, RP-AUC gives a more representative score for comparing detectors' performance¹³. As such, we report detection performance by RP-AUC, while RP-EER is given only when necessary for comparison with [149]. Figure 4.20(a) shows a RP curve illustrating these two measures. The second curve for evaluating detectors performance plots recall against the average number of false positives per image (RFPPI) [156]. Figure 4.20(b) shows an example of a RFPPI curve. Unlike a RP curve, a RFPPI curve is independent of the number of negative test images. This, together with its monotonic nature, makes interpretation of a RFPPI curve more intuitive and easier. However, a RFPPI curve lacks an overall comparative score and hence we use RP-AUC scores to compare between RFPPI curves.

For image classification, as in the manner of [91, 149], we consider an image to be classified correctly if it contains the object category, and localization by bounding box is not considered. Classification performance is shown by the Receiver Operating Characteristics (ROC) curve (Figure 4.20(c)) which plots the trade-off between the true positive rate and false positive rate, where

¹³See [91, 108, 236] for more detail motivations of using AUC over EER scores.

$$\begin{aligned}
\text{True positive rate} &= \frac{\text{Number of correctly classified positives}}{\text{Total number of positives in testing set}} \\
&= \text{Recall} ,
\end{aligned} \tag{4.15}$$

and

$$\text{False positive rate} = \frac{\text{Number of falsely classified positives}}{\text{Total number of negatives in testing set}} . \tag{4.16}$$

To compare classifiers performances, we take the whole ROC curve into account and report the area under it (ROC-AUC). This gives a more meaningful measure than the equal error rate (ROC-EER), which is reported only for comparison with [149] and other previously published results.

4.6.2 Datasets

Here, we introduce the datasets that are used for evaluating the performance of our technique, and for comparing against other state-of-the-art methods. As closely as possible, we use the same training and testing object images as other methods for comparing performance. Note that while most images in the datasets are given in color, neither color nor texture information is exploited as recognition cues in this chapter. Instead, the only image information used is the edge map extracted by the Canny edge detector [30].

4.6.2.1 Weizmann Multi-scale Horse Dataset [40]

This challenging dataset features 328 images containing one horse each. Horses in these images are of different breeds, textures, colors, and are shown under different lighting conditions. In addition, horses appear in a wide variety of poses (*e.g.* standing, galloping, grazing), and are often embedded in significant background clutter. The horses in these images are of different scales with the largest horse approximately 2.5 times bigger than the smallest horse. While generally seen from the side, moderate out-of-plane rotation of the horses can also be observed. We pair

these images with the background images of the Caltech-256 dataset [216]. These images represent a diverse collection of indoor and outdoor real-world scenes, and as reported in [216] are an improvement over the Caltech-101 background images (which consists of mainly synthetically generated pictures). Many clutter edge pixels are present in these images which pose a formidable challenge to our shape-only detector. We normalize all images to a maximum (either vertical or horizontal) dimension of 320 pixels. The first 100 horse and 100 background images are used for training, and the remaining 228 horse and 228 background images for testing.

4.6.2.2 ETH-80 Dataset [122]

This dataset contains eight different object categories (apple, car, cow, cup, dog, horse, pear, tomato) seen at controlled settings. Each category contains ten different objects. These objects are placed on a turntable in front of an uncluttered homogeneous blue background, and are photographed at equally spaced viewpoint intervals. We partition the first five objects of each category as training instances, and the remaining five as the testing instances for the category. All training and testing images are normalized to a maximum dimension of 320 pixels width or height. We do not report detection RP-AUC or classification ROC-AUC results on this dataset as it is too simple for our purpose, but use it to investigate quantitatively the robustness of our approach to viewpoint changes. More realistic experiments on viewpoint variations in complicated real-world images are given by testing on the Weizmann horse and the Graz-17 (introduced next) datasets.

4.6.2.3 Graz-17 Dataset [149]

This demanding dataset features a collection of 17 object and 1 background categories, in which images for each category have been pre-partitioned into training and testing sets. There are substantial intra-class variations in each category, and many images are severely cluttered with the objects occupying a small portion of the images. Each object category is composed by images containing one or more objects. In addition, objects present in each category are of different scales, in which the scale range is more than 7 times from the smallest object to the largest object

for some categories. As with the previous datasets, each image is normalized to a maximum dimension of 320 pixels. We learn one ensemble for each object category, and test each category separately by evaluating against the category test set and an equal number of background images (where possible since only 216 background images are available in the dataset).

4.6.3 Results

4.6.3.1 Evaluating on Weizmann Multi-scale Horse Dataset [40]

We now report experimental results, beginning by evaluating our boosted ensemble of shape token xCC on the complete Weizmann horse dataset. This and all experiments, unless otherwise stated, are performed with the following parameters settings (details about the parameters can be found throughout the chapter). Edge maps are extracted with Matlab implementation of the Canny edge detector at default settings. We construct line segments and ellipses from edge maps using the original parameter values in [31] and Chapter 3 respectively. The value of λ_ϵ , which is used to construct the appearance descriptors of shape tokens, is fixed as 0.8. We apply the adapted bisecting k -medoid clustering method to shape tokens with intra-class dissimilarity threshold th fixed at 20% of the maximum range of $D(\cdot)$. 20 best candidate codewords are collected in each of 30 sectors spreading radially from an object centroid, and the best 350 candidates are retained as codewords. To learn an ensemble, training hypotheses are taken with sliding step of 15 pixels in the horizontal and vertical directions across 8 training scales. Evaluation uses the same sliding and scale steps. We learn the ensemble of xCC with $T = 300$ boosting rounds, where the optimal number of splits in a tree is selected by $m = 3$ fold cross validation on the training data over values $\{1, 2, \dots, 10\}$.

We report quantitative results by our method in Figure 4.21. Additionally, the figure also includes the performance of the recent state-of-the-art contour based methods of Shotton *et al.* [91], which we termed as *Shotton-I* and *Shotton-II (Retrained-test)*. While *Shotton-I* used the same number of training images as this work, *Shotton-II (Retrained-test)* augmented a set of object hypotheses from the test images to the training data, and retrained their system on the augmented

data. In this aspect, *Shotton-II (Retrained-test)* learned from a larger and more diverse set of training images. We show image classification performances by the ROC curves in Figure 4.21(a), and report ROC-AUC scores obtained by the methods in the legend. The ROC-AUC scores reveal that our method performs better than *Shotton-I*, where we attained higher ROC scores of 0.9500 against 0.9251 obtained by *Shotton-I*. While *Shotton-II* has clearly benefitted from the increase in training data, our method, which does not exploit test images during training, nevertheless demonstrates slightly better classification capability.

We present object detection results by the RFPPI curves in Figure 4.21(b), and report RP-AUC scores in the legend. Object detection is comparatively harder than image classification since a strong but poorly localized bounding box contributes positively to classification but penalizes detection. Nevertheless, detection performance by our method surpasses that of both *Shotton-I* and *Shotton-II (Retrained-test)*, in which we attain RP-AUC scores of 0.9218, while their methods achieved a maximum of 0.8903 RP-AUC score. The RFPPI curves also reveal the superior performance of the proposed approach in which more than 85% of test objects are correctly detected by our ensemble at an extremely low FPPI of 0.067 (around 1 false detection for every 15 images). In contrast, at the same FPPI value, *Shotton-I* and *Shotton-II* manage to detect 78% and 80% of test objects respectively. Our detection result is also superior to that achieved very recently by the contour based method of Bai *et al.* [80], which obtained 0.8032 RP-AUC. While their method used a smaller subset of the training images, this is due to their necessity of obtaining figure-ground segmented training objects. In contrast, since our method needs only bounding boxes, we fully exploit all training images, which likely account for some part of the improved detection performance.

We show examples of correct and incorrect detections by our method on the Weizmann horse test set in Figure 4.22. Bounding boxes on test images represent detections, in which green indicates true positives, red false positives and yellow the ground truth for false negatives (miss). In addition, all codewords which contributed positively to detections are back-projected and shown on the right of test images. These codewords visualize parts of test objects which are useful for detection. We draw several conclusions. First, we observe the excellent performance of

Figure 4.21: Performance on the complete Weizmann horse test set, with comparison to Shotton *et al.* [91]. (a) Classification performance is shown by the ROC curves and quantitatively compared by the ROC-AUC scores given in legend. (b) Detection accuracy shown by RFPPI curves and compared by RP-AUC scores.

the detector where positive objects are accurately localized in scale-space, despite difficult challenges from extensive background clutter, and significant intra-class variability (even of a cartoon in fourth row, third image). Next, turning to incorrect detections, we observe false positives to arise when the pattern of cluttered edges appear very similar to the object model. For example, the curved right leg of the table (fifth row, first image) appears very similar to the rear end of a horse model with regards to image edges. This is confirmed by the back-projected code-words where a pattern which is representative of the back-legs, dock and croup of a horse model is visualized. Consequently, a bounding box centered on the table is falsely detected. Similarly, the outline of the box placed on a grass patch (fifth row, second image) is sufficiently similar to the neck and front legs of a horse model leading to a false detection on the test image. We show in Chapter 5 that one could further improve the discriminative power of this contour only ensemble by combining local appearance patches with object boundaries to obtain better detection results. Finally, false negatives (missed detections) are due either to considerable pose changes or to viewpoint variations which exceed the reasonable tolerance of our detector that is learned for only a single viewpoint of objects. We show an example of a horse which has significant pose change in the first test image of the bottom row. The second test image of the bottom row shows a missed detection in which test object undergoes (slight) viewpoint variation, and detection of the

object is further compounded by patterns of background edges which appear very similar to the object model (*e.g.* trunk of the tree in the background looks similar to the rear end and legs of a horse model, and the back of the test horse has similar appearance to the belly of a horse model in terms of edge pixels). A more extreme example of viewpoint variation is given in the third test image of the bottom row. We note that aspect ratio of ground truth boxes varies considerably as a result of viewpoint (and also pose) changes, where the vertical side is longer than the horizontal side. For accurate localization, we can extend the current multi-scale sliding window technique to four dimensions (spanned by space, aspect ratio, and scale dimensions). The use of a more extensive set of training images, covering different poses and viewpoints of the object category, will also mitigate such false negatives. Additionally, for robust detections across unconstrained viewpoints, one can combine a battery of detectors where each detector is trained for a specific viewpoint of the object, though we leave these improvements for future endeavors. An experimental evaluation of our approach towards viewpoint changes is reported next.

4.6.3.2 Detection under Viewpoint Variations

We now investigate quantitatively the tolerance by our boosted ensemble of xCC to viewpoint changes. Specifically, we learn detector from training objects seen in side-view and apply it to test objects that are rotated about a vertical axis. This experiment is performed on the ETH-80 dataset [122], and we report results on both rigid (car) and articulated (horse) categories. Given a target object category *e.g.* car, we use all (*i.e.* five) training car instances seen at 0° viewing angle as positive instances, and training instances from the remaining categories (*i.e.* apple, cow, cup, dog, horse, pear and tomato), seen also at 0° viewing angle, as negative instances. Both detectors for the car and horse categories are trained using a single object scale (as training instances from each category have similar size), and we keep all other parameter values unchanged. At test time, the side-on detector is applied to positive and negative test instances seen at various vertically rotated viewing angles.

Figure 4.23 shows the detection performance for the car detector. The top

Figure 4.22: Example detections on the Weizmann horse test set. Bounding boxes on test images depict detections where green indicates true positives, red false positives and yellow ground truth for false negatives. Codewords which contributed positively to detections are back-projected and visualized on the right of test images. Note excellent object localization despite significant intra-class variations, considerable background clutter, and articulation changes.

row of Figure 4.23(a) shows an example test car image with its degree of rotation about the vertical axis given in the top left corner. For each viewing angle, ground truth centroids of all test cars coincide at the same point (image center) which we show by yellow ‘+’ in the second row of Figure 4.23(a). Evaluating the car detector over a regular set of sliding windows in a test image gives a detection map (similar to the classifier activation map in [118], classification map in [115], and Hough voting map in [149]) for a car centroid. We smooth each map slightly (with a 15×15 averaging mask) to remove noise, and average the detection maps across all test car images for each viewpoint. The resulting detection maps (green denotes strong detection confidence and red weak confidence, where the transition

CHAPTER 4. CONTOUR BASED OBJECT RECOGNITION

Figure 4.23: Detecting car (rigid object category) under controlled viewpoint changes. (a) Top row: example test car with its rotation about vertical axis given in the top left corner, middle: ground truth centroids of all test cars at each viewpoint are located on the same yellow ‘+’, bottom: detection maps for a car centroid averaged across all test cars for each viewpoint. (b) Average detection confidence of test cars and other non-car instances plotted against viewing angle. See text for discussions.

from green to red indicate weakening detection confidence) are visualized for each viewpoint in the bottom row of Figure 4.23(a). Note that local maxima in the detection maps are well-defined (albeit slightly elongated) at substantial viewing angles, and also closely correspond to the positions of the ground truth centroids. Figure 4.23(b) plots the average detection confidence of test cars (bold green curve) and the average confidence of all non-car instances (bold red curve) across different viewing angles. Additionally, we also plot the average confidence for each individual non-car category (*i.e.* apple, cow, cup ...) in the same graph. As observed, the average car detection confidence (bold green curve) is not only above that of the average of non-car instances (bold red curve), but also above that for every other category across all viewing angles. This demonstrates that the side-on car detector is stable and can reliably discriminate against test categories across a wide range

CHAPTER 4. CONTOUR BASED OBJECT RECOGNITION

Figure 4.24: Detecting horse (articulated object category) under controlled viewpoint changes. Layout in this figure follows that for the ETH-80 car category in Figure 4.23.

of viewpoints.

Figure 4.24 reports the detection performance for the horse detector. As shown in Figure 4.24(a), there is greater smearing of votes for object centroid in the detection maps (though local maxima are still well-defined over a range of viewpoints) as compared to the car detector. This is due to the greater challenge of the articulated horse category, where object instances from the horse category exhibit greater intra-class variations. In particular, we show the average instance of each category in Figure 4.25, where each sub-figure is obtained by first normalizing the pixel values of each object instance to be from 0 to 1, and then averaging across all training (first column) and testing (remaining three columns) instances of a viewpoint. It can be observed that due to changes in object poses and body configurations, there is considerable intra-class variability in the horse category as compared to the car category. In Figure 4.24(b), it can be seen that the average horse detection confidence (bold green curve) is higher than that for non-horse instances (bold red

curve) at up to 68° in both directions. This drops to 22° when compared specifically against the dog and cow categories. This is not surprising since both dog and cow share similar shape boundaries with a horse. On the other hand, categories such as tomato and cup which have vastly different shape boundaries from a horse can be robustly discriminated across all tested viewing angles. An investigation of our method to detect and discriminate between categories sharing similar boundaries in real-world images with cluttered backgrounds is presented in Section 4.6.3.3.

Overall, these results demonstrate our method to have strong robustness in detecting rigid objects across a wide range of viewpoints. For other categories which have substantial viewpoint invariance *e.g.* apple, pear and tomato, detection performance is likely to be better. Performance on articulated categories is considerably weaker. A probable reason is due to the lack of positive training instances (only five horse instances available) coupled with significant variations in shape boundaries. Finally, while the car detector can discriminate against all non-car classes across the tested viewpoints, and the horse detector at up to 22° viewing angle, images tested here do not contain clutter. Consequently, for real images in cluttered background, tolerance to viewpoint variations is likely to be weaker (though viewpoint invariance is still given to a certain extent).

4.6.3.3 Detecting and Discriminating Similar Shaped Object Categories

Here, we investigate the ability of our method to detect and discriminate between categories whose objects share similar shape boundaries. Both cases on rigid, *i.e.* motorbike versus bike-side, and articulated, *i.e.* cow-side versus horse-side, categories are studied, in which these selected categories have many similar shape components such as wheels and handlebars (for the rigid case), and head and belly (for the articulated case). We use training and testing images from the Graz-17 dataset [149]. Parameter values are unchanged, although the number of scales is adjusted against the training data.

For the first case on rigid object categories, we prepare a training set comprising 90 motorbike and 90 bike-side images. A testing set which comprises 53 motorbike and 53 bike-side images is also collected, where these test images are different from the training images. We learn a motorbike detector and a bike-side detector on the

CHAPTER 4. CONTOUR BASED OBJECT RECOGNITION

Figure 4.25: Average object instance for the car (top row) and horse (bottom row) categories at specified viewpoints. First column shows the average training instances, and the remaining columns show average testing instances for selected viewing angles given in the top left corner. Yellow ‘+’ depict object centroids. Notice that object instances from the articulated horse category exhibit greater intra-class variations (due to different body configurations and object poses) than that from the rigid car category.

same 90+90 training set, and evaluate these detectors on the same 53+53 testing set. For the second case on articulated categories, we collect a training set which contains 45 cow-side and 45 horse-side images, and a testing set which contains 65 cow-side and 65 horse-side images. We learn a cow-side detector and a horse-side detector on the same 45+45 training images, and evaluate them on the same 65+65 testing images. Note that when learning and evaluating a detector for a target object category, images from the other category are regarded as background images. For example, to learn a motorbike (target object category) detector for the rigid case, we use the 90 motorbike images of the training set as positive images, and the 90 bike-side training images as background images. This motorbike detector is then tested on the test set, in which the 53 test motorbike images are considered as positive images, and the 53 test bike-side images as background images. Similarly, we consider horse-side images to be background images when learning/evaluating a cow-side detector for the articulated case.

Table 4.1 records the RP-AUC detection results of these detectors. For a target object category *e.g.* motorbike, a detection on a test image can either (i) localize an

object instance (*i.e.* motorbike), (ii) localize a background instance (*i.e.* bike-side), or (iii) fail to localize object or background instances (*i.e.* bounding box of detection does not satisfy the 50% overlap criteria with the bounding box of test instance from the image). The first two gives us an idea on the ability of the detector to discriminate between similar shape object categories, and we report them also in Table 4.1. Specifically, for each detector, we report (within brackets in Table 4.1) the number of object instances that are correctly localized by the most confident M detections, where M is the number of instances of the target object category in the test set. In addition, we include the number of background instances (also within the brackets) that have been falsely localized by the detector within the top M detections. Here, M is 53 for the rigid case, and is 65 for the articulated case. For example, referring to Table 4.1, the bike-side detector has correctly localized 41 bike-side instances and falsely localized 5 motorbike instances among the top 53 detections.

Table 4.1: Detection and discrimination results on similar shaped object categories.

		Background	
		Motorbike	Bike-side
Object	Motorbike	-	0.9956 (52, 0)
	Bike-side	0.8183 (41, 5)	-

		Background	
		Cow-side	Horse-side
Object	Cow-side	-	0.9959 (63, 2)
	Horse-side	0.9114 (56, 2)	-

Confusing rigid (left) and articulated (right) object categories. Each entry records RP-AUC detection results on categories which share similar shape boundaries. Values within brackets indicate the number of object instances and background instances that are localized by the most confident M detections, where M is the number of instances of the target category in the test set. Here, M is 53 for the rigid (left) case, and is 65 for the articulated (right) case. All training and testing images used here are from the training and testing sets, respectively, of the Graz-17 dataset [149].

We draw several conclusions. The motorbike detector is very discriminative in which no bike-side instances is localized within the top 53 detections. In addition, it has strong detecting capability where 52 detections among the top 53 correctly localize test motorbike instances. This strong detection performance is also evident from its high 0.9956 RP-AUC score. On the other hand, the bike-side detector demonstrates weaker discriminative ability in which it confused 5 motorbikes from the same test set to be bike-side. In addition, it has a weaker detection

CHAPTER 4. CONTOUR BASED OBJECT RECOGNITION

Figure 4.26: Example localizations by (a) motorbike, (b) bike-side, (c) cow-side and (d) horse-side detectors, among the top M detections. In each, we show localizations by bounding box on the left, and back-projected codewords on the right. Note that motorbike and bike-side detectors are evaluated on the same test set, as are cow-side and horse-side detectors.

performance where it localized 41 bike-side instances among the top 53 detections, and also obtained a lower RP-AUC score of 0.8183. This weaker performance is a consequence of greater challenges from the bike-side category in which the objects are characterized by holes and thin skeletal structures, and are shown embedded in substantial background clutter. For the articulated case, we observe the cow-side detector to have strong discriminative ability in which only 2 (out of 65) horse-side instances are confused as cow-side. Similar observation can be made for the horse-side detector. However, the horse-side detector has a weaker detection performance where only 56 horse-side instances are localized among the top 65 detections. This is due to greater variations of horse-side instances in the training and testing images. In contrast, the cow-side detector localized 63 cow-side instances among the top 65 detections and also attained a high RP-AUC score of 0.9959. On the whole, these results demonstrate that our method is good at detecting and discriminating between object categories having similar shape boundaries, but will benefit from either more consistent training objects or a larger set of training data. Additionally, edge detection by modern boundary detectors *e.g.* [70, 76], which extract meaningful object boundaries and eliminate clutter edges, will also improve performance. We show example detections in Figure 4.26. Detections on background instances by the bike-side, cow-side and horse-side detectors, among the top M detections, are shown in Figure 4.27.

CHAPTER 4. CONTOUR BASED OBJECT RECOGNITION

Figure 4.27: Localizations on background instances by (a) bike-side, (b) cow-side and (c) horse-side detectors, among the top M detections. Notice that these false detections are due to boundary patterns that are characteristics of the target object category, *e.g.* boundaries from the wheels of test motorbikes in (a) confused by bike-side detector as belonging to bike-side instances, and that from the head and belly of test cow-side instances in (c) as belonging to horse-side instances.

Table 4.2: Comparison of classification and detection performance of the boosted ensemble of shape token xCC with [91, 149] on the Graz-17 dataset.

Object category	Number of object images		Image classification results				Object detection results			
	Training	Testing	ROC-AUC	ROC-EER	ROC-EER	ROC-AUC	RP-AUC	RP-EER	RP-EER	RP-AUC
			Our method	Our method	[149]	[91]	Our method	Our method	[149]	[91]
Plane	100	400	0.9826	6.9%	2.6%	0.9953	0.9325	10.0%	7.4%	0.9310
Motorbike	100	400	0.9983	1.0%	3.2%	1.0000	0.9996	1.0%	4.4%	1.0000
Face	100	217	0.9974	2.6%	1.9%	0.9966	0.9895	2.7%	3.6%	0.9850
Car-rear	100	400	0.9883	6.9%	0.05%	0.9992	0.9797	4.0%	2.3%	0.9912
Car- $\frac{2}{3}$ -rear	32	14	0.9643	14.2%	-	0.9000	0.6843	35.7%	12.5%	0.6925
Car-front	34	16	0.9688	6.2%	-	0.9727	0.8256	18.7%	10.0%	0.7233
Bike-rear	29	13	0.9468	7.6%	-	0.9172	0.6042	40.0%	25.0%	0.6398
Bike-front	19	12	0.9584	8.3%	-	0.9375	0.7421	33.3%	41.7%	0.6344
Bike-side	90	53	0.9445	9.4%	-	0.9366	0.8299	18.8%	28.0%	0.6959
Bottle	54	64	0.9773	7.8%	-	0.9802	0.9009	12.5%	9.0%	0.9468
Cow-front	34	16	0.9844	6.2%	-	0.9727	0.8335	18.7%	18.0%	0.8575
Cow-side	45	65	0.9944	6.1%	-	0.9992	0.9945	6.1%	0.0%	0.9975
Horse-front	44	22	0.9918	4.5%	-	0.9566	0.7368	27.2%	13.8%	0.7852
Horse-side	55	96	0.9756	7.2%	-	0.9816	0.9361	11.4%	8.2%	0.9680
Person	39	18	0.9352	16.6%	-	0.9321	0.5730	47.6%	47.4%	0.4271
Mug	30	20	0.9525	5.0%	-	0.9600	0.9619	5.0%	6.7%	0.9035
Cup	31	20	0.9800	10.0%	-	0.9825	0.8964	15.0%	18.8%	0.9158
Averaged across categories			0.9730	7.4%	1.9%*	0.9659	0.8483	18.1%	15.1%	0.8291

*Average ROC-EER for [149] is calculated from the four categories whose classification results are reported by the authors.

4.6.3.4 Evaluating on Graz-17 Dataset [149]

In our final experiment, we evaluate image classification and object detection performance on the complete Graz-17 dataset [149]. Except for the number of scales (which we adjust against the training data), parameter values remain unchanged. We show example detections and codewords visualizations in Figure 4.28. Table 4.2 reports quantitative results, with comparison to recent state-of-the-art contour based recognition methods of Opelt *et al.* [149] and Shotton *et al.* [91]. Classifica-

Table 4.3: Comparison of ROC-EER results on the first four categories of the Graz-17 dataset to other published results.

Object category	Our method	Sivic '05 [98]	Crandall '06 [237]	Fergus '07 [107]	Bar-Hillel '08 [116]	Chen '09 [238]	Zhu '09 [239]
Plane	6.9%	3.4%	9.3%	6.3%	6.7%	8.2%	8.2%
Motorbike	1.0%	15.4%	3.0%	3.3%	4.9%	5.4%	7.1%
Face	2.6%	5.3%	2.0%	8.3%	6.3%	2.0%	2.3%
Car-rear	6.9%	21.4%	6.5%	8.8%	0.6%	-	-
Averaged across categories	4.4%	11.4%	5.2%	6.7%	4.6%	5.2%*	5.9%*

*Average ROC-EER scores for [238, 239] are calculated from the three categories whose classification results are reported by the authors.

tion and detection results are averaged over 17 categories and shown (in bold) in the last row of the table. Note that for [149], average ROC-EER is computed from the first four categories whose classification results are provided by the authors.

Even though AUC scores are more representative, we compare against Opelt *et al.* [149] by the EER scores (since only equal error rates are provided by the authors). This comparison is performed under the same experimental settings as [149], where a detector for each category is learned independently over the category and is tested on the category test set. For classification, ROC-EER scores show our method to perform better on the motorbike category, while [149] is better on the other three categories. Overall, our classifier achieves an average ROC-EER (across the four categories) of 4.4% which is not as good as that of their state-of-the-art method. Nevertheless, it is still quite competitive and performs better than other published results [98, 107, 116, 237–239] as shown in Table 4.3. It has to be mentioned that even though we use stronger supervision than the methods in Table 4.3, our results are obtained with 100 positive and 100 negative training images, which is less than half of what was used by these methods during training.

For object detection, our results, in comparison to Opelt *et al.* [149], show better performance on some categories (*e.g.* motorbike and face), but considerably worse for categories such as car- $\frac{2}{3}$ -rear and bike-rear which have very few test images. The number of test images, T_s , can sharply affect a RP-EER score and hence directly influences the significance of the RP-EER measure. Specifically, RP-EER reports detector performance at a single precision value (point at which precision equals recall) and hence even one false positive or missed detection can impact RP-EER by as much as $\frac{100}{T_s}\%$, as also pointed out in [91]. Consequently, even though a higher average RP-EER of 18.4% computed over all 17 categories is obtained by our method, much more significant are the detection results for categories with

more test images. In particular, considering object categories with more than 200 test images (*i.e.* first four categories), we attain an average RP-EER of 4.425%, exactly matching that obtained by [149] (Table 4.2).

We compare classification and detection performance with Shotton *et al.* [91] by the more representative AUC scores (which evaluate performance across all precision values). For detection, Shotton *et al.* [91] obtained higher RP-AUC scores on some categories (*e.g.* bottle, cow-front and horse-front), while we yield better results on other categories such as car-front, person and mug. More interesting are the detection results on the bike-rear, bike-front and bike-side categories for which test objects are characterized by holes and wiry structures, and are shown embedded in cluttered scenes. Of these three bicycle categories, [91] achieves better detection results on the bike-rear category where they obtain a 0.0356 higher RP-AUC. On the other hand, we achieve substantially higher RP-AUC on the two other bicycle categories (bike-front and bike-side), in which our average gain in RP-AUC is more than three times higher at 0.1209. Overall, we achieve better classification and detection results: [91] obtained an average ROC-AUC of 0.9659, which we improve upon with 0.9730, and an average RP-AUC of 0.8291, which is lower than 0.8483 attained by our method. Importantly, this improvement is achieved with a smaller training set, compared to [91] where hypotheses from the test images are augmented with the original training data for retraining (as in *Shotton-II* method of the previous Weizmann horse experiment in Section 4.6.3.1).

4.6.3.5 Computational time performance

We now present the runtime performance of our contour based recognition approach. Learning discriminative xCC for an object category takes approximately four hours with our Matlab implementation on a standard workstation. This training step can be done off-line and hence does not affect the processing time during query. Detecting object instances and classifying images at query time is much faster (see Table 4.4). Specifically, after preprocessing to extract shape primitives of line segments and ellipses, it takes an average time of around three minutes to construct shape tokens for a test image using our Matlab implementation. Finally, object detection and image classification by the multi-scale sliding window

CHAPTER 4. CONTOUR BASED OBJECT RECOGNITION

Figure 4.28: Example detections and codewords visualization on all object categories of Graz-17 test set. Note accurate localization despite large intra-class variations, poses and scale changes, illumination variations, background clutter and partial occlusion, and detection of multiple objects.

Table 4.4: Computational time to recognize objects in a single test image by our contour based method. Results shown are the mean and one standard deviation time obtained with a standard workstation on the first four Graz-17 categories.

Constructing shape tokens (Matlab implementation)	Detecting objects by multi-scale sliding window approach (C implementation)
3.34 ± 3.00 (minutes)	13.03 ± 7.06 (seconds)

approach can be achieved in under fifteen seconds (on average) by our C implementation for each test image. It has to be mentioned that this runtime performance is weaker than that of Shotton *et al.* [91], where their C# method takes around two hours to train and ten seconds per image to test, though we leave optimization for speed-up of our approach as a future endeavor.

4.7 Summary

In this chapter, we presented a novel method to object detection and image classification that rely only on the cue of edge information. Image contours are represented by very simple and generic shape primitives of line segments and ellipses, which are then paired together to form shape tokens. We learn a category-specific codebook of shape tokens from unsegmented and noisy training instances, with just bounding boxes provided. Combinations of codewords which reliably predict the location of object instances are learned by capitalizing on discriminative appearance, geometric and structural constraints of a category in a unified framework. Our approach does not restrict combinations to have a fixed number of codewords, but allow them to flexibly adapt to an object category. This, coupled with the generic nature of the primitives, empowers a combination with much flexibility to represent discriminative shape structures of varying complexity. We localize objects in unseen test images by exploiting a multi-scale sliding window technique, where a boosted ensemble evaluates each window to infer the presence or absence of a category instance.

Detailed experimental evaluations show excellent image classification and object detection performances of our method. In particular, we demonstrate the method to be robust towards inter-class similarity, in which it successfully detects and discriminates object categories having similar shape boundaries. While we build a detector for a single viewpoint, experiments also reveal our approach to cope well under slight changes in viewpoints. Additionally, we evaluate our method on challenging datasets across 17 demanding object categories, in which test objects seen under scale and illumination changes, intra-class variations, slight pose changes, partial occlusion and substantial background clutter are recognized. This further demonstrates the effectiveness of our approach. Finally, comparisons with recent state-of-the-art contour based recognition systems and other related methods show very favorable and competitive image classification and object detection results of our method.

Chapter 5

Fusion of Contour and Region Based Features

This chapter explores the combination of contour based features and region based features for image classification and object detection. Similar to the contour only method of the previous chapter, we tailor codewords to a target object category, and exploit decision tree and boosting as part of our learning module. Discriminative codeword combinations that are learned by the proposed hybrid method can contain variable number and variable types of codewords. These two degrees of variability empower a combination with much flexibility to represent discriminative features of the object category. We experimentally evaluate recognition performance using the fusion of contour and region features against (i) using contour only features, and (ii) using region only features. It is shown that the proposed hybrid method exploits the synergy between these complementary feature types to attain better recognition performance, above what is obtained by either feature type individually. Additionally, comparisons against other recent hybrid based methods demonstrate very favorable and competitive recognition performances of the proposed method.

5.1 Introduction

We presented in the previous chapter an approach to learn combinations of shape tokens (*i.e.* contour features) that are discriminative of an object category. While very good image classification and object detection results were achieved using these contour features, that chapter exploited only edge information of an image and completely ignored region based image information such as image intensity levels or texture. However, region based features offer valuable recognition cues as well, particularly for object categories which are well defined by their local appearance regions (*e.g.* spotted cat and zebra). Correspondingly, by relying on only one single source of image information, object models that are learned in the previous chapter may lack in their discriminative power.

In this chapter, we set out to explore the fusion of contour based features (*i.e.* shape tokens introduced in the previous chapter) with region based features (SIFT descriptors presented in [39]) in a unified framework to learn object models. Shape tokens and SIFT descriptors are largely orthogonal, where the former corresponds to shape boundaries and interior silhouettes, and the latter to sparse salient image patches. Correspondingly, the synergy between these two complementary features can result in enhanced recognition performance. Given our flexible approach to learn discriminative combinations of shape tokens in the previous chapter, the fusion of these two feature types is particularly straightforward. Figure 5.1 outlines the proposed method, with references to corresponding sections in this chapter. Specifically, we first separately learn a codebook of shape tokens and a codebook of SIFT descriptors from positive training images. Both codebooks are specific to a target object category in that the codewords correspond to distinctive and characteristic parts of the object category. Next, combinations of codewords that are representative of an object category are learned by harnessing discriminative appearance, geometric and structural constraints of the target object category in a unified framework. Each combination contains a variable number of x codewords, where these codewords can be either (i) all shape tokens, (ii) all SIFT descriptors, or (iii) a mixture of shape tokens and SIFT descriptors, (see ahead to Figures 5.3 and 5.4). The number and feature types to be combined together into a codeword

combination are learned automatically from the training images, and represent the more discriminative ones based on the training set. We term such a codeword combination as a *x-mixture-codeword-combination* or *xMCC*. By imparting these two degrees of variability (in both the number and types of codewords) to a *xMCC*, we empower it with greater flexibility to model discriminative appearance, geometric and structural constraints of the target object category. This improves on the *x-codeword-combination* (*i.e.* *xCC*) of the previous chapter where the codewords in a combination consists of only shape tokens. Finally, we learn an ensemble of discriminative *xMCC* using a boosting framework to distinguish object presence across scale space, and use the boosted ensemble to detect objects and classify images on the test set.

5.1.1 Related Works

Here, we briefly discuss related works that exploit both contour and region features as recognition cues. For a more comprehensive discussion of other object detection and image classification methods, we refer the reader to Chapter 2. Kumar *et al.* [138] employed both contour fragments and the texture enclosed within fragments as features to detect articulated object categories. Heitz *et al.* [146] developed a probabilistic method to learn a deformable shape model and an appearance model of the object category. Their work used image contours and SIFT descriptors as recognition features. While good detection results were demonstrated in [138] and [146], both works demand strong supervision where tracked video sequences or hand labeled parts are used for training in [138], and manual delineation of object outlines in [146]; we simply require bounding boxes. Leibe *et al.* [240] adopted a part based approach to find hypotheses for the presence of pedestrians using a set of DoG image patches. A post processing step was then used to match complete outline contours at the hypotheses. Very compelling results were demonstrated, though they require segmented object instances for training. Fergus *et al.* [159] used both curved contour fragments and PCA coefficients that describe salient interest regions for object recognition. As one weakness, edge information is not fully exploited in that work as they used only clean planar curved fragments that

Figure 5.1: Overview of our technique which harnesses shape tokens (contour features) and SIFT descriptors (region features) for the task of object detection and image classification. We first extract shape tokens and SIFT descriptors from training images from which codebooks of shape tokens and SIFT descriptors are separately learned. Next, these codebooks are exploited to learn an ensemble of discriminative codeword combination (*i.e.* $xMCC$) within a boosting framework. This boosted ensemble is used for object detection and image classification, and we evaluate recognition performance on test images. Notice that while the proposed framework is a very straightforward adaption of that proposed in the previous chapter (*c.f.* Figure 4.1), here complementary feature types are exploited as recognition cues instead.

are in between bitangent edge points.

More closely related to our technique, Opelt *et al.* [149] extended their contour only method and proposed a hybrid object detection method in the same paper which employed contour fragments and SIFT descriptors. They first learned a codebook of contour fragments and a codebook of SIFT descriptors from training

images. A fixed number of two contour fragment codewords are exhaustively paired together into a single contour based feature. Similarly, they exhaustively pair every two SIFT codewords into a single region based feature. Boosting is employed to select codeword pairings that are representative of an object category. At test time, these codeword pairings are used to vote on the presence of objects in a Hough map. In another closely related work, Shotton *et al.* [148] used dense texton and contour fragments for object detection in a multi-scale sliding window framework. Useful textons and contour fragments of an object category are selected in a boosting framework and are used to vote for the presence of an object in each window. As limitations, each boosted feature that is learned by both [149] and [148] is restricted to be either contour based or region based; their boosted feature cannot simultaneously contain both contour fragments and SIFT/texton. Additionally, to keep learning tractable, they constrain a boosted feature to contain a fixed number of contour fragments (two fragments in [149] and single fragment in [148]) or SIFT/texton (two SIFT descriptors in [149] and single texton in [148]). This limits their potential to represent object class models. In contrast, our method imposes neither restrictions; a boosted $xMCC$ that is learned is allowed to comprise a single feature type or a mixture of feature types, where each $xMCC$ can contain a varied number of features. Such flexibility allows a boosted $xMCC$ to naturally evolve to best represent an object category.

5.2 Category-specific Codebook of SIFT Descriptors

We learn a codebook of shape tokens and a codebook of SIFT descriptors separately from positive training images of the target object category. The method to learn a shape token codebook has been detailed previously in Section 4.3. Here, we describe our method to learn a category-specific codebook of SIFT descriptors.

We learn a SIFT codebook by a very similar method as that used to learn a shape token codebook. Specifically, we first detect salient image patches from within the bounding boxes of training objects using the DoG detector and represent

each salient patch with a SIFT descriptor¹⁴. Next, SIFT descriptors are clustered based on their appearances (as represented by the 128-dimension vectors) with the adapted bisecting k -medoid clustering method (as also done previously for shape tokens which is detailed in Section 4.3.1 and Algorithm 4.1). Dissimilarity between SIFT descriptors is computed by the Euclidean distance measure (in line with [92]) and the range of this measure is $[0, 2]$. Given that this distance measure differs from that used for shape tokens in equation (4.2) and affects only this clustering step, we experimentally determine the best value for the clustering parameter, giving dissimilarity threshold th to be 0.50. We apply a same second-step clustering for the relative positions of SIFT descriptors using mean-shift (see Section 4.3.2), and identify the medoid of each mean-shift sub-cluster as a SIFT candidate codeword. In the same way as that for a shape token candidate codeword, we associate each SIFT candidate codeword with a distance allowance τ indicating the range of appearance it represents, and a scale-normalized circular window of center $\bar{c}$ and radius $\bar{r}$ specifying where it is expected to be found relative to an object centroid. Note that previously in Section 4.3.3, we score a shape token candidate codeword based on four terms where one of the terms is the average edge strength represented by its members. Here, we score a SIFT candidate codeword as a product of the other three terms. Finally, using the same radial ranking method, we select the highest scoring 350 SIFT candidate codewords into a SIFT codebook.

We visualize a subset of the SIFT codewords for the horse category of the Weizmann dataset [40], and some other categories of the Graz-17 dataset [149] in the top row of Figure 5.2. Each SIFT codeword is represented as a square image patch whose center corresponds to a DoG keypoint, and whose area is proportional to the region size used to compute the SIFT descriptor. Interestingly, it can be observed that contextual information for the presence of object instances have been captured by SIFT codewords for some object categories, *e.g.* a plane often appear in front of a runway, and a car on a road (since the background contains repeatable region features which co-occur with the object instances). Note that such cues cannot be captured by shape tokens which are based on contours, unless

¹⁴SIFT descriptors are extracted from an image using the implementation by Lowe, available at [241].

Figure 5.2: Category-specific codebooks of SIFT and shape tokens that are learned for the Weizmann horse dataset [40] and the plane, motorbike, face and car-rear categories of the Graz-17 dataset [149]. For clarity of presentation, we show 200 SIFT codewords of each object category in the top row. Shape token codewords for the corresponding object categories are shown in the bottom row. Yellow ‘+’ in each codebook visualization denotes object centroid.

the background contains edges which appear repeatedly with the object instances *e.g.* the shadows cast by a horse or motorbike. For the sake of completeness, we also show the corresponding shape tokens codewords that are learned for these object categories in the bottom row of Figure 5.2, where line primitives are shown in blue and ellipse primitives in magenta.

5.3 Discriminative Shape Token and SIFT Codeword Combinations

In this section, we describe the proposed hybrid method to learn discriminative combinations of shape token and SIFT codewords *i.e.* $xMCC$. Each $xMCC$ that is learned can contain a variable number of codewords, and also contain a mixture of shape token and SIFT codewords, where the choices for the number and types of codewords to be included into a $xMCC$ are learned automatically from the training set. Consequently, such a flexible approach to combine codewords allows a $xMCC$ to naturally evolve to best represent an object category.

The learning of discriminative $xMCC$ follows a simple and straightforward adaptation of our method to learn xCC that was detailed in the previous chapter. In particular, at every object hypothesis that is sampled across all training (both positive and negative) images, we first find the $\mathcal{R}(\cdot)$ values of all shape tokens codewords by equation (4.8), and the $\mathcal{R}(\cdot)$ values of all SIFT codewords as

$$\mathfrak{R}_i(\hat{s}, \hat{x}) = d_{SIFT_{app}}(\gamma_i, t^*) + d_{geo}(\gamma_i, t^*), \quad (5.1)$$

where

$$t^* = \arg \min_{t'} (d_{SIFT_{app}}(\gamma_i, t') + d_{geo}(\gamma_i, t')), \quad (5.2)$$

$$d_{SIFT_{app}}(\gamma_i, t') = \begin{cases} \frac{\|A_i - A'\|_2}{\tau_i} & \text{if } \|A_i - A'\|_2 \leq \tau_i \\ \infty & \text{otherwise} \end{cases}, \quad (5.3)$$

and $\|\cdot\|_2$ is the L_2 norm. Here, γ_i denote a SIFT codeword, A_i its 128-dimension appearance descriptor, τ_i its appearance distance allowance, and t' denote a SIFT descriptor of an image whose appearance descriptor is A' . Computation of $d_{geo}(\cdot)$ follows that in equation (4.7). The $\mathfrak{R}(\cdot)$ values of all shape token codewords are arranged into a $N_c \times M$ matrix where N_c denotes the number of shape token codewords and M the number of object hypotheses, with an entry (i, j) in the matrix corresponding to the $\mathfrak{R}(\cdot)$ value of the i^{th} shape token codeword at the j^{th} object hypothesis. Similarly, we collect the $\mathfrak{R}(\cdot)$ values of all SIFT codewords into a $N_r \times M$ matrix, where N_r is the size of the SIFT codebook. These two matrixes are concatenated into a single $(N_c + N_r) \times M$ training matrix. Additionally, as done in the previous chapter, we also define a $1 \times M$ label vector where each entry in the vector has value +1 or -1 to indicate the presence or absence of an object in the training hypotheses, and a $1 \times M$ weight vector whose entries correspond to the weights of the training hypotheses. The training matrix and the label and weight vectors, together with a split number k (to indicate the number of splits in a tree), serve as input to learn a binary decision tree [74]. Hence, discriminative $xMCC$ can be found by simple path transversal from the root node to each leaf node of the learned decision tree, in which the prediction of a $xMCC$ at hypothesis $(\hat{s}, \hat{x})$ is,

$$xMCC[\hat{s}, \hat{x}] = \begin{cases} \ell & \text{if } p_i \mathfrak{R}_i(\hat{s}, \hat{x}) \leq p_i \vartheta_i \text{ and } p_j \mathfrak{R}_j(\hat{s}, \hat{x}) \leq p_j \vartheta_j \text{ and } \dots \\ 0 & \text{otherwise} \end{cases}, \quad (5.4)$$

where $\ell \in [-1, +1]$ is the predicted label given by the leaf node in the path and prediction of 0 implies that the $xMCC$ does not vote for or against an object presence at hypothesis $(\hat{s}, \hat{x})$, with the values of parameters p and ϑ learned automatically by the decision tree. Consequently, by boosting a set of decision trees using the same method in Algorithm 4.2, an ensemble of discriminative $xMCC$ can be learned in which the detection confidence at an object hypothesis is

$$H[\hat{s}, \hat{x}] = \sum_{j=1} \alpha_j \times xMCC_j[\hat{s}, \hat{x}], \quad (5.5)$$

with α_j as the weight of $xMCC_j$ and is learned during boosting.

Figures 5.3 and 5.4 visualize some $xMCC$ that are learned automatically for the Weizmann horse [40] and the Graz-17 motorbike [149] datasets respectively. Each row is for one $xMCC$, where the first column shows the codewords that are combined into a $xMCC$. The centroid of each codeword (*i.e.* midpoint between the centers of shape primitives for a shape token codeword, and the center of an image patch for a SIFT codeword) corresponds to the center of a colored circle, in which the circle defines the region where the codeword is expected to find matching features. A green-colored circle denotes the requirement that the codeword must find matching features and corresponds to $p = +1$ of predicate $p \Re(\hat{s}, \hat{x}) \leq p \vartheta$, while a red-colored circle depicts the requirement for the codeword to be unmatched (or be weakly matched) and corresponds to $p = -1$. The remaining three columns show positive hypotheses $(\hat{s}, \hat{x})$ in training images which satisfy the appearance, geometric and structural constraints modeled by the $xMCC$. We show shape token codewords of a $xMCC$ and its matching tokens in training images by the same color. SIFT codewords of a $xMCC$ and its matching SIFT descriptors in training images are depicted by square image patches, in which the center of each image patch is localized on the detected DoG keypoint, with the area of the patch corresponding to the region size at which the SIFT feature is computed. We indicate the location $\hat{x}$ of a hypothesis by a blue '+', and its scale $\hat{s}$ by the diameter of a blue-outlined circle. For clarity of presentation, we show $xMCC$ that are learned using 3 splits ($k = 3$) and hence each $xMCC$ has a variable number of at most 3 codewords. However, as in the previous chapter, rather than fixing the same split number for

Figure 5.3: Learning discriminative $xMCC$ for the Weizmann horse dataset. Each row is for one $xMCC$, in which the first column visualizes an $xMCC$ that is automatically learned by boosting, and the remaining columns show the positive hypotheses in training images which satisfy the appearance, geometric and structural constraints modeled by the $xMCC$. For clarity of presentation, we show $xMCC$ that are learned by decision trees with 3 splits; in practice, we find the optimal split number for an object category by 3-fold cross validation over values 1 to 10 on the training hypotheses, and use this split number to learn the $xMCC$. Notice that an $xMCC$ can contain purely SIFT codewords (first row), purely shape token codewords (second row), or a mixture of SIFT and shape token codewords (last two rows).

all object categories, we learn the optimal split number for each category using 3-fold cross validation over values 1 to 10. After finding the optimal split number for a category, we learn an ensemble of discriminative $xMCC$ over the entire training hypotheses, and use this ensemble to detect objects in test images. Notice that unlike xCC that were learned in the previous chapter which comprises only shape token codewords (*c.f.* Figures 4.18 and 4.19), each $xMCC$ that is learned in this chapter comprises codewords of types (i) purely SIFT (first row in Figures 5.3 and 5.4), (ii) purely shape token (second row), and (iii) a mixture of SIFT and shape token codewords (last two rows).

Figure 5.4: Learning discriminative $xMCC$ for the motorbike category of the Graz-17 dataset. Layout in this figure follows that of Figure 5.3. Note again that a learned $xMCC$ can comprise either a single feature type (*i.e.* shape token or SIFT descriptor), as shown in the first and second rows, or a mixture of complementary feature types, as shown in the bottom two rows.

5.4 Object Detection and Image Classification

We use the same multi-scale sliding window method that was detailed in Section 4.5 to detect objects in test images, except that now windows are evaluated using the boosted ensemble of $xMCC$ in equation (5.5). Similarly, we classify images in an identical way as that done in the previous chapter, in which we take the confidence of the strongest window (*i.e.* most confidence window) in each test image as the classification confidence of the image.

5.5 Experimental Evaluation

We evaluate our technique on the Weizmann horse dataset [40], and four other categories of the Graz-17 dataset [149]. Both image classification and object detection results are reported, and we compare against our contour only method of Chapter 4, and to other recent hybrid based methods. Additionally, we also eval-

uated against a benchmark region based method which is obtained by replacing shape tokens in our contour only method with SIFT descriptors. We adhere to the experimental procedures and setups described in Sections 4.6.1 and 4.6.2. Additionally, for fair comparison with the contour only method of Chapter 4, both the benchmark region based method and our proposed hybrid based method follow the exact parameter values as Section 4.6.3 for each object category. Importantly, our contour based, region based, and hybrid based methods are learned using 300 boosting rounds, where the optimal split number for each method is found by 3-fold cross validation over values 1 to 10 on training images of the target category. We report image classification performance of each method by the ROC-AUC score, which is obtained by first plotting the ROC curve¹⁵ and then computing the area under the curve. For object detection, we use the RP-AUC scores for quantitative comparisons between different methods. A RP-AUC score is computed as the area under a RP curve¹⁶. As pointed out in the previous chapter, the ROC-AUC and RP-AUC scores take the entire curves into account to give a more representative score to evaluate performance as compared to the ROC-EER¹⁷ and the RP-EER¹⁸ scores which report performance at a single point on the curves. Hence, as far as possible, we report the more meaningful AUC scores, and give EER scores only for comparison purposes with other hybrid based methods.

5.5.1 Image Classification

We begin by evaluating image classification performance of the proposed hybrid method of SIFT + shape token $xMCC$ ensemble on the complete Weizmann horse dataset, and on the planes, motorbike, face and car-rear categories of the Graz-17 dataset. Table 5.1 reports the ROC-AUC scores obtained by the method, with comparison to (i) our earlier contour only method of Chapter 4 which used only

¹⁵A ROC curve is obtained by plotting the true positive rate against the false positive rates as a global classification confidence is varied

¹⁶A RP curve plots the trade-off between recall and 1-precision as a global detection confidence is varied.

¹⁷A ROC-EER score is the false positive rate at a single point on the ROC curve *i.e.* the point at which true positive rate equals 1-false positive rate.

¹⁸A RP-EER score is the 1-precision value at a single point on the RP curve *i.e.* the point at which recall equals precision.

shape tokens, (ii) our benchmark region based method which used only sparse SIFT features, and (iii) the recent method by Shotton *et al.* [148] which adopted dense texton features and contour fragments as recognition cues. We average classification ROC-AUC scores over the five tested object categories and show them (in bold) in the last row of the table.

Table 5.1: Comparison of image classification performance to a benchmark region based method, our contour only method of the previous chapter, and to recent method by Shotton *et al.* [148]. First row shows ROC-AUC scores for the Weizmann horse dataset, and the next four rows shows the ROC-AUC scores for four categories of the Graz-17 dataset.

Object category	Our methods proposed in this thesis			Shotton <i>et al.</i> [148]		
	<i>Region based</i> SIFT xCC only ensemble	<i>Contour based</i> Shape token xCC only ensemble - Chapter 4	<i>Hybrid based</i> SIFT + shape token $xMCC$ ensemble	<i>Region based</i> Dense texton only features	<i>Contour based</i> Contour only features	<i>Hybrid based</i> Dense texton + contour features
Horse	0.9227	0.9500	0.9687	0.9323	0.9127	0.9655
Plane	0.8805	0.9826	0.9820	0.9900	0.9950	0.9940
Motorbike	0.9662	0.9983	0.9980	0.9960	1.0000	1.0000
Face	0.9419	0.9974	0.9961	0.9990	0.9970	0.9980
Car-rear	0.9751	0.9883	0.9942	1.0000	0.9990	0.9990
Averaged across five categories	0.9373	0.9833	0.9878	0.9835	0.9807	0.9913

Several interesting observations can be made from Table 5.1. Firstly, with reference to the methods proposed in this thesis, it can be observed that the SIFT xCC only ensemble (second column) is less effective in classifying images, in which it obtains ROC-AUC scores that are consistently below that attained by the shape token xCC only ensemble (third column), and the SIFT + shape token $xMCC$ ensemble (fourth column) for every tested category. The average ROC-AUC score that is computed across the five categories also reveals its inferior classification performance, in which its average ROC-AUC number is at least 0.04 lower than either ensemble. This weaker performance is very likely due to the fact that the SIFT xCC only ensemble adopts *only* the DoG interest region detector to extract salient patches from an image. Correspondingly, the only region based information is where the DoG detector fires. This obtains a very sparse set of image regions and does not provide dense enough regions to characterize an object category. Several other authors [94, 95, 97, 129] have also independently concluded that a single interest region detector cannot provide sufficient interest regions to represent an object category, and have instead resorted either to using a plethora of region detectors or to densely sampling image regions (see our earlier discussion on the region based

recognition methods in Section 2.2.1). The framework which we proposed here allows for a simple and flexible integration of features (*e.g.* interest regions detected by the Harris-Laplace, Hessian-Laplace and MSER detectors, and described by SIFT, PCA coefficients and spin images) and we believe that the use of such diverse features can improve classification performance, though we leave that for future endeavor. Turning to the performance by the proposed ensemble of $xMCC$ which combines SIFT features with shape tokens, we attain stronger classification performance in which a higher average ROC-AUC score is obtained by combining complementary features than using either shape tokens or SIFT descriptors alone. We highlight that all ensembles are obtained using the same codebook learning and boosting framework, and differ only in the type of features used. Correspondingly, the results obtained here demonstrate that our method of combining complementary feature types can improve classification performance.

In comparison to the classification results by the recent hybrid based method of Shotton *et al.* [148] which use dense texton and contour features (final column in Table 5.1), our proposed ensemble of SIFT + shape token $xMCC$ obtains 0.0035 lower average ROC-AUC score. We believe that the better classification performance by their method is due to their use of dense texton features; our contour based method (third column) attains slightly higher average ROC-AUC score than their contour based method (sixth column), but our region based method which use sparse SIFT features (second column) has substantially lower average ROC-AUC number as compared to their region based method which used dense texton features (fifth column). Consequently, it does appear that the dense texton features have contributed to the improved performance by their hybrid method, and we postulate that classification results by our method will also benefit by replacing sparse SIFT features with the dense texton features used in their work.

5.5.2 Object Detection

We now evaluate detection performance of the SIFT + shape token $xMCC$ ensemble on the same five test object categories. As highlighted in the previous chapter, object detection is a considerably harder task as compared to image classification

Figure 5.5: Example detections on the Weizmann horse test set (top row), and the plane, motorbike, face and car-rear categories of the Graz-17 test sets (remaining rows). Codewords which contributed positively to detections are back-projected and visualized on the right of each test image. Observe that test instances have been accurately localized in scale space despite substantial intra-class variations, background clutter, illumination variations, partial occlusion, and articulation changes.

since a poorly localized but strong bounding box contributes positively to classification but penalizes detection. We show example detections by the $xMCC$ ensemble in Figure 5.5. Shape token and SIFT codewords which contributed positively to detections are back-projected and visualized on the right of each test image. Table 5.2 reports detection performance by the hybrid method in terms of RP-AUC and RP-EER scores, with comparison against (i) the SIFT xCC only ensemble, (ii) the shape token xCC only ensemble, (iii) the hybrid based method of Shotton *et al.* [148] which used dense textron and contour features, and (iv) the recent hybrid based method of Opelt *et al.* [149] which used SIFT and contour features.

We draw several conclusions. Firstly, from Table 5.2, we observe that the SIFT xCC only ensemble has substantially weaker detection performance than either the shape token xCC only ensemble, or the SIFT + shape token $xMCC$ ensemble. This is not surprising since the SIFT only ensemble has already been shown to have a weaker classification performance as compared to the other two ensembles, and hence cannot be expected to have a stronger performance on the more

CHAPTER 5. FUSION OF CONTOUR AND REGION BASED FEATURES

Table 5.2: Comparison of object detection results to a benchmark region based method, our contour only method of the previous chapter, and to recent hybrid methods of Shotton *et al.* [148] and Opelt *et al.* [149]. First row is for the Weizmann horse dataset, and the next four rows for four other categories of the Graz-17 dataset.

Object category	Our methods proposed in this thesis						Shotton <i>et al.</i> [148]		Opelt <i>et al.</i> [149]
	<i>Region based</i> SIFT xCC only ensemble		<i>Contour based</i> Shape token xCC only ensemble - Chapter 4		<i>Hybrid based</i> SIFT + shape token $xMCC$ ensemble		<i>Hybrid based</i> Dense texton + contour features		<i>Hybrid based</i> SIFT + contour features
	RP-AUC	RP-EER	RP-AUC	RP-EER	RP-AUC	RP-EER	RP-AUC	RP-EER	RP-EER
Horse	0.8685	16.2%	0.9218	14.4%	0.9618	10.5%	0.9302	-	-
Plane	0.7168	27.5%	0.9325	10.0%	0.9341	9.5%	-	6.8%	13.4%
Motorbike	0.9779	6.0%	0.9996	1.0%	0.9981	1.0%	-	0.3%	3.7%
Face	0.9273	11.9%	0.9895	2.7%	0.9893	2.3%	-	2.3%	3.2%
Car-rear	0.9018	12.7%	0.9797	4.0%	0.9809	3.7%	-	1.0%	0.5%
Averaged across five categories	0.8785	14.9%	0.9646	6.4%	0.9728	5.4%	0.9302*	2.6%*	5.2%*

*Average RP-AUC and RP-EER scores for [148, 149] are calculated from the categories whose detection scores are reported by the authors.

challenging task of object detection. More importantly, we observe that among these three ensembles, the proposed hybrid method which exploits two sources of image information (*i.e.* edge and region based information) attains better detection performance as compared to relying on a single source of image information; the hybrid method obtains 0.9728 average RP-AUC score which is 0.0082 higher than that using only shape tokens, and 0.0943 higher than that using only SIFT features. This confirms our method which combines contour and region features can improve recognition performance, above what either feature type can obtain individually.

Next, turning to comparison with [148], our hybrid method attained higher RP-AUC score on the Weizmann horse dataset, but obtained weaker detection performance on three of four tested Graz-17 categories. Overall, we obtain 1.3% lower average RP-EER score (evaluated across the four Graz-17 classes) than that attained by [148]. We believe this weaker performance is due to insufficient image regions that are detected by the DoG interest region detector, and hence our method cannot fully exploit region based information of an image. On the other hand, [148] extracts dense texton features from an image, and thus has the ability to fully capitalize on the image region information. As future work, we plan to investigate how the choice of region based features can affect recognition performance by our framework. We also compare detection performance of our hybrid method against [149], which also used SIFT features (with contour fragments) as recognition cues. Their method obtains better RP-EER score on the car-rear cate-

Figure 5.6: Example incorrect detections on two categories (*i.e.* plane and horse categories) with the weakest detection performances. Incorrect detections for the plane category shown in Figures 5.6(a) and 5.6(d), while that for horse category shown in Figures 5.6(b), 5.6(c) and 5.6(e). In each figure, we show detections on test images on the left, where red indicates detected bounding box and yellow the ground truth for false negative. Back-projected codewords are shown on the right of each test image. The third image in Figures 5.6(a) and 5.6(b) show all SIFT features extracted from test images. Here, SIFT features are represented by scale oriented arrows on detected DoG keypoints.

gory, while we achieve better RP-EER scores on the other three Graz-17 categories. Overall, we attain 4.1% average RP-EER score (computed across the four Graz-17 categories), which is superior to the 5.2% obtained by their method.

It is useful to look at the plane and horse categories in which the proposed hybrid based method attains lower detection scores as compared to the other three object categories. We give some observations into the causes of the incorrect detections of these categories. Feature (both contour and region based) extraction is a crucial step in the accurate detection of test object instances. In particular, when no or little features are extracted from a test object instance, it leads to failed detection of the instance. Consider Figures 5.6(a) and 5.6(b). In each figure, we show detections on the left (where red bounding box indicates detection, and yellow the ground truth for false negative *i.e.* miss), back-projected codewords in the middle, and all extracted SIFT features on the right. Here, extracted SIFT features are represented as oriented scaled arrows on detected DoG keypoints. Notice that very few SIFT features are extracted from the object instances of the two test images. This failure in the feature extraction step contributes to the incorrect detections in the two images. Additionally, the proposed hybrid based method learns object

models for a single viewpoint. As such, it is unable to accurately localize the object instance in Figure 5.6(c), where substantial out-of-plane rotation of the test object exceeds the reasonable tolerance of our side-on horse detector. Finally, we shows false positives on negative images in Figures 5.6(d) and 5.6(e), in which Figure 5.6(d) shows the most confident false positive (on negative images) for the plane category and Figure 5.6(e) shows that for the horse category. These images contain features that are sufficiently similar to that seen in training images (*e.g.* textures of pavement in Figure 5.6(d) have similar appearance to that of a runway, and contours of the two vertical beams and horizontal billboard in Figure 5.6(e) appear sufficiently similar to the legs and belly of a horse). While it may not be entirely possible to eliminate all incorrect detections for every image (*i.e.* both positive and negative images and even those not tested here), the proposed hybrid based method has at least shown that combining complementary feature types can improve detection performances. In this light, we postulate that the combinations of more diverse feature types, perhaps by incorporating depth cues, can eliminate some of the incorrect detections shown here and further improve performances.

5.5.3 Computational time performance

We now give comments on the runtime performance of the proposed hybrid based method. Using an unoptimized Matlab implementation, a boosted ensemble of discriminative $xMCC$ can be learned for an object category in approximately nine hours on a standard workstation. After preprocessing to extract shape primitives, it takes under four minutes (on average) to extract shape tokens (with a Matlab implementation) and SIFT features (with a C implementation) for a test image (see Table 5.3). Finally, object detection and image classification by the multi-scale sliding window approach can be achieved in under a minute by our C implementation. We mention here that this runtime performance is weaker than that achieved by Shotton *et al.* [148] (which can process an image in around twelve seconds), though we leave code optimization of our hybrid based recognition approach for future improvements.

Table 5.3: Computational time to recognize objects in a single test image by our hybrid based method. Results shown are the mean and one standard deviation time obtained with a standard workstation on the four Graz-17 categories.

Constructing shape tokens and extracting SIFT features (combination of Matlab and C implementations)	Detecting objects by multi-scale sliding window approach (C implementation)
3.48 ± 3.01 (minutes)	32.70 ± 15.05 (seconds)

5.6 Summary

In this chapter, we combined contour and region based features for the tasks of image classification and object detection. Our hybrid approach exploits distinguishing appearance, geometric and structural constraints of the target object category to learn discriminative combinations of codewords. Each combination contains a variable number of codewords, where these codewords can either be (i) all shape tokens, (ii) all SIFT descriptors, or (iii) a mixture of shape tokens and SIFT descriptors. Consequently, these two degrees of variability (in the number and types of codewords in a combination) empower a codeword combination with much flexibility to model discriminative constraints of an object category. Evaluation on five object categories reveals that our method of combining contour and region based features achieve improved image classification and object detection performances as compared to using either feature type alone. Additionally, comparisons with other recent hybrid based method also yield very competitive and favorable recognition results of our method.

Chapter 6

Conclusion

This chapter summarizes the main contributions of this thesis. Limitations of the techniques developed in this research are discussed, and we suggest possible directions for future works. Finally, we conclude this thesis with a short final remark.

6.1 Summary of Main Contributions in this Thesis

In Chapter 3, we presented a novel and powerful method to detect ellipses from an edge image. A novelty of our approach was the development of a self-correcting mechanism. This allowed our system to automatically identify weak ellipses and regenerate improved ellipses that better represent edge information. We performed extensive experimental evaluations on synthetic images under the scenarios of image clutter, salt-and-pepper noise and occlusion, where we demonstrated convincingly that our method attained superior detection performance (in terms of both recall and precision measures) against other existing methods. Additionally, we applied the proposed detector on a set of challenging real-world images and achieved visually pleasing results. We are not aware of any other works that can detect ellipses from such difficult images. Consequently, this chapter contributed a significant step towards robust ellipse detection.

Chapter 4 presented a contour based method to classify images and to detect ob-

jects from complex real-world images, where we showed these two recognition tasks could be achieved with very simple shape primitives of line segments and ellipses. A key novelty of our method was the development of a principled method to learn combinations of a variable number of category-specific codewords as discriminative features. By allowing a combination to have variable number of codewords, we imparted much flexibility to it to represent complex class models. We learned each discriminative combination automatically from noisy and unsegmented training data with only bounding boxes provided. We performed a thorough investigation of our contour based recognition method. It was shown that our method possessed some robustness to viewpoint variations, and had a strong ability to discriminate between similar shaped object categories. Additionally, comparison against recent state-of-the-art contour based methods and other related methods yielded very competitive recognition performance of our method.

We exploited the recognition cues of contours and regions (image patches) in Chapter 5 for image classification and object detection. A major novelty of our hybrid method was that each learned discriminative feature could contain either (i) only contour based codewords, (ii) only region based codewords, or (iii) both contour and region based codewords. We tailored codewords explicitly to the target object category, in which both the number and the types of codewords in a discriminative feature were learned automatically from training images of the category, using just bounding boxes. It was shown that our method to combine codewords of different feature types could improve recognition performances, above what could be achieved by either feature type individually. Additionally, experimental evaluations also demonstrated that our recognition method compared very favorably against other recent hybrid based methods.

6.2 Limitations and Future Works

While the methods developed in this thesis have proven powerful, several limitations of the methods exist which we discuss here. These limitations set the directions for future improvements which we also highlight in this section.

The proposed ellipse detector was not implemented with speed in mind, in which

our current implementation (combination of unoptimized Matlab and C (mex) codes) takes an order of minutes to detect ellipses in an image (see Table 3.1). This is not suitable for real time applications. Detailed investigation of the algorithm reveals many areas where parallel computations can be applied. For example, in the initial stage, we can apply a single processor to each connection between line segments to break a line edge map (with the *Contour Fragments Extraction* procedure in Section 3.2.1). Additionally, it is also obvious that the *Elliptical-arcs Candidates Identification* procedure of Section 3.2.2 is readily parallelized, where a single processor can be applied to each contour fragment to find candidate elliptical-arcs. Similarly, in the second stage, a single processor can be used by the *Pair-wise cost Assignment* procedure (Section 3.3.1) to compute the cost of pairing every two elliptical-arcs. In the same way, the *Weak Hypothesis Identification* procedure can be applied with a multi-processor framework, in which each processor evaluates the goodness of an ellipse hypothesis. We hope to re-implement the ellipse detector in CUDA¹⁹ where the native instruction set and memory of the multi-processors in a CUDA graphic processing unit can be exploited to accelerate our current implementation.

Our current approach to learn object models (by clustering) and discriminative codeword combinations (by tightly coupling decision tree with AdaBoost) suffers from two restrictions. Firstly, it requires bounding boxes of training objects to be presented in which training objects are required to have similar viewing angles. Even though a capability to learn from unlabeled images is certainly desirable, we highlight that modern online datasets *e.g.* LabelMe [242] which harness the collective power of internet users to label training images, even giving a rough outline of object instances, provide billions of readily available annotated images. One can then make use of these annotations (*e.g.* by the aspect ratio of bounding boxes) to separate training objects into separate views. This mitigates the severity of this restriction to some extent. A more limiting restriction is that our approach only detects objects seen at the similar viewing angle as that during training, though some tolerance to viewpoint variations has been demonstrated. To achieve

¹⁹CUDA is an acronym for Compute Unified Device Architecture which is a parallel computing architecture.

greater tolerance to viewpoint variations, one can improve the representation power of the features *e.g.* by capitalizing on affine interest regions, and exploiting these regions to guide the search for test object instances. Extension of the sliding windows technique to four dimensions that are spanned by scale, position and aspect ratio will also achieve further invariance to viewpoint variations, at the cost of greater training effort. Other inspirations for developing viewpoint invariant recognition methods can be found in the recent works by Savarese and Fei-Fei [243] and Hoiem *et al.* [244].

Learning discriminative combinations of codewords is slow (order of hours). A reason for this is that we have implemented the entire learning algorithm in Matlab. It is well known that runtime speed of Matlab codes can be massively accelerated when the codes are written in C or C++ instead. Another reason is due to the dense sampling of object hypotheses from training images which results in a very large set of training hypotheses that is used as input in the learning algorithm. Perhaps the method of [91] which coupled a sparse sampling technique with boot-strapping on training data can be adopted to extract training hypotheses. Additionally, in this work, we have learned discriminative codeword combinations by exploiting the decision tree classifier. While excellent recognition results are demonstrated with the learned combinations, however a decision tree may not be the best approach to learn these combinations; other learning methods such as the separable functions network (using rectangular component functions) can also be applied. A detailed study in this area is certainly desirable.

Currently, we detect an object category from an image one at a time *i.e.* to detect horse and cow categories from an image, we have to *separately* learn boosted ensembles for each category. In this aspect, the effort for training grows linearly with the number of object categories. This linearly growing effort is clearly impractical for developing systems that can detect thousands of real-world object categories. Instead, a *strongly* sub-linear approach which shares ensembles between object categories is important for real-world applications. One possible method is to present training hypotheses of multiple object categories to the boosting algorithm. At each boosting round we can select a particular object category, and optimize a decision tree for the selected category. Each object category then as-

signs its own weight to the learned tree (from which codeword combinations are extracted) based on the weighted accuracy on its category training hypotheses. Subsequent boosting round can then focus on the object category whose training hypotheses have the least weighted accuracy. Consequently, after boosting, each object category shares the same ensemble but has separate weights for the codewords combinations of the ensemble. We plan to investigate further this approach to learn shared ensembles.

In this thesis, we have coupled decision trees with a boosting algorithm to learn the identities of codewords to be combined together. These combinations are then exploited to perform object detection and image classification tasks. One can also use other classical learning methods like neural network or SVM to learn a detector/classifier to perform these recognition tasks. Towards this end, we can concatenate the $\Re(\cdot)$ values (computed with equation (4.8) and equation (5.1)) of all codewords at each sampled location and scale of a training image. This forms a feature vector for the location and scale, and is paired with a binary label to indicate if the location and scale contains an object instance. By collecting the feature vectors and its labels across all locations and scales for all (both positive and negative) training images, we can then use this collected training data as input to learn a two class classifier to differentiate between positive and negative feature vectors. Such a method learns a hyperplane which separates between positive and negative feature vectors in the codeword space, rather than the identities of codewords to be combined together, and it would be interesting to experimentally evaluate which method perform best with the proposed features.

Our current work performs generic object recognition on a single image and can be readily extended for meaningful processing on video data. Specifically, by applying the proposed recognition approaches on individual frames of a video sequence, we can easily perform object recognition in the video data. One can also enhance recognition accuracy in video data by coupling frame rate information with the knowledge that object instances typically do not disappear or abruptly change positions across short sequence of video frames. In this aspect, by smoothing detection confidences of sliding windows across the temporal domain, it is likely that we can improve recall and precision scores, as opposed to considering each frame indepen-

dently. Additionally, we can further extend object *recognition* to object *tracking* by formulating the tracking problem as a detection problem. In particular, a detector can be applied on each video frame to locate an object instance. Consequently, by collecting the locations of the object instances across video frames, we can then obtain the trajectory of the object across spatial and temporal domains. Note that such a tracking approach can also be exploited to discriminate between various poses of an object instance (*e.g.* a person sitting or standing), and hence supports high level activity recognition in video data.

We have explored the fusion of contour and region features for image classification and object detection. This attempt can be substantially improved by exploiting dense region features or diverse collections of interest region detectors/descriptors. The flexible nature of our learning framework supports a simple and straightforward extension along this direction. Additionally, other than contour and region features, we are also interested in the addition of more recognition cues. For example, we can extract features from the depth map of an image and combine these features with contour/region features to form more discriminative codeword combinations. Additionally, we believe the role of semantic cues (*i.e.* contextual features) have not been fully exploited in this thesis and plan to study how such features can be harnessed for improved recognition performances. Finally, our future plans also include testing on other publicly available datasets, *e.g.* PASCAL VOC datasets [234].

6.3 Concluding Remark

We have developed a number of techniques in this thesis that have proven powerful in recognizing object categories. While there is still a long way away from replicating human visual system which can recognize many thousands of categories under incredibly difficult settings (*e.g.* extreme intra-class appearance variations), we believe this thesis has taken a positive step towards a solution.

Bibliography

- [1] A.E. Booth and S. Waxman, “Object names and object functions serve as cues to categories for infants,” *Developmental Psychology*, vol. 38, no. 6, pp. 948–957, 2002.
- [2] A.E. Booth, “Object function and categorization in infancy: Two mechanisms of facilitation,” *Infancy*, vol. 10, no. 2, pp. 145–169, 2006.
- [3] I. Biederman, “Recognition-by-components: A theory of human image understanding,” *Psychological Review*, vol. 94, no. 2, pp. 115–147, 1987.
- [4] S. Thorpe, D. Fize, and C. Marlot, “Speed of processing in the human visual system,” *Nature*, vol. 381, no. 6582, pp. 520–522, 1996.
- [5] M.V. Peelen, L. Fei-Fei, and S. Kastner, “Neural mechanisms of rapid natural scene categorization in human visual cortex,” *Nature*, vol. 460, no. 7251, pp. 94–97, 2009.
- [6] P.C. Quinn and P.D. Eimas, “Perceptual organization and categorization in young infants,” *Advances in Infancy Research*, vol. 10, pp. 1–36, 1996.
- [7] L.B. Cohen and C.H. Cashon, “Infant perception and cognition,” *Comprehensive handbook of psychology*, vol. 6, pp. 65–89, 2003.
- [8] L.B. Cohen, “Unresolved issues in infant categorization,” *Early category and concept development*, pp. 193–209, 2003.
- [9] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.
- [10] Flickr, “<http://www.flickr.com/>,” 2010, [Online; accessed 26-January-2010].
- [11] Google image search engine, “<http://images.google.com/>,” 2010, [Online; accessed 26-January-2010].

BIBLIOGRAPHY

- [12] PicSearch image search engine, “<http://www.picsearch.com/>,” 2010, [Online; accessed 26-January-2010].
- [13] Yahoo! image search engine, “<http://images.search.yahoo.com/>,” 2010, [Online; accessed 26-January-2010].
- [14] Blinkx video search engine, “<http://www.blinkx.com/>,” 2010, [Online; accessed 26-January-2010].
- [15] Truveo video search engine, “<http://www.truveo.com/>,” 2010, [Online; accessed 26-January-2010].
- [16] VideoSurf video search engine, “<http://www.videosurf.com/>,” 2010, [Online; accessed 26-January-2010].
- [17] YouTube video database, “<http://www.youtube.com/>,” 2010, [Online; accessed 26-January-2010].
- [18] I.C. Sluimer, M. Prokop, I. Hartmann, and B.V. Ginneken, “Automated classification of hyperlucency, fibrosis, ground glass, solid and focal lesions in high resolution CT of the lung,” *Medical Physics*, vol. 33, no. 7, pp. 2610–2620, 2006.
- [19] J. Nayak, P.S. Bhat, and U.R. Acharya, “Automatic identification of diabetic maculopathy stages using fundus images,” *Journal of Medical Engineering and Technology*, vol. 33, no. 2, pp. 119–129, 2009.
- [20] K. Mombaur, “Using optimization to create self-stable human-like running,” *Robotica*, vol. 27, no. 3, pp. 79–85, 2009.
- [21] T.P. Sattar, H.L. Rodriguez, and B. Bridge, “Climbing ring robot for inspection of offshore wind turbines,” *Industrial Robot*, vol. 36, no. 4, pp. 326–330, 2009.
- [22] X.S. Xu, F.M. Sun, and G.P. Wang, “The control and optimization design of the fish-like underwater robot with the aid of the giant magnetostrictive material actuator,” *Journal of Vibration and Control*, vol. 15, no. 10, pp. 1443–1462, 2009.
- [23] F. Bullo and A.D. Lewis, “Kinematic controllability and motion planning for the snakeboard,” *IEEE Transactions on Robotics and Automation*, vol. 19, no. 3, pp. 494–498, 2003.

BIBLIOGRAPHY

- [24] R. Balasubramanian, A.A. Rizzi, and M.T. Mason, “Legless locomotion: A novel locomotion technique for legged robots,” *International Journal of Robotics Research*, vol. 27, no. 5, pp. 575–594, 2008.
- [25] Google map, “<http://maps.google.com/>,” 2010, [Online; accessed 26-January-2010].
- [26] Flash Earth, “<http://www.flashearth.com/>,” 2010, [Online; accessed 26-January-2010].
- [27] E. Hussain, K. Kim, and J. Shan, “Object-based image classification and web-mapping techniques for flood damage assessment,” *American Society for Photogrammetry and Remote Sensing Conference*, 2009.
- [28] Y.E. Shimabukuro, V. Duarte, E. Arai, R.M. Freitas, A. Lima, D.M. Vale-riano, I.F. Brown, and M.L.R. Maldonado, “Fraction images derived from Terra Modis data for mapping burnt areas in brazilian amazonia,” *International Journal of Remote Sensing*, vol. 30, no. 6, pp. 1537–1546, March 2009.
- [29] M. Brand and P. Pletscher, “A conditional random field for automatic photo editing,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–7, 2008.
- [30] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.
- [31] M.K.H. Leung and Y.H. Yang, “Dynamic two-strip algorithm in curve fitting,” *Pattern Recognition*, vol. 23, no. 1–2, pp. 69–79, 1990.
- [32] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaf-falitzky, T. Kadir, and L.V. Gool, “A comparison of affine region detectors,” *International Journal of Computer Vision*, vol. 65, no. 1–2, pp. 43–72, 2005.
- [33] K. Mikolajczyk and C. Schmid, “Indexing based on scale invariant interest points,” *IEEE International Conference on Computer Vision*, pp. 525–531, 2001.
- [34] K. Mikolajczyk and C. Schmid, “Scale and affine invariant interest point detectors,” *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

BIBLIOGRAPHY

- [35] C. Harris and M. Stevens, “A combined corner and edge detector,” *Alvey Vision Conference*, pp. 147–151, 1988.
- [36] T. Lindeberg, “Feature detection with automatic scale selection,” *International Journal of Computer Vision*, vol. 30, no. 2, pp. 79–116, 1998.
- [37] T. Kadir and M. Brady, “Scale, saliency and image description,” *International Journal of Computer Vision*, vol. 45, no. 2, pp. 83–105, 2001.
- [38] D.G. Lowe, “Object recognition from local scale-invariant features,” *IEEE International Conference on Computer Vision*, pp. 1150–1157, 1999.
- [39] D.G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [40] E. Borenstein and S. Ullman, “Class-specific, top-down segmentation,” *European Conference on Computer Vision*, pp. 639–641, 2002.
- [41] W.L. Zhao, “<http://www.cs.cityu.edu.hk/~wzhao2/lip-vireo.htm>,” 2010, [Online; accessed 25-June-2010].
- [42] J. Matas, O. Chum, M. Urban, and T. Pajdla, “<http://www.robots.ox.ac.uk/vgg/research/affine/detectors.html>,” 2010, [Online; accessed 25-June-2010].
- [43] A. Baumberg, “Reliable feature matching across widely separated views,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 774–781, 2000.
- [44] K. Mikolajczyk and C. Schmid, “An affine invariant interest point detector,” *European Conference on Computer Vision*, pp. 128–142, 2002.
- [45] T. Kadir, A. Zisserman, and M. Brady, “An affine invariant salient region detector,” *European Conference on Computer Vision*, pp. 345–357, 2004.
- [46] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions,” *British Machine Vision Conference*, pp. 384–393, 2002.
- [47] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions,” *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004.

BIBLIOGRAPHY

- [48] M. Perdoch, J. Matas, and S. Obdrzalek, “Stable affine frames on isophotes,” *IEEE International Conference on Computer Vision*, pp. 1–8, 2007.
- [49] P. Moreels and P. Perona, “Evaluation of features detectors and descriptors based on 3D objects,” *International Journal of Computer Vision*, vol. 73, no. 3, pp. 263–284, 2007.
- [50] M.J. Swain and D.H. Ballard, “Color indexing,” *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.
- [51] S. Lazebnik, C. Schmid, and J. Ponce, “A sparse texture representation using local affine regions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1265–1278, 2005.
- [52] A.E. Johnson and M. Hebert, “Using spin images for efficient object recognition in cluttered 3D scenes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 433–449, 1999.
- [53] S. Marcelja, “Mathematical description of the responses of simple cortical cells,” *Journal of the Optical Society of America*, vol. 7, no. 11, pp. 1297–1300, 1980.
- [54] J.G. Daugman, “Two-dimensional spectral analysis of cortical receptive field profile,” *Vision Research*, vol. 20, no. 10, pp. 847–856, 1980.
- [55] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.
- [56] J.J. Koenderink and A.J.V. Doorn, “Representation of local geometry in the visual system,” *Biological Cybernetics*, vol. 55, no. 6, pp. 367–375, 1987.
- [57] W.T. Freeman and E.H. Adelson, “The design and use of steerable filters,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891–906, 1991.
- [58] A. Bosch, A. Zisserman, and X. Munoz, “Scene classification via plsa,” *European Conference on Computer Vision*, pp. 517–530, 2006.
- [59] T. Geodeme, T. Tuytelaars, G. Vanacker, M. Nuttin, and L.V. Gool, “Omni-directional sparse visual path following with occlusion-robust feature tracking,” *IEEE International Conference on Computer Vision Workshop*, pp. 207–215, 2005.

BIBLIOGRAPHY

- [60] Y. Ke and R. Sukthankar, “PCA-SIFT: A more distinctive representation for local image descriptors,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 506–513, 2004.
- [61] A. Stein and M. Hebert, “Incorporating background invariance into feature-based object recognition,” *IEEE Workshop on Applications of Computer Vision*, vol. 1, no. 5–7, pp. 37, 2005.
- [62] J.M. Morel and G. Yu, “Asift: A new framework for fully affine invariant image comparison,” *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 438–469, 2009.
- [63] L.G. Roberts, *Machine perception of three-dimensional solids*, pp. 159–197, 1980.
- [64] J.M.S. Prewitt, *Object enhancement and extraction*, pp. 75–149, 1970.
- [65] I.E. Sobel, “Camera model and machine perception,” *Ph.D. thesis, Stanford University*, 1970.
- [66] D. Marr and E. Hildreth, “Theory of edge detection,” *Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.
- [67] T. Lindeberg, “Edge detection and ridge detection with automatic scale selection,” *International Journal of Computer Vision*, vol. 30, no. 2, pp. 117–154, 1998.
- [68] I. Kokkinos, “<http://www.mas.ecp.fr/vision/personnel/iasonas/code.html>,” 2010, [Online; accessed 25-June-2010].
- [69] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [70] D.R. Martin, C.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*, vol. 26, no. 5, pp. 530–549, 2004.
- [71] B. Sumengen and B.S. Manjunath, “Multi-scale edge detection and image segmentation,” 2005.

BIBLIOGRAPHY

- [72] N. Coudray, J.L. Buessler, H. Kihl, and J.P. Urban, “Multi-scale and first derivative analysis for edge detection in TEM images,” *International Conference on Image Analysis and Recognition*, pp. 1005–1016, 2007.
- [73] L. Ruihua and M. Yang, “Multiscale edge extraction using the fuzzy rules,” *Journal of Algorithms and Computational Technology*, vol. 2, no. 3, pp. 349–358, 2008.
- [74] L. Breiman, J. Friedman, C.J. Stone, and R.A. Olshen, *Classification and Regression Trees*, Wadsworth and Brooks, Monterey, CA, 1984.
- [75] V.N. Vapnik, *Statistical Learning Theory*, New York: Wiley, 1998.
- [76] P. Dollar, Z. Tu, and S. Belongie, “Supervised learning of edges and object boundaries,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1964–1971, 2006.
- [77] D.P. Huttenlocher, G.A. Klanderman, and W.A. Rucklidge, “Comparing images using the hausdorff distance,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 850–863, 1993.
- [78] H.G. Barrow, J.M. Tenenbaum, R.C. Bolles, and H.C. Wolf, “Parametric correspondence and chamfer matching: Two new techniques for image matching,” *International Joint Conference on Artificial Intelligence*, pp. 659–663, 1977.
- [79] A.M. Bronstein, M.M. Bronstein, A.M. Bruckstein, and R. Kimmel, “Analysis of two-dimensional non-rigid shapes,” *International Journal of Computer Vision*, vol. 78, no. 1, pp. 67–88, 2008.
- [80] X. Bai, X. Wang, L.J. Latecki, W.Y. Liu, and Z. Tu, “Active skeleton for non-rigid object detection,” *IEEE International Conference on Computer Vision*, 2009.
- [81] H. Blum, “Biological shape and visual science,” *Journal of Theoretical Biology*, vol. 38, no. 2, pp. 205–287, 1973.
- [82] W.P. Choi, K.M. Lam, and W.C. Siu, “Extraction of the euclidean skeleton based on a connectivity criterion,” *Pattern Recognition*, vol. 36, no. 3, pp. 721–729, 2003.

BIBLIOGRAPHY

- [83] X. Bai, L.J. Latecki, and W.Y. Liu, “Skeleton pruning by contour partitioning with discrete curve evolution,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 449–462, 2007.
- [84] D.G. Lowe, “Three-dimensional object recognition from single two-dimensional images,” *Artificial Intelligence*, vol. 31, no. 3, pp. 355–395, 1987.
- [85] X. Ren, C.C. Fowlkes, and J. Malik, “Scale-invariant contour completion using conditional random fields,” *IEEE International Conference on Computer Vision*, vol. 2, pp. 1214–1221, 2005.
- [86] I. Kokkinos, P. Maragos, and A. Yuille, “Bottom-up and top-down object detection using primal sketch features and graphical models,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1893–1900, 2006.
- [87] R.O. Duda and P.E. Hart, “Use of the Hough transformation to detect lines and curves in pictures,” *ACM Communications*, vol. 15, no. 1, pp. 11–15, 1972.
- [88] P. Meer, D. Mintz, A. Rosenfeld, and D.Y. Kim, “Robust regression methods for computer vision: A review,” *International Journal of Computer Vision*, vol. 6, no. 1, pp. 59–70, 1991.
- [89] Y. Chi, *Shape Retrieval And Matching Based On Line Segments*, Ph.D. thesis, Nanyang Technological University, 2006.
- [90] V.S. Nalwa, “Line-drawing interpretation: Straight lines and conic sections,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 4, pp. 514–529, 1988.
- [91] J. Shotton, A. Blake, and R. Cipolla, “Multi-scale categorical object recognition using contour fragments,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1270–1281, 2008.
- [92] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, and C. Bray, “Visual categorization with bags of keypoints,” *European Conference on Computer Vision Workshop*, pp. 1–22, 2004.
- [93] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification*, Wiley, 2001.
- [94] E. Nowak, F. Jurie, and B. Triggs, “Sampling strategies for bag-of-features image classification,” *European Conference on Computer Vision*, pp. 490–503, 2006.

BIBLIOGRAPHY

- [95] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer, “Generic object recognition with boosting,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 416–431, 2006.
- [96] L.V. Gool, T. Moons, and D. Ungureanu, “Affine/photometric invariants for planar intensity patterns,” *European Conference on Computer Vision*, pp. 642–651, 1996.
- [97] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, “Local features and kernels for classification of texture and object categories: A comprehensive study,” *International Journal of Computer Vision*, vol. 73, no. 2, pp. 213–238, 2007.
- [98] J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, and W.T. Freeman, “Discovering objects and their location in images,” *IEEE International Conference on Computer Vision*, pp. 370–377, 2005.
- [99] T. Hofmann, “Probabilistic latent semantic indexing,” pp. 50–57, 1999.
- [100] L.J. Li and L. Fei-Fei, “Optimol: Automatic object picture collection via incremental model learning,” *International Journal of Computer Vision*, pp. 1–22, 2009.
- [101] H. Jegou, M. Douze, and C. Schmid, “Packing bag-of-features,” *IEEE International Conference on Computer Vision*, 2009.
- [102] A. Pinz, “Object categorization,” *Foundations and Trends in Computer Graphics and Vision*, vol. 1, no. 4, pp. 255–353, 2006.
- [103] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2169–2178, 2006.
- [104] M.C. Burl, M. Weber, and P. Perona, “A probabilistic approach to object recognition using local photometry and global geometry,” *European Conference on Computer Vision*, pp. 628–641, 1998.
- [105] M. Weber, M. Welling, and P. Perona, “Towards automatic discovery of object categories,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 101–108, 2000.

BIBLIOGRAPHY

- [106] L. Fei-Fei, R. Fergus, and P. Perona, “One-shot learning of object categories,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594–611, 2006.
- [107] R. Fergus, P. Perona, and A. Zisserman, “Weakly supervised scale-invariant learning of models for visual recognition,” *International Journal of Computer Vision*, vol. 71, no. 3, pp. 273–303, 2007.
- [108] M. Weber, M. Welling, and P. Perona, “Unsupervised learning of models for recognition,” *European Conference on Computer Vision*, pp. 18–32, 2000.
- [109] R.M. Haralick and L.G. Shapiro, *Computer and Robot Vision II*, Addison-Wesley, 1993.
- [110] R. Fergus, A. Zisserman, and P. Perona, “Object class recognition by unsupervised scale-invariant learning,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 264–271, 2003.
- [111] L. Fei-Fei, R. Fergus, and P. Perona, “A bayesian approach to unsupervised one-shot learning of object categories,” *IEEE International Conference on Computer Vision*, pp. 1134–1141, 2003.
- [112] A.D. Holub, M. Welling, and P. Perona, “Hybrid generative-discriminative visual categorization,” *International Journal of Computer Vision*, vol. 77, no. 1–3, pp. 239–258, 2008.
- [113] Y. Kamiya, T. Takahashi, I. Ide, and H. Murase, “A multimodal constellation model for object category recognition,” *International Multimedia Modeling Conference*, pp. 310–321, 2009.
- [114] X. Ma and W.E.L. Grimson, “Edge-based rich representation for vehicle classification,” *IEEE International Conference on Computer Vision*, pp. 1185–1192, 2005.
- [115] J. Shotton, A. Blake, , and R. Cipolla, “Contour-based learning for object detection,” *IEEE International Conference on Computer Vision*, pp. 503–510, 2005.
- [116] A. Bar-Hillel and D. Weinshall, “Efficient learning of relational object class models,” *International Journal of Computer Vision*, vol. 7, no. 1–3, pp. 175–198, 2008.

BIBLIOGRAPHY

- [117] M. Vidal-Naquet and S. Ullman, “Object recognition with informative features and linear classification,” *IEEE International Conference on Computer Vision*, pp. 281–288, 2003.
- [118] 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*, vol. 26, no. 11, pp. 1475–1490, 2004.
- [119] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, “Learning object categories from google’s image search,” *IEEE International Conference on Computer Vision*, pp. 1816–1823, 2005.
- [120] A. Bar-Hillel, T. Hertz, and D. Weinshall, “Object class recognition by boosting a part-based model,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 702–709, 2005.
- [121] G. Carneiro and D.G. Lowe, “Sparse flexible models of local features,” *European Conference on Computer Vision*, pp. 29–43, 2006.
- [122] B. Leibe and B. Schiele, “Interleaved object categorization and segmentation,” *British Machine Vision Conference*, pp. 759–768, 2003.
- [123] B. Leibe and B. Schiele, “Scale invariant object categorization using a scale-adaptive mean-shift search,” *German Association for Pattern Recognition Symposium*, pp. 145–153, 2004.
- [124] B. Leibe, A. Leonardis, and B. Schiele, “Robust object detection by interleaving categorization and segmentation,” *International Journal of Computer Vision*, vol. 77, no. 1–3, pp. 259–289, 2008.
- [125] D.H. Ballard, “Generalizing the Hough transform to detect arbitrary shapes,” *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.
- [126] J. Gall and V. Lempitsky, “Class-specific Hough forests for object detection,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1022–1029, 2009.
- [127] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [128] L. Karlinsky, M. Dinerstein, D. Levi, and S. Ullman, “Unsupervised classification and part localization by consistency amplification,” *European Conference on Computer Vision*, pp. 321–335, 2008.

BIBLIOGRAPHY

- [129] F. Jurie and B. Triggs, “Creating efficient codebooks for visual recognition,” *IEEE International Conference on Computer Vision*, pp. 604–610, 2005.
- [130] T. Quack, V. Ferrari, B. Leibe, and L.V. Gool, “Efficient mining of frequent and distinctive feature configurations,” *IEEE International Conference on Computer Vision*, pp. 1–8, 2007.
- [131] R. Agrawal, T. Imielinski, and A. Swami, “Mining association rules between sets of items in large databases,” *ACM International Conference on Management of Data*, pp. 207–216, 1993.
- [132] Y.J. Lee and K. Grauman, “Foreground focus: Unsupervised learning from partially matching images,” *International Journal of Computer Vision*, vol. 85, no. 2, pp. 143–166, 2009.
- [133] E. Mach, *Analysis of Sensations (1897)*, Dover Edition, 1959.
- [134] A. Slater and V. Morison, “Shape constancy and slant perception at birth,” *Perception*, vol. 14, no. 3, pp. 337–344, 1985.
- [135] B.C. Gundeep, “Basic-level and superordinate-like categorical representations in early infancy,” *Cognition*, vol. 60, no. 2, pp. 105–141, 1996.
- [136] B. Landau, L. Smith, and S. Jones, “Object perception and object naming in early development,” *Trends Cognitive Science*, vol. 2, no. 1, pp. 19–24, 1998.
- [137] P.C. Quinn, P.D. Eimas, and M. Tarr, “Perceptual categorization of cat and dog silhouettes by 3 to 4 month old infants,” *Journal of Experimental Child Psychology*, vol. 79, no. 1, pp. 78–94, 2001.
- [138] P.M. Kumar, P.H.S. Torr, and A. Zisserman, “Extending pictorial structures for object recognition,” *British Machine Vision Conference*, pp. 789–798, 2004.
- [139] P.F. Felzenszwalb and D.P. Huttenlocher, “Efficient matching of pictorial structures,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 66–73, 2000.
- [140] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference*, 1998.

BIBLIOGRAPHY

- [141] K.B. Sun and B.J. Super, “Classification of contour shapes using class segment sets,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 727–733, 2005.
- [142] X. Bai, W.Y. Liu, and Z. Tu, “Integrating contour and skeleton for shape classification,” *IEEE International Conference on Computer Vision Workshop*, 2009.
- [143] P.F. Felzenszwalb and J.D. Schwartz, “Hierarchical matching of deformable shapes,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.
- [144] C. Lu, L.J. Latecki, N. Adluru, X. Yang, and H. Ling, “Shape guided contour grouping with particle filters,” *IEEE International Conference on Computer Vision*, 2009.
- [145] K. Schindler and D. Suter, “Object detection by global contour shape,” *Pattern Recognition*, vol. 41, no. 12, pp. 3736–3748, 2008.
- [146] G. Heitz, G. Elidan, B. Packer, and D. Koller, “Shape-based object localization for descriptive classification,” *International Journal of Computer Vision*, vol. 84, no. 1, pp. 40–62, 2009.
- [147] N.H. Trinh and B.B. Kimia, “Category-specific object recognition and segmentation using a skeletal shape model,” *British Machine Vision Conference*, 2009.
- [148] J. Shotton, A. Blake, and R. Cipolla, “Efficiently combining contour and texture cues for object recognition,” *British Machine Vision Conference*, 2008.
- [149] A. Opelt, A. Pinz, and A. Zisserman, “Learning an alphabet of shape and appearance for multi-class object detection,” *International Journal of Computer Vision*, vol. 80, no. 1, pp. 16–44, 2008.
- [150] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
- [151] P. David and D. DeMenthon, “Object recognition in high clutter images using line features,” *IEEE International Conference on Computer Vision*, pp. 1581–1588, 2005.

BIBLIOGRAPHY

- [152] X. Ren, “Learning and matching line aspects for articulated objects,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.
- [153] Y. Gao and M.K.H. Leung, “Face recognition using line edge map,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, pp. 764–779, 2002.
- [154] F. Li and M.K.H. Leung, “Hierarchical identification of palmprint using line-based Hough transform,” *IEEE International Conference on Pattern Recognition*, pp. 149–152, 2006.
- [155] V. Ferrari, T. Tuytelaars, and L.V. Gool, “Object detection by contour segment networks,” *European Conference on Computer Vision*, pp. 14–28, 2006.
- [156] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid, “Groups of adjacent contour segments for object detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 1, pp. 36–51, 2008.
- [157] V. Ferrari, F. Jurie, and C. Schmid, “From images to shape models for object detection,” *International Journal of Computer Vision*, vol. 87, no. 3, pp. 284–303, 2009.
- [158] K. Koffka, *Principles of Gestalt Psychology*, New York: Harcourt, Brace and Company, 1935.
- [159] R. Fergus, P. Perona, and A. Zisserman, “A visual category filter for Google images,” *European Conference on Computer Vision*, pp. 242–256, 2004.
- [160] R. Fergus, P. Perona, and A. Zisserman, “A sparse object category model for efficient learning and exhaustive recognition,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 380–387, 2005.
- [161] Y. Chi and M.K.H. Leung, “Part-based object retrieval in cluttered environment,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 890–895, 2007.
- [162] S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition using shape contexts,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

BIBLIOGRAPHY

- [163] R. Jonker and A. Volegenant, “A shortest augmenting path algorithm for dense and sparse linear assignment problems,” *Computing*, vol. 38, no. 4, pp. 325–340, 1987.
- [164] Q. Zhu, L. Wang, Y. Wu, and J. Shi, “Contour context selection for object detection: A set-to-set contour matching approach,” *European Conference on Computer Vision*, pp. 774–787, 2008.
- [165] Q.H. Zhu, G. Song, and J.B. Shi, “Untangling cycles for contour grouping,” *IEEE International Conference on Computer Vision*, pp. 1–8, 2007.
- [166] F. Jurie and C. Schmid, “Scale invariant shape features for recognition of object categories,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 90–96, 2004.
- [167] O. Carmichael and M. Hebert, “Shape-based recognition of wiry objects,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 12, pp. 1537–1552, 2004.
- [168] A.C. Berg, T.L. Berg, and J. Malik, “Shape matching and object recognition using low distortion correspondence,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 26–33, 2005.
- [169] A.C. Berg and J. Malik, “Geometric blur for template matching,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 607–614, 2001.
- [170] K. Mikolajczyk, A. Zisserman, and C. Schmid, “Shape recognition with edge-based features,” *British Machine Vision Conference*, pp. 779–788, 2003.
- [171] C.E. Brodley and P.E. Utgoff, *Multivariate Decision Trees*, Machine Learning, University of Massachusetts, 1995.
- [172] C.P. Erick and C. Kamath, “Inducing oblique decision trees with evolutionary algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 1, pp. 54–68, 2003.
- [173] J.R. Quinlan, “Induction of decision trees,” *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.
- [174] J.R. Quinlan, *C4.5: Programs for machine learning*, Morgan Kaufmann Publishers, 1993.

BIBLIOGRAPHY

- [175] K. Grabczewski and W. Duch, “A general purpose separability criterion for classification systems,” *Conference on Neural Networks and Their Applications*, pp. 203–208, 1999.
- [176] B. Schoelkopf, C.J.C. Burges, and A.J. Smola, *Advances in kernel methods: Support vector learning*, 1999.
- [177] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 1, pp. 21–27, 1967.
- [178] Y. Freund and R.E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.
- [179] Y. Freund, “Boosting a weak learning algorithm by majority,” *Information and Computation*, vol. 121, no. 2, pp. 256–285, 1995.
- [180] J. Friedman, T. Hastie, and R. Tibshirani, “Additive logistic regression: A statistical view of boosting,” *Annals of Statistics*, vol. 38, no. 2, pp. 337–407, 2000.
- [181] M. Nakamura, H. Nomiya, and K. Uehara, “Improvement of boosting algorithm by modifying the weighting rule,” *Annals of Mathematics and Artificial Intelligence*, vol. 41, no. 1, pp. 95–109, 2004.
- [182] A. Demiriz, K.P. Bennett, and S.T. John, “Linear programming boosting via column generation,” *Machine Learning*, vol. 46, no. 1–3.
- [183] R.A. Servedio, “Smooth boosting and learning with malicious noise,” *Journal of Machine Learning Research*, vol. 4, pp. 633–648, 2003.
- [184] J.G. Wang, E. Sung, and R. Venkateswarlu, “Eye gaze estimation from a single image of one eye,” *IEEE International Conference on Computer Vision*, pp. 136–143, 2003.
- [185] M. Ferri, F. Mangili, and G. Viano, “Projective pose estimation of linear and quadratic primitives in monocular computer vision,” *Computer vision graphics and image processing: Image Understanding*, vol. 58, no. 1, pp. 66–84, 1993.

BIBLIOGRAPHY

- [186] T. Ellis, A. Abbood, and B. Brillault, "Ellipse detection and matching with uncertainty," *Image and Vision Computing*, vol. 10, no. 5, pp. 271–276, 1992.
- [187] S. Masciangelo, "3-d cues from a single view: detection of elliptical arcs and model based perspective backprojection," *British Machine Vision Conference*, pp. 223–228, 1990.
- [188] C.A. Rothwell, A. Zisserman, D.A. Forsyth, and J.L. Mundy, "Planar object recognition using projective shape representation," *International Journal of Computer Vision*, vol. 16, no. 1, pp. 57–99, 1995.
- [189] S. Kothari, Q. Chaudry, and M.D. Wand, "Automated cell counting and cluster segmentation using concavity detection and ellipse fitting techniques," *IEEE International Symposium on Biomedical Imaging*, pp. 795–798, 2009.
- [190] S. Badiei, S.E. Salcudean, J. Varah, and W.J. Morris, "Prostate segmentation in 2d ultrasound images using image warping and ellipse fitting," *IEEE International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 17–24, 2006.
- [191] P. Hough, "Method and means for recognizing complex patterns," *US Patent*, 3069654, 1962.
- [192] H.K. Yuen, J. Illingworth, and J. Kittler, "Detecting partially occluded ellipses using the Hough transform," *Image and Vision Computing*, vol. 7, no. 1, pp. 31–37, 1989.
- [193] C.T. Ho and L.H. Chen, "A fast ellipse/circle detector using geometric symmetry," *Pattern Recognition*, vol. 28, no. 1, pp. 117–124, 1995.
- [194] A.Y.-S. Chia, M.K.H. Leung, H.L. Eng, and S. Rahardja, "Ellipse detection with Hough transform in one dimensional parametric space," *IEEE International Conference on Image Processing*, pp. 333–336, 2007.
- [195] N. Guil and E.L. Zapata, "Lower order circle and ellipse Hough transform," *Pattern Recognition*, vol. 30, no. 10, pp. 1729–1744, 1997.
- [196] S.C. Zhang and Z.Q. Liu, "A robust, real-time ellipse detector," *Pattern Recognition*, vol. 38, no. 2, pp. 273–287, 2005.
- [197] N. Kiryati, Y. Eldar, and A.M. Bruckstein, "A probabilistic Hough transform," *Pattern Recognition*, vol. 24, no. 4, pp. 303–316, 1991.

BIBLIOGRAPHY

- [198] L. Xu, E. Oja, and P. Kultanen, “A new curve detection method: Randomized Hough transform,” *Pattern Recognition Letters*, vol. 11, no. 5, pp. 331–338, 1990.
- [199] R.A. McLaughlin, “Randomized Hough transform: Improved ellipse detection with comparison,” *Pattern Recognition Letters*, vol. 19, no. 3–4, pp. 299–305, 1998.
- [200] T.C. Chen and K.L. Chung, “An efficient randomized algorithm for detecting circles,” *Computer Vision and Image Understanding*, vol. 83, no. 2, pp. 172–191, 2001.
- [201] W. Lu and J. Tan, “Detection of incomplete ellipse in images with strong noise by iterative randomized Hough transform (IRHT),” *Pattern Recognition*, vol. 41, no. 4, pp. 1268–1279, 2008.
- [202] R.A. McLaughlin, “Intelligent algorithms for finding curves and surfaces in real world data,” *Ph.D. Thesis*, 2000.
- [203] Y.C. Cheng, “The distinctiveness of a curve in a parameterized neighborhood: Extraction and applications,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1215–1222, 2006.
- [204] P.Y. Yin, “A new circle/ellipse detector using genetic algorithms,” *Pattern Recognition Letters*, vol. 20, no. 7, pp. 731–740, 1999.
- [205] J. Yao, N. Kharna, and P. Grogono, “Fast, robust GA-based ellipse detection,” *IEEE International Conference on Pattern Recognition*, pp. 859–862, 2004.
- [206] S. Procter and J. Illingworth, “A comparison of the randomized Hough transform and a genetic algorithm for ellipse extraction,” *Pattern Recognition in Practice IV: Multiple Paradigms, Comparative Studies and Hybrid Systems*, pp. 449–460, 1994.
- [207] Y. Qiao and S.H. Ong, “Arc-based evaluation and detection of ellipses,” *Pattern Recognition*, vol. 40, no. 7, pp. 1990–2003, 2007.
- [208] K.U. Kasemir and K. Betzler, “Detecting ellipses of limited eccentricity in images with high noise levels,” *Image and Vision Computing*, vol. 21, no. 2, pp. 221–227, 2003.

BIBLIOGRAPHY

- [209] W. Gander, G.H. Golub, and R. Strebel, “Least squares fitting of circles and ellipses,” *BIT Numerical Mathematics*, vol. 34, no. 4, pp. 558–578, 1994.
- [210] P.L. Rosin, “A note on the least squares fitting of ellipses,” *Pattern Recognition Letters*, vol. 14, no. 10, pp. 799–808, 1993.
- [211] G. Danuser and M. Stricker, “Parametric model fitting: from inlier characterization to outlier detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 263–280, 1998.
- [212] P.J. Huber, *Robust Statistics*, Wiley, New York, 1981.
- [213] A. Fitzgibbon, M. Pilu, and R.B. Fisher, “Direct least squares fitting of ellipses,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 476–480, 1999.
- [214] E. Kim, M. Haseyama, and H. Kitajima, “Fast and robust ellipse extraction from complicated images,” *IEEE International Conference on Information Technology and Applications*, pp. 357–362, 2002.
- [215] F. Mai, Y.S. Hung, H. Zhong, and W.F. Sze, “A hierarchical approach for fast and robust ellipse extraction,” *Pattern Recognition*, vol. 41, no. 8, pp. 2512–2524, 2008.
- [216] G. Griffin, A. Holub, and P. Perona, “Caltech-256 Object Category Dataset,” *California Institute of Technology Technical Report*, , no. 24, 2007.
- [217] D. Eberly, “Distance from a point to an ellipse in 2d,” *Magic Software, Inc*, 2004.
- [218] R. Fabbri, L.D.F. Costa, J.C. Torelli, and O.M. Bruno, “2D euclidean distance transform algorithms: A comparative survey,” *ACM Computing Surveys*, vol. 40, no. 1, pp. 1–44, 2008.
- [219] H.N. Gabow, “Data structure for weighted matching and nearest common ancestors with linking,” *Annual ACM Symposium on Discrete Algorithms*, pp. 434–443, 1990.
- [220] A.Y.-S. Chia, S. Rahardja, D. Rajan, and M.K.H. Leung, “Structural descriptors for category level object detection,” *IEEE Transactions on Multimedia*, vol. 11, no. 8, pp. 1407–1421, 2009.

BIBLIOGRAPHY

- [221] F. Perronnin, “Universal and adapted vocabularies for generic visual categorization,” *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 30, no. 7, pp. 1243–1256, 2008.
- [222] S. Belongie, J. Malik, and J. Puzicha, “Matching shapes,” *IEEE International Conference on Computer Vision*, pp. 454–461, 2001.
- [223] D. Gavrilu, “A bayesian, exemplar-based approach to hierarchical shape matching,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1408–1421, 2007.
- [224] P. Tan, M. Steinback, and V. Kumar, *Introduction to Data Mining*, Addison-Wesley, 2005.
- [225] T. Ishioka, “Extended k-means with an efficient estimation of the number of clusters,” *International Conference on Intelligent Data Engineering and Automated Learning, Data Mining, Financial Engineering, and Intelligent Agents*, pp. 17–22, 2000.
- [226] D. Pelleg and A. Moore, “X-means: Extending k-means with efficient estimation of the number of clusters,” *International Conference on Machine Learning*, pp. 727–734, 2000.
- [227] Y. Cheng, “Mean shift mode seeking and clustering,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, 1995.
- [228] D. Dementhon and R. Megret, “Spatio-temporal segmentation of video by hierarchical mean shift analysis,” *University of Maryland, College Park, Technical Report*, 2002.
- [229] E. Seemann, B. Leibe, and B. Schiele, “Multi-aspect detection of articulation objects,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1582–1588, 2006.
- [230] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 886–893, 2005.
- [231] D.Q. Zhang and S.F. Chang, “A generative-discriminative hybrid method for multi-view object detection,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2017–2024, 2006.

BIBLIOGRAPHY

- [232] B. Ommer and J.M. Buhmann, “Learning the compositional nature of visual objects,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.
- [233] W. Zhang, G. Zelinsky, and D. Samaras, “Real-time accurate object detection using multiple resolutions,” *IEEE International Conference on Computer Vision*, pp. 1–8, 2007.
- [234] The PASCAL Visual Object Classes Homepage, “<http://pascallin.ecs.soton.ac.uk/challenges/voc/voc2009/index.html>,” 2009, [Online; accessed 20-May-2009].
- [235] B. Wu and R. Nevatia, “Detection and segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses,” *International Journal of Computer Vision*, vol. 82, no. 2, pp. 185–204, 2009.
- [236] P. Carbonetto, G. Dorko, C. Schmid, H. Kunk, and N.D. Freitas, “A semi-supervised learning approach to object recognition with spatial integration of local features and segmentation cues,” *Toward Category-Level Object Recognition*, pp. 277–300, 2006.
- [237] D. Crandall and D. Huttenlocher, “Weakly supervised learning of part-based spatial models for visual object recognition,” *European Conference on Computer Vision*, pp. 16–29, 2006.
- [238] Y. Chen, L.L. Zhu, A. Yuille, and H. Zhang, “Unsupervised learning of probabilistic object models (POMs) for object classification, segmentation and recognition using knowledge propagation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1747–1761, 2009.
- [239] L. Zhu, Y. Chen, and A. Yuille, “Unsupervised learning of probabilistic grammar-markov models for object categories,” *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 31, no. 1, pp. 114–128, 2009.
- [240] B. Leibe, E. Seemann, and B. Schiele, “Pedestrian detection in crowded scenes,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 878–885, 2005.
- [241] D.G. Lowe, “<http://www.cs.ubc.ca/~lowe/keypoints/>,” 2008, [Online; accessed 28-September-2008].

BIBLIOGRAPHY

- [242] LabelMe dataset, “<http://labelme.csail.mit.edu/>,” 2010, [Online; accessed 3-March-2010].
- [243] S. Savarese and L. Fei-Fei, “View synthesis for recognizing unseen poses of object classes,” *European Conference on Computer Vision*, pp. 602–615, 2008.
- [244] D. Hoiem, C. Rother, and J. Winn, “3D layoutcrf for multi-view object class recognition and segmentation,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.