To the Graduate Council:

I am submitting herewith a dissertation written by Chang Cheng entitled "Scene Segmentation and Object Classification for Place Recognition." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Electrical Engineering.

Mongi A. Abidi, Major Professor

We have read this dissertation
and recommend its acceptance:

Seddik M. Djouadi

Andreas Koschan

Hairong Qi

Timothy M. Young

Accepted for the Council:

Carolyn R. Hodges
Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Scene Segmentation and Object Classification for Place Recognition

Dissertation
Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Chang Cheng
August 2010

Copyright © 2010 by The Graduate School
The University of Tennessee, Knoxville
All rights reserved.

Copies of this document may be printed from this website for
personal use without permission.

Acknowledgements

This dissertation could not have been completed without the inspiration and support from many people. First and foremost, I would like to express my deepest thanks to my parents, Xuhua Tu and Zizu Cheng. Without their inspiration and sacrifice, I would never have achieved this lifelong dream. I also own a special thank to my brother, Qiang Cheng, who have gave me tremendous supports and encouragements for all these years. For each of them, I am forever gratitude.

I additionally would like to thank my advisor, Dr. Mongi Abidi. His willingness to support my work and his inspiration and guidance throughout my studies has helped me to develop my skills as a researcher. I thank him for giving me the instructions and suggestions that have been indispensable to the completion of this dissertation. I can never thank Dr. David L. Page enough. His advice and counsel over the years have had a tremendous impact on my research and myself as a person. Special thanks indeed go to Dr. Andreas Koschan, whose regular technical comments provided me valuable guidance during these last few years. I would further like to thank the other members of my committee, Dr. Seddik M. Djouadi, Dr. Hairong Qi and Dr. Timothy M. Young, for their invaluable advice that helped to improve this work. I greatly appreciate their time and input to this dissertation.

Within the Imaging, Robotics and Intelligent Systems Laboratory (IRIS lab), specially, I would like to give my sincerest thank to Chung-hao for his constant encouragement and helpful suggestions on my work. I also own many thanks to my fellow graduate students, Rangan, Roselyne, Jessica (YaoYi), Jacob, Michael, Zhiyu, Wei, Harishwaran (Hari), Muharrem for their help and support. It was a pleasure working with you all. The administrative staff Justin Acuff deserves a special mention. Throughout my work, Justin has worked miracles by solving various hardware and software issues.

Finally, I must express my appreciation to the many friends. I sincerely thank Yuan and Dayu. Your friendships and encouragements through both good times and hard times have been a source of strength for me. I would further like to thank Shaotao, Wei, Qian, Teng, Ling, Xia, Jun, Yun, Huadong, Siyuan and many others. I am profoundly grateful for your friendships as well.

Sincere thanks to you all.

Abstract

This dissertation addresses the place recognition and loop detection problem in large scale outdoor environments. It is noticeable that humans are capable of recognizing places with ease even in large complex environments. Many psychological works support that humans perceive a scene based on the perception of objects. Instead of creating a detailed representation of all the objects in a scene, human visual systems build an economic scene representation by putting emphasis on the extraction of a few key 'aspects' of the scene information, such as an inventory of salient objects and the layout of these objects, etc. This economic representation results in an enormous saving of processing and memory resources, which plays a key role for the success of human visual system on place recognition.

This dissertation tries to solve the place recognition and loop closing problem in a way similar to human visual system. First, a novel image segmentation algorithm is developed. The image segmentation algorithm is based on a Perceptual Organization model, which allows the image segmentation algorithm to 'perceive' the special structural relations among the constituent parts of an unknown object and hence to group them together without object-specific knowledge.

Then a new object recognition method is developed. Based on the fairly accurate segmentations generated by the image segmentation algorithm, an informative object description that includes not only the appearance (colors and textures), but also the parts layout and shape information is built. Then a novel feature selection algorithm is developed. The feature selection method can select a subset of features that best describes the characteristics of an object class. Classifiers trained with the selected features can classify objects with high accuracy.

In next step, a subset of the salient objects in a scene is selected as landmark objects to label the place. The landmark objects are highly distinctive and widely visible. Each landmark object is represented by a list of SIFT descriptors extracted from the object surface. This object representation allows us to reliably recognize an object under certain viewpoint changes. To achieve efficient scene-matching, an indexing structure is developed. Both texture feature and color feature of objects are used as indexing features. The texture feature and the color feature are viewpoint-invariant and hence can be used to effectively find the candidate objects with similar surface characteristics to a query object. Experimental results show that the object-based place recognition and loop detection method can efficiently recognize a place in a large complex outdoor environment.

Table of contents

Table of contents ..... **v**
List of figures ..... **vii**
List of tables ..... **ix**
1 Introduction **1**
1.1 Motivation 4
1.2 Contributions 8
1.3 Document organization 9
2 Related work **10**
2.1 Image segmentation 10
2.1.1 Region based methods 10
2.1.2 Contour-based segmentation methods 12
2.1.3 Boundary detection approaches 12
2.1.4 Class segmentation methods 13
2.1.5 Image segmentation methods based on Gestalt laws 13
2.2 Object classification and feature selection 14
2.2.1 Object classification methods 14
2.2.2 Feature selection 16
2.3 Simultaneous localization and mapping (SLAM) and place recognition 17
2.3.1 Robot mapping and SLAM 17
2.3.2 Data association and loop closing problem for SLAM 21
2.3.3 Vision-based methods 22
2.3.4 Object representation methods 23
2.3.5 Content-based image retrieval technologies 24
3 Image segmentation based on Perceptual Organization **27**
3.1 Perceptual Organization phenomenon 28
3.2 Gestalt psychology and Perceptual Organization 29
3.3 Gestalt cues in real-world objects 31
3.4 Background identification for outdoor scene images 35
3.5 Perceptual Organization model 38
3.6 Get good spatial support for object parts 45
3.7 Image segmentation algorithm 45
3.8 Discussion 47
3.9 Summary 48
4 Object classification based on appearance, parts layout and shape **50**

4.1 Building informative object description 50
4.1.1 Build appearance features 52
4.1.2 Build parts layout features 53
4.1.3 Build shape features 55
4.2 Feature selection 57
4.3 Summary 59
5 Object-based place recognition and loop closing 60
5.1 Scale Invariant Feature Transform (SIFT) 61
5.2 Local feature based object representation 63
5.3 Landmark object detection and scene representation 64
5.4 Color feature 65
5.4.1 Color representation 66
5.4.2 Color invariance 67
5.5 Texture feature 67
5.6 Range tree database: combination of color and texture features 70
5.7 Place recognition and loop closing algorithm 72
5.8 Summary 73
6 Experimental results 74
6.1 Image segmentation 74
6.1.1 Evaluation measures 74
6.1.2 LabelMe database 76
6.1.3 Gould09 database 83
6.1.4 Geometric Context (GC) database 87
6.1.5 Qualitative assessment 90
6.2 Object classification 92
6.2.1 Experimental setup 92
6.2.2 Results 92
6.3 Place recognition and loop closing 99
6.3.1 Experimental setup 99
6.3.2 Results 99
6.4 Summary 101
7 Conclusion 106
7.1 Summary of contribution 106
7.2 Directions for future research 108
7.2.1 Improvement on the Perceptual Organization model 108
7.2.2 Object recognition based on semantic parts 108
7.2.3 Place recognition for dynamic environments 109
7.2.4 Other applications 109
7.3 Discussion with closing remarks 110
Bibliography 112

List of figures

Figure 1.1. An architecture for the visual perception of scenes, from [Ronald00]..... 3
Figure 1.2 An illustration of the big loop closing problem for SLAM from [Newman05]..... 4
Figure 1.3 An example of large number of affine covariant visual features visible at many viewpoints in outdoor environments. 5
Figure 1.4 An example that recognizing place by comparing two images based on local feature matching has difficulty dealing with dynamic environments. 6
Figure 1.5 The block diagram for our object-based place recognition and loop closing system. 8
Figure 2.1 Some examples of different image segmentation methods 11
Figure 2.2 An example of using texton for object classification. 15
Figure 2.3 An example of an occupied grid map for an office environment. From [Burgard96] 18
Figure 2.4 An example of a feature map. from [Bailey02]..... 19
Figure 2.5 An example of a topological map. from [Jefferies05]..... 20
Figure 3.1 A synthetic example of Perceptual Organization phenomena..... 29
Figure 3.2 Gestalt laws.. 30
Figure 3.3 Symmetry law in real-world objects..... 32
Figure 3.4 Continuation law in real-world objects. 34
Figure 3.5 Convexity law in real-world objects..... 34
Figure 3.6 Proximity law in real-world objects. 36
Figure 3.7 Similarity law in real-world objects. 36
Figure 3.8 examples of background and structured objects..... 37
Figure 3.9 (a) Examples of notch from [Vasselle93]..... 40
Figure 3.10 Examples of shape regularity. 40
Figure 3.11 (a) Symmetry relations: the red dots indicate the centroids of the components.. 42
Figure 3.12 Example of Convexity relation..... 42
Figure 3.13 Get spatial support for object parts..... 46
Figure 3.14 Illustration of our segmentation pipeline..... 49
Figure 4.1 Illustration of object classification pipeline. 51
Figure 4.2 Illustration of building appearance features.. 52
Figure 4.3 Illustration of building parts-layout features. 54
Figure 4.4 Examples of the solutions of poisson equation for silhouettes..... 56
Figure 4.5 An example of a Bayesian Network. 57
Figure 4.6 Illustration of our feature selection method..... 58
Figure 5.1 This figure illustrates SIFT keypoint detection and keypoint descriptor generation, from [Lowe04]. 62

Figure 5.2 Examples of object representation..... 63
Figure 5.3 Examples of landmark objects detection..... 65
Figure 5.4 (a) Illustration of the energy distribution of the incident light..... 68
Figure 5.5 Illustration of a range tree search.. 71
Figure 6.1 Region-based segmentation measurements..... 77
Figure 6.2 Boundary-based evaluation results..... 79
Figure 6.3 Region-based evaluation results..... 79
Figure 6.4 Comparison of the four tested methods..... 80
Figure 6.5 Examples of our POM segmentation algorithm on different environments..... 81
Figure 6.6 Examples of the object classes that our POM segmentation algorithm can handle.. 82
Figure 6.7 Examples of Gould09 database. 85
Figure 6.8 The performance of the baseline method of Gould08 (masked by a red arrow) is comparable to three other class segmentation methods on MSRC-21 database based on pixel-level accuracy..... 85
Figure 6.9 Examples of our POM segmentation algorithm on Geometic Context dataset..... 86
Figure 6.10 Examples of Geometic Context (GC) database..... 88
Figure 6.11 Examples of our POM segmentation algorithm on Geometic Context dataset... 89
Figure 6.12 Examples of where our POM segmentation algorithm does well. 91
Figure 6.13 Examples of where our POM segmentation algorithm makes mistakes. 91
Figure 6.14 Examples of MSRC-21 database [Shotton09]..... 94
Figure 6.15 Examples of car, building and face images in MSRC-21 database [Shotton09]. 97
Figure 6.16 Classification accuracy confusion matrixs. 98
Figure 6.17 Testing environment..... 102
Figure 6.18 An Example of loop closing..... 103
Figure 6.19. Examples of loop detection under scene changes. 104
Figure 6.20 Query efficiency: SIFT database vs. range tree..... 105
Figure 7.1 Summary of contributions. 107

List of tables

Table 6.1 Segmentation accuracy score on LabelMe dataset 78
Table 6.2 Segmentation accuracy score on Gould09 dataset..... 83
Table 6.3 Summary of segmentation score on GC dataset 90
Table 6.4 Summary of results of background objects classification accuracy. 95
Table 6.5 Summary of results obtained by different combination of features on classifying car, building and face. 95
Table 6.6 Summary of results obtained by different feature selection methods on classifying car, building and face. 95
Table 6.7 Summary of results obtained by different classification methods on classifying car, building and body..... 96
Table 6.8 Storage efficiency 102

1 Introduction

Many people may have experienced similar situations like this in their lives – when you are exploring a strange environment without a map at hand, after a while, you may move back to the point where you start. Once that happens, you can immediately find that you have visited the place before. It seems that our visual system creates a stable and detailed representation of each place we have visited and stores it in our brain. When we revisit a place, we can quickly match the detailed representation stored in our brain to the current scene and hence recognize the place accordingly. Our visual systems can help us to recognize many places with ease even in large complex environments. Why is our visual system so powerful on place recognition and how does our visual system really work?

The seemingly easy task actually requires many complex vision activities in our brain. According to recent work [Rensink00], human visual systems do not keep a detailed representation of a scene in the brain at all. This is because the gathering of visual information is done using a retina that has high resolution only over a few degrees of visual angle. Therefore, a complete representation of a scene requires the contents of individual eye fixations to be integrated via a high-capacity visual buffer [Feldman85, Trehub91]. However, many experimental works [Irwin96, Simons97] do not find evidence that such an integrative visual buffer exists – for example, changes in an image of scene become difficult to detect during a blink, eye movement or other such interruption. This "change blindness" shows that people are bad at accumulating visual detail. Otherwise, change would be easily detected by comparing immediate visual input with the contents of the visual buffer. The fact that change blindness can be induced under various conditions indicates that dropping visual details may be central to the way people represent the world around them.

A large body of evidence [Rensink00, Moore98, Rufin03] supports that visual systems perceive scenes based on the perception of objects. Essential properties of an object include the requirement that it should be discrete, be differentiated from its background and be an individual, something that cannot be divided without losing its integrity. The whole perception process of scenes is stated as the follows [Rensink00]: the period of first a few hundred milliseconds is called "early vision" stage. Early vision refers to the aspect of low-level processing carried out regardless of higher-level context. Low-level processes are concerned with separating out the various physical factors in the scene that give rise to the pattern of luminance intensities in the image. The output of early vision is a set of "proto-objects". Proto-objects refer to relatively complex assemblies of fragments that correspond to localized structures in the world. Many works [Rensink95, Rensink98] suggest that proto-objects are the lowest-level structures directly accessible to attention, with much of their

underlying detail being accessed only by deliberate effort. The proto-objects only have limited coherence in space and time. Right after proto-objects in a scene are formed by low-level processes, high-level processes start. First, the abstract meaning, or gist of the scene (e.g. whether the scene is a city, harbor, picnic, etc.) is extracted. At the same time, another important aspect of scene structure called layout is also extracted. Layout refers to the spatial arrangement of the objects in a scene. The gist and layout are then used to prioritize attention, directing it to the objects that are most important in the context. Visual system usually focuses attention on one object at a time [Deubel96, Garavan98, Rensink98]. Once an object gets attended, the constituent parts of the object are grouped together and a summary description of the object is formed, like its size, overall shape, dominant colors, etc. Among many discrete objects in a scene, only a few of them will get attention. For the attended objects, only a subset of them (4~6) will be explicitly represented in short-term visual memory [Rufin03]. Many other objects will be ignored or forgotten soon. Finally, an abstract scene schema is formed. Scene schemas are long-term structures that may last indefinitely. Scene schemas are believed to include an inventory of objects that are explicitly represented in short-term memory, along with the relative locations of these objects. Figure 1.1 shows the architecture of the visual perception of scenes.

Therefore, our visual systems take a very elegant strategy to process the large amounts of information surrounding us. Instead of creating a detailed representation of all the objects in a scene, our visual systems build a more economic scene representation by putting emphasis on the extraction of a few key 'aspects' of the scene information. This economic representation results in an enormous saving of processing and memory resources, which plays a key role for the success of our visual system on place recognition.

Now we want to ask if we can build a similar visual system for a robot? When a robot is moving around in a large complex environment, can its visual system also detect the various objects in a scene? Can its visual system also create an economic representation for a scene by extracting a few key aspects of the most important objects in that scene? Can its visual system also recognize a place by efficiently and accurately matching the objects stored in its "long-term memory" to the objects detected in the current scene? The answers to these questions are the goal of this dissertation.

The key for vision-based place recognition is of objects perception (segmentation) and object recognition. The two problems are the fundamental unsolved problems in computer vision. The two problems are tightly related. Object perception (segmentation) tells where objects are in a scene while object recognition tells what the percept objects are. Not only for place recognition, many other computer vision tasks like scene categorization, content-based image retrieval etc., essentially relay on reliable object perception and recognition. We have developed a novel image segmentation and object recognition system. Based on these two works, we develop an object-based place recognition and loop detection method. Like the human visual system, our place recognition method also operates on object-level. Experimental result shows that our method can recognize places efficiently and accurately in a large complex outdoor environment.

This dissertation presents the details of our image segmentation, object recognition and object-based place recognition and loop detection methods. The remainder of this chapter outlines the motivation for this research in section 1.1. Section 1.2 gives the pipeline of our

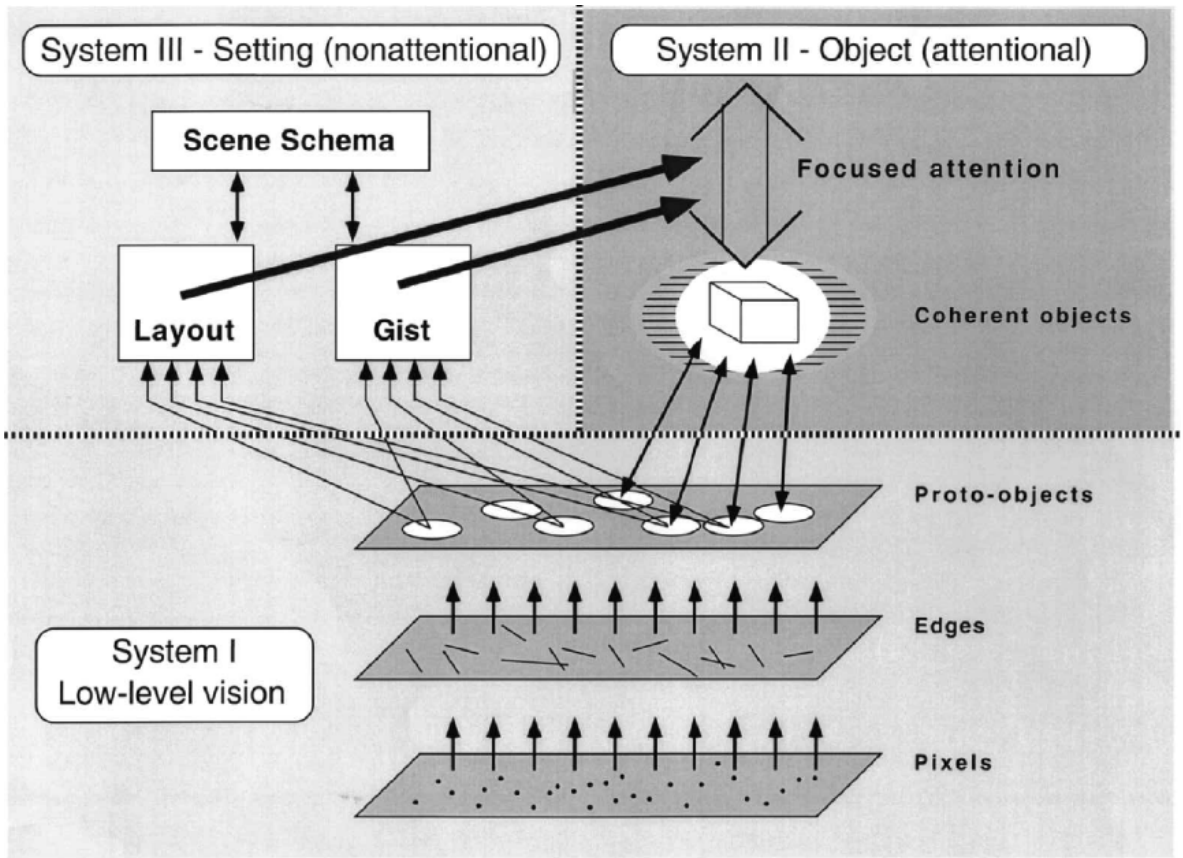


Figure 1.1. An architecture for the visual perception of scenes, from [Ronald00]. It is suggested that the visual perception of scenes may be carried out via the interaction of three different systems. System I: Early-level process produce volatile proto-objects (object parts). System II: Focused attention "assemble" different object parts to form an individuated object with both temporal and spatial coherence. System III: Setting information – obtained via a non-attentional stream – guides the allocation of focused attention to various parts of the scene, and allows priorities to the various possible objects.

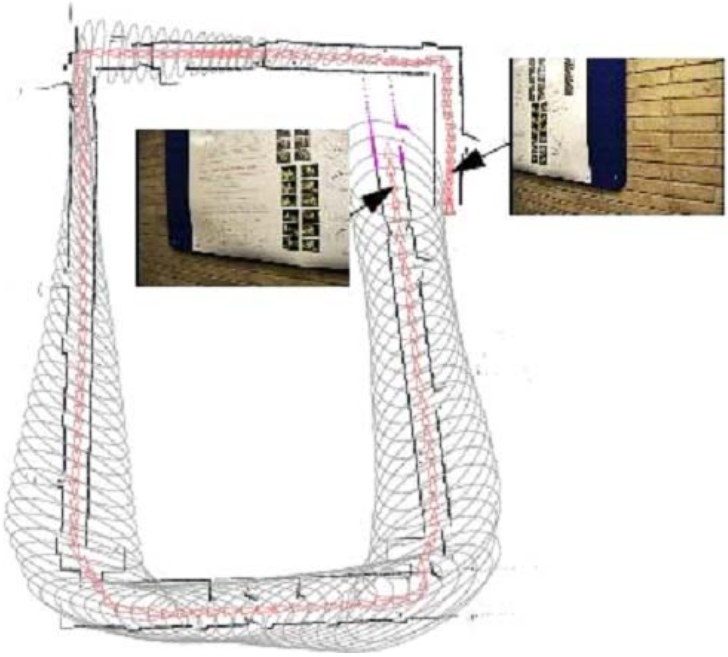


Figure 1.2 An illustration of the big loop closing problem for SLAM from [Newman05]. When a vehicle is closing a big loop, such as around a building or hallway corridors, it cannot detect the loop due to the gross error of location estimates and hence is unable to build a consistent map.

system and contributions of this dissertation. Section 1.3 concludes this chapter with the document organization.

1.1 Motivation

Reliable place recognition is an important computer vision task for autonomous navigation of mobile vehicles, particularly, for the *simultaneous localization and mapping* (SLAM) problem. Although much progress has been made [Tomatis03, Bosse04, Montemerlo01], today's SLAM technology still faces some serious problems that prevent it from becoming a robust and practical application system. A major problem SLAM faces is the big loop closure problem [Newman05]: when a vehicle is closing a big loop, such as around a building or hallway corridors, it cannot detect the loop due to the gross error of location estimates and hence is unable to build a consistent map (see Figure 1.2 for an example).

The key to solving the loop closure problem is enabling vehicles to quickly detect a loop while revisiting an area. One possible way to achieve this is to capture some special geometric or visual features from each scene so that the appearance dissimilarity of different scenes can be accurately measured. Most existing SLAM systems use range devices such as a range scanner and laser scanner to measure the geometric primitives (e.g., corners and edges) of a workspace to build a map. These simple geometric features are often not special enough

Figure 1.3 An example of large number of affine covariant visual features visible at many viewpoints in outdoor environments. For each row, left is original scene image, right shows the SIFT features (green arrows) detected from the image

to build a unique "signature" for a place. In contrast, it is generally agreed that vision is one of the richest sources of information and it has the potential to provide enough information to uniquely identify the robot's position [Lamon01].

Good visual features should have the following properties:

(1) robust to changes in view point;

(2) robust to changes in scale;

(3) robust to changes in illumination.

This is because when mobile vehicles are moving around in a complex environment, they usually observe many landmarks over time from different angles, distances or illuminations. Recently the computer vision community has adopted a class of affine covariant visual features [Lowe04, Matas02, Kadir01]. Many of these features are invariant, to certain degree, to changes in illumination, scale, translation, rotation and viewpoint. This makes them ideally suited to be used as visual landmarks. The performance evaluation of different affine covariant features can be found in [Mikolajczyk03, Mikolajczyk05]. Promising results have been obtained by recent appearance-based approaches [Newman05, Se02, Se05] that are based on the affine covariant features. In [Newman05], the combination of saliency [Kadir01] and MSER [Matas02] features are used as visual landmarks. In [Se02, Se05] SIFT features [Lowe04] are used as the basic visual features.

Typically, a challenge for these methods is that they are not scalable to large environments. Although the affine covariant visual features have some properties that make them suited to be used as visual landmarks, they also cause a serious information redundancy problem. From any given viewpoint, there may be hundreds or even thousands of affine covariant features visible (see Figure 1.3, for an example). Features detected at a sequence of positions along a long path quickly form a large database. In a large environment, it is often the case that a database may contain hundreds of thousand or even millions of features. When a robot attempts to match the features visible at current position to the features that has been

(a) (b)

(c)

Figure 1.4 An example that recognizing place by comparing two images based on local feature matching has difficulty dealing with dynamic environments. (a) and (b) are two images taken from a same place. It can be observed that the scene contains many mobile objects like cars. 3423 SIFT features are extracted from (a) and 3379 ISFT features are extracted from (b). (c) shows that there are only 27 matched SIFT features between the two images.

stored in its database, the space of features that must be searched might be very large. This makes scene-matching very inefficient and hence makes on-line loop detection difficult.

Therefore, the critical problem for vision-based loop detection is how to detect a loop without a costly search in large environments. There are two ways to increase scene-matching efficiency. Firstly, we can reduce the information amount for labeling a place.

Although there may be hundreds or even thousands of features visible at a given viewpoint of an environment, only a few of them are necessary for place recognition. If we only keep a subset of such features that is good enough for recognizing the place, then we can significantly compress the feature database size. Secondly, more importantly, we can add some indexing information for the features in the database so that we can greatly reduce the search space of features when executing a query. Good indexing information can make the real-time query achievable even in a very large database.

Another challenge for these methods is the dynamic environments. Most works recognize a place by comparing two images based on local feature matching. The method can only handle fairly static environments. For dynamic environments where a scene contains many mobile objects like cars and people, the method may not work well. If a scene contains many mobile objects, then some mobile objects may move out and some new mobile objects may move in. As a result, images taken from the same place at different time may have big visual

difference. Under this situation, only a small portion of two images may get matched. An example is shown in Figure 1.4. 3423 and 3379 SIFT features are extracted from (a) and (b) images respectively. Only 27 SIFT features are matched between the two images. It is difficult to decide if the two images are taken from the same place only based on local feature matching information.

We propose to use an object-based place recognition method to address these problems. As mentioned in the introduction section, human visual system perceives scenes based on the perception of objects. Many psychological works support that human visual system takes a very elegant strategy to process the large amounts of information surrounding us. Instead of creating a detailed representation of all the objects in a scene, human visual system builds a very economic scene representation by extracting a few key 'aspects' of the scene information. This economic representation results in an enormous saving of processing and memory resources, which plays a key role for the success of our visual system on place recognition. Thus, we believe that vision-based place recognition methods should follow a similar strategy. Place recognition based on object level has many desired properties. Firstly, we can greatly simplify scene representation. Compared to representing a scene by recording global scene property, representing a scene with a set of salient objects in the scene requires much less information. In addition, place recognition based on object level can increase the place recognition persistence. When a vehicle revisits a place, most likely it will see the scene from a different viewpoint. Therefore, it is inevitable that there may exist some changes in the scene caused by different backgrounds or illuminations. By representing a scene with a set of salient objects, we can ignore the background information and hence make place recognition more robust. More importantly, many objects are appearance-distinguishable since objects have various surface characteristics like colors and textures. Thus, the surface characteristics of objects provide good indexing information for objects. This can help us to avoid many unnecessary comparisons. For example, we do not need to compare an object in red color to an object in green color when performing a query in a database. The indexing information can significantly improve scene-matching efficiency. Besides, if we can recognize some static objects like buildings and use these static objects as landmark, we may also be able to handle the dynamic environments.

Recognizing places on the level of objects requires stable perception on various objects in different scenes. In other words, we need to segment scene images into their constituent objects. This turns out to be one of the fundamental and challenging problems in computer vision community. Usually many structured objects are composed of different parts. Since different parts of an object have different functions, it is natural that different parts of an object may have totally different surface characteristics (e.g., colors and textures). Most existing image segmentation methods are designed to detect homogeneous units in images. These homogeneous units may only approximately correspond to object parts. As results, these image segmentation methods often cause an over-segment problem – an object in an image is segmented into multiple parts. Since these image segmentation methods do not know what objects are, they cannot regroup the object parts back. It is recently argued that, to achieve that, one might need to recognize the object first. This forms a *chicken-and-egg* problem since segmentation is supposed to be a pre-processing for object recognition. Thus, the basic problem that needs to be solved for object-based place recognition is how to detect object boundaries without object-specific knowledge in scene images.

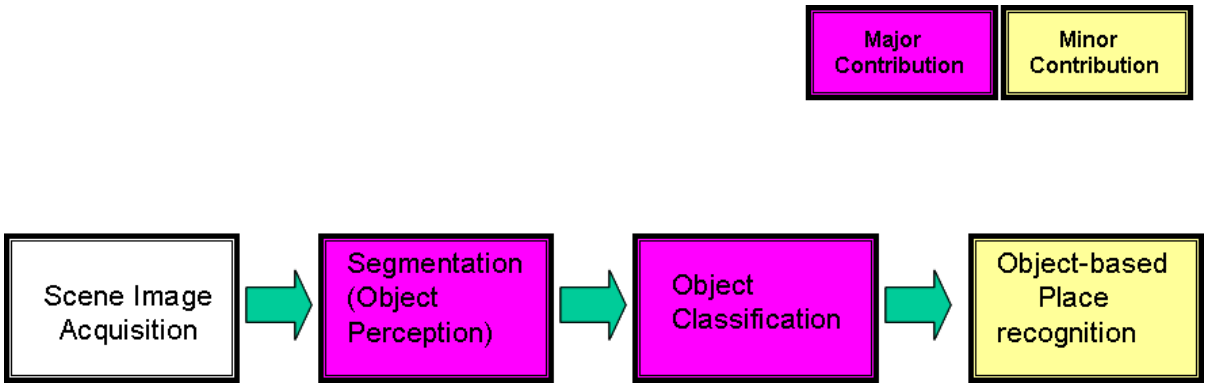


Figure 1.5 The block diagram for our object-based place recognition and loop closing system.

1.2 Contributions

The pipeline of this dissertation work is illustrated in Figure 1.5. An object-based place recognition and loop closing method has been developed. This system includes a novel image segmentation algorithm and an object recognition method and an object-based place recognition algorithm. Accordingly, our research contributions are listed as follows.

- **A novel image segmentation algorithm**: The most significant contribution is the development of a novel image segmentation algorithm. The image segmentation algorithm is based on a Perceptual Organization model. Like the human visual system, the Perceptual Organization model can 'perceive' various special structural relations that obey the principle of *non-accidentalness* among the constituent parts of an object and hence can group them together without object-specific knowledge. Experimental results showed that the Perceptual Organization model outperforms two recent methods in the literature and can stably detect various salient objects for different scenes.
- **A new object classification method**: We then develop a new object classification method. Based on the fairly accurate segmentations generated by our image segmentation algorithm, we build an informative object description that include not only the appearance (colors and textures), but also the parts layout and shape information. We also develop a novel feature selection algorithm. Our feature selection method can select a subset of features that can characterize an object class well. With the selected features, we can classify an object with high accuracy.
- **Object-based place recognition method:** Instead of using large number of affine covariant features to represent a scene, we use a set of objects that are both distinctive and widely visible to represent a place. This object-based scene representation method greatly reduces the amount of information for labeling a place. Thus, we achieve a high degree of compression on the size of visual feature database. We then

designed a range tree database to organize the detected objects. This range tree database makes use of the texture and color information to index the stored objects. The indexing information can help us to quickly find the range of potential matching objects for a query object. Thus, the range tree database allows us to detect a loop without a costly search in large environments. Experimental results showed that our method can quickly detect a loop in most cases in a large complex outdoor environment.

1.3 Document organization

The remainder of this document is organized as follows:

- Chapter 2 reviews existing work relevant to this dissertation, including image segmentation, object classification and place recognition and loop closing methods.
- Chapter 3 describes our image segmentation algorithm for scene images.
- Chapter 4 presents our object classification method.
- Chapter 5 presents our object-based place recognition and loop closing algorithm.
- Chapter 6 presents the experimental results.
- Chapter 7 concludes with a summary of accomplished work and future works for this dissertation.

2 Related work

This chapter discusses research works in two relevant areas: existing image segmentation algorithms are addressed in section 2.1; Section 2.2 discusses current object classification methods; Section 2.3 reviews various SLAM methods and the data associate problem;

2.1 Image segmentation

Over the last 30 years, a large number of image segmentation approaches have been proposed in literature. In this section, we review some representative methods. The reviewed algorithms are divided into five categories: (1) region-based image segmentation techniques, (2) contour-based segmentation methods, (3) boundary detection approaches based on statistical learning, (4) class segmentation methods, (5) methods based on Gestalt laws. Region-based methods and contour-based methods belong to bottom-up methods which segment images based on low-level features like colors, textures and edges, etc. Boundary detection methods, class segmentation methods, and methods based Gestalt laws belong to top-down methods which segment images based on higher-level knowledge. Boundary detection methods detect object boundary based on boundary models learned from training images. Class segmentation methods segment images based on object-specific knowledge learned from training images. Methods based on Gestalt laws segment images based on generic knowledge (Gestalt laws) of real-world objects. Our image segmentation method (POM) belongs to this category. Figure 2.1 shows some examples of the five categories.

2.1.1 Region based methods

To overcome the shortcomings of early graph-based methods that only segment images based on local properties, Wu and Leahy [Wu93] first introduced a minimum cuts criterion in graph. This criterion is designed to minimize the similarity between pixels in two groups. The weakness of their work is that their method favors small components. To fix this bias, Shi and Malik [Shi00] proposed the normalized cut criterion. The normalized cut criterion re-scales the cut weight by computing the cut cost as a fraction of the total edge connections to all the nodes in the graph, which removes the trivial solutions of cutting small sets of isolated nodes in the graph. The main problem of normalized cut methods is the computational efficiency. Minimizing normalized cut yields a NP-hard problem. Although the authors

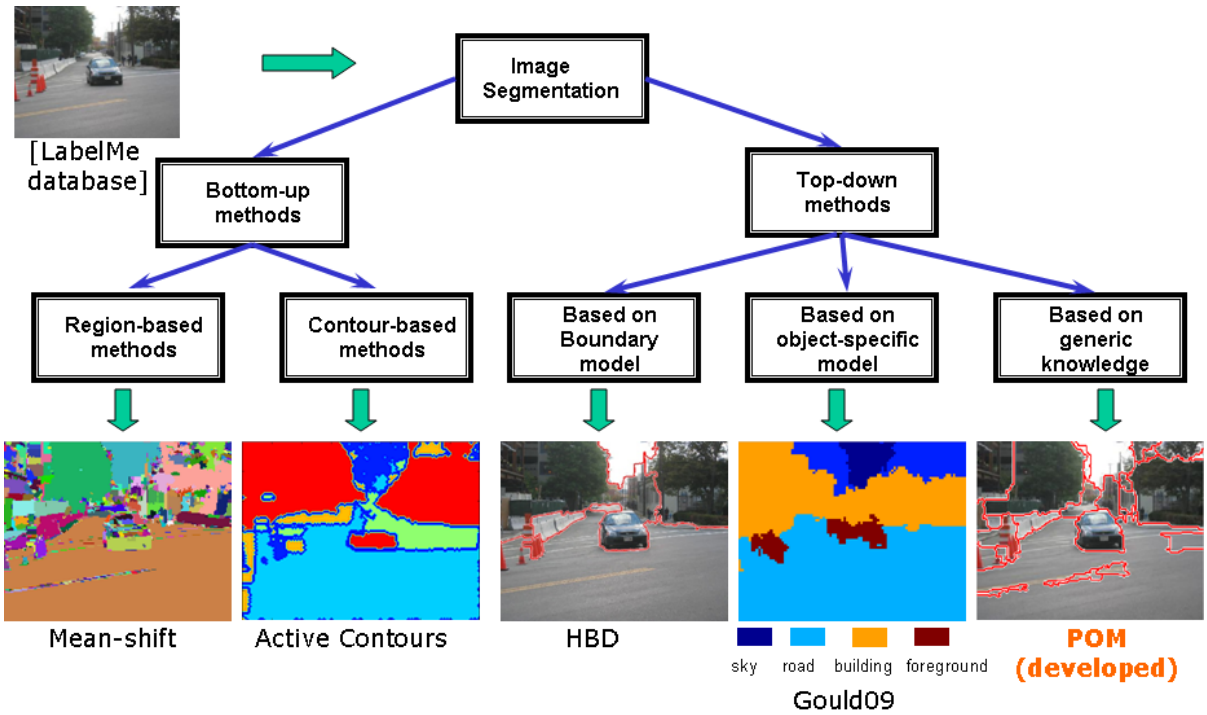


Figure 2.1 Some examples of different image segmentation methods

developed several approximation methods to compute the minimum normalized cut, the computational costs of these approximations are still too high for many practical applications and the error in these approximations is not well understood.

Felzenszwalb and Huttenlocher [Felzenszwalb04] proposed an efficient graph-based generic image segmentation algorithm. As with the graph cut methods, this method also tries to capture the non-local image characteristics. This is achieved by adaptively adjusting the segmentation criterion based on the degree of variability in neighboring regions of the image. A boundary is detected when the degree of variability across the boundary of two regions are large relative to the degree of variability inside at least one of the regions. Due to the high efficiency and good segmentation quality, this method has been widely used by many applications to generate an initial partition. This initial partition often provides good spatial support for the subsequent region-merging processes.

Comaniciu and Peter [Comaniciu02] treated image segmentation as a cluster problem in a spatial-range feature space. Their mean-shift segmentation algorithm has illustrated excellent performance on different image datasets and has been considered as one of the best bottom-up image segmentation methods (see Figure 2.1- Mean-shift for an example). Some of these region-based methods have been widely used to generate small coherent regions called superpixels for many applications [Malisiewicz07, Micusik09, Pantofaru08, Yang07]. Our method also makes use of this algorithm to provide space support for detecting the shape and size information of object parts.

2.1.2 Contour-based segmentation methods

Contour closure is one of the important grouping factors identified by Gestalt psychologists. Early contour-based works like active contour methods [Chan01] only utilize boundary properties such as intensity gradients (see Figure 2.1- Active contours for an example). Zhu and Yuille [Zhu96] first used both boundary and region information within an energy optimization model. They assume an image consists of a set of homogenous regions. The homogeneity of the regions is defined on some low-level properties such as intensity, color or texture. For their method to achieve good performance, a set of initial seeds needs to be placed correctly inside each homogenous region. In addition, they use the gradient descent algorithm to compute the minimum energy, which in many cases can only find the local minima.

Jermyn and Ishikawa [Jermyn01] proposed a new form of energy function. Their energy function is defined on the space of boundaries in the image domain in the form of a ratio of two integrals around the boundary. The numerator of the energy is a measure of the 'flow' of some quantity into or out of the region. The denominator is a generalized measure of the length of the boundary. The main contribution of this form of energy is that the general types of region information are allowed to be incorporated into the energy function. Our method is also based on this form of energy function. We extend the energy function to incorporate the different Gestalt laws together to capture structure context information. Our energy function is addressed in detail in Chapter 3.

2.1.3 Boundary detection approaches

For comparison purposes, we briefly consider a class of boundaries detection methods based on statistical learning. Martin et al. [Martin04] treated boundary detection as a supervised learning problem. They used a large data set of human-labeled boundaries in natural images to train a boundary model. Their model can then predict the possibility of boundary at each pixel based on a set of low-level cues like brightness, color and texture extracted from local image patches. One big challenge for their method is that object parts have wide variation of surface characteristics (e.g., brightness, color, texture etc.). For example, people wear clothes with different colors and textures; cars of same model can be painted into totally different colors, etc. For many objects, some of their parts may have similar colors to the backgrounds around them. It is very difficult to detect the real boundaries for these objects only based on surface characteristics information. In addition, their model detects a boundary solely based on local information, which often cannot provide enough evidence for the appearing of boundaries in a location. As a result, their model often fails to detect some true boundaries and meanwhile detect a lot of false positives in practice.

Noticing the importance of context information, Dollar *et al*. [Dollar06] designed their supervised learning algorithm for edge and boundaries detection based on a large number of generic features calculated over a large image patch. The generic features include gradients, difference of offset Gaussian, and so on at multiple scales and locations. The context information is expected to be provided in a very large aperture and some knowledge of

Gestalt laws are also implicitly incorporated into the model through some well-designed training samples that obey some Gestalt laws. Their method can only output a soft boundary map. To find the close-contour of object boundaries, further knowledge is required.

In the recent work, Hoiem *et al*. [Hoiem07] take a similar statistical approach. They claimed that people recognize objects intrinsically based on 3D interpretation. Therefore, their statistical mode is based on both 2D perceptual cues like color, position, strength and length of boundaries, and 3D cues such as surface orientation estimates, and depth estimates. As with other statistical methods, their model also suffers from the wide variation of the surface properties of object parts. Additionally, recovering 3D information from a single image often causes errors. As a result, their model also has the problem of accurately detecting object boundaries in many cases (see Figure 2.1- HBD for an example).

2.1.4 Class segmentation methods

Multi-class image segmentation (or semantic segmentation) has become an active research area in recent years. The goal here is to label each pixel in the image with one of a set of predefined object class labels. Many works operate on pixel level. Shotton *et al*. [Shotton09] assign a class label to a pixel based on a joint appearance, shape and context model. Shotton *et al*. [Shotton08] proposed to use semantic texton forests for fast classification. A number of works utilize superpixels as a starting point for their task. Gould *et al*.[Gould08] proposed a superpixel-based conditional random field (CRF) to learn the relative location offsets of categories. In their recent work [Gould09], they developed a classification model which is defined in terms of a unified energy function over scene appearance and scene geometry structure (see Figure 2.1- Gould09 for an example). Fulkerson *et al*. [Fulkerson09] use superpixels as the basic unit. They construct a classifier on the histogram of local features detected in each superpixel and regularize the classifier by aggregating histograms in the neighborhood of each superpixel. Other notable works in this area includes Micusik *et al*. [Micusik09], Yang *et al*. [Yang07] and He *et al*. [He04].

2.1.5 Image segmentation methods based on Gestalt laws

Finally, we review some previous efforts attempting to apply Gestalt Laws to guide image segmentation. A great challenge for applying Gestalt Laws of Organization to real-world applications is to find quantitative and objective measures of these grouping laws. The Gestalt Laws of Organization are descriptive. Therefore, one needs to quantify them for scientific applications. Another challenge to incorporate Gestalt Laws of Organization into practical applications is how to combine the various grouping factors. People seem to apply multiple Gestalt laws at once when performing grouping. This is because different parts of an object are usually attached in different ways and a single Gestalt law only captures a single structural relation of object parts. Thus, only by systematically applying the various Gestalt laws, one may be able to group all the constituent parts of an object together. Previous systems cannot find an elegant way to handle these two challenges. Some works [Lowe85;

Jacobs96] only implemented a small subset of Gestalt Laws of Organization for particular tasks.

A number of works [Lowe85, Jacobs96, Mahamud03, Ren08] only applied one or two Gestalt Laws (e.g. *proximity*, *curvilinear, continuity*, *closure* or *convexity* etc.) on one-dimensional image features (e.g. lines, curves, edges) to find closed contours in images. Lowe [Lowe85] and Mahamud *et al.*[Mahamud03] integrated *proximity* and *continuity* laws to detect smooth closed contour bounding unknown objects in real images. Ren *et al*. [Ren08] developed a probabilistic model of *continuity* and *closure* built on a scale-invariant geometric structure to estimate object boundaries. Jacobs [Jacobs96] emphasized that *convexity* plays an important role in perceptual organization and in many cases overrules other laws like *closure*.

Mohan and Nevatia [Mohan92] incorporated several Gestalt Laws to detect a group of collated features describing objects. Their segmentation algorithm, however, is simply based on a set of ad hoc geometric relations among these collated features and is not based on the optimization of a measure of the value of a group. McCafferty [McCafferty90] formulated the grouping problem in Perceptual Organization as an energy minimization problem where the energy of a grouping is defined as a function of how well it obeys the Gestalt Laws of Organization. He treated the total energy of a grouping as the linear combination of the individual grouping energies corresponding to the Gestalt Laws of Organization. This form of group energy requires finding the value of the weighting term denoting the relative importance of each grouping factor for overall perceptual grouping, which is difficult to decide since no systematic attempt has been made to categorize the relative importance of each Gestalt Laws of Organization.

2.2 Object classification and feature selection

2.2.1 Object classification methods

The recognition of object categories is one of the fundamental and challenging problems in computer vision. Object classification methods can be roughly divided into two categories based on their object representation model – *part-based methods* and *orderless bag-of-keypoints* methods [Zhang07].

A number of works are based on modeling objects by parts. Lazebnik *et al*. [Lazebnik04] model objects with geometrically invariant parts. In [Fergus03], objects are modeled as flexible constellations of scale-invariant parts. Felzenszwalb *et al*. [Felzenszwalb05] represent an object by a collection of parts arranged in a deformable configuration. The deformable configuration is represented by spring-like connections between pairs of parts. Amit *et al*. [Amit07] proposed *patchwork of parts* (POP) model. They use edge features as parts. The deformation of an object is defined in terms of locations of a number of reference points. Each reference point is associated with a part. Other notable works include [Craddall05, Epshtein07, Schneiderman04]. Although from a conceptual point, part-based models provide an appealing way of representing many real-world objects, their values have

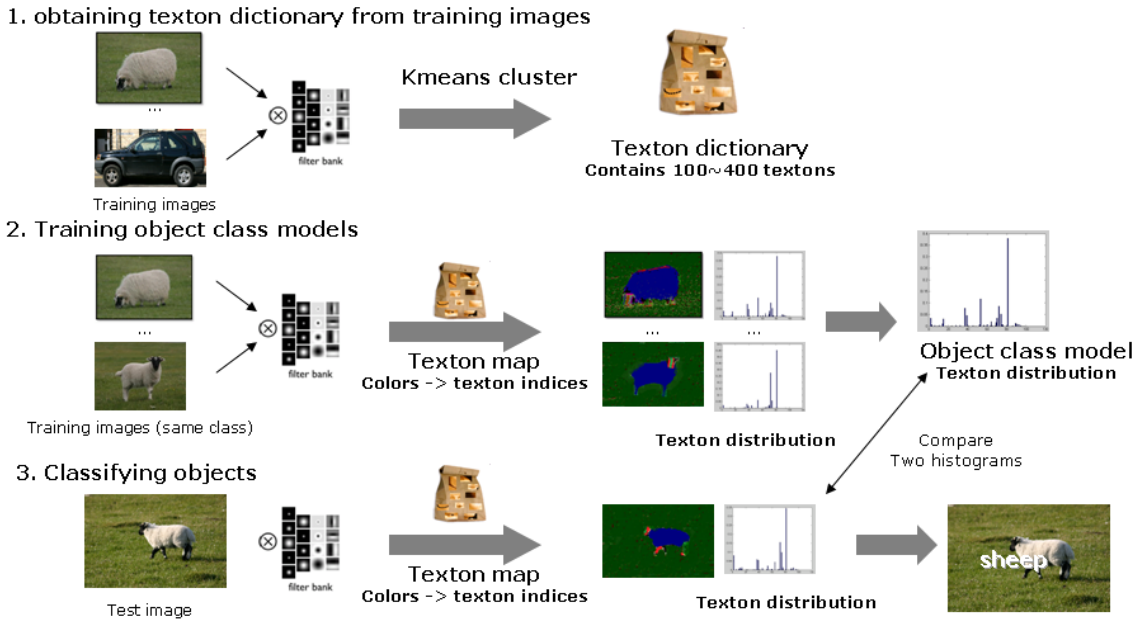


Figure 2.2 An example of using texton for object classification. In first step, convolving each training image with a filter bank. Each pixel gets a vector of responses of the filter bank. Then using kmeans method to cluster these vectors to a set of textons. In the second stage, mapping each training image of the same class to textons learned in the first stage and using these texton distribution to learn an object class model for each class. In third stage, mapping a testing image to texton and comparing the texton distribution of the testing image with the learned object class models to classify an image. Some images and figures are from [Shotton09].

not been demonstrated in practical applications. It has been reported that part-based models are often outperformed by conceptually simple models like *orderless bag-of-keypoints* methods on difficult datasets. This is mainly caused by the difficulty of matching a model to an object instant in an image. Due to occlusion and viewpoint changes, the object instants may yield vastly different shapes, which make reliably model matching difficult in practice. In addition, learning spatial relations remains complex and computational expensive. It often requires training objects to be well segmented from the background in an image, which currently can only be achieved by hand. Recently, Felzenszwalb *et al*. [Felzenszwalb08] proposed a discriminatively trained, multiscale, deformable part model. Their models include both a coarse global template and higher resolution templates. The global template covers an entire object while higher resolution templates cover salient parts. Unlike other part-based methods, their model can be trained from weakly-labeled data (e.g. a training object is specified by a bounding box, the accuracy outline of the object is not available).

In contrast to part-based methods, the *orderless bag-of-keypoints* methods have achieved more successes in practical applications. The *orderless bag-of-keypoints* methods have shown good invariance to pose, viewpoint and occlusion. In addition, they have the advantage of simplicity and computation efficiency. Early works recognize objects with a sparse set of local features. The local features are often extracted by specialized interest operators from texture regions. Csurka *et al.* [Csurka04] proposed to a *bag-of-keypoints* method based on vector quantization of affine invariant descriptor of image patches. They first use harris affine detector to detect a set of harris points from an image. Then build a

SIFT descriptors for each harris points and cluster the SIFT descriptors to form keypoints. Therefore a *bag-of-keypoints* corresponds to a histogram of the number of occurrences of particular image patterns in a given image. Grauman *et al.* [Grauman05] use harris and SIFT detectors to detect interest points and build PAC-SIFT descriptors for each detected interesting point. They then use earth mover's distance (EMD) to compute the correspondence between two bags of features. Other notable works include [Opelt04, Willamowski04]. Since these methods only extract information from texture regions, they can not handle objects with uniform surfaces like sky, road, water, etc.

After realizing that uniform regions also contain important information for object recognition, people proposed many methods based on dense set of features. Different with methods based on sparse set of features, methods based on dense set of features extract features from both texture regions and uniform regions. Malik *et al.* [Malik01] used texton for texture recognition. The term textons were originally used to describe human textural perception and in the literature usually refers to the clusters of feature vectors in a high dimensional space [Winn05]. Winn *et al*. [Winn05] first used texton for object recognition. The specific textons and the particular proportions of texton in each class are learned from a segmented training set (see Figure 2.2 for an example). Although totally discarding the spatial layouts of textons, the learned model perform well on both shape-free and structured objects. Their ideals are followed by many other works [Lazebnik06, Savarese06, Shotton09]. One drawback of the *bag-of-keypoints* methods is that they do not represent the geometric structure of the object class and also do not distinguish between foreground and background features. Due to this reason, the performance of *bag-of-keypoints* methods may be affected by clutter. Some works have been proposed to fix the problem. Savarese et al. [Savarese06] present an approach that augment *bag-of-keypoints* method and incorporate shape information. They use correlograms to capture spatial co-occurrences of features. Their method can encode both local and global shape and can be robust to occlusions. Grauman and Darrell [Grauman05] propose to use pyramid match kernels to compute the correspondence of two sets of unordered features. Their method maps unordered feature sets to multi-resolution histograms and computes a weighted histogram intersection in this space. Lyu [Lyou05] design a new kernel function that obeys the mercer condition to compute the correspondence between two unordered features. All these methods augment the *orderless bag-of-keypoints* methods by incorporating certain spatial information and have obtained promising results.

2.2.2 Feature selection

Classification tasks usually involve representing the patterns as feature vectors. In many practical applications, the number of features can be very large. Feature selection refers to search algorithms that select a subset of most characterizing features from an initial larger set of features. When the initial set of features is small, optimal subset of features can be found by exhaustively searching. When the initial set of features is large (usually > 50), however, feature selection becomes challenging since the exhaustively search is infeasible due to the excessively large numbers of possible feature sets. Many sequential-search-based approximation schemes like best individual features, sequential forward search, sequential forward floating search etc. have been proposed in the literature [Jain00]. Peng *et al*. [Peng05]

proposed a criterion of max-dependency, max-relevance and min-redundancy for feature selection. The measurement of their criteria is based on mutual information.

A big family of feature selection works is based on Markov Blanket. The Markov Blanket of the target feature *T*, denoted as *MB*(*T*), is defined as the set of features conditioned on which all other variables are probabilistically independent of *T*. Margaritis *et al*. [Margaritis99] presented the first provably correct algorithm that discovers the Markov Blanket of a feature from data. They presented a two-stage Markov Blanket called Grow-shrink Markov Blanket algorithm (GSMB). Xing *et al*. [Xing01] follows a similar scheme in their work. Tsamardinos *et al*. [Tsamardinos03] introduced several variants of GSMB such as the Incremental Association Markov Blanket (IAMB) and the Interleaved IAMB (Inter-IAMB). One drawback of GSMB and its variants is that they require sample exponential to the size of *MB*(*T*), which seriously limits the applications of their methods. Aliferis *et al*. [Aliferis05] proposed a sample-efficient Markov Blanket algorithm. However their method is based on an assumption that the maximum number of conditioning and conditioned variables is bounded. This assumption may not hold for complex datasets.

2.3 Simultaneous localization and mapping (SLAM) and place recognition

2.3.1 Robot mapping and SLAM

Mapping is the problem of generating models of robot environments from sensor data. The reason why robot mapping is important is because many successful mobile robot systems require maps for their operation. To map an environment, a robot has to deal with two types of sensor noise: Noise in perception (e.g., range measurements) and noise in odometer (e.g., wheel encoders) [Thrun01].

Because of the latter, the problem of mapping creates an inherent localization problem, which is the problem of determining the location of a robot relative to its own map. The mobile robot mapping problem is therefore often referred to as the *simultaneous localization and mapping problem* (SLAM) [Thrun01].

The field of mapping was roughly divided into metric and topological approaches [Thrun02] based on their map representations. Metric maps capture the geometric properties of the environment. A representative metric map representation is the occupancy grid [Moravec88, Elfes87, Elfes88]. Occupancy grids represent a region as a matrix of cells. Each cell describes a small rectangular area in the environment and indicates the probability that the area is occupied by a value in the range (0,1). An example of occupancy grid map is shown in figure 2.3. This method has been used in many robotic systems such as [Buhmann95, Burgard96, Burgard99, Borenstein91, Guzzoni97, Schneider94, Thrun00, Yamauchi97].

There are several difficulties concerning occupancy grids. The first issue is the tradeoff between grid resolution and computational complexity. To capture environment detail, small grid size is desired. On the other hand, for feasible and efficient computation, larger grid size is required, especially in large environments. Occupancy grids represent uncertainty at a local

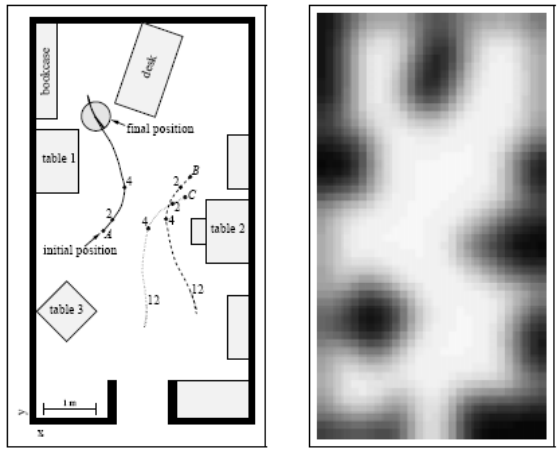


Figure 2.3 An example of an occupied grid map for an office environment. From [Burgard96]

(vehicle-centric) level, but not at a global level, which is essential for map convergence. Besides, data association, a main issue all SLAM methods are facing, is difficult operating on occupancy grids. These issues make occupancy grid methods only applicable to reasonable small environments [Bailey02].

Another representative metric representation is feature map. Feature maps represent the environment by the global locations of parametric features. Appropriate feature may be landmarks, distinctive objects or shapes in the environment. An example of feature map is shown in Figure 2.4. Feature maps are widely used in probabilistic approaches. Smith and Self [Smith88] first introduced a statistical framework to generate maps. In their approach, a map is represented by the Cartesian coordinates of sets of features. The map is combined with robot's pose to form a state vector. They then used Kalman filter to estimate the joint posterior over all the features and robot pose. Based on assumptions that all the noises are in Gaussian distribution, the joint posterior is represented by a state mean vector and a state covariance matrix. In the following years, many people [Csorba97, Castellanos99, Guivant01, Dissanayake01, Newman00, Leonard92, Williams01] further developed this approach. Today, the Kalman filter based methods are in still widespread use.

The main weakness of the Kalman filter based methods is the high computation and storage costs. The stochastic SLAM methods need to maintain correlations in the state covariance matrix. For an environment containing n map features, there are $n \times n$ elements stored in the covariance. These $n \times n$ elements need to be updated with each new observation. Therefore, the computation and storage are specified as being $O(n^2)$.

A further problem concerning the feature map based methods is that they are only applied to structured environments where the observed objects can be reasonably depicted by basic geometric feature models. For the unstructured environments, objects may have arbitrary shapes or curves. More complex models are required to well describe these general objects.

Besides, like the occupancy grids, feature maps also have a problem for robust data association. Objects depicted by basic geometric feature models are often not unique enough to label a place. This makes observation-to-map correspondence is difficult. All these issues make feature map based methods only apply to small-scale environments where stable landmarks are observable, computation is tractable, and accumulated state uncertainty does

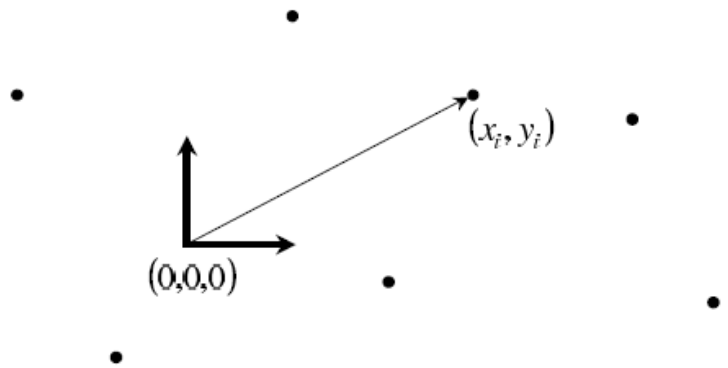


Figure 2.4 An example of a feature map. from [Bailey02]

not exceed conservative limits. For feasible and convergent operation in larger areas, modifications to the basic stochastic SLAM algorithm are required [Bailey02].

To handle large-scale environments, some researchers proposed topological map representation. Occupancy grids and feature maps are both metric maps where location is defined as a set of coordinates in Cartesian space. Topological maps do not rely on metric measurements. Instead, they represent the environment in terms of places and connecting paths. Topological maps are generally depicted by a graph structure where the graph nodes define distinctive places in the environment and the graph edges define procedural information for traveling from one node to another. The topological maps are built on the assumptions that distinctive places are locally distinguishable from the surrounding area and the procedural information is sufficient to enable the robot to travel from one place to next specified place. Examples of topological methods include [Jefferies05, Choset96, Engelson92 Kortenkamp94, Pierce94, Shatkay97, Yamauchi96, Zimmer96].

Topological maps are attractive for their efficient and compact representation, which makes them suitable for large-scale environments. The most critical weakness of topological maps, however, is place recognition. Place recognition is a form of data association where the observation-to-map correspondence is based on the apparent similarity between two data sets: observation information and a graph node description. It is a common case that some places in a large environment may have similar appearances. Without the aid of some form of metric location measure, it might be difficult to distinguish these places. If a place is not recognized or an alternate location is mistaken for a place then the topological sequence is broken and the robot becomes lost [Bailey02].

The qualities of metric and topological maps are complementary. Metric maps, with an appropriate uncertainty representation, can capture the geometric properties of the environment. The captured geometric properties can help constrain data association and permit non-qualitative trajectory planning. On the other hand, topological maps break the world up into locally connected regions and avoid the problems of maintaining a global reference frame. For this reason, most working topological methods use hybrid topological and metric maps. Hybrid topological metric maps are basically topological frameworks where the place definition and /or path definitions contain metric information, which means

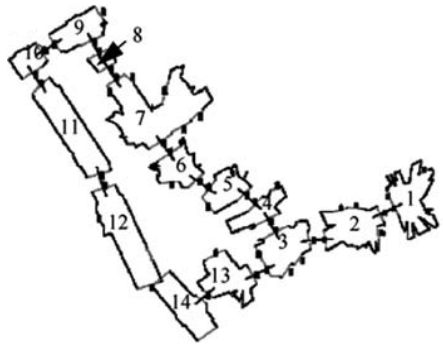


Figure 2.5 An example of a topological map. from [Jefferies05]

that places are no longer restricted to discrete locations but can describe regions of arbitrary size and shape as local metric maps. An example of hybrid topological metric maps is shown in Figure 2.5.

Although many progresses have been made in past twenty years, robotic mapping still faces many problems. A key challenge in robotic mapping arises from the nature of the measurement noise. The measurement errors are statistically dependent. This is because errors in control systems accumulate over time. As results, the control errors affect the way future sensor measurements are interpreted. This causes many existing mapping algorithms surprisingly complex, both from a mathematical and from an implementation point of view [Thrun02].

Another difficulty of the robot mapping problem is caused by the high dimensionality of the entities that are being mapped. A detailed two-dimensional floor plan, which is an equally common representation of robotic maps, often requires thousands of numbers. But a detailed 3D visual map of a building may easily require millions of numbers. From a statistical point of view, each such number is a dimension of the underlying estimation problem. Thus, the mapping problem can be extremely high-dimensional [Thrun02].

The dynamism of robot environments also creates a big challenge. Some changes in an environment may be relatively slow, such as the change of appearance of a tree across different seasons. Others are faster, such as the change of door status or the location of furniture items. The dynamism of robot environments often makes the explanation of sensor measurement inconsistent. For example, imagine a robot facing a closed door that previously was modeled as open. Such an observation may be explained by two hypotheses, namely that the door status changed, or that the robot is not where it believes to be. Thus, most methods rely on a static world assumption, in which the robot is the only time-variant quantity and everything else that moves is just noise. Consequently, most techniques are only applied in relatively short time windows, during which the respective environments are static [Thrun02].

A challenge also arises from the fact that robots must choose their way during mapping. The task of generating robot motion during building a map is commonly referred to as *robotic exploration*. While optimal robot motion is relatively well-understood in fully modeled environments, exploring robots have to cope with partial and incomplete models.

For this reason, exploration is often solved sub-optimally via simple heuristics. When choosing where to move, various quantities have to be traded off: the expected gain in map information, the time and energy it takes to gain this information, the possible loss of pose information along the way, and so on. Furthermore, the underlying map estimation technique must be able to generate maps in real-time, which is an important restriction that rules out many existing approaches [Thrun02].

Among all the problems in today's robotic mapping, however, the hardest one is the *correspondence problem*, also known as the *data association problem* [Thrun02]. Most SLAM methods build a map by incrementally integrating new data into the map. Every time a new sensor measurement is acquired, the robot must determine if:

1) It is associated with an unknown region of workspace?
2) It is associated with a previously mapped region? or
3) It is spurious and should be ignored?

The correspondence problem becomes challenging when the robot's pose is in gross error. Figure 1.2 shows an instance of this problem, in which a robot attempts to map a large cyclic environment. When closing the cycle, the robot has to find out where it is relative to its previously built map. This problem is complicated by the fact that at the time of cycle closing, the robot's accumulated pose error might be unboundedly large. The correspondence problem is difficult, since the number of possible hypotheses can grow exponentially over time. Most scientific progress on the correspondence problem has emerged in the past several years, after a long period in which the problem was basically ignored in the robot mapping community [Thrun02]. The work presented in this dissertation is mainly designed to help handle this big challenge in SLAM.

2.3.2 Data association and loop closing problem for SLAM

Early works [Smith88, Castellanos99, Feder99] use the gated Nearest Neighbor algorithm to associate sensor observations with map features given map and pose estimates. This type of algorithms fails when the pose estimate is in gross error, which is often the case when a robot is closing a big loop. Neira and Tardos [Neira01] present a Joint compatibility test algorithm. Instead of considering each matching between sensor observation and map features independently, their algorithm takes into account the correlations between features in a local region. This technique presents some degrees of robustness against pose estimate error. Bosse *et al*. [Bosse04] developed a hybrid metrical/topological approach. They represent a large environment as a graph of multiple local submaps with limited size. The loop closure event is detected by performing matching with constant-size local submaps that are expected to have some non-empty intersection with the true local area. Other hybrid metrical/topological approaches include [Tomatis03, Choset01, Bailey01]. Thrun *et al*. [Thrun01] propose a SLAM method to avoid the need for exact data association. Instead of only keeping the most likely pose, their method maintains the full posterior over poses. By doing so, their method can accommodate large errors of pose estimate. With this technique, their method is able to close moderately sized loops. A similar work is developed by Gutmann *et al*. [Gutmann99]. However, all the methods rely on pose estimate to associate data or detect loop. Therefore, these methods still struggle when the pose estimate is in gross

error.

2.3.3 Vision-based methods

Newman and Ho [Newman05] point out that the fundamental solution for the big loop closing problem of SLAM is to allow a robot to be able to recognize place in an independent way. i.e., a robot should be able to quickly decide if it is revisiting a place or exploring a new place only based on the current sensor measurement and the pre-built map, and not rely on its pose estimates. They believe that camera is an ideal tool for this purpose because it can provide rich information about a place and allow a robot to be able to build a unique "signature" for each place. They combine the Saliency and "Maximally stable extremal regions" (MSER) techniques together to detect the regions that are not only salient but also robust to changes in view point in an image. Then, they build a SIFT descriptor to encode each detected region and store the SIFT descriptors in a database. Loop detection is achieved by matching the SIFT descriptors extracted from the current image to the SIFT descriptors stored in the database. They conduct the experiment in an indoor environment. The result shows that using visual information alone to detect possible loop closure events can make traditional SLAM algorithm more robust. However, their query mechanism has a complexity linear in database size, which makes their method not scalable to large environments.

Se *et al*. [Se02] developed a vision-based global localization and mapping method. They use SIFT features as visual landmarks. By keeping the distinctive SIFT features in a database, they are able to build a 3-D map of an environment and use these SIFT features for localization at the same time. By matching the SIFT features detected in current frame to the pre-built SIFT database map, they can globally localize the robot and detect a loop. The distinctiveness of SIFT features eliminates the data association problem struggled by the traditional SLAM methods. Once detect a loop, they can determinate the accumulated position error and employ a global minimization procedure to do backward correction on all the previous submaps. Their work has been tested in a simple indoor office environment. As with Newman and Ho's work, their loop detection method may require a high computational cost search for large outdoor environments.

Cummins and Newman [Cummins07] propose a probabilistic appearance based navigation and loop closing method. They apply a bag-of-words representation in a probabilistic framework. Words are built as SIFT descriptors around the regions of Harris-affine interest points. Each word represents some small aspect of the local workspace. The underlying map representation is a set of discrete places. Each place is parameterized by a probability distribution over the existence of words at that place. They train a generative model of observations based on a large number of images collected from a large urban region. The generative model of observations is used to capture the co-occurrence statistics of the visual words. If an observation contains some distinctive words and has a match for one of the mapped place, then a loop is detected. Otherwise, a new place is added into the map. They experiment in a large outdoor environment and achieve good performance on place recognition. However, their method requires a time-consuming training stage. If a robot

moves to a totally different environment, the generative model of observations trained before may not be accurate enough and a new training process is required for the new environment. In addition, they use a collection of visual words to represent a place. Thus, their scene-matching method is essentially based on the global scene image properties. This might cause scene-matching fail due to the presence of some changes in the scene in many cases.

Sala *et al*. [Sala06] notice the importance of eliminating redundancy from landmark-based map. They claim that more attention needs to be put on the size of landmark database or the number of landmark lookups required for localization. They present a method to automatically select an optimal set from the entire set of features visible in an environment. They decompose the world into a small number of maximally sized regions. At each position in a given region, the same small set of features is visible for navigation. They collect images at known discrete points (e.g., the points of a virtual grid overlaid on the floor of the environment) in a small indoor environment. By doing so, for each of the grid points, they know which features in the database are visible; and for each feature in the database, they know from which grid point it is visible. Therefore, their method is only applied to localization problem where the map of an environment is already available.

2.3.4 Object representation methods

Objects in the real world have description of infinite complexity. In practice, however, object representation is often application-related. Most applications need only small amounts of information to perform a task adequately. The key is to choose representations for each task that encapsulates necessary information to perform the task in an efficient manner.

In our case, our goal is to recognize a scene by recognizing a set of salient objects in the scene. When mobile vehicles are moving around in a complex environment, it is often the case that the vehicles may observe many objects over time from different angles, distances or under different illuminations. Therefore, for our application, object representation should have the following properties:

1. Robust to changes in view point;
2. Robust to changes in scale;
3. Robust to changes in illumination.

There are many object representation methods in the literature. Early works represent objects with a list of expected properties, such as colors, shapes, rough sizes, etc. The success of these methods in object recognition has mainly been in domains where imaging conditions and object appearance are restricted or well-controlled. As for the recognition of many common objects in natural settings, such as cars and people in outdoor scenes, these techniques often cannot produce satisfied results. The main difficulty, as pointed out by Ullman [Ullman96], lies in the tremendous view variability associated with the images of a given object. For example, depending on the viewing angle, the pictures of a car may look very different. As a result, an algorithm designed based on a single or a few views may not work on a picture of the object taken from a new view [Zhang03].

Most works in view-invariant object recognition can be classified into three approaches [Ullman96], namely, using invariants, part decomposition, and alignment. In using

invariants, the idea is to use object features that are invariant over a wide range of views, such as various geometric invariants [Mundy92]. Despite its success in some applications, using invariants also has several limitations. First, for a given object, it might not always be obvious as to how to find its invariants. Even if they can be found, weighing their relative importance can be difficult. Second, many invariants are sensitive to noise and occlusion. Finally, because many invariants are obtained by many-to-one mappings, different objects may have the same invariant feature values [Zhang03].

Part decomposition approach decomposes an object into a set of simple and generic shapes, such as boxes, cylinders, and generalized cylinders. Then, the object is represented by a graph. The nodes of the graph correspond to the decomposed parts and the edges of the graph encode their spatial relations. By doing so, part decomposition approach can represent very complicated objects. Besides, this approach also achieves some degree of invariance through the spatial relations embedded in the graph representation. For example, the wheels of a car are always below its windows from different viewing angles. This approach, however, also has some practical limitations. For example, at relatively coarse levels, many similar objects may have the same part decomposition. A fine level decomposition, on the other hand, may produce very complex models for relatively simple objects. Similarly, stable and consistent part decomposition is often not easy and may even be ill-defined for many natural objects [Ullman96].

Alignment approach achieves invariance by compensating for the transformations that an object may have undergone during the imaging process. The advantage of this approach is that it uses the entire shape, rather than its partial (as in using invariants) or coarse (as in part-decomposition) representations. Thus it promises better recognition performance. The disadvantage of this approach is that sometimes, the optimization involved in the alignment operation may be computation- intensive[Zhang03].

Recently the computer vision community has adopted a class of affine covariant visual features [Lowe04, Matas02, Kadir01]. Many of these features are invariant, to certain degree, to changes in illumination, scale, translation, rotation and viewpoint. This makes them ideally suited to be used as visual landmarks. Among them, SIFT [Lowe04] has gained noticeable attention and been widely used in applications.

2.3.5 Content-based image retrieval technologies

Our goal is to quickly recognize a place by matching the set of landmark objects in the place to the landmark objects recorded in a large database. To achieve this goal, we need to build an efficient indexing schema so that for any query object, we can quickly find the most similar objects to the query object from the database. This application is similar to the content-based image retrieval (CBIR) technologies. CBIR aims at retrieving images based on their visual similarity to a user specified query image. Many CBIR systems are based on using visual features like shape, color, texture, etc to index and retrieve relevant images from image database.

Shape is an important feature for perceptual object recognition and classification of images [Prasad04]. Different shape representations such as chain code, polygonal approximations, curvature, Fourier descriptors, radii method, and moment descriptors have been proposed and used in various applications [Gonzalez87]. Features such as moment invariants and area of

region do not give perceptual shape similarity. The weakness of chain codes is that they are not normalized and hence are not invariant to shape scale. A region-based shape representation and indexing scheme that is translation, rotation, and scale invariant is proposed in [Lu99]. Compared to Fourier method, this method achieves better retrieval performance. The drawback of this method is that it works only on binary images and has been applied on only 2D planar images. Moreover the shapes with similar eccentricity but different shapes are also retrieved as matching images. A retrieval system using combined color and shape indexing has been developed [Prasad04]. This method combines the earlier grid-based shape representation with the dominant color-based index to provide better retrieval efficiency and effectiveness.

Color is one of the most important image indexing features employed in CBIR. Bimbo [Bimbo99] provides a comprehensive survey of various methods employed for color image indexing and retrieval in image databases. Some of the popular methods to characterize color information in images are color histograms [Hafner95], color moments [Stricker97], and color correlograms [Huang97]. Though all these methods provide good characterization of color, they have the problem of high-dimensionality. This leads to more computational time, inefficient indexing, and low performance[Prasad04]. To overcome these problems, Ravishankar *et al*.[Ravishankar99] propose dominant color regions approach. The dominant color descriptor gives the distribution of the salient colors in the image. Unlike the bin quantization in the histograms, the specification of colors in a dominant color descriptor is limited only by the color space quantization. Hence dominant color descriptor provides an effective, compact, and intuitive representation of colors present in a region of interest. Similar methods include color clustering [Wan98].

Texture, like color, is also considered to be a powerful low level feature for image search and retrieval application. Generally, approaches in the analysis of image based on textural contents use either the spatial or frequency-based techniques. Spatial approaches include structural and statistical models such as the co-occurrence matrix and autoregressive models [Abbadeni00]. In the frequency-based approach, popular techniques include wavelet-based models such as the Gabor wavelet decomposition model. There is another class of models called perceptual models. In these models, textures are represented by a set of features that have a perceptual meaning such as contrast, coarseness, directionality, regularity, etc. [Abbadeni00, Bimbo99]. Other approaches include image decomposition by filtering with a subband or wavelet filter bank [Theoharatos06, Do02] and appliance of a linear transformation by a Fourier or discrete cosine transform [Leung01, Zhong00].

Although our application is similar to CBIR, there are some differences between these two applications. The most important difference is that our application requires indexing features to be viewpoint-invariant. This is because when vehicles are revisiting a place, most likely they will see the scene from a different viewpoint. This means that objects in a scene might be observed from different distances, different viewing angles and under different illuminations. The viewpoint-invariant requirement makes shape feature not applicable to our application. In CBIR, shape features are usually applied to some types of image datasets where images are taken under control conditions. In these images, objects are usually placed in uniform backgrounds so that the geometric shape of objects can be accurately extracted. This condition does not hold in our application. In our case, objects are not controlled and appear in their natural habitat. From different viewing angles, object's shape looks quite different (e.g. cars). In addition, sometimes the perceptual organization model used in our

image segmentation algorithm might make small error on grouping objects, which also affects the accuracy of object shape. All these issues make shape features instable. Therefore, in our work, we do not use shape as an indexing feature. Compared to shape feature, appearance-based features like color and texture are more stable for viewpoint changes, which makes them suited as indexing features for our application.

3 Image segmentation based on Perceptual Organization

In this chapter, we want to tackle a challenging problem – how to detect object boundaries without object-specific knowledge in scene images. As we mentioned in Chapter 1, this turns out to be one of the fundamental and challenging problems in computer vision community. Usually many structural-rich objects are composed of different parts. Since different parts of an object have different functions, it is natural that different parts of an object may have totally different surface characteristics (e.g., colors and textures). Most existing image segmentation methods are designed to detect homogeneous units in images. These homogeneous units may only approximately correspond to object parts. As a result, these image segmentation methods often cause over-segment problem – an object in an image is segmented into multiple parts. Since these image segmentation methods do not know what objects are, they cannot regroup the object parts back. It is recently argued that, to achieve that, one might need to recognize the object first. This forms a chicken-and-egg problem since segmentation is supposed to be a pre-processing for object recognition. In this chapter, we developed a novel image segmentation algorithm. The image segmentation algorithm is based on a Perceptual Organization model. With the Perceptual Organization model, our image segmentation method can 'perceive' the special structural relations among the constituent parts of an unknown object and hence can group them together without object-specific knowledge. The image segmentation algorithm allows us to 'perceive' the salient objects in a place. The work presented in this chapter is the most significant contribution for the dissertation.

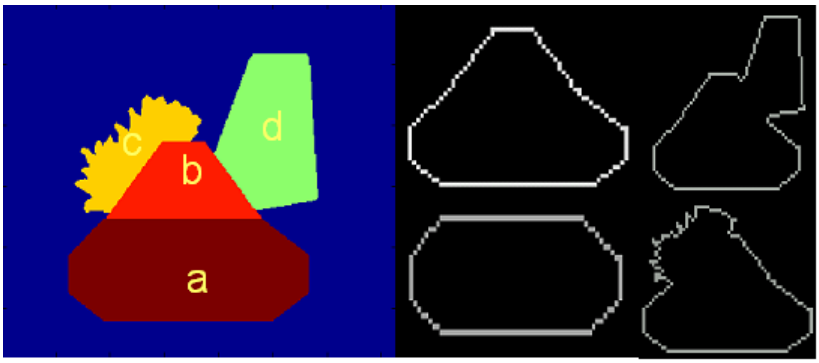
The remainder of the chapter is organized as follows: we first present an example of Perceptual Organization phenomenon in Section 3.1. We then briefly introduce the Gestalt laws and Perceptual Organization in Section 3.2. In Section 3.3, we illustrate a list of Gestalt cues with real-world objects, our Perceptual Organization model mainly rests on the list of Gestalt cues. Then, we describe the generic knowledge about natural scene images in Section 3.4. In Section 3.5, we present our Perceptual Organization model and the boundary detection algorithm. The relation of our Perceptual Organization model and the Gestalt laws is discussed in Section 3.6. Finally, the image segmentation algorithm is described in Section 3.7 and Section 3.8.

3.1 Perceptual Organization phenomenon

As we mentioned in Chapter 1, detecting object boundaries in scene images is a challenging problem. Natural scene images usually contain multiple objects. In many cases, the objects in a scene are cluttered and occluded. Different scenes are often associated with different groups of characteristic objects. Many structural-rich objects like vehicles and buildings etc. are composed of several parts. Since different parts have different functions, it is natural that different parts of an object may have totally different surface characteristics (e.g., color, texture, brightness, etc.). Most bottom-up methods are designed to detect homogeneous image regions, which may only correspond approximately to the object parts. As a result, bottom-up methods may inevitably cause an over-segmenting problem – an object may be segmented into multiple regions (parts) and it is difficult to regroup these regions back. It is recently argued [Malisiewicz07, Russell 06] that in order to regroup these regions back, one might need to solve the object recognition problem first, which forms a chicken-and-egg problem since segmentation is supposed to be a pre-processing step for object recognition. Some works [Borenstein04, Rutishauser04] attack this difficulty by using object-specific models to help identify object boundaries. These methods are generally called top-down methods. However, these methods do not perform well when the images contain objects that have not been seen before. Therefore, top-down methods cannot apply to scene images since scene images usually contain various kinds of objects and it is impossible to learn all the object models in advance. Therefore, our research objective is to explore detecting object boundaries in an image without any object-specific knowledge.

It has long been known that Perceptual Organization plays a powerful role in human visual perception. Perceptual Organization in general refers to a basic capability of the human visual system to derive relevant groupings and structures from an image without prior knowledge of its contents. The Gestalt psychologists summarized some underlying principles (e.g., proximity, similarity, continuation, closure, symmetry, etc.) that lead to human perceptual grouping. They believed that these laws capture some basic abilities of the human mind to proceed from the whole to the part [Lowe85]. A simple synthetic example of perceptual grouping phenomena is shown in Figure 3.1. In this image, although the four parts have different surface characteristics and shapes, most people would prefer to group part *a* and part *b* together. The main reason is that these two parts are approximately symmetric along a vertical axis and also aligned along the vertical direction which makes the boundary of the union of these two parts has a smooth continuation. Part *d* is separated from the group of *a* and *b* because it is weakly attached to part *b.* Although there is also a strong attachment between part *c* and part *b*, unlike part *a* and *b*, part *c* has an irregular shape. Notice that in this process, people seem to simultaneously apply four Gestalt laws to help them do grouping: symmetry and continuation laws for grouping part *a* and part *b* together; proximity law for separating part *d* from part *a* and *b*; similarity law for separating part *c* from part *a* and *b*.

The simple example illustrates that people have the ability to make a correct decision on grouping the different parts of an unknown object together even under certain degrees of clutter and occlusion. This kind of capability is desired for natural scene image segmentation. Although different objects in natural scenes have different configurations of parts, in most



(a) (b)

Figure 3.1 A synthetic example of Perceptual Organization phenomena. (a) A synthetic image. (b) The boundary energy of four regions containing component *a*. Brighter line represents lower boundary energy value. The region that contains *a* and *b* (Upper left) has the minimal boundary energy.

.

cases an object part should be at least tightly attached to one of the remaining parts of that object. Therefore, if an object is composed of a set of parts, there always exist some special structural relations among most of these parts. As demonstrated in the above example, in many cases, these kinds of special structural relations can be captured by the human visual system. Our goal is to develop an image segmentation method that has a similar behavior like human visual system. For most of the salient objects in a scene, even though without any specific knowledge about these objects, our algorithm should be able to group the different parts of these unknown objects together and find the correct boundaries for these objects. We believe that we can achieve this goal by applying the Gestalt laws in image segmentation. However, a great challenge is that all Gestalt laws are descriptive and therefore are difficult to be applied to scientific applications directly.

3.2 Gestalt psychology and Perceptual Organization

Gestalt is a German word meaning configuration or pattern. Different from the previous theory that assumed that perception could be explained solely as a combination of individual components, the school of Gestalt Psychology stresses that the whole is greater than the sum of its parts. This is because when a group of parts are considered as a whole, many new properties will emerge. For example, the property of orientation emerges only when two or more dots are considered and each dot does not have this property individually [McCafferty90].

The major contribution of Gestalt psychology to today's understanding of Perceptual Organization was to highlight a number of factors, which they considered to lead to visual grouping. These factors are summarized as follows [Wertheimer38, Lowe85]: *Proximity*: components that are close together tend to be grouped together; *Similarity*: components that are similar in physical attributes like size, shape, color or others, are grouped together; *Continuation*: components that lie along a common line or smooth curve are grouped together; *Symmetry*: components that are bilaterally symmetric about some axis are grouped

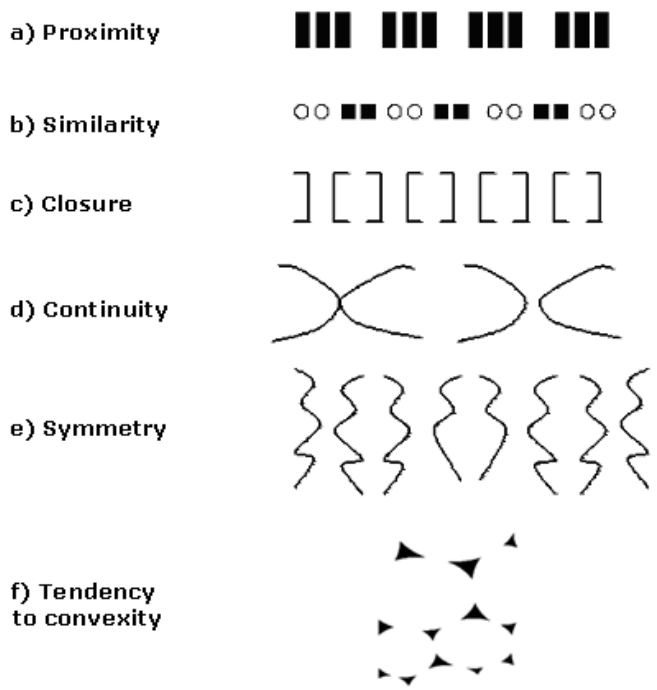


Figure 3.2 Gestalt laws. (a) black blocks are grouped on the basis of proximity; (b) circles and squares are paired based on similarity in shape; (c) incomplete squares are perceived due to closure; (d) two crossing lines rather than two non-crossing curves are perceived due to good continuation and (e) curves are paired on the basis of symmetry; (f) four ellipses are perceived due to convexity.

together; *Closure*: components that seem to complete some entity are grouped together. These grouping factors are known as the Gestalt laws of Organization and used to explain how smaller objects are grouped to form larger ones. Examples of Gestalt laws are shown in Figure 3.2. In the later stage, Gestaltists tried to summarize their laws of organization with the single law of *Pragnanz*, which is a German term meaning "good of form". This law is also referred as law of *simplicity*. The law is based on the observation that in any situation, objects in the environment are perceived in a way that makes as simply as possible. However, many psychologists realized that this law of *simplicity* is unsatisfactory because it is not well-defined and too subjective for scientific use. In 1980's, Witkin and Tenenbum proposed an alternative principle called *non-accidentalness* [Witkin83]. Specifically, from a probabilistic point of view, the Gestalt laws are relatively unlikely to occur by accident, but most likely originate in a single object or process. In other words, there exist causal relations between Gestalt laws and scene structure. For example, the *Proximity* law can be explained by the fact that if two components are closer together in an image, they are also very likely to be adjacent in the 3D world. Besides the above classic Gestalt laws, other Gestalt laws like *Convexity* also draws a large notice by many researchers [Jacobs96, Jabobs03].

As we have mentioned in the Chapter 3.1, there are two challenges for applying Gestalt Laws of Organization to real-world applications. The first challenge is to find quantitative and objective measures of these grouping laws. The Gestalt Laws of Organization are in descriptive forms. Therefore, one needs to quantify them for scientific use. Another challenge is that of how to combine the various grouping factors. People seem to apply the

Gestalt laws systematically in a grouping process. This is because different parts of an object are usually attached in different ways. If one applies Gestalt laws independently without incorporating other laws, he may only capture a subset of the parts that the structural relations of them only obey the applied Gestalt laws and miss the remaining. Thus, only by systematically applying the various Gestalt laws, one may be able to group all the constituent parts of an object together. Our solutions for solving these two challenges are presented in the following subsections.

3.3 Gestalt cues in real-world objects

As shown in Figure 3.2, classic Gestalt laws are usually illustrated with a set of artificial elements like dots and lines. As results, most works in the literature mainly apply Gestalt laws on zero- or one-dimensional image features (e.g. points, lines, curves, etc.). Different with these works, our method applies Gestalt laws on two-dimensional image features – object parts. Following Jacobs [Jacobs03], we treat perception organization as a process of finding features in 2-D image that provide cues about the 3-D structure that produced the image. Since real-world objects are composed of different parts, the properties of object parts in 2-D images provide vital clues about the real-world objects that produce the 2-D objects in the images. We first give the formal definition of object parts in images:

Definition: an object part refers to a homogenous portion of an object surface in an image.

Based on our empirical observation, most object parts have approximately homogenous surfaces (e.g. colors, textures, etc.). Therefore, the homogenous patches in an image approximately correspond to the parts of the real-world 3-D objects that produce the 2-D image. Throughout this dissertation, we use this definition for object parts.

As Bruce [Bruce90] pointed out that the Gestalt grouping laws are believed to reflect the general properties of the world. In other words, the Gestalt laws may work because they reflect a set of sensible assumptions that can be made about the world of physical and biological objects. In the remaining of this section, we illustrate a set of Gestalt cues with real-world objects to show the validation of these assumptions. Our perceptual Organization model presented in Section 3.5 then rests on these Gestalt cues.

The *symmetry* law is based on an assumption that many natural and man-made objects are symmetrical. The symmetrical relations can be observed in various natural and man-made objects. Examples of symmetry relations among different parts of real-world objects are illustrated in Figure 3.3 – in (a), the person's head and body are approximately symmetric along a vertical central axis; in (b)-(d), different parts of the buildings are approximately symmetric along their vertical central axis; in (e), the head and body of the owl are approximately symmetric along the vertical central axis; in (f), different parts of the bowling are approximately symmetric along the vertical central axis; in (g), the top red portion and the bottom black base of the mailbox are approximately symmetric along the vertical central axis; in (h), the top red portion and the white base of the fire-hydrant are approximately symmetric along the vertical central axis. Therefore, based on the *symmetry* law, if two object parts are symmetrical along a central axis (most likely along a vertical axis in natural

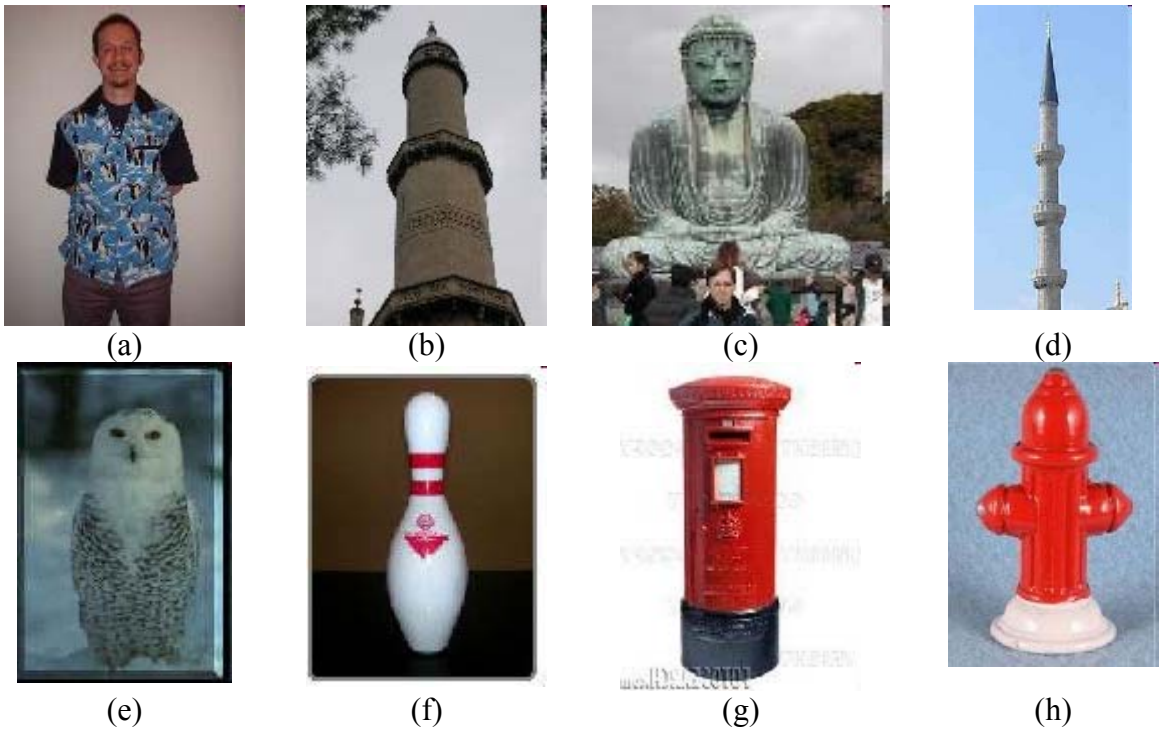


Figure 3.3 Symmetry law in real-world objects. (a) the person's head and body are approximately symmetric along a same vertical central axis; (b)-(d) different parts of the buildings are approximately symmetric along the vertical central axis; (e) the head and body of the owl are approximately symmetric along the vertical central axis; (f) different parts of the bowling bottle are approximately symmetric along the vertical central axis; (g) the top red portion and the bottom black base of the mailbox are approximately symmetric along the vertical central axis; (h) the top red portion and the white base of the fire-hydrant are approximately symmetric along the vertical central axis. (images from [Griffin07] and [Russell08])

scenes), then the two object parts may belong to a single object.

The *continuation* law is based on an assumption that the shapes of natural objects and many man-made objects tend to vary smoothly rather than having abrupt discontinuous. Examples of continuation relations among different parts of real-world objects are illustrated in Figure 3.4 – in (a), the beak of the goose is aligned with the head; in (b), the central white portion of the ibis is aligned with the black tail; in (c), the blue portion of the umbrella is aligned with the central white portion; in (d), the blue portion of the bag is aligned with the black portion of the bag; in (e), the top black portion of the bread-maker is aligned with the bottom white portion; in (f), the top white portion of the paper shredder is aligned with the black bottom; in (g)-(h), from the front and rear views, the windows of the vehicles are aligned with the bodies. Therefore, we conclude that like the symmetrical relation, alignment is also a strong indication of two object parts belonging to a single object.

Convexity is a common characteristic for object parts and object surfaces. The surfaces of many natural or man-made objects, such as buildings, furniture, many kinds of vehicles, etc., are known to be convex. For some objects, even if their structures are not fully convex, they may consist of at least some convex parts. Given the importance of convexity, Liu et al [Liu99] claimed that theories of perceptual organization must take into account the role of convexity. And convexity has been considered in grouping by many researchers [Jacobs96, Liu99, Jabobs03]. Examples of convexity relations among different parts of real-world objects are illustrated in Figure 3.5 – in (a), the panel of the gas pump is embedded in the blue portion of the gas pump; in (b), the red portion of the gold fish is embedded in the body of the fish; in (c), the label of the winebottle is embedded in the bottle; in (d), the black portion of the welding mask is embedded in the red portion of the mask; in (e), the yellow head of the blimp is embedded in the remaining white part; in (f), the central blue portion of the video projector is embedded in the remaining part of the projector; in (g)-(h), the windows and wheels of the vehicles are embedded in the bodies of the vehicles. The embedded components of these objects help increase the convexities of the objects. Thus, according to the *Convexity* law, an object part that is embedded into an entity should belong to the entity due to its contribution of increasing the degree of convexity of the entity.

The *Proximity* law is based on an assumption that if two object parts are closer together in an image, they are also very likely to be adjacent in the 3D world. Since matter is cohesive, adjacent regions are likely to belong together. In classic Gestalt laws, the *Proximity* law is often illustrated by a group of artificial components that are spatially closer together. This type of illustration, however, cannot hold in real-world images. A real-world image is actually the 2-D projection of a 3-D scene. Due to the projection transformation, the depth information of the scene is lost. Therefore, in a 2-D scene image plane, neighboring object parts do not necessarily mean that they are adjacent in 3-D world! To judge if two neighboring object parts in a 2-D image are adjacent, we need to take into account depth information. One depth cue that is widely used in art is the relative size of different objects. This is based on the fact that the size of an image cast by an object is small if the object is far away and large if the object is closer. Thus, if the size of an object part is much smaller than its neighbors, the object part may be far away from its 2-D neighbors in 3-D world. On the other hand, if an object part is much bigger than its neighbors, the object part may belong to background according to the figure/ground law. Besides, if two object parts are cohesive, they need to share certain amount of boundaries. Some examples of proximity relations are

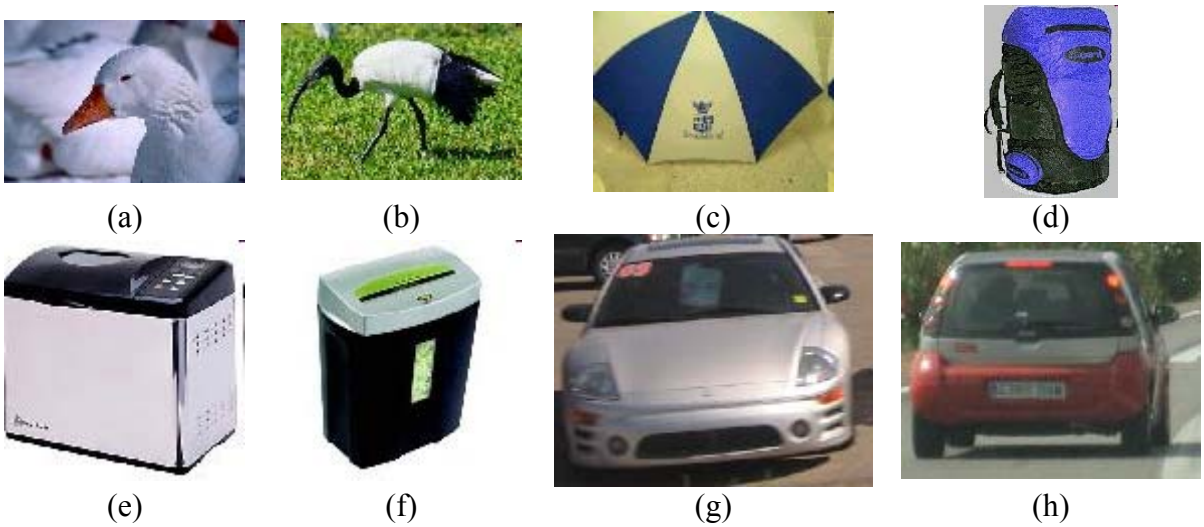


Figure 3.4 Continuation law in real-world objects. (a) the beak of the goose is aligned with the head; (b) the central white portion of the ibis is aligned with the black tail; (c) the blue portion of the umbrella is aligned with the central white portion; (d) the blue portion of the bag is aligned with the black portion of the bag; (e) the top black portion of the bread-maker is aligned with the bottom white portion; (f) the top white portion of the paper shredder is aligned with the black bottom; (g)-(h) from the front and rear views, the windows of the vehicles are aligned with the bodies. (images from [Griffin07] and [Russell08])

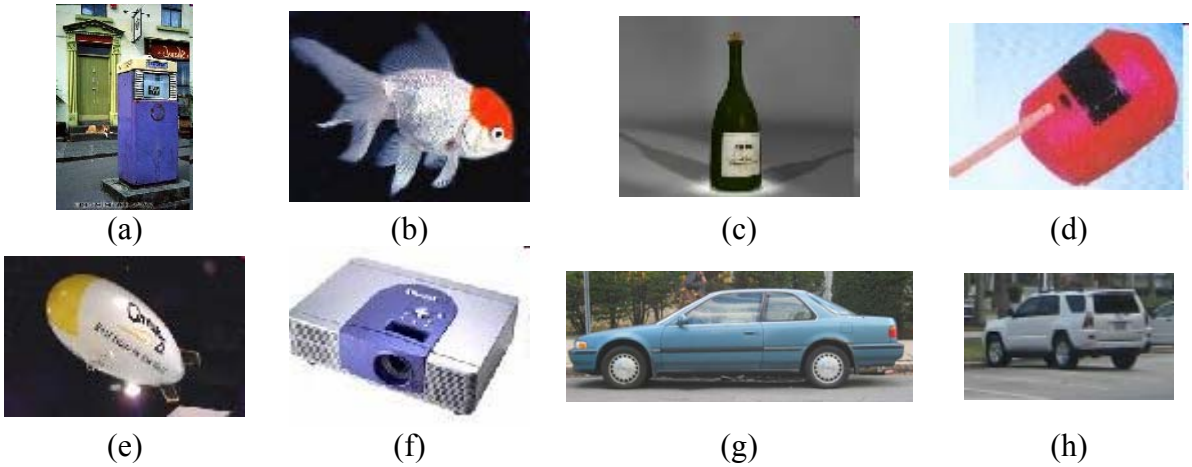


Figure 3.5 Convexity law in real-world objects. (a) the panel of the gas pump is embedded in the blue portion of the gas pump; (b) the read portion of the gold fish is embedded in the body of the fish; (c) the label of the whine bottle is embedded in the bottle; (d) the black portion of the welding mask is embedded in the red portion of the mask; (e) the yellow head of the blimp is embedded in the remaining white part; (f) the central blue portion of the video projector is embedded in the remaining part of the projector; (g)-(h) the windows and wheels of the vehicles are embedded in the bodies of the vehicles. The embedded components of these objects help increase the convexities of the objects. (images from [Griffin07] and [Russell08])

presented in Figure 3.6 – in (a)-(d) the windows of the vehicles have similar size with the bodies and share certain amount of boundaries with the bodies; in (e)-(f), the roofs of the buildings have similar size with the remaining parts of the buildings and share certain amount of boundaries with the remaining parts of the buildings. Therefore, if two neighboring object parts have relatively similar sizes and share certain amount of boundaries, then these two neighboring object parts are considered to be adjacent in 3-D world.

One challenging problem for segmentation is to determinate which image regions are seen as the foreground and which are seen as background. It has been known that convexity plays an important role in figure/ground separation – the regions perceived as foreground tends to be more convex than those assigned as background. Besides, as we mentioned earlier, the relative sizes plays a role in the figure/ground separation problem as well – small size objects tend to be perceived as foreground and large size objects tend to be perceived as background. The other cue that we find is useful for the figure/ground separation is the boundary complexity of image regions. Based on our observation, vegetations are common background objects in many natural scenes. Many vegetations have irregular shapes while most man-made objects have regular shapes. Examples of the similarity cue is shown in Figure 3.7 – in (a)-(f), the man-made objects in the images have smooth boundaries while the vegetations in the images have complex boundaries. Therefore, the similarity in boundary complexities can also plays a role on separating the man-made foreground objects from the background vegetations. This problem is addressed in our Perceptual Organization model.

3.4 Background identification for outdoor scene images

In order to detect objects' boundaries in a natural scene image, one needs to answer two basic questions [Malisiewicz07]: 1. Where may an object begin in a scene? 2. Where may an object end in that scene? We tackle the first question in this section.

According to [Shotton09], objects appearing in natural scenes can be roughly divided into two categories: unstructured objects and structured objects. Unstructured objects usually have nearly homogenous surfaces while structured objects usually consist of multiple parts with each part having distinct appearances (e.g. colors, textures). The common backgrounds in outdoor natural scenes are those unstructured objects like skies, roads, trees, grasses, etc (see Figure 3.8 for examples). These background objects have low visual variability and in most cases are distinguishable from other structured objects in an image. For instance, a sky usually has a uniform appearance with blue or white colors; a tree or grass usually has a textured appearance with green colors. Therefore, these background objects can be accurately recognized solely based on appearance information.

Suppose we can use a bottom-up segmentation method to segment an outdoor image into uniform regions (Superpixels). Then some of the regions must belong to the background objects. To recognize these background regions, we use a technique similar to [Shotton09]. The key for this method is to use *textons* to represent object appearance information. The term *texton* is first presented by [Malik01] for describing human textural perception. The whole textonization process proceeds as follows: Firstly the training images are converted to the perceptually uniform CIE Lab color space. Then the training images are convolved with a

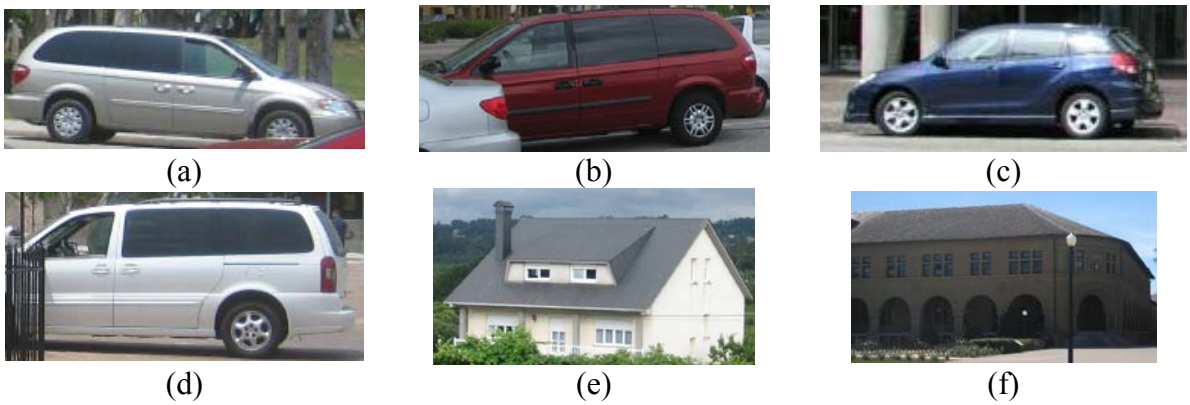


Figure 3.6 Proximity law in real-world objects. (a)-(d) the windows of the vehicles have similar size with the bodies and share certain amount of boundaries with the bodies; (e)-(f) the roofs of the buildings have similar size with the remaining parts of the buildings and share certain amount of boundaries with the remaining parts of the buildings. (images from [Griffin07] and [Russell08])

Figure 3.7 Similarity law in real-world objects. (a)-(f) the man-made objects in the images have smooth boundaries while the vegetations in the images have complex boundaries. (images from [Griffin07] and [Russell08])

.

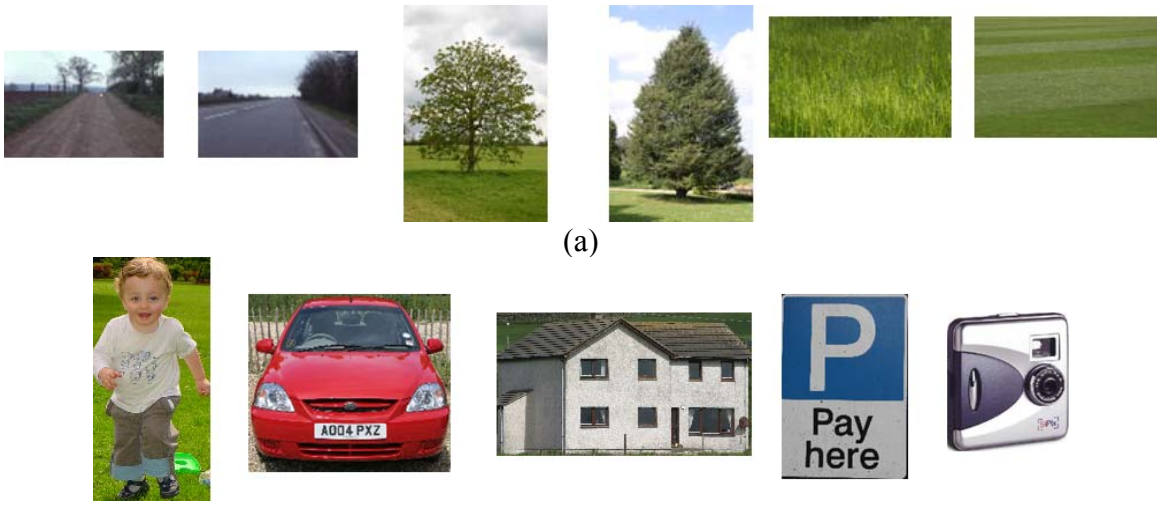


Figure 3.8 examples of background and structured objects. (a) background objects (sky, road, tree and grass); (b) structured objects; (images from [Griffin07] and [Shatton09])

17-dimensional filter-bank. We use the same filter-bank as [Winn05], which consists of Gaussians at scales 1, 2, 4; x and y derivatives of Gaussians at scales 2 and 4 and Laplacians of Gaussians at scales 1, 2, 4, 8. The Gaussians are applied to all three color channels, while the other filters are applied only to the luminance (L) channel. By doing so, we obtain a 17-dimensional response for each training pixel. The 17-dimensional response is then augmented with the CIE L, a, b channels to form a 20-dimensional vector. This is different from [Shotton09] because we find that after augmenting the three color channels, we can achieve slightly higher classification accuracy. Then the Euclidean-distance K-means clustering algorithm is performed on the 20-dimensional vectors collected from the training images to generate K cluster centers. These cluster centers are called *textons*. Finally each pixel in each image is assigned to the nearest cluster center, producing the *texton map*. More details about the textonization process are referred to [Winn05]. After this textonization process, each image region of the training images is represented by a histogram of textons. In addition, we calculate the centroid of the segment and augment the centroid location (in upper, middle or bottom area of an image) with the 150-dimensional texton histogram to form a 151-dimensional appearance feature vector. The reason to augment region location is that some objects like sky tends to occur in upper area of the images, while road or water tends to appear at the bottom of the images. Therefore the absolute location of a region also provides useful information for some object classes.

We then use these training data to train a set of binary Adaboost classifiers [Friedman00] to classify the unstructured objects (skies, roads, trees, grasses, etc.). Similar to the result from [Shotton09], our classifiers also achieve high accuracy on classifying these background objects on outdoor images. Each of the superpixels that are classified as non-background objects may correspond to a portion of an object surface. Therefore, after background identification, we are able to find the beginning part of an object in the image (see Fig. 3.14-(b) for an example). The question left unanswered is given the beginning part of an object, where the remaining parts of that object are?

3.5 Perceptual Organization model

Now the problem that we are interested in is: Given an object part of an unknown object, can we find the remaining parts of the object in an image without any object-specific knowledge of the object?

To tackle this problem, we develop a Perceptual Organization model. Our strategy works as follows: Since there always exist some special structural relations that obey the principle of *non-accidentalness* among the constituent parts of an object, we may be able to group the set of parts together by capturing these special structural relations. This problem is formalized as follows:

Problem definition: Let Ω be the domain of definition of a scene image, $\Omega = R_B \cup R_S$. R_B denotes the regions that belong to backgrounds. After the background separation, we known that all the structured objects in the image are contained in a sub-region $R_S \subset \Omega$. Let P_0 be the initial partition of Ω from a bottom-up segmentation method. Let a denote a uniform patch from the initial partition P_0. For $\forall\ (a \in P_0) \wedge (a \in R_S)$, we want to find the maximum region $R_a \subset R_S$ so that, $a \in R_a$ and for any uniform patch i, where $(i \in P_0) \wedge (i \in R_a)$, i should have some special structural relations that obey the *non-accidentalness* principle with the remaining patches in R_a. This is formulated as follows:

$$R_a = \arg\min_R (E[\partial R]) \quad \text{with} \quad a \in R \wedge R \subset R_S \tag{3.1}$$

where R is a region in R_S, ∂R is the boundary of R, $E[\partial R]$ is a boundary energy function, which is defined as follows [Jermyn01]:

$$E[\partial R] = \frac{\int_{O^1} \gamma'(t)^{\perp} \cdot v(\gamma(t))dt}{L(\partial R)} \tag{3.2}$$

where $L(\partial R)$ is the length of the boundary of R and γ is an injection from a circle O^1 into Ω. t is an arbitrary parameterization of O^1. Therefore, $\gamma(t)$ is a point in Ω. $\gamma'(t)$ is a tangent vector to the boundary of R. $\gamma'(t)^{\perp}$ is a normal vector, oriented according to the orientation of the boundary of R. v is an user-defined vector field on Ω derived from image data. According to Green's theorem:

$$\int_{O^1} \gamma'(t)^{\perp} \cdot v(\gamma(t))dt = -\iint_R \nabla \cdot v(x,y)dxdy \tag{3.3}$$

This relates the line integral of a vector field v over the boundary of a region R to a double integral of its divergence over the interior. Therefore, instead of encoding the information along a boundary, we are able to encode the information over the region enclosed by the boundary. Let function $f(x,y) = \nabla \cdot v(x,y)$, the boundary energy becomes:

$$E[\partial R] = \frac{-\iint_R f(x,y)dxdy}{L(\partial R)} \tag{3.4}$$

Thus, the function $f(x,y)$ encodes data from image region R. By choosing $f(x,y)$ appropriately, one can extract any interesting region from an image. For example, let IT represents image intensity, $f(x,y)=IT$ looks for bright areas and $f(x,y)=e^{-IT}$ prefers dark areas. The general form of f is defined as follows (Jermyn and Ishikawa 2001):

$$f(x,y) = M(\rho(T[I](x,y), Y_s)) \tag{3.5}$$

where M is a monotonically-decreasing function of its argument. Given an image I, T maps a point in an image domain to a feature space $T[I]$: $\Omega \rightarrow Y$. Y is a metric space with metric ρ, Y_s is a point in Y. Thus, the value $f(x,y)$ is large when the feature value at point (x,y), $T[I](x,y)$, is close to Y_s. In our case, we want to use $f(x,y)$ to encode Gestalt laws. Therefore, we define $f(x,y)$ as follows:

$$f(x,y) = e^{-\theta \bullet abs(S_i - S_a)} \quad \text{with } (x,y) \in i, i \in R \tag{3.6}$$

where θ is a weight vector and abs denotes the absolute value. S_i is a point in the structural context space and S_i=[B_i C_i]. S_a is a reference point in the structure context space and it encodes structure information of a. Since a is a starting point of an unknown object and it is the only known information that we have about the unknown object, we use a as reference point. C_i is the connectedness strength, which we will define later. B_i is the boundary complexity of image patch i, which can be measured as [Su06]:

$$B_i = \frac{1}{N}\sum_{d=1}^{N} A(s,k) \cdot F(s,k) \tag{3.7}$$

$$A(s,k) = 1 - \frac{\| p_{d+ks} - p_d \|}{\sum_{c=1}^{k} \| p_{d+cs} - p_{d+(c-1)s} \|} \tag{3.8}$$

$$F(s,k) = 1 - 2 * \left| 0.5 - \frac{n}{N-3} \right| \tag{3.9}$$

where N is the number of pixels of the boundary of image patch i, k is the length of a sliding window over the entire boundary of patch i. $A(s,k)$ and $F(s,k)$ are the strength and frequency of the singularity at scale(step) s. p_d and p_{d+ks} are the two ends of a segment of the boundary in the window. p_{d+cs} and $p_{d+(c-1)s}$ are the pixels between p_d and p_{d+ks}. n is the number of the notches in the window. A notch means a non-convex portion of a polygon, which is defined as a vertex with an inner angle larger than 180° . The detail definitions of notch, $A(s,k)$ and $F(s,k)$ are given in [Vasselle93]. See figure 3.9 for examples for notch, $A(s,k)$ and $F(s,k)$. Small B_i value means patch i has a regular shape (i.e., a smooth boundary). Large B_i value means patch i has an irregular shape. Examples of regular shapes and irregular shapes are shown in Figure 3.10. If patch i has similar shape regularity to patch a, then the function f might have large value inside patch i (depends on the connectedness C_i).

After obtaining the boundary complexity of patch i, we need to measure how tightly image patch i is connected to the parts of the unknown object that contain image patch a. C_i is the cohesiveness strength and is calculated as:

$$C_i = \begin{cases} 1 & \text{for } i = a \\ \max_j (e^{-\phi_{ij}\varphi_{ij}\lambda_{ij}} C_j) & \text{for } i \neq a \wedge j \in neighbor(i) \end{cases} \tag{3.10}$$

where j is a neighboring patch of patch i, ϕ_{ij} measures the symmetry of i and j along a vertical axis and is defined as:

$$\phi_{ij} = 1 - \delta(y_i, y_j) \tag{3.11}$$

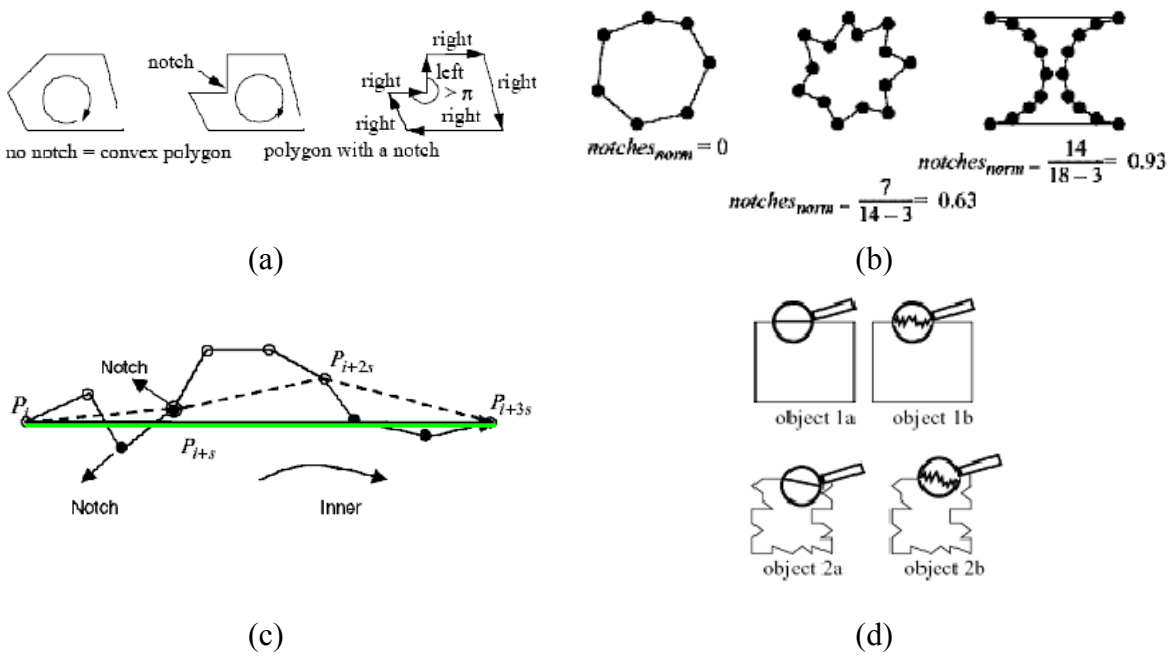


Figure 3.9 (a) Examples of notch from [Vasselle93]. (b) Normalized number of notch can be used to measure the frequency of vibration of a boundary from [Vasselle93]. (c) Illustration of strength of vibration from [Su06]: the solid line is the original curve; the green line is the chord from p_i to p_{i+3s}, the dashed line is the curve at scale s. It can be observed that the ratio between the length of green line and dashed line can be used to measure the vibration strength. (d) Examples of local vibration and global vibration from [Vasselle93].

.

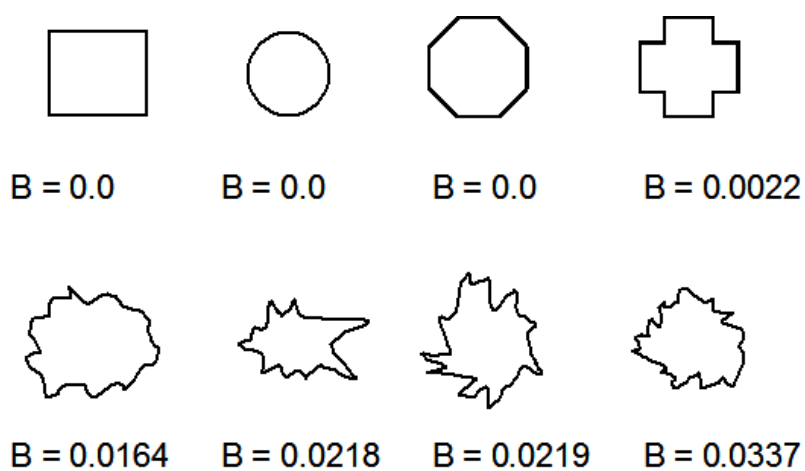


Figure 3.10 Examples of shape regularity. First row: regular shapes. Second row: irregular shapes. Notice that regular shape objects have smaller B values than irregular shape objects.

.

where δ is *Kronecker delta* function; y_i and y_j are the *y-coordinate* of the centroids of i and j. If y_i and y_j are very close, then ϕ_{ij} will be zero and image patch i will have the same connectedness to a as its neighbor j. This is because parts that are approximately symmetric along a vertical axis are very likely belonging to the same object. Examples of symmetry relation are shown in Figure 3.11-(a).

φ_{ij} measures the alignment of i and j:

$$\varphi_{ij} = \begin{cases} 0 & if\ e(\partial ij) \cap \partial i = \Phi \wedge e(\partial ij) \cap \partial j = \Phi \\ 1 & if\ e(\partial ij) \cap \partial i \neq \Phi \vee e(\partial ij) \cap \partial j \neq \Phi \end{cases} \tag{3.12}$$

where ∂i and ∂j are the boundary of i and j respectively; $e(\partial ij)$ is the extension of the common boundary between i and j. Φ denotes empty set. If two parts are strictly aligned along a direction, then, it is a strong indication that the two parts may belong to the same object. Examples of alignment relation are shown in Figure 3.11-(b).

If i and j are neither symmetric nor aligned, then the connectedness of image patch i depends on how it is attached to j. λ_{ij} measures the attachment strength between i and j. It is defined as:

$$\lambda_{ij} = \beta * e^{-\alpha \frac{(\cos\omega) * L(\partial ij)}{L(\partial i) + L(\partial j)}} \tag{3.13}$$

where α and β are constants. The attachment strength depends on the ratio of the common boundary length between i and j to the sum of the boundary lengths of i and j. If there is a large difference in size between i and j (i.e., $L(\partial i) >> L(\partial j)$ or $L(\partial j) >> L(\partial i)$), it usually means that the large one belongs to background like a wall or a big vegetation. Thus, i and j have a weak connectedness. If i and j have similar sizes and share a long common boundary, then i and j are strongly connected. ω is the angle between the line connecting the two ends of ∂ij and the horizontal line. Many objects may stand next to each other in a natural scene. Therefore, even if patch i and patch j are tightly attached along horizontal direction, patch i and patch j may still belong to two neighboring objects. We use "cos ω" in Equation (10) to control the connectedness strength of two attaching patches according to the orientation of the attachment. Examples of strong attachments and weak attachments are shown in Figure 3.11-(c) and Figure 3.11-(d).

In our implementation, we first set the cohesiveness of image patch a to one and then gradually spread the connectedness outward until it reaches all the patches in R. We set parameters $\theta = [18, 3.5]$, $\alpha = 20$ and $\beta = 3$ for our Perceptual Organization model. The parameters are set empirically. We tested our Perceptual Organization model on 50 street scene images selected from Labelme dataset [Russell08]. We tried different combinations of θ, α and β. The combination of $\theta = [18, 3.5]$, $\alpha = 20$ and $\beta = 3$ gives us the best performance of segmentation accuracy.

We have explicitly encoded four Gestalt laws (i.e., *Similarity, Symmetry, Continuation and Proximity*) into our Perceptual Organization model. The *Convexity* law is actually implicitly encoded in the Perceptual Organization model. This is shown in an example in Figure 3.12. In the synthetic image, patch b has a weak connectedness to patch a due to a big difference in sizes. Therefore, the two patches should not be treated as one entity. However, the boundary energy of the region that contains patches a and b is smaller than that of the region that only contains patch a. Therefore, patches a and b are treated as one entity by our Perceptual Organization model. The reason is that, the boundary length of the region that contains a and

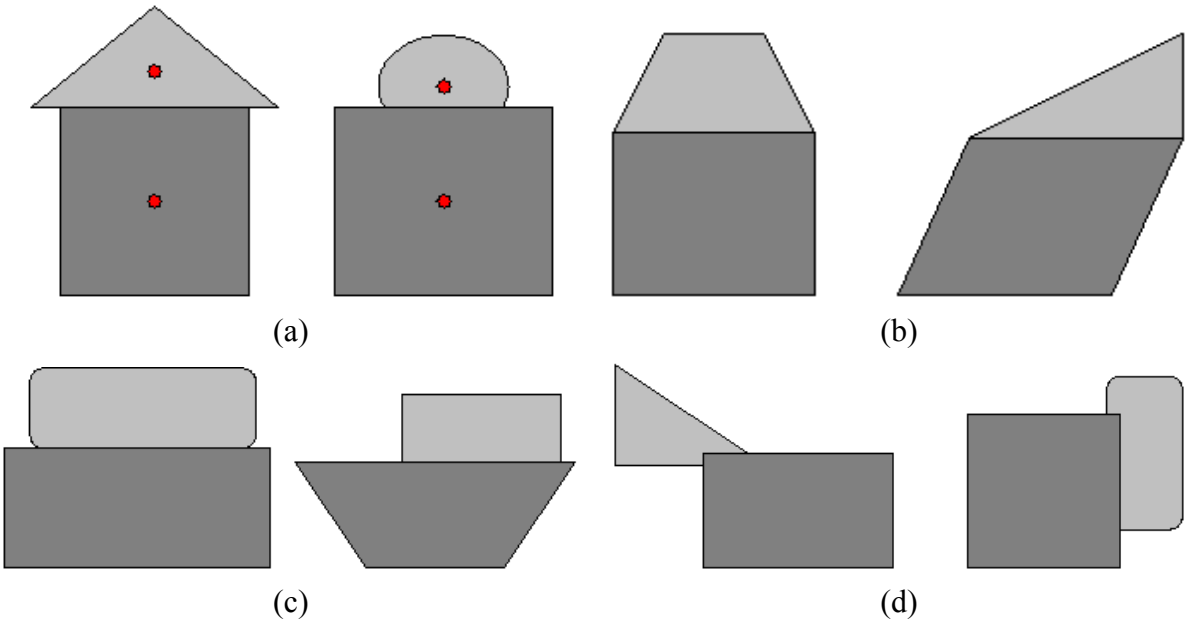


Figure 3.11 (a) Symmetry relations: the red dots indicate the centroids of the components. (b) Alignment relations. (c) Two components are strongly attached. (d) Two components are weakly attached.

.

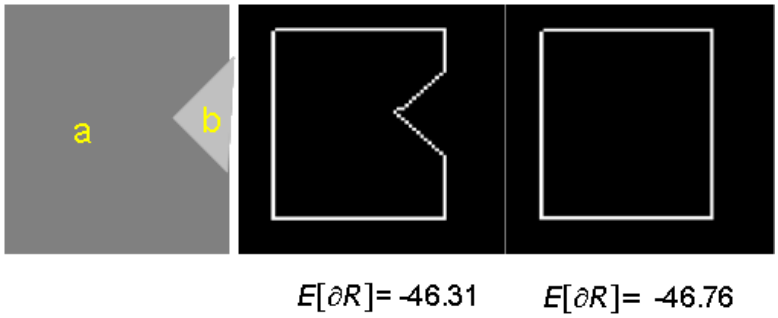


Figure 3.12 Example of Convexity relation. The boundary energy ($E[\partial R]$) of region *a* (middle) is larger than that of the region that contains *a* and *b* (right). Thus, our Perceptual Organization model will group *a* and *b* together.

.

b is shorter than that of the region that contains only patch *a*, since the patch *b* helps complete a big entity. As a result, the density of the flux of structural potential through the boundary of region containing *a* and *b* is higher than that of the region containing only *a*. In sum, any components that help complete a big entity will be grouped together with the other components composing that big entity due to their contribution of reducing the boundary length of the new-formed big entity.

Therefore, similar to human visual system, our Perceptual Organization model can 'perceive' a list of special structural relations that obey the principle of *non-accidentalness* like similarity in shape regularity, symmetry, alignment, adjacency, tendency of convexity, etc. The 'perception' is quantified by boundary energy – whenever a new member is added in a group, if the new member has some structural relations obeying the principle of *non-accidentalness* with other members in the group, then the boundary energy of the new-formed group is smaller than that of the old group. Otherwise, the boundary energy of the new-formed group is higher than that of the old group.

The remaining task is to find the region R_a in Equation (3.1), which has the minimum boundary energy among all the regions that contain image patch *a*. In other words, we need to find the maximum region that contains image patch *a* and all image patches contained in the region have some special structural relations obeying the principle of *non-accidentalness* with each other. This region may be the region occupied by the whole object. The challenge is that there may exist a large number of possible regions that contain image patch *a* and it is computationally expensive to search all the possible regions to find the one with global minimum boundary energy. Therefore, we develop a boundary detection algorithm based on a breadth-first search strategy. Instead of finding the region with the global minimum boundary energy, the algorithm tries to find a region with the local minimum boundary energy. Although it is not guaranteed that the algorithm is always able to find the region with the optimal boundary energy, we have found that it works quite well in practice.

Algorithm 1 Boundary Detection

INPUT: P_0 , R_S, and the reference image patch *a*

OUTPUT: region R_a that contains *a* with the minimal boundary energy in a local area of $R_a \cup neighbors(R_a)$

1. Let $R_a = a$.
2. Let $NR_a = \{v_n \mid (v_n \in P_0) \wedge (v_n \in R_S) \wedge (v_n \in neighbors(R_a))\}$.
3. Repeat step 4-7 for $q = 1, \ldots, n$.
4. Select a subset of NR_a with q regions: $\mu = (u_1, \ldots, u_q)$. So that $\forall\ x, y <= q$, there exists a path in μ connecting u_x to u_y.
5. Measure the boundary energy of $R_a \cup \mu$ with Equation (4).
6. If $E[\partial(R_a \cup \mu)] < E[\partial(R_a)]$, set $R_a = R_a \cup \mu$, GOTO step 2.
7. Otherwise select the next set of μ from NR_a and repeat step 4-7 until all possible μ have been tested.
8. Return R_a.

At the beginning, R_a only contains image patch a. Then, the algorithm measures the boundary energy of the combination of R_a and its immediate neighbors. The algorithm stops when no combination of R_a and its immediate neighbors have smaller boundary energy than that of R_a. The "$q = 1$" in Step 3 tests the combination of R_a and a single neighboring region of R_a. The "$q = 2$" tests the combination of R_a and a pair of connected neighboring regions of R_a, and so on. In practice, we have found that even when $n = 2$, the algorithm performs well in general.

An example of our boundary detection algorithm is shown in Figure 3.1. In this synthetic image, all four patches have different surface characteristics (i.e., colors) and shapes. We want to detect the boundary of the object that contains patch a. The f values (in Equation 6) in patch a decides the boundary energy of patch a, which is evaluated as: $f(x,y) = exp(-\theta * abs(S_a - S_a)) = 1$. Then, the boundary energy of patch a can be calculated from Equation (4): $E[\partial(a)] = -\iint_a 1\ dxdy\ /\ L(\partial(a))$. We then test the boundary energy of the region that contains a and its immediate neighbor b. Patch b has a strong connectedness to patch a, because they are aligned and also symmetric along a vertical axis. In addition, like patch a, patch b also has a smooth boundary. Thus, the f value in patch b also equals to one and the boundary energy $E[\partial(a\cup b)] < E[\partial(a)]$. According to our boundary detection algorithm (Step 6), the object now contains both patch a and b and the new neighbors of the object are patch c and d. The next step is to test if part d also belongs to the object. The f value in patch d is very small because patch d only shares a small amount of boundary with patch b. Therefore, parch d has a weak connectedness to patch b, and hence, $E[\partial(a\cup b\cup d)] > E[\partial(a\cup b)]$. So patch d might not be part of the object. The remaining neighbor is patch c. Although patch c has a strong connectedness to patch b, unlike patch a, it has an irregular shape. Therefore, the f value in patch c is much smaller than one and $E[\partial(a\cup b\cup c)] > E[\partial(a\cup b)]$.So patch c might not belong to this object either. Since patch d and patch c are not connected, the algorithm stops and return $R_a = a\cup b$, which has the minimum boundary energy in the area of $R_a \cup c\cup d$. It is not necessary to measure $E[\partial(a\cup b\cup c\cup d)]$ because we have proven that, if "$E[\partial(a\cup b\cup d)] > E[\partial(a\cup b)]$" and "$E[\partial(a\cup b\cup c)] > E[\partial(a\cup b)]$" and "$d$ and c are not connected", then "$E[\partial(a\cup b\cup c\cup d)] > E[\partial(a\cup b)]$".

Lemma 1 Let region C and region D be neighbors of region A. Regions C and D are not connected. If "$E[\partial(A\cup D)] > E[\partial(A)]$" and "$E[\partial(A\cup C)] > E[\partial(A)]$", then "$E[\partial(A\cup C\cup D)] > E[\partial(A)]$ ".

Proof. Let $F_A = \iint_A f(x,y)\ dxdy$; $F_C = \iint_{C+\partial(AC)} f(x,y)\ dxdy$; $F_D = \iint_{D+\partial(AD)} f(x,y)\ dxdy$;

According to Equation (4), $E[\partial(A\cup C)] > E[\partial(A)] \rightarrow (F_A+F_C)/(L(\partial A) + L(\partial C) - 2L(\partial AC)) < F_A/L(\partial A) \rightarrow F_C < F_A(L(\partial C) - 2L(\partial AC))/L(\partial A)$;

Similarly, $E[\partial(A\cup D)] > E[\partial(A)] \rightarrow (F_A+F_D)/(L(\partial A) + L(\partial D) - 2L(\partial AD)) < F_A/L(\partial A) \rightarrow F_D < F_A(L(\partial D) - 2L(\partial AD))/L(\partial A)$;

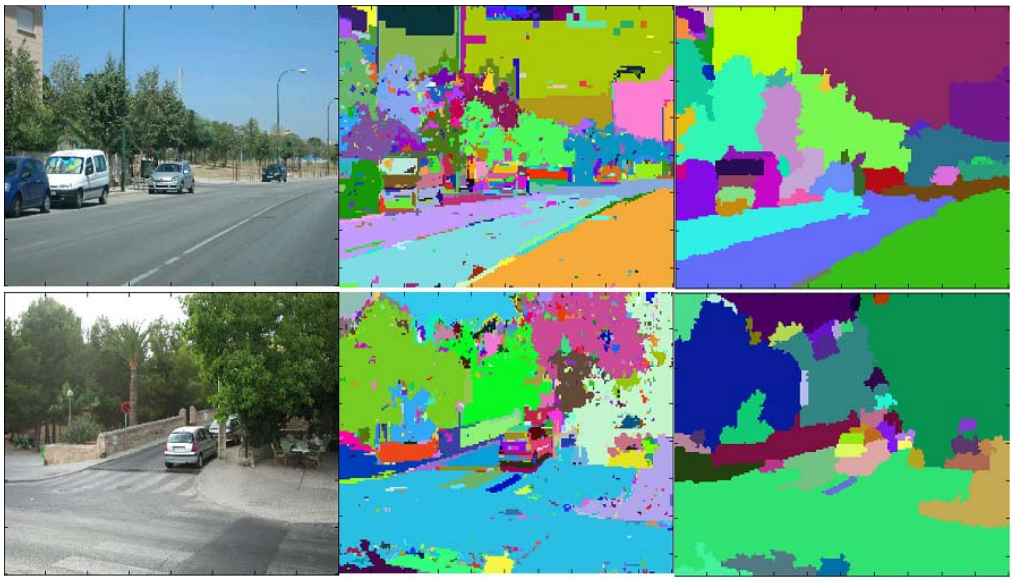
Finally, $E[\partial(A\cup C\cup D)] = -(F_A+F_C+F_D)/(L(\partial A)+L(\partial C)+L(\partial D) - 2L(\partial AC)-2L(\partial AD)) > -F_A(L(\partial A)+L(\partial C)+L(\partial D)-2L(\partial AC)-2L(\partial AD))/((L(\partial A)+L(\partial C)+L(\partial D)-2L(\partial AC)-2L(\partial AD)\ L(\partial A)) = -F_A/L(\partial A) = E[\partial(A)]$. □

3.6 Get good spatial support for object parts

The Perceptual Organization model introduced in Section 3.4 can 'perceive' various special structural relations that obey the principle of *non-accidentalness* among the constituent parts of an object. Therefore, to apply the proposed Perceptual Organization model to real-world natural scene images, we need to have good spatial support for the object parts. In other words, we need to segment an image into regions so that each region approximately corresponds to an object part. This task is achievable because most object parts have nearly homogenous surfaces and many existing bottom-up image segmentation methods can efficiently detect the homogenous units in an image. In our implementation, we make use of Felzenszwalb and Huttenlocher's [Felzenszwalb04] approach to generate an initial partition for a natural scene image. We choose their method because it is not only very efficient, but also has the ability to preserve detail in low-variability image regions while ignoring detail in high-variability regions. Their method has the advantage of being able to handle different object parts with various surface properties. However, the raw segmentation result of their method in many cases are still too noisy (refer Figure 3.13 – (b)). To further increase the segmentation quality, we apply a segment-merge method on the initial superpixels to merge the small size regions (i.e., region size < 0.03% of the image size) with their neighbors. These small size regions are often caused by the texture of surfaces or by the inhomogeneous portions of some part surfaces. Since these small size image regions contribute little to the structure information (shape and size) of object parts, we merge them together with their larger neighbors to improve the performance of our Perceptual Organization model. In addition, if two adjacent regions have similar colors, we also merge them together. By doing so, we obtain a set of improved superpixels. Most of these improved superpixels correspond to object parts. Examples of improved Superpixels are presented in Figure 3.13. The whole process takes about 6 seconds in our Matlab implementation.

3.7 Image segmentation algorithm

We now turn into image segmentation algorithm. Given an outdoor scene image, we firstly apply the segment-merge technique described above to generate a set of improved superpixels. Most of the superpixels correspond approximately to object parts in that scene. We build a graph to represent these superpixels: Let $G = (V, E)$ be an undirected graph. Each vertex $v_e \in V$ corresponds to a superpixel and each edge $(v_e, v_f) \in E$ corresponds to a pair of neighboring vertices. We then use our background classifiers described in Section 3.4 to divide V into two parts: backgrounds like skies, roads, grasses, trees, etc. (R_B) and structured parts (R_S). All the structured objects in the scene are therefore contained in R_S. We then apply Perceptual Organization on R_S . At the beginning, all the components in R_S are marked as unprocessed. Then, for each unprocessed component v_u in R_S, we use the boundary detection algorithm described in the Section IV-B to detect the best region O_u that contains vertex v_u. The region O_u may correspond to a single object or the semantically more



(a) (b) (c)

Figure 3.13 Get spatial support for object parts. (a) Original images. (b) The initial partitions from Felzenszwalb and Huttenlocher's (2004) approach. (c) The final superpixels after applying a segment-merging method on the initial partition. The final superpixels approximately correspond to object parts.

meaningful part of an object. We mark all the components composing the O_u as processed. The algorithm gradually moves from the ground plane up to the sky until all the components in R_S are processed. Then we finish one round of Perceptual Organization procedure and use the grouped regions in this round as a new set of components to start a new round of Perceptual Organization on R_S. At the beginning of a new round Perceptual Organization, we merge the adjacent components if they have similar colors and build a new graph for the new components in R_S. This Perceptual Organization procedure is repeated for multiple rounds until no components in R_S can be grouped with other components. In practice, we find that the result of two rounds grouping is good enough in most cases. At last, in a post-processing step, we merge all the adjacent sky objects and ground objects together to generate final segmentation. An illustration of the algorithm's pipeline is shown in Fig. 3.14.

Algorithm 2 Image segmentation

INPUT: a graph $G = (V, E)$, and a division of V: Structured parts (R_S) and Backgrounds (R_B) .

OUTPUT: a segmentation of V into objects or regions $SS = (O_1, \ldots, O_k)$

1. Set $SS = R_S$;
2. Repeat steps 2-9 until members in SS do not change;
3. Let known regions $R_k = R_B$ and unknown regions $R_{uk} = R_S$; Set $ST = \varnothing$;
4. Let outer regions $R_o = \{v_o \mid (v_o \in V) \wedge (v_o \in R_{uk}) \wedge (v_o \in neighbors(R_k))\}$;

5. While $R_o \neq \emptyset$, repeat steps 5-7s;
6. For each $v_o \in R_o$,
7. Call the boundary detection algorithm in Section 3.3 to detect the region of the object O_{vo} that contains v_o.
8. Let $R_o = R_o \setminus O_{vo}$; $R_{uk} = R_{uk} \setminus O_{vo}$; $ST = ST \cup O_{vo}$ and $R_k = R_k \cup O_{vo}$; Let $R_t = \{v_t \mid v_t \in V \wedge v_t \in R_{uk} \wedge v_t \in neighbors(O_{vo})\}$ and set $R_o = R_o \cup R_t$;
9. Finish one round grouping, Let $SS = ST$ and use SS members to rebuild G;
10. Let $SS = SS \cup R_B$ and return SS;

3.8 Discussion

It is well accepted that segmentation and recognition should not be separated and should be treated as an interleaving procedure. Our method basically follows the scheme. Our method requires identifying some background objects as a starting point. Compared to the nearly unlimited number of structured object classes, there are only several common background objects in outdoor scenes. These background objects have low visual variety and hence can be reliably recognized. After background objects are identified, we can roughly know where the structured objects are and delimit the grouping in certain areas of an image. For many objects like the major object classes appearing in street (buildings, vehicles, signs, people, etc.) and many other objects illustrated in Section 3.3, our method can piece the whole objects or the main portion of the objects together without requiring recognizing them. In other words, for these object classes, our method provides a way to separate segmentation and recognition. This is the major difference between our method and other class segmentation methods which require recognizing an object in order to segment it. Our work shows that for many fairly articulated objects, recognition may not be a requirement for segmentation. The geometric relations of the constitute parts of the objects provide useful cues indicating the memberships of these parts.

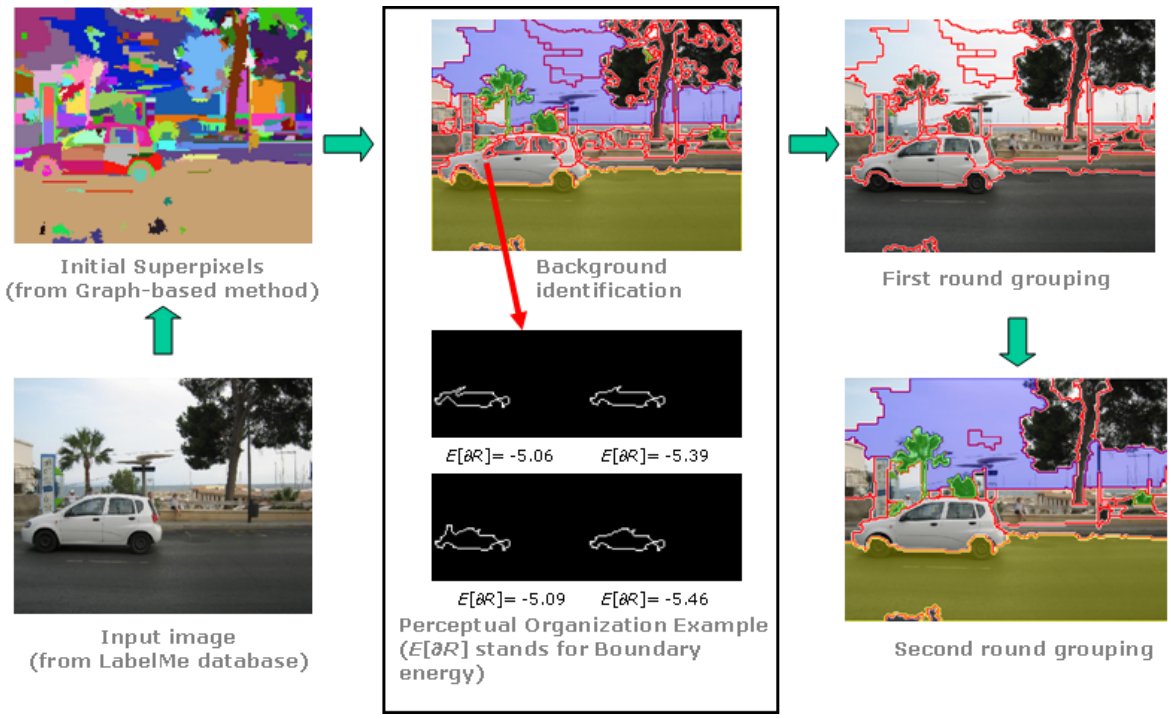
For some objects that have complex structures like bicycles, motorcycles, some complex buildings etc., our simple Perceptual Organization model may not be able to piece the whole objects together. Instead, it may only piece some semantically meaningful parts like wheels, windows or doors together. For these objects, object recognition is still required to generate good segmentations. The results of our method can greatly help the recognition by providing some semantically meaningful parts of these objects.

Future extension to this work includes integrating more Gestalt laws into the Perceptual Organization model. In current model, we only check the symmetry relations between two vertically attached parts. Symmetry along other directions should also be considered. The closure law which can be used to handle occlusions is also desired for a Perceptual Organization model. Besides, we also want to find better way of combining different Gestalt laws. Our Perceptual Organization gives more concerns on symmetry, continuity and convexity. This is because some psychology experiments show that sometimes convexity law can overrule other laws like closure. Symmetry law and continuity law sometimes can overrule proximity law. However, currently there are no conclusive theories in the

psychology literature showing which law overrules other laws and how to overrule other laws. A systematical research on this problem will guide us to develop better ways to combining different Gestalt laws together. We also want to incorporate more object classification techniques in our method to recognize the common objects appearing in outdoor scene. We believe that segmentation and recognition should be interleaved to generate good segmentation. Recognizing some objects can greatly help segment and recognize other objects in an image. Every round, our Perceptual Organization model may group some objects or semantically meaningful parts of objects together. Recognizing these well segmented objects will be helpful for next round Perceptual Organization. This will further enhance the segmentation quality of our method for complex scene environments.

3.9 Summary

In this chapter, we developed a novel image segmentation algorithm. First we assume that we are handling outdoor scene images. The common backgrounds of outdoor scene images are skies, grounds, trees, grasses, etc. The backgrounds are nearly uniform and have special colors and textures. Therefore the backgrounds can be identified based on colors and textures information. Once backgrounds are identified, the remaining regions should belong to structured objects which are difficult to segment. Here another assumption we make is that most constituent parts of structured objects are nearly uniform. Our experiments show that the two assumptions hold for most outdoor scene images. The key component of the image segmentation algorithm is a Perceptual Organization model. We first illustrate a list of Gestalt cues with real-world objects. We then encode the list of Gestalt laws into a boundary energy model to build a Perceptual Organization model. By doing so, the Perceptual Organization model can ‘perceive’ the special structural relations among the constituent parts of an unknown object and hence can group them together without object-specific knowledge. The image segmentation algorithm developed in this chapter allows us to ‘perceive’ the salient objects in a place.



(a) (b) (c)

Figure 3.14 Illustration of our segmentation pipeline. (a) Down: Input images. Up: Initial superpixels from [Felzenszwalb04]. (b) Up: Improved superpixels with background objects identified. Sky is labeled as blue, ground is labeled as yellow and vegetations (tree or grass) are labeled as green. Down: an example of Perceptual Organization process. $E[\partial R]$ stands for boundary energy. Firstly the bottom white part of the white car is selected. The $E[\partial R]$ for the part is measured as -5.06. Then our Perceptual Organization model groups the two pieces of front windows with the white part based on convexity laws. The $E[\partial R]$ for these two regions is measured as -5.39 and -5.46 respectively. Except for these two parts and the small segment of the front wheel, other parts do not have special geometric relations with the white part, such as the bottom part of the white sign behind of the front part of the car. The $E[\partial R]$ for the region containing the sign part is measured as -5.09. Therefore, the region with $E[\partial R]$ as -5.46 is detected as the best region for the white part of the car. (c) Up: the result of the first round Perceptual Organization. Notice that the different parts of the white car now have been grouped into two big pieces. These two big pieces are aligned. Down: final segmentation result after second round perceptual Organization. Notice the different parts of the white car are grouped together as a single object. (This figure is best viewed in color.)

.

4 Object classification based on appearance, parts layout and shape

The image segmentation algorithm described in Chapter 3 can stably detect the boundaries of many salient objects under different outdoor scenes. Therefore, like human visual system, our system can 'perceive' the salient objects in scenes. The next step is to recognize the salient objects.

In this chapter, we present a novel method for object classification. Our main contributions are two-fold. Firstly, we build an informative object description, which consists of not only appearance, but also certain structural information like parts layout and shape of objects. All these information are combined into a high-dimensional vector. Secondly, we develop a new information-based wrapper feature selection method. Feature selection refers to search algorithms that select a subset of most characterizing features from an initial larger set of features. For many pattern recognition applications, feature selection is critic to minimize the classification error, especially when the original feature set is very large. With this feature selection method, for each object class, we can find a small subset of features that characterize the object class well. We then train a binary classifier for each object class based on the subset of predictive features selected by our feature selection method for classification (see Figure 4.1 for an example).

The remainder of this chapter is organized as follows: We first describe how to build an informative object description that consists of appearance, parts layout and shape in section 4.1. Our feature selection method is presented in section 4.2.

4.1 Building informative object description

In this section, we introduce our object description method. Our goal is to build an informative object description that encodes appearance, parts layout and shape information of an object. Intuitively, each object class has a specific characteristic of the combination of the three aspects of information. Therefore, with this informative object description, we may be able to find the specific information combination that characterizes an object class well. The remaining part of this section details how to quantify the three aspects of information respectively.

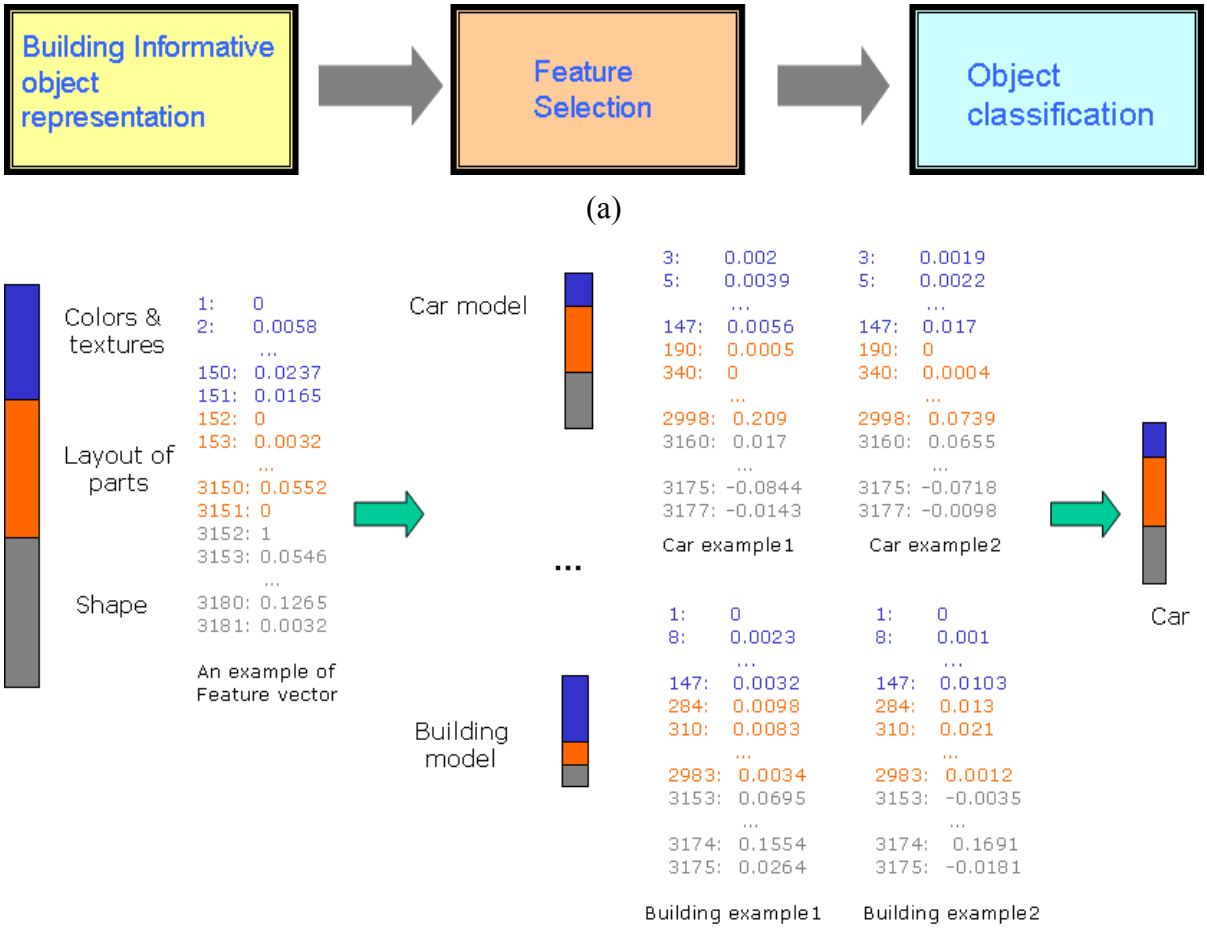


Figure 4.1 Illustration of object classification pipeline. (a) The object classification pipeline; (b) the numerical example of object classification pipeline. For each object, we first build a 3181-dimensional feature vector that describes appearance, parts layout and shape of objects. Then for each object class, we select a subset of predictive features. At last we train a binary classifier for each class with the selected predictive subset of features to classify the object class.

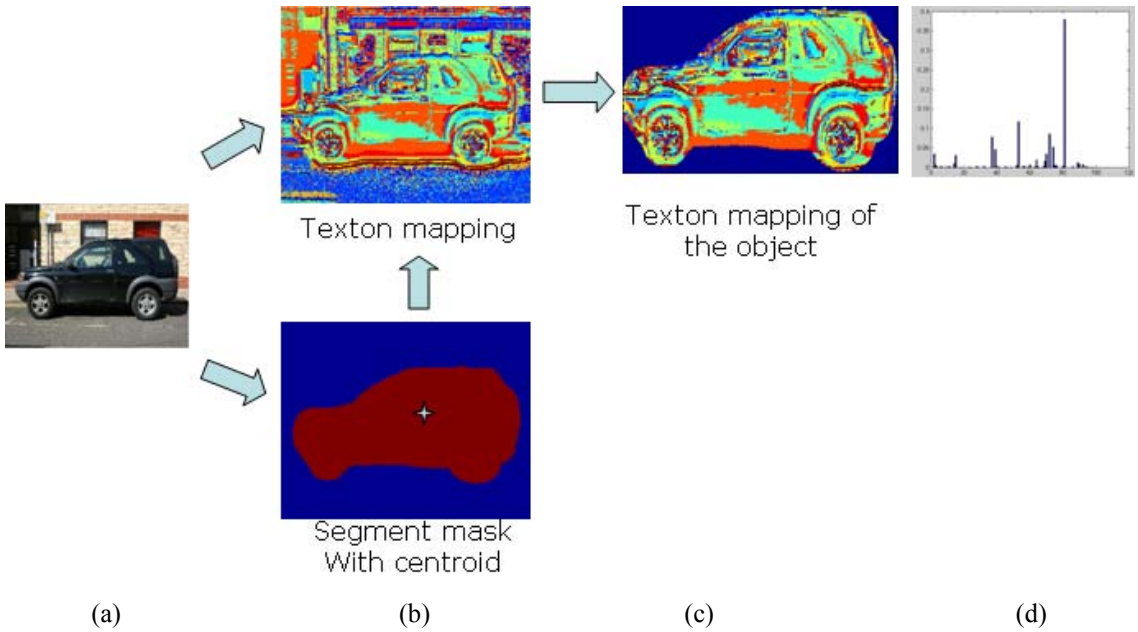


Figure 4.2 Illustration of building appearance features. (a) The input image; (b) upper: texton mapping of the input image; Down: segment mask of the object. The star is the centroid. (c) the texton mapping of the input object extracted with the segment mask. Different colors indicate different textons. (d) The normalized histograms of texton.

4.1.1 Build appearance features

Appearance information has been widely used in the literature for object classification. Among various appearance representations, *textons* has drowned many attentions [Shotton09, Winn05, Malik01, Varma05]. The term textons were originally used to describe human textural perception and in the literature usually refers to the clusters of feature vectors in a high dimensional space [Winn05]. Textons is well known for providing a compact representation for the range of different appearance of an object.

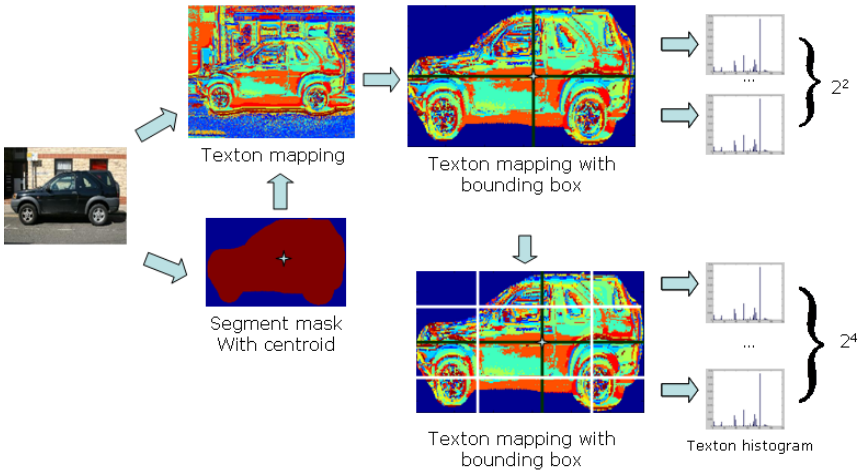
The process of textonization proceeds as follows. For each training image, firstly the RGB color space is converted to the perceptually uniform CIE Lab color space. Then a filter bank is applied to convolve the L, a, and b channels respectively. The filter bank is made of 3 Gaussians, 4 Laplacian of Gaussians (LoG) and 4 first order derivatives of Gaussians. The three Gaussian kernels (with σ = 1, 2, 4) are applied to each CIE L, a, b channel, thus producing 9 filter responses. The four LoGs (with σ = 1, 2, 4, 8) are applied to the L channel only, producing 4 filter responses. The four derivatives of Gaussians are divided into the two x and y aligned sets, each with two different values of (σ = 2, 4). Derivatives of Gaussians are also applied to the L channel only, thus producing 4 filter responses. Totally a pixel is associated a 17 dimensional filter responses. Then the 17 dimensional filter responses is augmented with the L, a, b channel values to form a 20 dimensional feature vector. This is slightly different with the works in [Winn05]. In our experiment, we find that after

augmenting the three color channels, we can achieve slightly higher classification performance. Each 20 dimensional feature vector records the local color and texture information around the pixel. Then the Euclidiean-distance *K*-means clustering algorithm is employed on the 20 dimensional feature vectors of all training pixels to produce k cluster centers (in our work, we used $k = 150$). The k cluster centers (textons) define a visual vocabulary. Then each pixel in each image is assigned to the nearest cluster center; thus generating a map of *textons* (see Figure 4.2-b for an example). Finally, assuming that a fairly accurate segment for each object can be obtained by our POM segmentation method, then a normalized histogram of texton occurrences across the region (segment) is computed. The histogram of texton occurrences captures the statistic appearance information of the object surface (see [Winn05] for details). In addition, we calculate the centroid of the segment and augment the centroid location (in upper, middle or bottom area of an image) with the 150-dimensional texton histogram to form a 151-dimensional appearance feature vector. The reason to augment region location is that some objects like sky tends to occur in upper area of the images, while road or water tends to appear at the bottom of the images. Therefore the absolute location of a region also provides useful information for some object classes.

4.1.2 Build parts layout features

One difficult problem for object classifications is the intra-class variations. Usually structured objects consist of multiple parts. Even for objects belonging to the same class, many parts of these objects have wide variations on appearances. For example, people often wear various kinds of clothes in different scenes and cars can be painted into different colors, etc. This often confuses the appearance-based classification methods and causes them to make incorrect decisions. However, some parts of the objects in the same class do tend to present relatively stable appearances. For instant, the appearances of wheels, windshields and headlights do not vary very much across different cars. People's faces relatively have more stable appearances than their bodies, etc. Besides, for objects belonging to the same class, the spatial arrangement of parts also has certain patterns. For example, the wheels of cars always appear in the bottoms, people's faces always appear on the top, etc. Therefore object parts and their spatial arrangements encode important information about object classes and can be used to handle the difficult intra-class variations problem.

One natural way to extract the parts layout information is to represent an object with parts and record the geometric relations among the parts [Ullman01, Lazebnik05, Fergus03]. However, partitioning an object into parts and detecting the geometric correspondence among the parts turns out to be difficult and computationally expensive [Lazebnik06]. Besides, this approach cannot handle large viewpoint variations and object deformations. For examples, many bottom-up methods can segment images into uniform regions (superpixels). But many constituent parts of structured objects are just nearly uniform. Under different viewpoints and illuminations, a constituent part may consist of different superpixels. Therefore directly using superpixels as object parts is not stable. We propose a solution that can efficiently extract the approximate parts layout information meanwhile be robust to certain degree for the viewpoint variations and object deformations. Our method is inspired by the 'subdivide and disorder' techniques [Lazebnik06]. The operation of 'subdivide and



(a) (b) (c)

Figure 4.3 Illustration of building parts-layout features. (a) Left: input object; Right-up: Texton mapping of input image; Right-down: segment mask with a bounding box. The star is the centroid. (b) the texton map of the car object. Different colors indicate different textons. Upper: coarse subdivision. Bottom: fine subdivision. (c) The normalized histograms of texton occurrences for each subblock. In total $2^2+2^4=20$ subblocks and each histogram has 150 dimensions. The 20 histograms are concatenated in order to form a 3000 dimensional feature vector.

disorder' involves partitioning an object into subblocks and compute histograms of local features. Specifically, given an object, first the bounding box of the object region is extracted. Then the centroid of the object is calculated. Centered on the centroid, the bounding box is divided into four subblocks. This is because the position of object centroid is stable under certain deformations. Each subblock is further equally divided into four subblocks. In total, $2^2+2^4=20$ subblocks are obtained. For each subblock, a normalized histogram of texton occurrences is computed. Finally the 20 normalized histograms are concatenated in order to form a 20x150=3000 dimensional feature vector (see Fig. 4.3 for an example).

Our method has some advantages. Firstly, we apply both a coarse and a fine subdivision on the object surface. This allows us to capture the appearance-stable parts with difference sizes. The major portion of large-size parts like windshields may be contained in the subblocks of coarse resolution while the major portion of small-size parts like wheels or headlights may be contained in the subblocks of fine resolution. Therefore the histograms corresponding to these subblocks will exhibit certain patterns. Since the histograms are concatenated in order, the arrangements of the appearance-stable parts are also approximately recorded. In addition, our method is robust to certain degree for viewpoint variations and object deformations. This is mainly because for each subblock, we totally ignore the spatial information and only record the statistic appearance information in the local area. As a result, local geometric relation variations caused by viewpoint variations or deformations will not cause too much change in the histogram of the appearance information drawn from the local area.

4.1.3 Build shape features

Shapes also provide useful information for object classifications. In many cases, people can readily identify an object solely based on its silhouette without any other extra information. Although many object classes like cars, people etc., have wide intra-class variations on the appearances, there always exist certain similarities in shapes for the objects in the same class. Therefore, shape property is also very useful for overcoming the intra-class variations problem for object classifications.

Gorelick et. al. [Gorelick05] propose an efficient method to extract the shape information of objects based on poisson equation. We briefly review their work as follows.

Consider a silhouette S embedded in a grid with mesh size h surrounded by a closed contour ∂S. For every internal point, place a set of particles at the point and let them move in a random walk until they hit the boundary contour. Let $U(x,y)$ denotes the mean time required for a particle to hit the boundaries from point (x,y). Then $U(x,y)$ is equal to the average value of its immediate four neighbors plus a constant:

$$U(x,y) = 1 + \frac{1}{4}(U(x+h,y) + U(x-h,y) + U(x,y+h) + U(x,y-h)) \tag{4.1}$$

Then (4.1) is a discrete form approximation of the poisson equation:

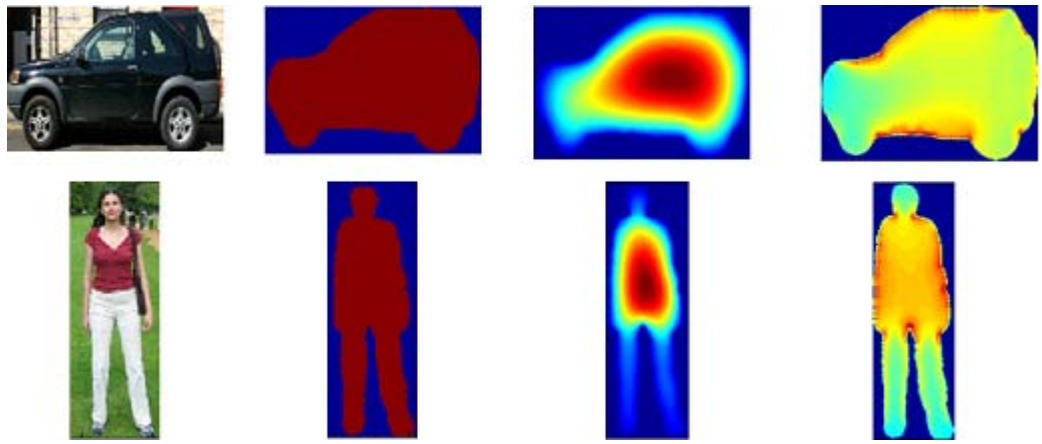
$$\Delta U(x,y) = -\frac{4}{h^2} \tag{4.2}$$

With $\Delta U = U_{xx} + U_{yy}$ denoting the Laplacian of U and $4/h^2$ denoting the overall scaling. The boundary condition of (2) is subject to Dirichlet boundary conditions $U(x,y) = 0$ at the bounding contour ∂S. Fig. 4.4 shows couple of examples of the solutions of poisson equation (2) obtained for the silhouettes. It can be observed that U consists of a set of contours which represent smoother versions of the bounding contour, which allows each internal point to 'feel' some properties of the boundary contour (shape). One property is the local orientation of the boundary shape. Let $M(x,y)$ be the Hessian matrix of U at point (x,y). It is known that the principal orientation of a shape is the orientation of the leading eigenvector of $M(x,y)$. Let $\alpha(x,y)$ denote the principal direction felt locally at (x, y). We can use $\alpha(x,y)$ to identify the vertical and horizontal regions of a shape by checking if $\alpha(x,y)$ is close to either zero or $\pi/2$:

$$W_0(x,y) = e^{-r\left|0-\left|\alpha(x,y)\right|\right|} \tag{4.3}$$

$$W_{\pi/2}(x,y) = e^{-r\left|\frac{\pi}{2}-\left|\alpha(x,y)\right|\right|} \tag{4.4}$$

Where γ is a constant (we used $\gamma = 3$). Based on an observation that U grows faster with distance from concavities than from convexities, the gradient magnitude should be higher in concave regions. So define:



(a) (b) (c) (d)

Figure 4.4 Examples of the solutions of poisson equation for silhouettes. (a) Input objects. (b) The silhouettes (shapes) of input objects. (c) The solutions of poisson equation *U*. (d) The log(1/ Φ) for two silhouettes. Notice the concave regions are marked as dark colors.

$$\Phi(x,y) = U(x,y) + \left\|\nabla U(x,y)\right\|^2 \tag{4.5}$$

Φ should have high value in concave regions (see Fig. 4.3-(d)). We then can use Φ to identify the concave regions of a shape:

$$W_c(x,y) = \frac{1}{1+e^{-\delta\Phi_n(x,y)}} \tag{4.6}$$

Where Φ_n is obtained by normalizing *Φ* so that its maximal absolute value is 1. We used $\delta = 4$. With the W_0, $W_{\pi/2}$ and W_c, we can use the following weighted moments to capture the statistic information of an object shape:

$$m_{pq} = \frac{\sum_x \sum_y w(x,y) x^p y^q}{\sum_x \sum_y w(x,y)} \tag{4.7}$$

Where p,q are the moments orders. $p,q \in \{0,1,2,3\}$ and $p+q \leq 3$. We substitute w with vertical measure W_0, horizontal measure $W_{\pi/2}$ and concave measure W_c, resulting in 10 moments per measure and 30 moments in total. More detailed explanation of this shape information extraction method can be found in [Gorelick05].

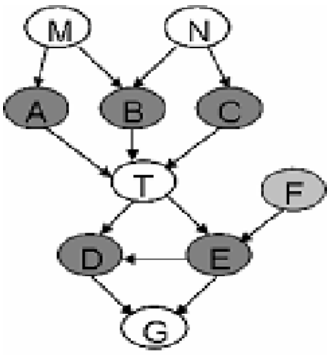


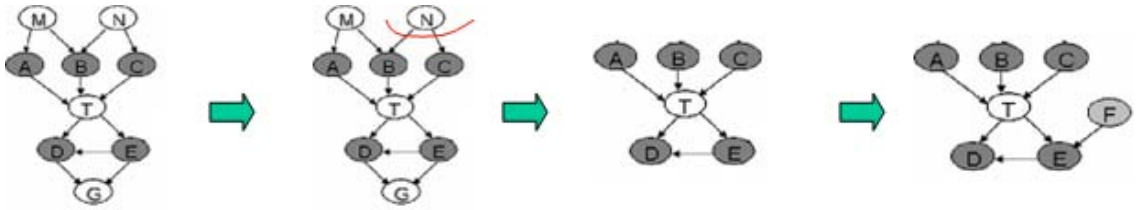
Figure 4.5 An example of a Bayesian Network. The parents and children of *T* are variables in dark gray. The Markov Blankets of *T* includes an additional variable *F*.

4.2 Feature selection

In this section, we introduce our feature selection approach. Feature selection refers to search algorithms that select a subset of most characterizing features from an initial larger set of features. When the initial set of features is small, optimal subset of features can be found by exhaustively searching. When the initial set of features is large (usually > 50), however, feature selection becomes challenging since the exhaustively search is infeasible due to the excessively large numbers of possible feature sets. In Section 4.1, we build a high-dimensional object representation for each object. This object representation consists of 151 dimensional appearance features, 3000 dimensional parts-layout features and 30 dimensional shape features. Thus in this case, only suboptimal feature selection algorithm can be applied. For each object class, our goal is to select the subset of features that 'best' characterizes the object class from the initial 3181 features.

Feature selection has been the focus of many statistical pattern recognition works [Tsamardinos03, Peng05, Yaramakala05]. Tsamardinos and Aliferis [Tsamardinos03] show that feature selection problem can be modeled as finding the Markov Blankets (MB) in Bayesian Networks (BN). A Bayesian network is a directed acyclic graph whose nodes represent random variables and edges represent conditional dependence. Nodes which are not connected represent variables that are conditionally independent of each other. The MB of a node in a BN is its parents, children and children's parents (see Fig. 4.5 for an example). In other words, the MB is the minimum conditioning set that makes all other random variables independent for a target variable. If we set the object label as a binary random variable with 1 indicating the target object class and 0 indicating other object classes and also set the appearance features, parts layout features and shape features as random variables, then our task is to find the MB of the binary label variable.

Many Markov blanket discovery algorithms have been proposed [Tsamardinos03, Aliferis03, Yaramakala05]. These methods all follow a two-phase scheme. In the first stage use some sorts of association measurements to select a set of candidate parents and children of target variable. This set of candidates is the superset of MB which contains some non-



(a) (b) (c) (d)

Figure 4.6 Illustration of our feature selection method.

members of MB. Then in the second stage, apply statistic independent tests to remove the false positives. These methods have some limitations. Firstly, most of association measurements of the methods are based on mutual information. Mutual information is biased on features that have large uncertainties. For instance, in our case, the parts-layout features reflect the local appearance information of objects, which tend to have larger uncertainties than features reflecting the global properties of objects. This may cause these methods miss some real MB members. Secondly, the conditional tests of independence they apply are G^2 test or *Fisher z* test. The number of the training instances to reliable estimate of the statistic tests is exponential to the size of the conditioning set. When the size of MB is large, the required number of training samples may reach to hundreds of thousands for these statistic tests to be reliable. However, for most image datasets, the available training samples are only several hundreds or several thousands. As a result, these methods may not be able to remove the false positives in the second stage and return an incorrect MB. Our feature selection method overcomes both of the limitations to some degrees.

Like other approaches, the first step of our method is to select a set of candidate parents and children of the target variable. Prior knowledge of features is very valuable for feature selection. Intuitively, each object class can be accurately described by a specific combination of appearance, parts layout and shape information. Thus, a good feature subset that well characterizes an object class should be the combination of the three classes of features. Based on this prior knowledge, for each class of features, we select the set of most predictive features as the candidate parents and children. The predictability of a feature for an object class is measured by the information gain [Israels83]:

$$G(X,T) = \frac{H(X) - H(X \mid T)}{H(X)} = \frac{I(X,T)}{H(X)} \tag{4.8}$$

Where T is the target binary class label variable. X is a feature variable. $H(X)$ is the entropy of X. $H(X|T)$ is the conditional entropy. $I(T,X)$ is the mutual information between T and X. Compared to mutual information, information gain reduces the bias on features that have large uncertainties and can be used to find the features that have large effect on target class label variable T. For appearance features and shape features, we select the features whose information gains are larger than the mean information gain of features in the class. For parts-layout features, we select 50 features with highest information gains. Since all parents and children should have large effect on the target variable, by selecting features with large information gains from each feature class set, we can cover most parents and children of the

target label variable. Let PC_1 denote this initial candidate parents and children set (see Figure 4.6 – a for an example).

In the second stage, we apply a wrapper to remove the false positives in PC_1. A wrapper is a feature selector that convolves with a classifier (we used Adaboost classifier [Friedman00]), with direct goal to minimize the classification error of the particular classifier [Peng05]. We keep removing a feature from PC_1 and check the cross-validated performance of Adaboost classifier. If the classification error remains the same or reduces, permanently remove the feature. Otherwise, keep the feature in PC_1. Our strategy is first to remove the features that have relatively small information gains and large mutual information with other members in PC_1. Then remove features with increasing information gains. The idea here is that features that have relatively small information gain and large mutual information with other members in PC_1 likely are the false positives. When the real members of MB are remained, removing the redundant features will not affect or may increase the classification accuracy. When most redundant features are removed, removing a member of MB will lower the classification accuracy. After this stage, we get a compact candidate parents and children PC_2 (see Figure 4.6 – b, c for an example).

In the last stage, first add all the members of PC_2 into the MB set. Then for each $X \in PC_2$, select a feature $Y \notin PC_1$ with highest $G(X,Y)$. Add Y into the MB set and check the cross-validated performance of Adaboost classifier. If the classification error reduces, keep Y in the MB. Otherwise, remove Y. The goal for this operation is to find the children's parents (like the variable F in Fig. 4.4). The children's parents may not have direct predictability on target T but combining them with the corresponding children of T can further increase the classification performance (see Figure 4.6 – d for an example).

4.3 Summary

The image segmentation algorithm described in Chapter 3 allows us to 'perceive' the salient objects in scenes. The next step is to recognize what the salient objects are. In this chapter, we present a novel method for object classification. Our main contributions are two-fold. Firstly, we build an informative object description, which consists of not only appearance, but also certain structural information like parts layout and shape of objects. All these information are combined into a high-dimensional vector. Secondly, we develop a new information-based wrapper feature selection method. Feature selection refers to search algorithms that select a subset of most characterizing features from an initial larger set of features. For many pattern recognition applications, feature selection is critic to minimize the classification error, especially when the original feature set is very large. With this feature selection method, for each object class, we can find a small subset of features that well characterize the object class. We then train a binary classifier for each object class based on the subset of predictive features selected by our feature selection method for classification.

5 Object-based place recognition and loop closing

The image segmentation algorithm described in Chapter 3 can stably detect the boundaries of many salient objects under different scenes. Therefore, like human visual system, our system can also 'perceive' the salient objects in scenes. As introduced in Chapter 1, based on the perception on the objects in a scene, human visual system can build a very economic object-based scene representation. The object-based scene representation usually consists of 4~6 salient objects. Each object is summarized with a description, like its size, overall shape, dominant colors, etc. This economic representation results in an enormous saving of processing and memory resources, which plays a key role for the success of human visual system on place recognition. Our place recognition approach takes a similar strategy. In this chapter, we will first discuss how to build an economic object-based scene representation. We mainly address two problems:

1. How to represent an object so that it can be reliably identified under different illuminations, from different distances and from different viewpoints?
2. Among a group of objects appearing in a scene, which subset of them is more valuable to be used as landmark objects to label the place?

After each place is represented by a set of distinctive and widely visible landmark objects, we then need to address how to recognize a place based on the landmark objects. To recognize a place, we need to match the set of landmark objects detected from current place to the landmark objects detected from previous places. It is often the case that thousands of landmark objects might be detected in a large environment. Thus, the landmark objects detected previously might form a large object database. Object query in the large database will be very inefficient without good indexing information. As we mentioned in the introduction section, scene-matching efficiency is the key for on-line loop detection in large environments. To increase query efficiency, we need to use some stable characteristics of these objects as indexes to quickly find the group of candidates that are similar to the query object from the database. Therefore, another goal of this chapter is to develop an efficient indexing schema for fast object retrieval from a large object database. We mainly address two problems: 1) what kinds of characteristics of landmark objects can be used as indexing information? 2) how to organize the landmark objects according to indexing information to achieve efficient scene matching?

The remainder of this chapter is organized as follows: a typical affine covariant visual feature – the Scale Invariant Feature Transform (SIFT) is introduced in section 5.1. Our SIFT

based object representation method and landmark objects selection method are described in section 5.2 and section 5.3 respectively. Then we give a detail introduction of a color model in the literature – color invariant in section 5.4. Some typical texture models in the literature and our texture descriptor are presented in section 5.5. Our range-tree object database that combines both texture and color features as indexing information is introduced in section 5.6. Finally we present the whole object-based place recognition and loop closing method in section 5.7.

5.1 Scale Invariant Feature Transform (SIFT)

The Scale Invariant Feature Transform (SIFT) [Lowe04] basically consists of four stages: (1) Scale-space extrema detection; (2) Keypoint localization; (3)Orientation assignment; (4) Generating keypoint descriptor. The first stage is to identify locations and scales that can be repeatedly assigned under differing views of the same object. To achieve this goal, a Gaussian-scale space is first built by keeping using Gaussian kernels with different scale level σ to smooth the original image. With the series of Gaussian smoothed images, the difference-of-Gaussian (DoG) images are constructed by subtracting the adjacent images in the Gaussian-scale space (see Figure 5.1-a). Each sample point is compared to its eight neighbors in the current image and nine neighbors in the scale above and below in DoG. A sample is selected only if it is larger than all of these neighbors or smaller than all of them (see Figure 5.1- b). These local maxima and minima of DoG are selected as candidate keypoints and some of them will be eliminated if found to be unstable. To achieve orientation invariant, each keypoint is assigned with an orientation. This orientation is decided by the peak of a gradient orientation histogram formed from the local area around that keypoint. Thus, a SIFT feature consists of the location of a keypoint, the scale and the assigned orientation. The last step is to compute a descriptor for the local image region. This descriptor is highly distinctive and also invariant to change in illumination or 3D viewpoint. Similar as the orientation assignment, the descriptor is also created based on a patch of pixels in the local neighborhood of the keypoint. First the gradient magnitude and orientation at each image sample point in a region around the key point location is computed and weighted by a Gaussian window. These samples are then accumulated into 4×4 array of histograms with 8 orientation bins in each. Therefore a standard SIFT descriptor is a 4×4×8=128 element feature vector (see Figure 5.1-c).

SIFT keypoints have some excellent properties: (1) They are very distinctive: the distinctiveness is achieved by using a high-dimensional vector to represent the image gradients within a local image region; (2) They are surprisingly resilient to image rotation and scale and resilient to substantial range of affine distortion and change in illumination; (3) Their computation is efficient. Extracting keypoints from a typical image can be finished in near real-time. These characteristics make SIFT have wide applications such as object recognition, motion tracking, image panorama assembly, place recognition, robot localization and SLAM.

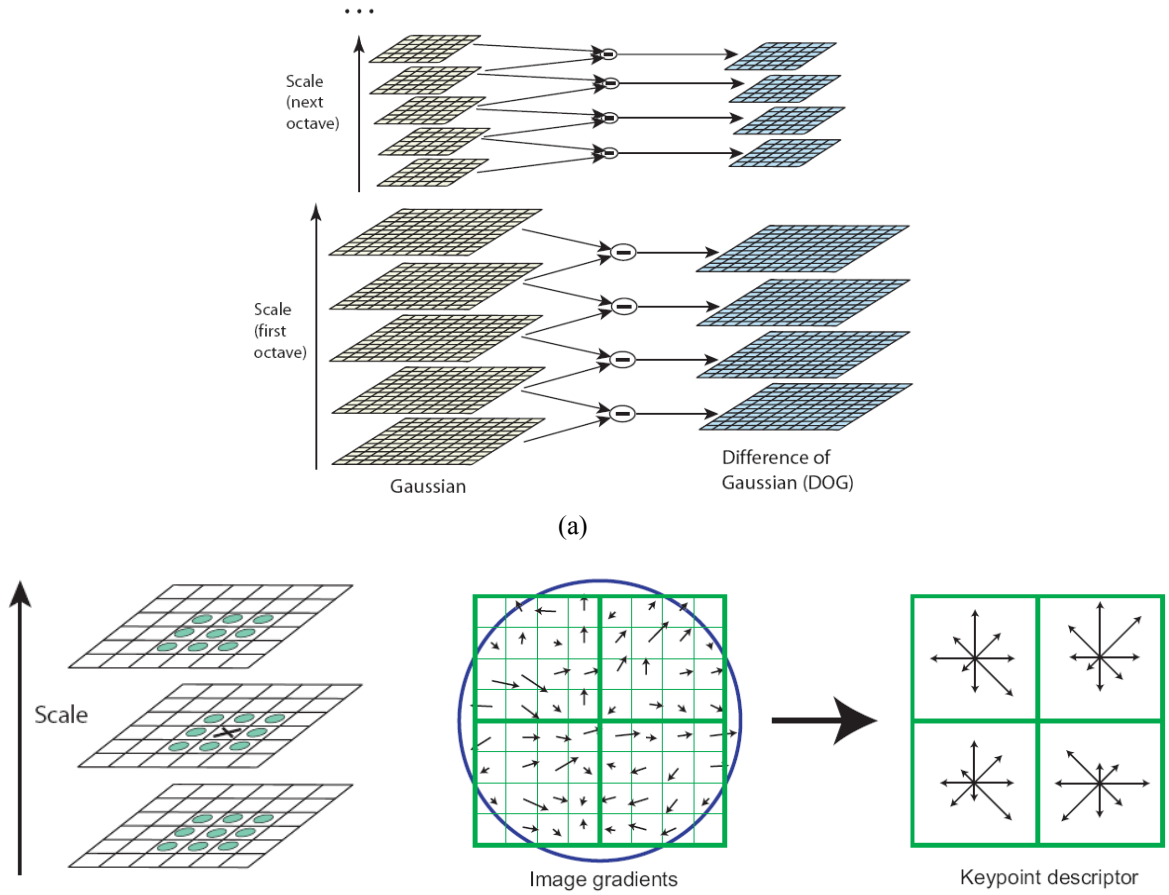


Figure 5.1 This figure illustrates SIFT keypoint detection and keypoint descriptor generation, from [Lowe04]. (a) Construction of the difference of Gaussian space (DOG). (b) Local extrema detection in the difference of Gaussian space. (c) Generation of keypoint descriptor.

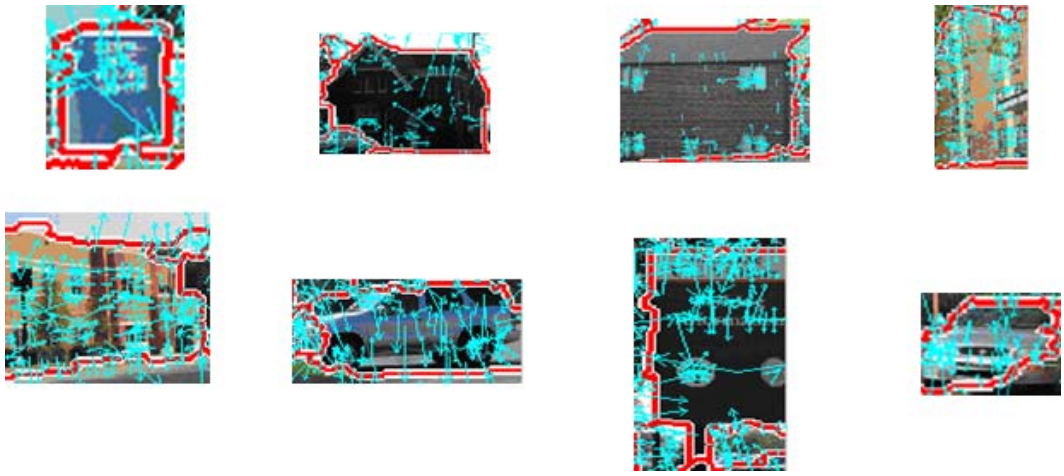


Figure 5.2 Examples of object representation. For each object, the green arrows enclosed in the red boundary of the object are SIFT keypoints. Each SIFT keypoint is assigned with a high dimensional vector as descriptor, which records the image gradients within a local image region around the keypoint. Each object is then represented with a list of SIFT descriptors of the SIFT keypoints.

5.2 Local feature based object representation

As introduced in Chapter 5.1, SIFT has many attractive properties that can fulfill the requirements of our application. Thus, in our work, we choose to represent an object based on SIFT. In most cases, we can extract a set of SIFT keypoints from object surfaces. For each SIFT keypoint, we can build a high dimensional vector as descriptor. The SIFT descriptor records the image gradients within a local image region around the keypoint. An object is then represented by a list of such SIFT descriptors (see Figure 5.2 for examples).

To evaluate the goodness of object representations, Marr [Marr82] proposed the following five criteria:

1. Accessibility - needed information should be directly available from the representation rather than derivable through heavy computation;
2. Scope - object representation method should be able to represent a wide range of objects;
3. Uniqueness - an object should have a unique representation;
4. Stability - small variations in an object should not cause large variations in the representation;
5. Sensitivity - detailed features should be represented as needed.

Our object representation generally satisfies these criteria except for *sensitivity*, which is not a big concern in our application. Our goal is to reliably recognize a place by matching the set of objects appearing in the place to the recorded objects. This requires object representation to be unique so that any object is distinguishable from others. Object

representation also needs to be stable because an object may be observed from different viewpoints. In addition, object representation needs to be capable of representing various objects appearing in natural scenes. Therefore, for our application, the most important requirements for object representation are uniqueness, stability and scope. The *uniqueness* is achieved by the highly distinctive SIFT descriptors. The distinctiveness of SIFT descriptors have been verified by many works including [Lowe04, Mikolajczyk05, Newman05, Se02, Se05]. The stability is also fulfilled. SIFT keypoints are invariant, to certain degree, to scale changes, rotation changes and illumination changes. This means that from different viewpoints, we can extract a same set of SIFT keypoints from an object and hence create a same set of SIFT descriptors. The *scope* is satisfied because we can extract a set of SIFT keypoints from most object surfaces. The number of SIFT keypoints is directly related to the degree of surface texture of objects. Most object surfaces have certain degree of textures. Therefore, this object representation can represent most objects appearing in natural scenes.

5.3 Landmark object detection and scene representation

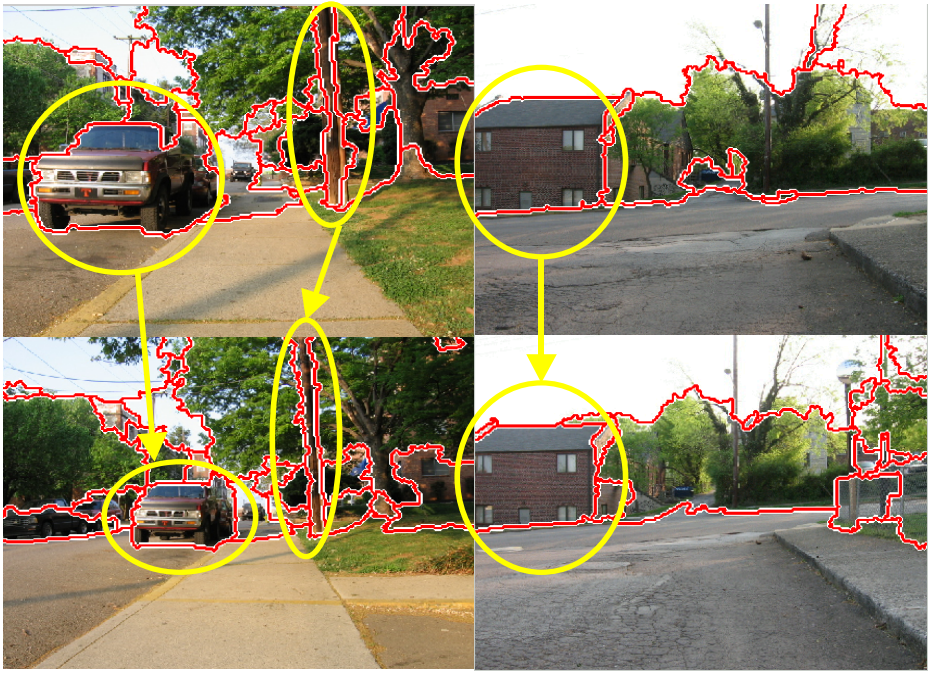
After creating representations for the objects in a scene, the next step is to represent the scene with a subset of objects. As mentioned in Chapter 1, human visual system only keeps 4~6 "impressive" objects in long-term memory. These "impressive" objects are often good enough for place recognition. This economic scene representation results in an enormous saving of processing and memory resources, which plays a key role for the success of human visual system on place recognition. Therefore, we want to take a similar strategy. The question then is: among a group of objects appearing in a scene, which subset of them is more valuable as landmarks to label the scene? According to [Sala06], we define two criteria to select such landmark objects:

1. The objects should be distinguishable; and
2. The objects should be widely visible.

The first criterion is common sense: as landmarks, these landmark objects should be distinctive enough to uniquely label a place. The reason for the second criterion is because when a vehicle revisits a place, most likely it will see the scene from a different viewpoint. Therefore, objects visible only from a single location are not suitable as landmarks since they will cause place recognition not persistent.

The first criterion is well addressed by our object representation. We use a set of SIFT descriptors to represent an object. Since SIFT descriptors are highly distinctive, as long as an object contains certain number (e.g., >= 8 in our case) of SIFT features (keypoints), the object is distinguishable enough.

To check if an object is widely visible, we compare the set of objects detected in current position with those objects detected in several previous positions. The comparison between two objects is based on matching the set of SIFT descriptors contained in one object to the set of SIFT descriptors contained in another object. If at least six pairs of SIFT descriptors are matched, then the two objects are considered to be the same object in the scene. If an object can be stably observed from multiple positions (e.g., >= 2 in our case), the object is



(a) (b)

Figure 5.3 Examples of landmark objects detection. For each column, upper is the image taken at the current position, bottom is the image taken in previous position. Objects that can be observed from multiple positions are selected as landmark objects (marked with yellow circles).

selected as a landmark object. Examples of landmark objects detection are shown in Figure 4.3. By doing so, each detected landmark object is highly distinguishable and widely visible. A set of these kinds of landmark objects can be expected to be able to uniquely label a place. In our implementation, each place is represented by a set of landmark objects.

5.4 Color feature

Although color is well known as a powerful cue in the distinction and recognition of objects, it also has some weaknesses. For example, one drawback of using color histogram model for object recognition is when the illumination circumstances are not equal, the object recognition accuracy degrade significantly. Besides, the changes in the surface orientations of objects also affect color perception. Therefore, Gevers *et al*. [Gevers99] propose some criteria for good color models, such as being robust against changes in viewing direction, changes in illumination, changes in surface orientation of the object (i.e. the geometry of the object), etc. These criteria well match the requirement of our application. Geusebroek *et al*. [Geusebroek01] investigated invariant properties of colors and discriminative power under different illumination conditions. We will give a brief introduction of their work.

5.4.1 Color representation

Geusebroek *et al*. claim that colors are only defined in terms of human observation [Geusebroek00]. A perceived color does not directly correspond to the spectral content of the stimulus. There is no one-to-one mapping of spectral content to perceived color. Instead, perceived color is related to both the spectral energy distribution and the spatial configuration of colors. Therefore, they propose to use a 3D spectral-spatial space to represent color.

The spectral structure of color can be formulized as the follows [Geusebroek00]:

Let $D(\lambda)$ be the energy distribution of the incident light, where λ denotes wavelength. Let $G(\lambda_0;\sigma_\lambda)$ be the Gaussian at spectral scale σ_λ positioned at λ_0. The spectral energy distribution is approximated by a Taylor expansion at λ_0:

$$D(\lambda) = D^{\lambda_0} + \lambda D_\lambda^{\lambda_0} + \frac{1}{2}\lambda^2 D_{\lambda\lambda}^{\lambda_0} + \ldots \tag{5.1}$$

The observed spectral energy in the Gaussian color model, at infinitely small spatial resolution and spectral scale is in second order to

$$\hat{D}^{\sigma_\lambda}(\lambda) = \hat{D}^{\lambda_0,\sigma_\lambda} + \lambda \hat{D}_\lambda^{\lambda_0,\sigma_\lambda} + \frac{1}{2}\lambda^2 \hat{D}_{\lambda\lambda}^{\lambda_0,\sigma_\lambda} + \ldots \tag{5.2}$$

where

$$\hat{D}^{\lambda_0,\sigma_\lambda} = \int D(\lambda) G(\lambda;\lambda_0,\sigma_\lambda) d\lambda \tag{5.3}$$

measures the spectral intensity,

$$\hat{D}_\lambda^{\lambda_0,\sigma_\lambda} = \int D(\lambda) G_\lambda(\lambda;\lambda_0,\sigma_\lambda) d\lambda \tag{5.4}$$

measures the first order spectral derivative, and

$$\hat{D}_{\lambda\lambda}^{\lambda_0,\sigma_\lambda} = \int D(\lambda) G_{\lambda\lambda}(\lambda;\lambda_0,\sigma_\lambda) d\lambda \tag{5.5}$$

measures the second order spectral derivative. G_λ and $G_{\lambda\lambda}$ denote derivatives of the Gaussian with respect to λ. The coefficients of equation 5.1 are also called Gaussian color model. The first three components D, D_λ and $D_{\lambda\lambda}$ of Gaussian color model well approximate the CIE 1964 XYZ basis when taking $\lambda_0 = 520$ nm and $\sigma_\lambda = 55$ nm. Conversion from RGB color model to Gaussian color model is achieved by a linear transformation [Geusebroek00]:

$$\begin{bmatrix} \hat{D} \\ \hat{D}_\lambda \\ \hat{D}_{\lambda\lambda} \end{bmatrix} = \begin{pmatrix} 0.06 & 0.63 & 0.27 \\ 0.3 & 0.04 & -0.35 \\ 0.34 & -0.6 & 0.17 \end{pmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{5.6}$$

The spatial structure of color can be formulized as the follows [Geusebroek00]:

Introduction of spatial extent in the Gaussian color model yields a local Taylor expansion at wavelength λ_0 and position x_0. Now the measurement of a spatio-spectral energy distribution has a spatial as well as a spectral resolution. The measurement is obtained by probing an energy density volume in a three-dimensional spatio-spectral space. The size of the probe is determined by the observation scale σ_λ and σ_x.

$$\hat{D}(\lambda, x) = \hat{D} + \begin{pmatrix} x \\ \lambda \end{pmatrix}^T \begin{bmatrix} \hat{D}_x \\ \hat{D}_\lambda \end{bmatrix} + \frac{1}{2}\begin{pmatrix} x \\ \lambda \end{pmatrix}^T \begin{bmatrix} \hat{D}_{xx} & \hat{D}_{x\lambda} \\ \hat{D}_{\lambda x} & \hat{D}_{\lambda\lambda} \end{bmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} + \ldots \tag{5.7}$$

Where

$$\hat{D}_{x^i\lambda^j}(\lambda,x) = D(\lambda,x) * G_{x^i\lambda^j}(\lambda,x;\sigma_\lambda,\sigma_x) \tag{5.8}$$

Where $G_{x^i\lambda^j}(\lambda,x;\sigma_\lambda,\sigma_x)$ are the spatio-spectral probes.

5.4.2 Color invariance

The spectral components of an equal energy illumination source (daylight) are constant over the wavelength. Under assumption of an equal energy illumination, the energy distribution of the incident light is defined as follows [Geusebroek01]:

$$D(\lambda,x) = h(x)\left\{p_f(x) + (1-p_f(x))^2 W_\infty(\lambda,x)\right\} \tag{5.9}$$

Where $h(x)$ denotes intensity variations, $p_f(x)$ denotes the Fresnel reflectance at x, $W_\infty(\lambda,x)$ denotes the material reflectivity (see figure 5.1 – a for an example). Differentiating $D(\lambda,x)$ with respect to λ get:

$$D_\lambda(\lambda,x) = h(x)(1-p_f(x))^2 \frac{\partial W_\infty(\lambda,x)}{\partial\lambda} \tag{5.10}$$

Differentiating $D_\lambda(\lambda, x)$ with respect to λ get:

$$D_{\lambda\lambda}(\lambda,x) = h(x)(1-p_f(x))^2 \frac{\partial^2 W_\infty(\lambda,x)}{\partial\lambda^2} \tag{5.11}$$

Then the ratio of D_λ and $D_{\lambda\lambda}$ equates:

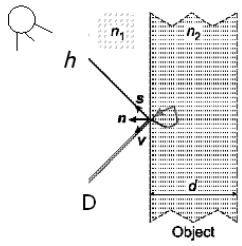
$$H = \frac{D_\lambda(\lambda,x)}{D_{\lambda\lambda}(\lambda,x)} = \frac{\dfrac{\partial W_\infty(\lambda,x)}{\partial\lambda}}{\dfrac{\partial^2 W_\infty(\lambda,x)}{\partial\lambda^2}} \tag{5.12}$$

The H only depends on the derivatives of the material reflectivity of objects. Thus, H defines a color invariance. According to Geusebroek [Geusebroek01], H is invariant to viewing direction, surface orientation, highlights, illumination direction and illumination intensity (see figure 5.4 –b and c for an example). These properties make color invariance H suited to be used as an indexing feature for our application. The color invariance H can be approximated with the approximation of the second and third components of Gaussian color model in equation 5.6:

$$\hat{H} = \frac{\hat{D}_\lambda}{\hat{D}_{\lambda\lambda}} \tag{5.13}$$

5.5 Texture feature

Approaches in the analysis of image based on textural contents use either the spatial or frequency-based techniques. A typical spatial-based technique is called edge histogram descriptor [Manjunath01]. The edge histogram descriptor captures the spatial distribution of edges. The distribution of edges is a good texture signature. The computation of this descriptor is as follows: A given image is first sub-divided into 16 sub-images. Each of the 16 sub-images is further subdivided into image blocks. The size of these image blocks scale



(a)

(b) (c)

Figure 5.4 (a) Illustration of the energy distribution of the incident light. (b) Input image. (c) Color invariant *H*.

with the image size and is assumed to be a power of 2. The number of image blocks per sub-image is kept constant, independent of the original image dimensions, by scaling their size appropriately. A simple edge detector is then applied to each of the macro-block, treating the macro-block as a pixel image. The pixel intensities for the partitions of the image block are computed by averaging the intensity values of the corresponding pixels. The edge-detector operators include four directional selective detectors and one isotropic operator. Thus, edges are broadly grouped into five categories: vertical, horizontal, 45 diagonal, 135 diagonal, and isotropic (nonorientation specific). Those image blocks whose edge strengths exceed a certain minimum threshold are used in computing the histogram. For each sub-image, a local edge histograms is computed. Each local histogram has five bins corresponding to the above five edge categories. Thus, the edge histogram for the whole image consists of 80 bins. These bins are nonuniformly quantized using 3 bits/bin, resulting in a descriptor of size 240 bits. Edge histogram descriptor is useful for image to image matching even when the underlying texture is not homogeneous.

A typical frequency-based technique is called Homogeneous Texture Descriptor (HTD). This descriptor is computed by first filtering the image with a bank of orientation and scale sensitive filters, and then computing the mean and standard deviation of the filtered outputs in the frequency domain. The computation of this descriptor is as follows [Manjunath01]. The frequency space is partitioned into 30 channels with equal divisions in the angular direction (at 30 intervals) and octave division in the radial direction (five octaves). The center frequencies of the feature channels are spaced equally in 30 in angular direction. In the radial

direction, the center frequencies of the neighboring feature channels are spaced one octave apart. The individual feature channels are modeled using 2-D Gabor functions. Gabor functions are modulated Gaussian functions. For the bank of filters used, the filter parameters are selected such that the half-maximum contours of the 2-D Gaussians of adjacent filters in the radial and angular directions touch each other. The image texture energy and the deviation of the energy in each of the filtered channels are then computed. Both the energy and the energy deviation are then logarithmically scaled to obtain two numbers, e_j and d_j for jth feature channel. The HTD is formed by a feature vector with 32 elements [Manjunath01]:

$$TD = [f_{DC,} f_{SD,} e_1, e_2, ..., e_{30}, d_1, d_2, ..., d_{30}] \quad (5.14)$$

The first two elements of the feature vector are the mean intensity and the standard deviation of the image texture, respectively. The HTD provides a quantitative characterization of texture for similarity-based image-to-image matching.

Some texture models can characterize perceptual attributes such as directionality, regularity, and coarseness of a texture. An example of this kind of texture model is referred to as the "texture browsing descriptor" [Manjunath01]. This is a compact descriptor that requires only maximum 12 bits to characterize a texture's regularity (2 bits), directionality (3bits×2), and coarseness (2bits×2). The regularity of a texture is graded on a scale of 0 to 3, with 0 indicating an irregular or random texture. A value of 3 indicates a periodic pattern with well-defined directionality and coarseness values. The directionality of a texture is quantized to six values, ranging from 0 to 150 in steps of 30. A coarseness component is associated with each dominant direction. Coarseness is related to image scale or resolution. It is quantized to four levels, with 0 indicating a fine grain texture and a value of 3 indicating a coarse texture. To compute the texture browsing descriptor, the image is first filtered using a bank of scale and orientation selective band-pass filters. This part is very similar to the one for HTD introduced above. The filtered outputs are then used to compute the texture browsing descriptor components. This texture descriptor is valuable because it is related to human perception of texture, which allows users to manually specify a descriptor for query.

Basically, the texture feature extraction processes of both spatial and frequency-based techniques involve some operations with certain computational cost. This will increase the computational complexity. In our application, computational complexity is one of the main factors in choosing an indexing feature. In addition, like shape features, most of the texture features are used in image datasets where images are taken under control conditions. In these images, the resolution is high enough to allow detailed texture properties to be accurately extracted. Some of texture properties such as coarseness are sensitive to scale change. Therefore, these texture features are not viewpoint-invariant, which will cause low indexing performance in our application. For these reasons, the texture features in the literature are not well suited for our application.

A texture-related characteristic of landmark objects is the degree of surface texture. The degree of surface texture can be measured by the number of SIFT features extracted from the surface. Usually highly textured surfaces generate large number of SIFT features and nearly uniform surfaces generate little SIFT features. Using the number of SIFT extracted from an object surface as texture descriptor has some merits. First, this texture descriptor satisfies the needs of viewpoint-invariant, which is critical to our application. According to [Lowe04], the stability of detection for SIFT features is around 70%~90% under different degrees of affine distortion. Thus, the number of SIFT features extracted from an object surface is stable under

viewpoint changes. Besides, unlike the texture descriptors in the literature, this texture descriptor is low-dimensional (only one dimension). Thus, the similarity measurement of this texture descriptor is very simple and efficient. Although the discriminative power of this texture descriptor is not very strong, it is sufficient for our application. Notice that in our case, the retrieval mechanism does not need to solve the object recognition task. We use a set of highly distinctive SIFT descriptors to represent an object. The set of high dimensional SIFT descriptors extracted from an object surface records enough detail information of the object surface. Thus, in our application, our strategy is to use indexing features to find a set of candidates with similar perceptual properties and then use highly distinctive SIFT descriptors to get a precise similarity match list among the candidate objects. Thus, using the number of SIFT as texture descriptor can basically satisfy our needs.

5.6 Range tree database: combination of color and texture features

For each landmark object, we can extract a color feature (color invariance H) and a texture feature (the number of SIFT features) from the object surface. Both of them satisfy the viewpoint-invariant requirement and can be efficiently extracted. Now we can design an efficient indexing structure based on these two features. In current effort, we have implemented a system that indexes and retrieves similar objects based on a cascaded texture-color indexing method. Object retrieval is initially indexed by texture and then indexed by color.

According to [Lowe04], the stability of detection for SIFT features is around 70%~90% under different degrees of affine distortion. Thus, the number of SIFT features extracted from an object surface is stable in a certain range. For a query object containing 30 SIFT features, most likely the matching object will be among those objects that contain 20~40 SIFT features. For this reason, we design a range-tree [Samet06] structure database to help us quickly find a range of candidates with similar degree of surface texture in a large database.

A range tree is a balanced binary search tree where the data are sorted in the leaf nodes and these leaf nodes are linked in sorted order by use of a doubly linked list. The non-leaf nodes contain midrange values of their left and right subtrees. The midrange value contained in a non-leaf node enables discriminating between the left and right subtrees of the non-leaf node. A range search for $[B:Z]$ can be performed by searching the tree and finding the node with either the largest value $\leq B$ or the smallest value $\geq B$, and then following the links until reaching a leaf node with a value greater than Z. For T points, this process takes $O(\log_2 T + W)$ time. W is the number of objects found. An example of a range tree search is shown in Figure 5.5. In this figure, the query object (a car) contains 30 SIFT features. We can quickly find a group of objects that contains SIFT features range from 20~40 in the object database through range tree.

After retrieving a group of candidate objects with similar degree of surface texture to the query object, we further select a subset of the candidate objects that have similar color to the query object.

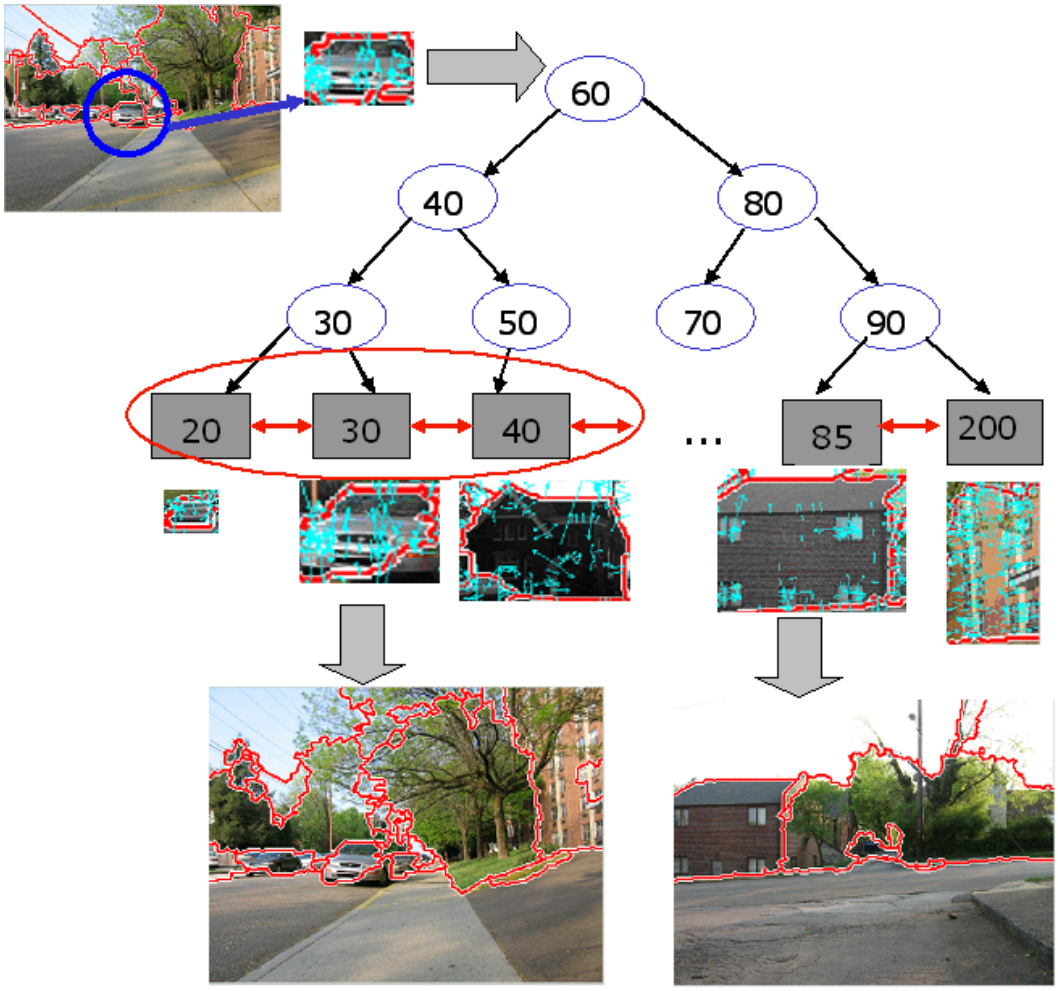


Figure 5.5 Illustration of a range tree search. The green arrows in the objects are SIFT features. The query object contains 30 SIFT features. Those objects that contain 20~40 SIFT features in the database can be quickly found.

For each object, we create a histogram for the color invariance H. The histogram entries with the highest Peak in the histogram are recorded as the object's dominant color. Any entry in the histogram with local peak that is within 80% of the highest peak is also recorded for the object. If an object has single peak in the histogram, then the object may have an approximately homogenous surface. Otherwise, the object may consist of multiple parts with totally different colors. If the difference of dominant color invariance H values of two objects is less than 0.2, then the two objects are considered to have similar colors. In addition, if the query object has multiple color invariance H values, then selected objects should also have multiple color invariance H values.

By combining the surface texture and color information of objects, we are able to compare a query object only to the objects that have similar surface characteristics to the query object. This avoids many unnecessary comparisons and therefore, significantly increases the query efficiency.

The final stage of the retrieval process is to determinate if the query object can be matched to the candidate objects with similar perceptual properties (texture and color) to the query object. This is achieved as follows: For each candidate object, we further compare the set of SIFT descriptors extracted from the candidate object surface to those SIFT descriptors extracted from the query object surface. If more than 6 of SIFT descriptors of the query object get matched to the SIFT descriptors of a candidate object, then the query object is matched to the candidate object, which means that the query object has been seen before and the vehicle may be revisiting a place.

5.7 Place recognition and loop closing algorithm

The complete place recognition and loop closing algorithm is summarized as follows:

INPUT: image I_c taken from current scene c with current time t_c

OUTPUT: place recognition (loop detection) result

1. Detect SIFT features from I_c;
2. Run the image segmentation algorithm presented in Chapter 3.7 to segment I_c;
3. Detect a set of objects O_c. Each object $o_k \in O_c$ contains more than 8 SIFT features;
4. Let O_p contains the objects detected from three immediate previous positions: $O_p = O_{c-1} \cup O_{c-2} \cup O_{c-3}$; Let landmark objects $LO_c = \phi$; Compare O_c with O_p, if $(o_l \in O_c) \wedge (o_l \in O_p)$, let $LO_c = LO_c \cup o_l$; let $O_p = O_p \setminus O_{c-3}$ and $O_p = O_p \cup O_c$; If $LO_c = \phi$, move to the next position. Otherwise, represent scene c with a set of landmark objects $LO_c \subseteq O_c$;
5. Set $M = \phi$; For each landmark object $o_l \in LO_c$, search in the range tree. If a match o_m is found, output the scene number set s (records the these scenes that o_m appears) of o_m and let $M = M \cup s$; Otherwise, insert o_l into the range tree with scene number set c;
6. For each scene number $m \in M$, compare LO_c and LO_m, if $|LO_c|$ is close to $|LO_m|$ and around 50% of LO_c get matched with LO_m and there is a big difference between t_c and t_m, then detect a loop. (t_m is the time when m was visited); if $M = \phi$ or scene c is not

matched with any scene in M ,output that c is a new scene.

5.8 Summary

In this chapter, we address how to recognize a place based on the landmark objects.

The image segmentation algorithm described in Chapter 3 allows us to 'perceive' the salient objects in scenes. As introduced in Chapter 1, based on the perception on the objects in a scene, human visual system can build a very economic object-based scene representation. The object-based scene representation usually consists of 4~6 salient objects. Each object is summarized with a description, like its size, overall shape, dominant colors, etc. This economic representation results in an enormous saving of processing and memory resources, which plays a key role for the success of human visual system on place recognition.

Our place recognition approach takes a similar strategy. In this chapter, we mainly tackle how to build an economic object-based scene representation. We first represent each object with a list of SIFT descriptors. The SIFT based object representation make the object can be reliably identified under different illuminations, from different distances and from different viewpoints. We then select a subset of objects that are widely visible among the detected objects from each place as landmark objects. A set of these kinds of landmark objects can be expected to be able to uniquely label a place.

To recognize a place, we need to match the set of landmark objects detected from current place to the landmark objects detected from previous places. Object query in the large database will be very inefficient without good indexing information. To increase query efficiency, we need to use some stable characteristics of these objects as indexes to quickly find the group of candidates that are similar to the query object from the database. We first figure out that for place recognition, colors and degree of textures of object appearances are two good characteristics to be used as indexing information. Then we design a range-tree database to organize the detected objects. By combining the surface texture and color information of objects, we are able to compare a query object only to the objects that have similar surface characteristics to the query object. This avoids many unnecessary comparisons and therefore, significantly increases the query efficiency. Finally, we give the whole object-based place recognition algorithm that is based on all the techniques we develop in chapter 3, 4 and 5.

6 Experimental results

In this chapter we present the experimental results for our image segmentation method, object classification method and place recognition method. Section 6.1 tests our image segmentation method on three challenging scene image datasets and compare the results with several state-of-art methods of reference. In section 6.2, we evaluate our object classification method on the MSCR-21 dataset. The place recognition experimental results are illustrated in section 6.3.

6.1 Image segmentation

6.1.1 Evaluation measures

Quantifying the performance of a segmentation algorithm remains a challenging task [Unnikrishnan05]. Segmentation errors can be classified roughly into three categories: *over-segment* – an object is segmented into multiple regions; *under-segment* – multiple objects are segmented together; and *arbitrary-segment* – different parts of a group of objects are segmented together. It is widely agreed that humans differ in the level of details at which they perceive images [Unnikrishnan05] and over-segment in certain degree may often correspond to this kind of difference in granularity. This is also called selective refinement [Martin02]. In many cases over-segment in certain degree is still highly valuable for the subsequent processing in many applications. Under-segment basically means that a segmentation algorithm contributes little in the image region. Arbitrary-segment is the worst case. It means that the content of an image is totally distorted in the region. Therefore, a good segmentation performance measure should be tolerant to the refinement to certain degrees and meanwhile be sensitive to the under-segment and arbitrary-segment errors.

Many performance measures have been proposed in the literature [Martin02, Estrada05, Unnikrishnan05, Zhang06] for the segmentation evaluation task. Under human-aided segmentation evaluation framework, boundary matching measure and region differencing measure are widely used. The former is based in the comparison of machine-detected boundaries with respect to human-marked boundary and the latter operates on computing the degree of overlap between machine-generated segmentations and human-generated reference segmentation. If segment membership information is discarded, then a segmentation can be regarded as a boundary map. The boundary matching measure, when applying on the

boundary map, is tolerant to the refinement to certain degree. However, it is also insensitive to under-segment. Although the region difference measure can appropriately punish the under-segment, it is not tolerant to the refinement. Since it is not an easy task to determine which performance measure is superior, Martin [Martin02] recommended that people should use multiple performance measures to validate their results. Therefore, we have used both boundary matching measure and region differencing measure.

For the boundary matching measure, we use the Precision-Recall framework developed by Martin [Martin02]. A precision-recall curve is a parameterized curve that captures the trade-off between accuracy and noise as a segmentation algorithm's parameters varies. *Precision* is the fraction of detections that are true boundaries, while *recall* is the fraction of true boundaries that are detected. Thus, precision is the probability that the segmentation algorithm's signal is valid, and recall is the probability that the ground truth data is detected. These two quantities can be combined in a single quality measure, *F-measure*, defined as the weighted harmonic mean of *Precision* and *Recall*:

$$F = \frac{1}{\pi K^{-1} + (1-\pi)U^{-1}} \tag{6.1}$$

where K and U represent Precision and Recall, respectively. π is a weight that is usually set to 0.5. The reason of choosing the same weights for Precision and Recall is that both of them are equally important to generate good segmentations. The F-measure is valued between 0 and 1, where larger values are more desirable. The location of the maximum F-measure along a precision-recall curve provides the optimal parameters for a segmentation algorithm [Estrada05]. In our case, each tested algorithm uses the default parameters released by the authors. The only thing we changed in Ncuts and MBD is setting the parameter about the number of desired regions based on the ground truth data for each image, which we believe should yields better results than a fixed number for these two algorithms. Thus, the precision-recall curve for each algorithm degenerates into a single point.

For the region difference measure, we choose the method proposed by Russell [Russell06]. Unlike the region difference measure in [Martin02], this measure does not have the problem of favoring under-segments or over-segments. A segmentation accuracy score is defined as:

$$\kappa = \frac{GT \cap RM}{GT \cup RM} \tag{6.2}$$

where GT and RM represent the set of pixels in the ground truth segment of an object and the machine-generated object segment, respectively. If more than one ground truth segmentation intersects RM, then we use the one that results in the highest score. The score is then averaged over all the detected object segments. The measurement is often called object-level accuracy. Another measurement that is often used for evaluating classification segmentations is pixel-level accuracy. Pixel-level accuracy is also based on equation 6.2. The difference between object-level accuracy and pixel-level accuracy is that for the ground truth class segmentation, multiple objects belonging to the same class are assigned with the same label. As a result, if a method merges some physically apart same class objects together, it still achieves high pixel-level accuracy. Therefore, methods achieving high pixel-level

accuracy do not mean that they can also achieve high object-level accuracy (See Figure 6.3 for an example).

6.1.2 LabelMe database

6.1.2.1 Experimental setup

We first test the image segmentation algorithm described in Section 3.8 on the LabelMe [Russell08] scene image database. We selected 100 typical natural scene images from the LabelMe database as test images. The test images cover various environments like urban areas, suburban areas, residence areas, roads, airports, etc. Most salient objects in these images have already been labeled by different peoples. Therefore, the labels can be used as ground truth. The labeled objects cover a variety of object classes like sky, roads, buildings, vehicles, vegetations, people, signs, etc. This provides more flexibility to us: we can not only evaluate the overall performance of a segmentation algorithm but also measure the segmentation quality of the algorithm on the individual object class.

We benchmarked three methods of reference: the recent version of *Ncuts* [Cour05], Martin's [Martin04] boundary detector and Hoiem's [Hoiem07] boundary detector. We compared their segmentation results with ours. The four tested methods represent four different strategies of boundaries detection: The *Ncut* algorithm detects boundaries based on appearance homogeneity; Martin's boundary detector (MBD) predicts the probability of boundary at each pixel using local brightness, texture, and color gradient cues; Hoiem's boundary detector (HBD) identifies and labels boundaries using the traditional edge and region cues together with 3D surface and depth cues; and our method detects object boundaries based on the Perceptual Organization Model (POM) presented in this paper.

In all the experiments, we used the implementation provided by the authors. Both *Ncuts* and MBD require users to set a parameter about the number of regions that an input image needs to be segmented into. To make a fair comparison, these parameters are set based on the ground truth segmentation for each image so that these two methods can yield approximately the best segmentation results. Specifically, in our implementation, we set parameters $\sigma=0.5$, $k=300$ and $min=20$ for Felzenszwalb's [Felzenszwalb04] algorithm to generate the initial partition for an input image and we set parameters $\theta = [18, 3.5]$, $\alpha = 20$ and $\beta = 3$ for the Perceptual Organization model used in our image segmentation algorithm.

6.1.2.2 Experimental result

Figure 6.2 shows the precision-recall curve and the distribution of recall and precision for the four tested algorithms on the 100 testing images from LabelMe dataset. In the precision-recall curve graph (Figure 6.2 – Left), it can be observed that the F-measure of our method (POM) dominates significantly over the others. The distribution of recall (Figure 6.2 – middle) shows that for most images, our POM can detect more than 60% true boundaries. The distribution of precision (Figure 6.2 - right) shows that majority of detected boundaries by our POM in most images are true boundaries. Notice that the HBD achieves higher precision. However, HBD obtains higher precision in the cost of lower recall rate. This

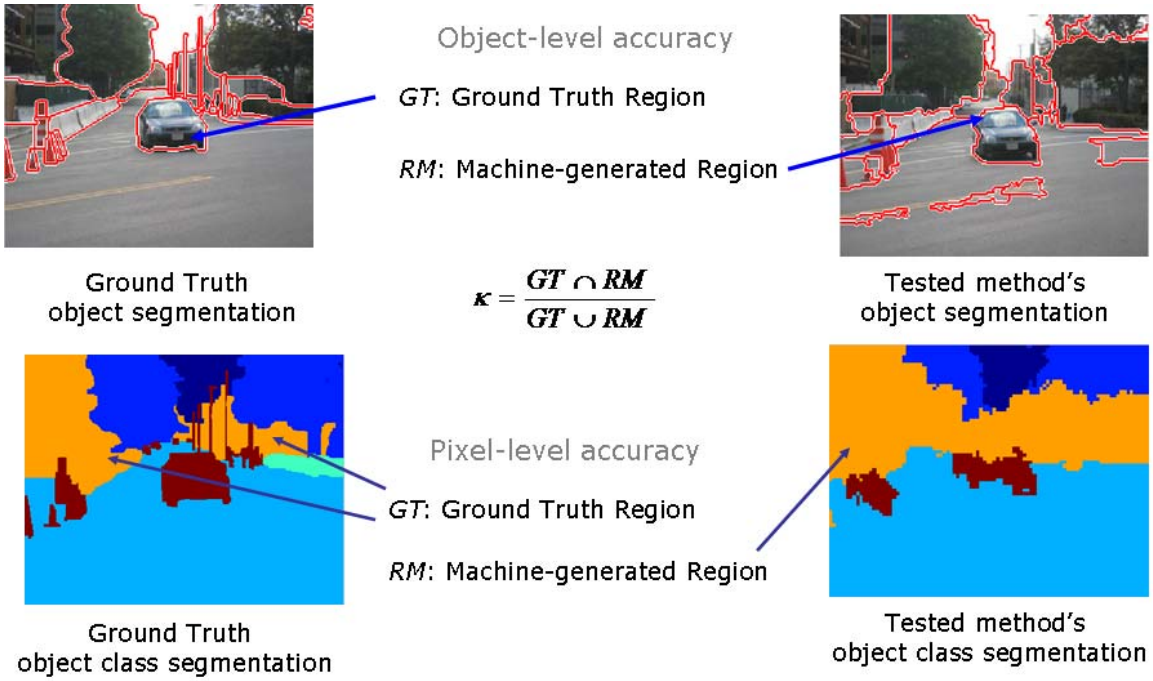


Figure 6.1 Region-based segmentation measurements. The first row: object-level accuracy. The second row: pixel-level accuracy.

Table 6.1 Segmentation accuracy score on LabelMe dataset

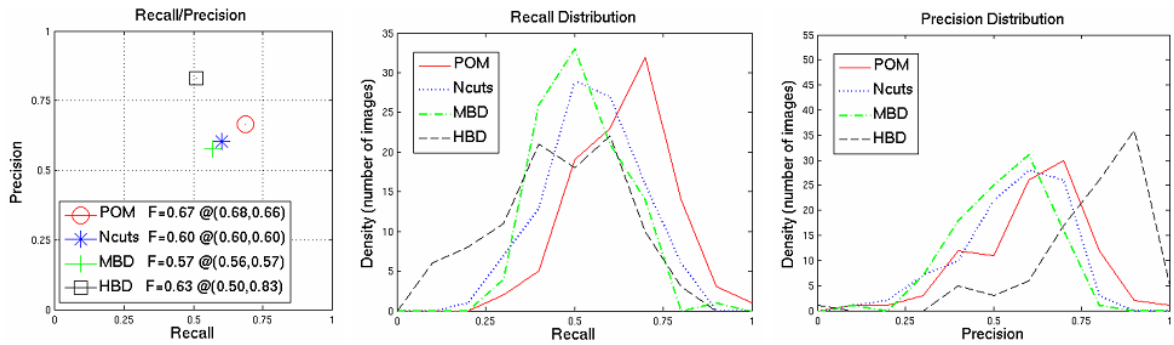
	Ncuts	**MBD**	**HBD**	**POM**
Overall (%)	40.2	34.6	36.1	53.2
Vehicle (%)	25.3	25.2	14.2	49.5
Building (%)	40.7	38.5	33.8	47.6
Vegetation (%)	38.4	34.0	33.6	43.4

means that although the majority of detected boundaries by HBD are true boundaries, lots of true boundaries are not detected in the images. Thus, HBD causes some under-segment problems in many images.

The results of averaged segmentation accuracy score are summarized in Table 6.1. The distribution of the segmentation accuracy of the overall objects, vehicles, buildings and vegetation are presented in Figure 6.3. We applied the Student's T-test to the accuracy results for the POM compared against other three methods. This test confirms that the differences in these results are statistically significant, with a confidence level of 99.5% (except for the confidence level of the difference between POM and Ncuts on vegetation is 95%). Our method scores better than the other methods on the overall objects and the three object classes. Especially for the vehicles, our method achieves significantly better results than the other methods. We have closely inspected the results of our method and found that most of the vehicles that our method gets low segmentation accuracy score on are those occluded vehicles or small-size distant vehicles in the background (see the middle image in Figure 6.4 for examples). As shown in Figure 6.4 and Figure 6.5, segmentation errors on these vehicles actually do not affect the visual quality of segmentation too much. For the salient front-stage vehicles, our method can group the whole vehicles together in most cases. As for the buildings, most of the low-score buildings are caused by the over-segment error. Unlike the vehicles, some buildings have complicated structures. Some parts of a building are not very tightly attached to the main part of that building. As results, our method often over-segments a big building into two or three connected components. Since the region difference measure we used here is not tolerant to refinement, these buildings achieve very low score (around 30% ~ 50%), which affects the performance of our method on the buildings.

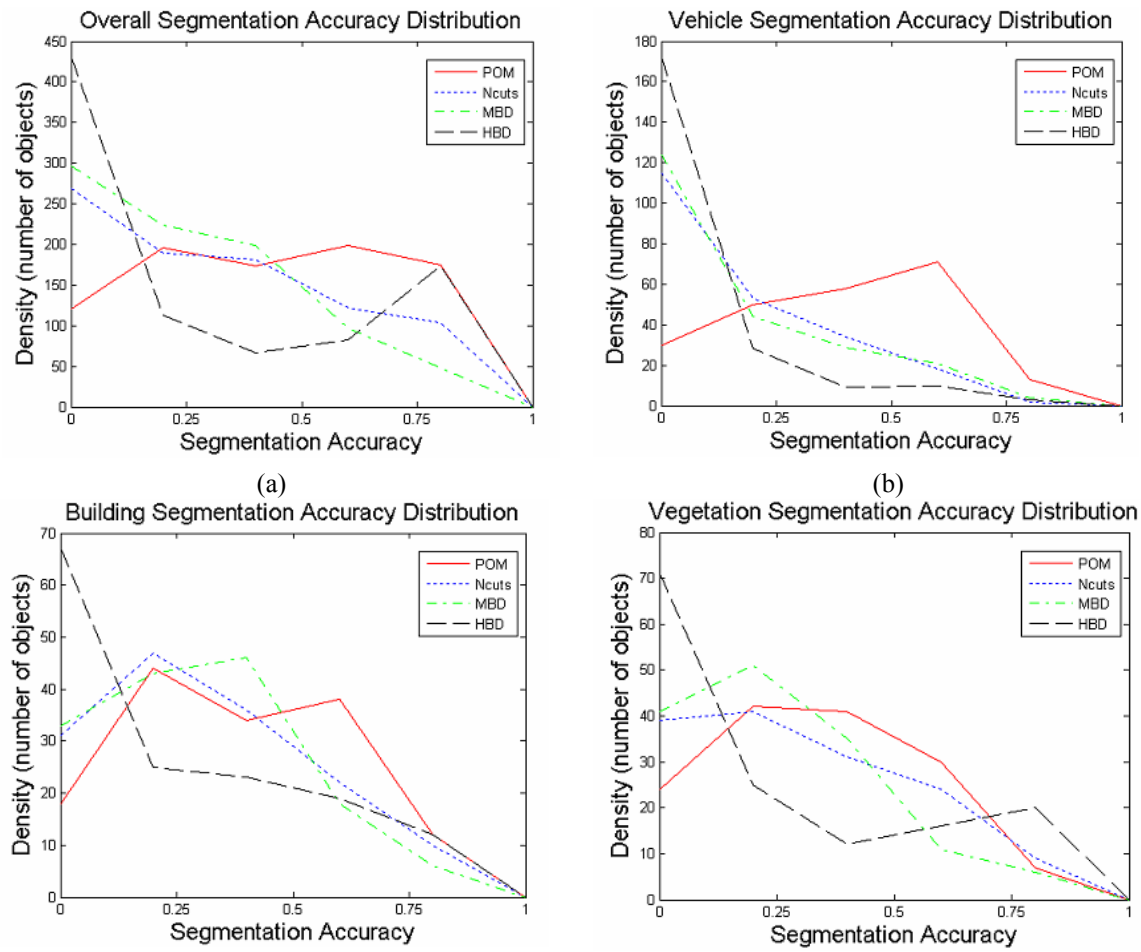
Figure 6.4 and Figure 6.5 present some qualitative testing results. Figure 6.4 compares the four tested methods on three examples. It can be observed that *Ncuts* algorithm presents good performance on detecting homogenous objects like vegetations and some buildings. For objects with multiple parts with different colors like vehicles, however, the *Ncuts* algorithm often merges some parts of these objects with the background that has close color to them.

The results of MBD and HBD show that both methods have a problem of finding the close-contours of the non-homogenous object boundaries. It can be observed that while the two methods can accurately detect the boundaries in the areas where the object parts are significantly different from the background, they both often make mistakes in the areas where object parts have the similar surface properties to the background. Compared to the other tested methods, our POM achieves the best visual qualities. The segments obtained by our method often coincide with the physical objects or their connected parts. Unlike the other methods, our method incorporates psycho-physical perceptions, which makes its output close



(a) (b) (c)

Figure 6.2 Boundary-based evaluation results. (a) Precision-Recall curve. (b) The distribution of Recall. (c) The distribution of Precision.



(a) (b)

(c) (d)

Figure 6.3 Region-based evaluation results. (a) The distribution of overall segmentation accuracy. (b) The distribution of vehicle segmentation accuracy. (c) The distribution of building segmentation accuracy. (d) The distribution of vegetation segmentation accuracy.

(a)

(b)

(c)

(d)

(e)

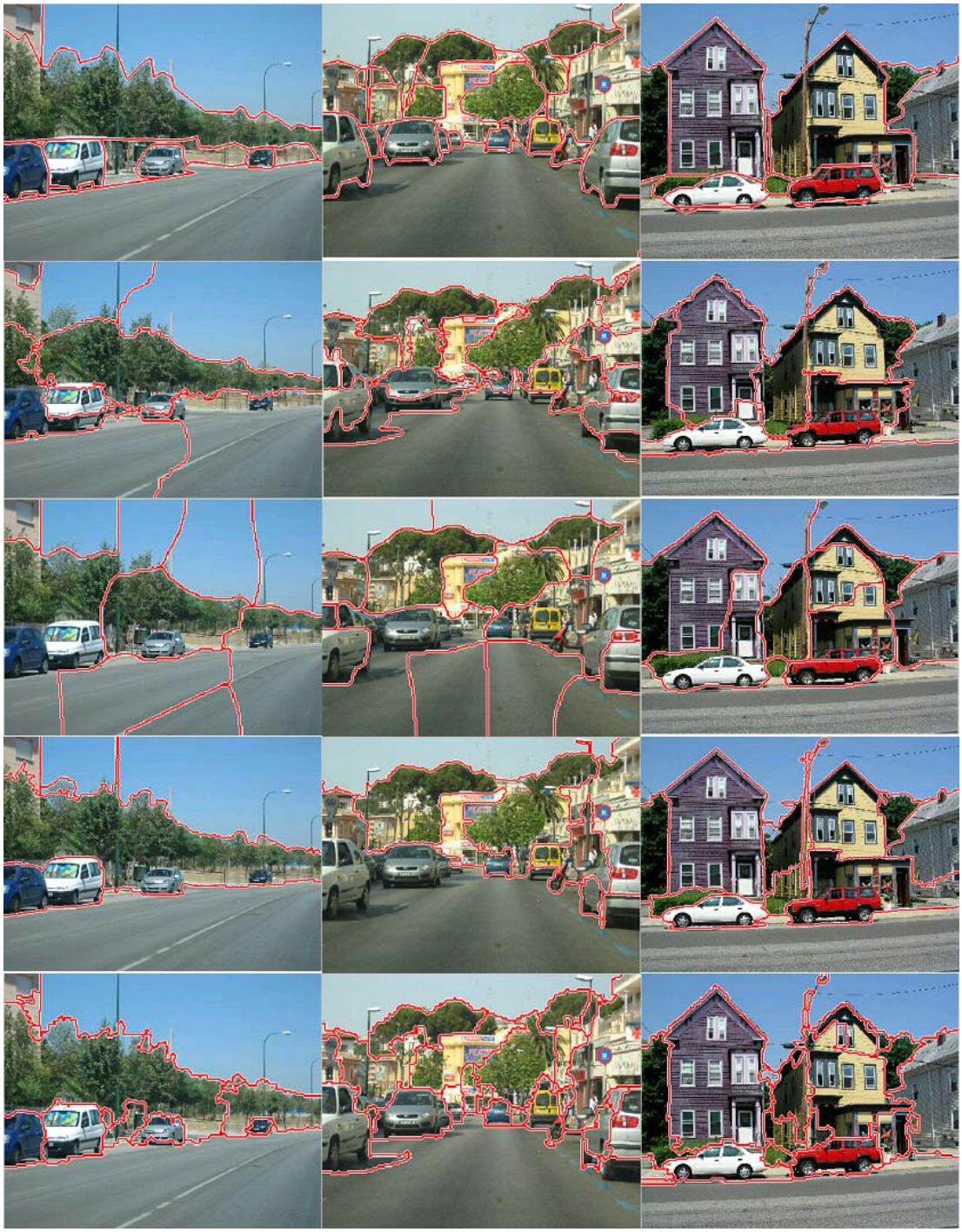


Figure 6.4 Comparison of the four tested methods. (a) An example of ground truth segmentations. (b)-(e) Segmentation results from the *Ncuts*, Martin's boundary detector (MBD), Hoiem's boundary detector (HBD), and our method based on the Perceptual Organization model (POM), respectively. Segmentation results of POM are much closer to the ground truth segmentations compared to the results from the other three methods. (This figure is best viewed in color).

Figure 6.5 Examples of our POM segmentation algorithm on different environments. (a) Roads. (b) Airports. (c) Suburban areas. (d) Residence areas and (e) Urban areas. It can be observed that the boundaries of most salient front-stage objects in these scenes are accurately detected. (This figure is best viewed in color).

Figure 6.6 Examples of the object classes that our POM segmentation algorithm can handle. (a) Vehicles; (b) buildings. (c) Other man-made objects with simple structures. (This figure is best viewed in color).

to human perception (the ground truth segmentation).

Figure 6.5 presents the segmentations of our image segmentation algorithm on different natural scene environments. In simple environments (e.g., roads) where there is no too much occlusion and clutter, the boundaries of almost all salient objects in the images are accurately detected. In complex environments (e.g. suburban and urban areas), even though our method might make some grouping errors due to occlusion and clutter in some regions of the images, it still generates good segmentation results – most salient objects in these scenes are accurately segmented. This shows that our proposed Perceptual Organization model can stably detect the boundaries of various objects under different environments.

Figure 6.6 summarizes the object classes our Perceptual Organization model can handle in the street. Our Perceptual Organization model favors vehicles very well. Vehicles mainly consist of three parts: windows, bodies and wheels. As shown in Figure 3.4, 3.5, and 3.6, the different parts of vehicles are approximately aligned, embedded or share certain amount of boundaries with each other. Our Perceptual Organization model can capture all these spatial relations and hence can group the whole vehicles together in most cases. Similarly, different parts of many buildings are symmetric, aligned or share certain amount of boundaries with each other. Therefore, our Perceptual Organization model can handle these types of buildings as well. For other many man-made objects with simple structures, such as fire-hydrants, ad boards, signs, etc. These objects either have approximately homogenous surfaces or the structural relations of the different parts of the objects obey the Gestalt cues presented in Chapter 3.3. Our Perceptual Organization model also can handle these objects well. In general, the experimental result shows that our Perceptual Organization model can handle the

Table 6.2 Segmentation accuracy score on Gould09 dataset

	sky	tree	road	grass	water	building	mountain	foreground	**average**	**Pixel %**
Gould09	32.5	13.5	28	26.4	37.6	11.8	20.4	9.3	16.3	**75.4**
POM (ours)	**46.1**	**29.3**	**38.8**	**42.9**	**58.8**	**29.6**	**28.8**	**27.8**	**32.5**	**-**

major object classes that appear in outdoor scenes, which builds a foundation for our object-based place recognition and loop closing method.

6.1.3 Gould09 database

6.1.3.1 Experimental setup

In second experiment, we test our image segmentation algorithm on a recently released outdoor image database Gould-09 [Gould09] (see examples in Figure 6.7). This database contains 715 images of urban and rural scenes assembled from a collection of public image datasets: LabelMe [Russell08], MSRC-21 [Shotton09], PASCAL [Everingham07], and Geometric Context (GC) [Hoiem05]. The images on this database are down-sampled to approximately 320x240 pixels. The images contain wide variety of physical and biological objects like buildings, sign, cars, people, cows, sheep, etc. This dataset provides ground truth object class segmentations that associate each region with one of 8 semantic classes (sky, tree, road, grass, water, building, mountain, or foreground). Besides the object class labels, the ground truth object segmentations that associate each segment with one physical object are also provided. Therefore we can evaluate our POM performance on segmenting different object classes on this dataset. The database is publically available from the first author's [Gould09] website. Following the same setup in [Gould09], we randomly split the database into 572 training images and 143 testing images.

We benchmarked a state-of-art class segmentation method – Gould *et al*. [Gould09b] for references on this dataset. The Gould's method also used superpixels as a starting point for their methods. The Gould's method we tested is a slight variant of the baseline method described in [Gould08]. The baseline method of [Gould08] achieved comparable result against the relative location prior method in [Gould08], Shotton's method [Shotton09] and Yang's method [Yang07] on the MSCR-21 [Shotton09] dataset (see figure 6.8 for an example). All two methods are trained on the training set and tested on the testing set. For our Perceptual Organization method (POM), we set parameter $\sigma=0.5$, $k=80$ and $min=80$ for Felzenszwalb's algorithm [Felzenszwalb04] to generate the initial superpixels for an input image and set parameters $\theta = [18, 3.5]$, $\alpha = 20$ and $\beta = 3$ for our Perceptual Organization model. We used the 572 training images to learn five binary Adaboost [Friedman00] classifiers to identify 5 background object classes (sky, road, grass, tree, water).

6.1.3.2 Experimental result

Table 6.2 compares the performance of our method with the baseline methods (Gould *et al*. [Gould09b]) on the Gould09 dataset. The segmentation accuracy measurement is still based on equation (6.2). The score is averaged over all the salient object segments. If the size of a ground truth object segment < % 0.5 of the image size, it is not a salient object and will not be accounted for segmentation accuracy. In total, we detect 2757 salient objects from 143 testing images and on average 19 objects per image. We used Normalized cut algorithm [Cour05] to generate 400 superpixels per image for Gould's method. We are able to achieve on average improvement of 16.2% over the performance of Gould's method.

The Gould09 database only provides 8 semantic classes (sky, tree, road, grass, water, building, mountain, or foreground) for training. The foreground semantic class on Gould09 actually includes a wide variety of object classes like cars, buses, people, signs, sheep, cows, bicycles, motorcycles, etc., which have totally different appearance and shape characteristics. This makes training an accurate classifier for classifying the foreground class become challenging. Gould's method seems to be adaptable to the variation of the number of semantic classes. Their method achieved 70.1% pixel-level accuracy on the 21-class MSCR database and achieved impressive 75.4% pixel-level accuracy on the 8-class Gould09 database. However, since the foreground class includes a wide variety of object classes, Gould's method cannot well handle clutter environments where multiple foreground objects may appear in close locations. In such cases, Gould's method often merges a group of different object instances like people, car, sign etc. in close location together. This affects the performance of their method on the object-level segmentation. If the foreground class can be further subdivided into more semantic object classes, the performance of Gould's method can be expected to improve on the Gould09 database.

The problem of small number of semantic classes does not affect our method. Our method only requires identifying 5 common homogenous background objects (sky, tree, road, grass, water). The remaining object classes are treated as structured objects. Our Perceptual Organization model can piece the major portion of the structured objects together without requiring recognizing them. From this point of view, our method is easy-to-train compared to the class segmentation methods in the literature. Among 2757 salient objects detected in the testing images, the structured objects (buildings + foregrounds) accounts for 52.6%. Our method significantly outperforms Gould's method on segmenting the structured objects. This shows that our Perceptual Organization model can well handle various structured objects appearing in outdoor scenes.

To gain a qualitative perspective of the performance of the three testing methods, we present several representative images (first row), along with the ground truth segmentations, and Gould's results (second row and third row), as well as the results of our method (fourth row) in Figure 6.9.

The first example (the first column) contains a nearly centered people with clean backgrounds. It can be observed that all three methods can well handle this type of images. The Gould's method accurately classifies most part of the centered people as foreground. Our method also pieces the major portion of the people together. The second example is a typical street scene which contains several structured objects (vehicle, building), and background objects. These structured objects are well separated in the image. Our method well segmented most of the vehicles and the buildings. The third image is a clutter street scene

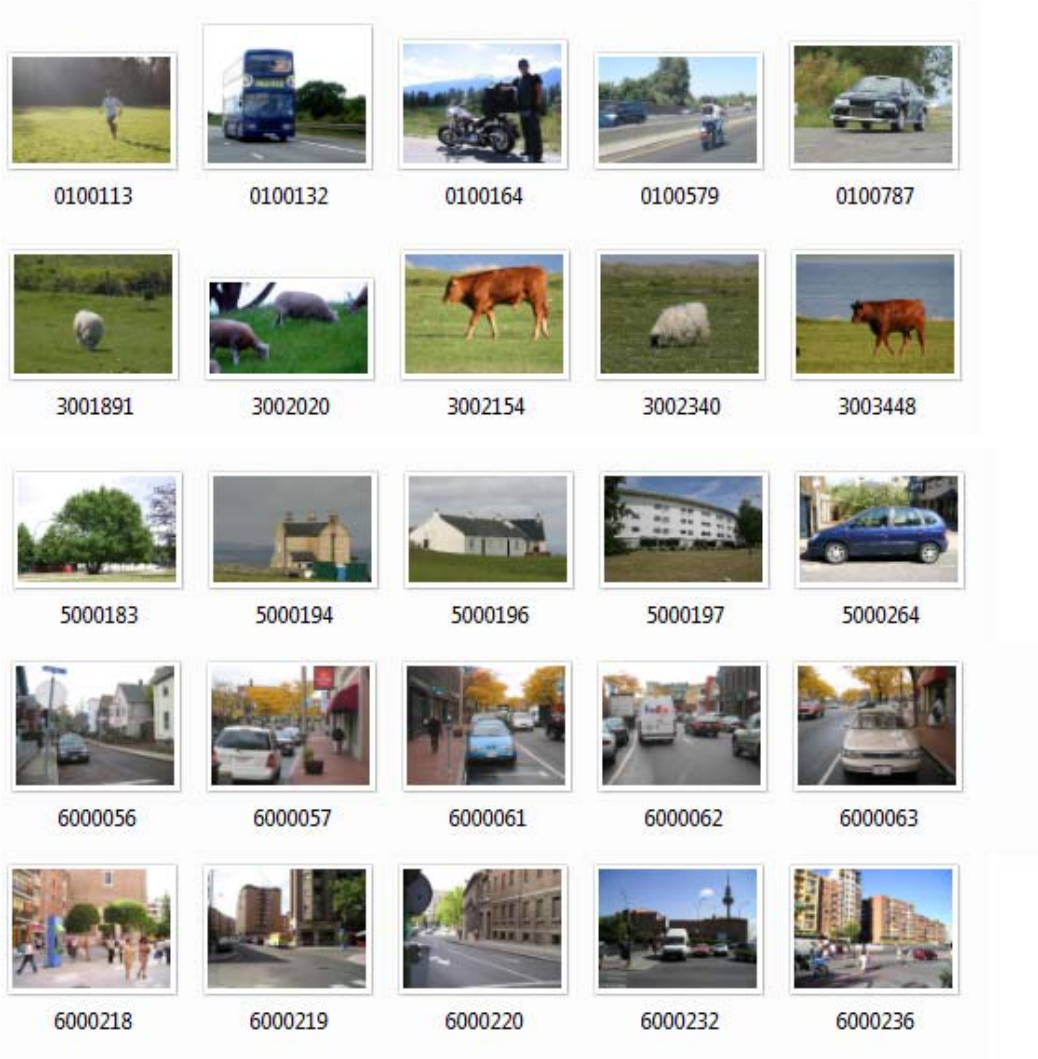


Figure 6.7 Examples of Gould09 database.

Algorithm	21-class MSRC Accuracy Min.	Avg.	Max.	Std.	9-class MSRC Accuracy Min.	Avg.	Max.	Std.
Shotton et al.		72.2%		n/a		-		n/a
Yang et al.		75.1%		n/a		-		n/a
Schroff et al.		-		-		75.2%		-
Baseline Logistic	61.8%	63.6%	65.4%	1.67%	77.5%	78.9%	79.8%	1.05%
Baseline CRF	68.3%	70.1%	72.6%	1.81%	81.2%	83.0%	81.4%	1.28%
Logistic + Rel. Loc	73.5%	75.7%	77.4%	1.74%	87.6%	88.1%	88.9%	0.67%
CRF + Rel. Loc.	74.0%	76.5%	78.1%	1.82%	87.8%	88.5%	89.5%	0.82%

Figure 6.8 The performance of the baseline method of Gould08 (masked by a red arrow) is comparable to three other class segmentation methods on MSRC-21 database based on pixel-level accuracy. The table is from [Gould08].

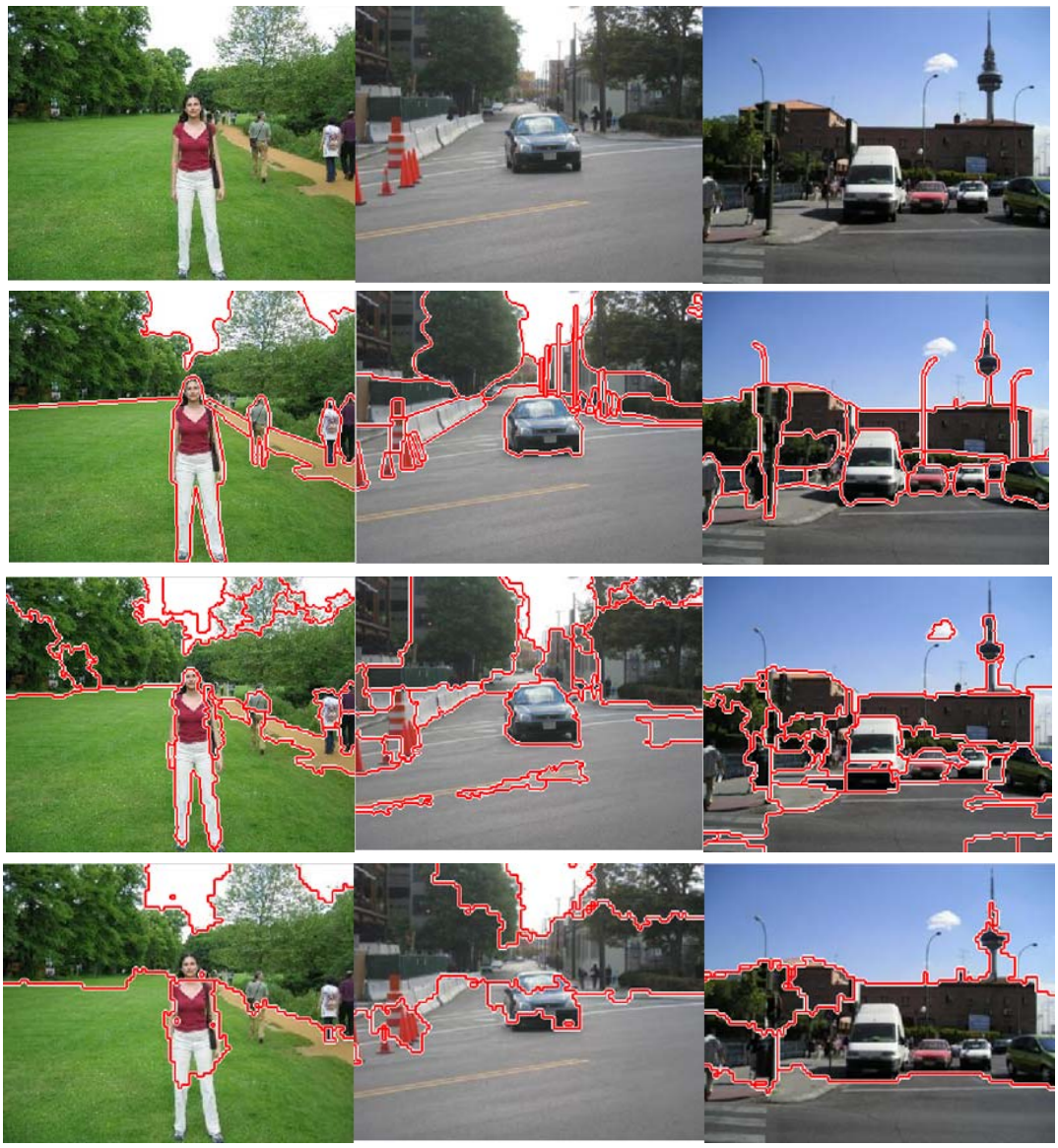


Figure 6.9 Examples of our POM segmentation algorithm on Geometic Context dataset. Row 1: Input images. Row 2: Ground Truth segmentations. Row 3: POM (ours) results. Row 4: Gould's result. It can be observed that our method can stably segment some salient objects in different scenes. (This figure is best viewed in color).

which contains several vehicles parked in front of a building. These vehicles stay closely to each other. Our method well segmented most of the vehicles and the background building. The Gould's method miss-classified some parts of the buildings as foreground. Besides, Gould's method merges the whole group of vehicles together.

6.1.4 Geometric Context (GC) database

6.1.4.1 Experimental setup

To further test the performance of our method under different outdoor scenes, we evaluate our image segmentation method on the Geometric Context database (GC) [Hoiem05], which consists of a wide variety of outdoor scenes including beaches, forest, hills, suburbs and urban streets etc (see examples in figure 6.10). 101 images of the dataset are assigned with ground truth object segmentations by the authors in [Hoiem05] (each ground truth object segment corresponds to one physical object instance). Therefore we can evaluate our POM method's performance on handling different outdoor scenes on this dataset. Since the ground truth object class information is not provided on this database, which is required for the class segmentation methods in the training stage, to make a fair comparison, we only compare our method with the Hoiem's boundary detector (HBD)[Hoiem07] on overall object segmentation. Hoiem's boundary detector has been trained and tested in the same dataset and hence can be used as a baseline for this dataset.

In this dataset, we set parameter $\sigma=0.5$, $k=280$ and $min=100$ for Felzenszwalb's algorithm [Felzenszwalb04] to generate the initial superpixels for an input image and set parameters $\theta = [18, 3.5]$, $\alpha = 20$ and $\beta = 3$ for our Perceptual Organization model. We use the same background classifiers trained in the Gould-09 database to identify background objects on this dataset.

6.1.4.2 Experimental result

Table 6.3 compares the performance of our method with the baseline HBD method in the Geometric Context (GC) dataset. The segmentation accuracy measurement is still based on equation (6.2). The score is averaged over all the salient object segments. If the size of a ground truth segment size < % 0.5 of the image size, it is not a salient object and will not be accounted for segmentation accuracy. In total, we detect 786 salient objects from 101 images and on average 7.8 objects per image. We are able to achieve on average improvement of 16% over HBD's performance.

To gain a qualitative perspective of the performance of the two testing methods, we present several representative images (first row), along with the ground truth segmentations (second row) and HBD's results (third row), as well as the results of our method (fourth row) in Figure 6.11. The first example (the first column) shows that both methods can well handle salient objects with clean backgrounds. HBD correctly detect the major boundaries of the big tower. Our method also pieces the major parts of the tower together. The second example is a street scene which contains many rigid objects (vehicles) in rear view. Our method well segment most of the vehicles while HBD treats the whole scene as a big object. The third

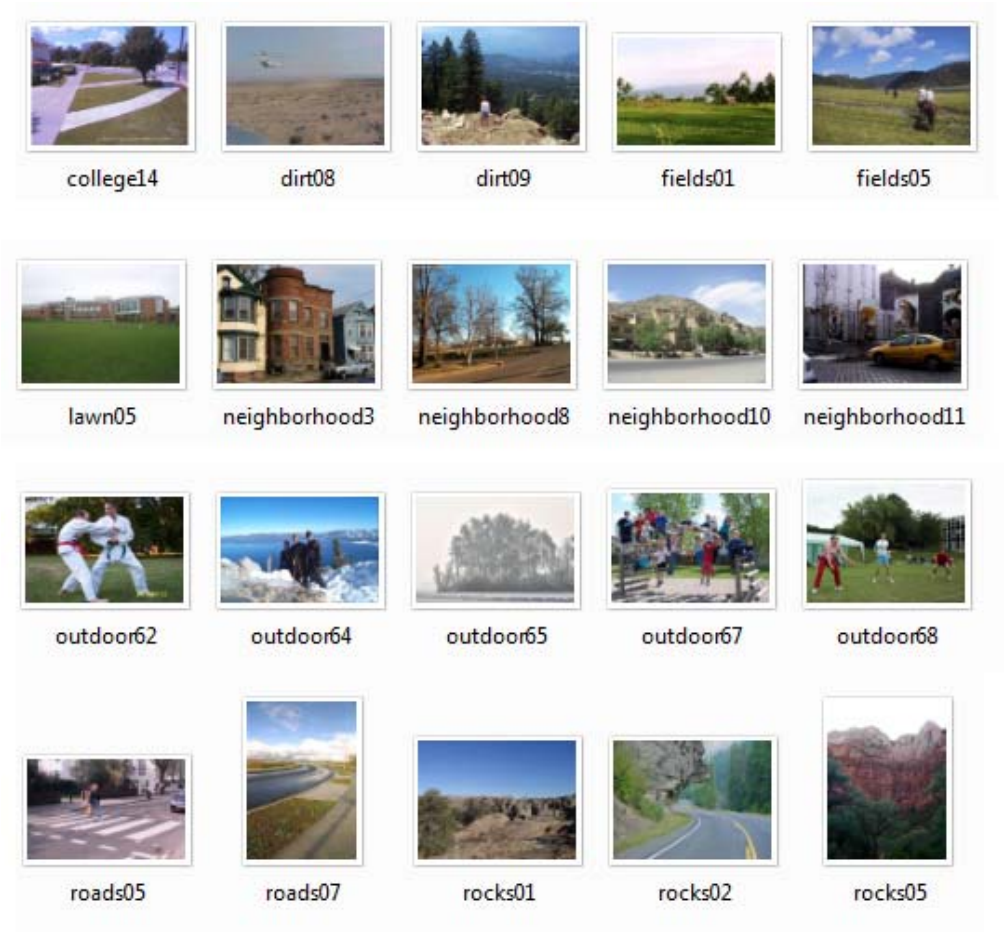


Figure 6.10 Examples of Geometic Context (GC) database.

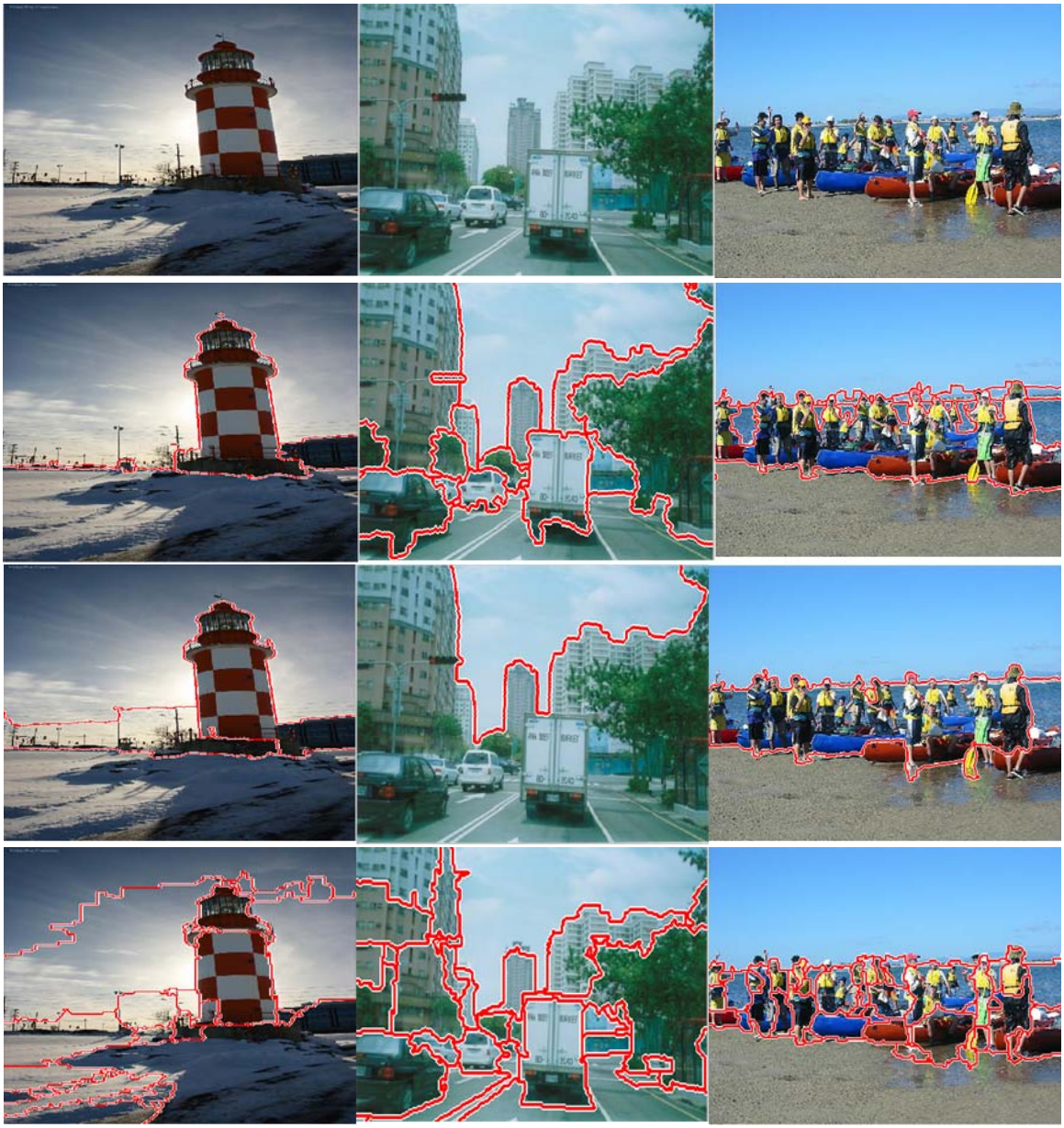


Figure 6.11 Examples of our POM segmentation algorithm on Geometic Context dataset. Row 1: Input images. Row 2: Ground Truth segmentations. Row 3: Hoiem's results. Row 4: POM (ours) results. It can be observed that our method can stably segment some salient objects in different scenes. (This figure is best viewed in color).

Table 6.3 Summary of segmentation score on GC dataset

	HBD	Ours
Overall (%)	36.8	52.8

image is a clutter environment. HBD treats the group of people as a big object and accurately detect the boundary of the group of people. Our method generates finer segmentations. Although several people in middle are grouped together due to clutter, our method generates several good segments for people standing on the edge of the group of people.

6.1.5 Qualitative assessment

Figure 6.12 and Figure 6.13 show some good and bad examples, respectively. From Figure 6.12, it can be noticed that our method can well segment many kinds of structured objects even under clutter environments.

There are still some mistakes that we would like to address in the future. The inference of our Perceptual Organization model is mainly based on the geometric relations between different object parts. This requires obtaining the geometric properties (shape, size, etc.) of object parts. We assume that object parts have nearly homogenous surface and hence the uniform regions in an image correspond to object parts. Although this assumption holds in most cases, there are still some exceptions. For example, in Figure 6.13, the black car body is painted into different patterns. As a result, the car body is over-segmented to many small parts. Under this situation, our Perceptual Organization model could not detect any special relations between the small parts and hence could not piece them together. We hope that we can handle this problem in the future by redefining the *proximity* law. A group of neighboring parts with similar size actually obeys the definition of *proximity* law in many Perceptual Organization works.

Our method can handle objects with polygon shapes well. But some object classes like bicycles, motorcycles, some buildings, etc. have very complex structures. Some parts of the objects are not strongly attached to other parts. For these object classes, our Perceptual Organization model may not be able to piece the whole object together. Instead, it may only piece some semantic meaningful parts of the objects together. For these objects, higher-level object-specific knowledge is still required to segment the whole objects.

Another problem is caused by strong reflection. An example is shown in Figure. Due to the strong reflection, the upper rear part of the blue bus shows extremely bright white color. Our method identified the region as sky and hence did not piece the part with the bus. In clutter environments, one structured object may stand in front of another structured object. From some viewpoints, the parts of the front object may coincidently have special geometric relations with the parts of the back object. Under these situations, our Perceptual Organization model may be confused and merge these parts together. This problem can be addressed by recognizing the background structured objects. Currently our method can only identify five homogenous background object classes (sky, road, tree, grass and water).

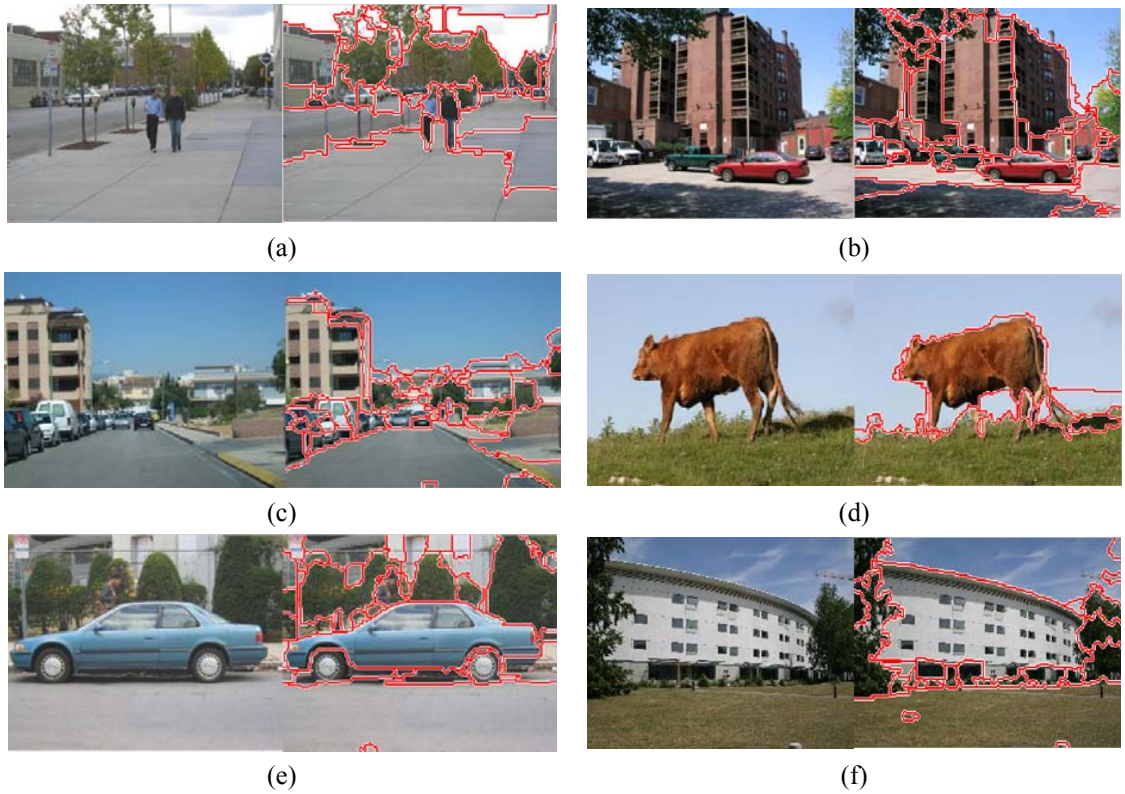


Figure 6.12 Examples of where our POM segmentation algorithm does well. (This figure is best viewed in color).

Figure 6.13 Examples of where our POM segmentation algorithm makes mistakes. (This figure is best viewed in color).

Actually, the mountains, buildings and walls are also common background objects in outdoor scenes. We plan to enhance the background identification capability of our method in the future by training more classifiers to identify mountains, buildings and walls etc. With the ability of identifying more background object classes, the performance of our method can be expected to be further improved.

6.2 Object classification

6.2.1 Experimental setup

We evaluated the performance of discrimination of the proposed method with respect to the Microsoft Research Cambridge (MSRC) database[Shotton09]. The MSRC database is composed of 591 images with 21 object classes: building, grass, tree, cow, sheep, sky, airplane, water, face, car, bicycle, flower, sign, bird, book, chair, road, cat, dog, body and boat. The images are segmented and objects are hand-labeled with colors assigned as indices into the list of object classes. All the object classes are viewed under general lighting conditions and poses (see Figure 6.14 for examples). We used the GML Adaboost classifier [Friedman00] for both feature selection and object classification. In the feature selection stage, for each object class, we applied the feature selection method described in Chapter 4.2 to select a subset of features that characterizes the object class well. In the object classification stage, first, the database is split into 50% training and 50% testing sets. Then for each class, we applied the corresponding selected features on the training set to train a binary classifier. Each binary classifier predicts if an instance belongs to one particular class. We directly use five binary classifiers to recognize five common background object classes (skies, roads, trees, grasses and water). Finally, with the collection of structured object binary classifiers, we used the one-per-class method [Guruswami99] to build a multiclass classifier by classifying an instance to the class whose binary classifier returns highest confidence level.

For all our experiments, we use region-based recognition accuracy – i.e. given a ground truth segment, assign a single class label to it and then measure classification accuracy.

6.2.2 Results

As we mentioned in Chapter 3.4, the common backgrounds in outdoor natural scenes are those unstructured objects like skies, roads, trees, grasses, etc. These background objects have low visual variability and in most cases are distinguishable from other structured objects in an image. For instance, a sky usually has a uniform appearance with blue or white colors; a tree or grass usually has a textured appearance with green colors. Therefore, these background objects can be accurately recognized solely based on appearance information.

We first randomly split database into 50% training and 50% testing set. We then use the 151-dimensional appearance features described in Chapter 4.1.1 to train five binary Adaboost

classifiers [Friedman00] for the five background object classes (skies, roads, trees, grasses and water) using the training set. Table 6.4 reports the classification accuracies of the five background object classifiers on the testing set. It can be observed that the trained classifiers can accurately recognize the five background objects. For both training and testing, we use the ground truth segments for feature extraction.

After recognizing the unstructured background object classes, we then test the performance of our method on structured object classes. Since appearance-based methods have been widely used in literature, we want to access how much the parts-layout and shape information, combined with appearance information, can help with object class recognition. Specifically, we are interested in the following aspects: (1) the effect of combing appearance and parts-layout features; (2) the effect of combing appearance and shape features; (3) the effect of combing appearance, parts-layout and shape features.

We selected three typical structured classes: car, building and face. Car and building have large variability on appearance but have certain pattern on parts-layout and shape. Face has stable variability on appearance and certain pattern on shape. The database contains enough training samples for the three classes (see Figure 6.15 for examples). Thus, the three classes should be well characterized by combination of appearance, parts-layout and shape information.

Table 6.5 reports the classification accuracies on the three selected object classes with different combination of features. The results are classification accuracy of the corresponding binary classifiers of the three classes. The one-per-class method was not applied in this experiment. It can be observed that the three classes cannot be accurately classified only based on appearance information. Both parts-layout and shape information helps improving the classification accuracy. The use of joint appearance, parts-layout and shape information gains the highest classification accuracies for the three object classes.

We also evaluated our feature selection method on the three selected object classes. We compared our method against the HITON feature selection method [Aliferis03], which has achieved successes on many complex large datasets. We also tested the results when no feature selection was applied (i.e. using the full 3181 features for classification). Table 6.6 reports the classification accuracies on the three selected object classes with different feature selection methods. The results are classification accuracies of the corresponding binary classifiers of the three classes. The one-per-class method was not applied in this experiment. It can be observed that with full set of features, even though all the information is contained, the learned classifiers are still not accurate for the three object classes because of too many noises (redundant features) in the data. For the HITON feature selection method, due to the small sample size, the conditional tests of independence are not reliable. As a result, HITON returned incorrect MB. We found that HITON picked the majorities of features from the parts-layout features and thus basically ignore the important appearance and shape information. Compared to HITON, our feature selection methods always picked a compact combination of appearance, parts-layout and shape features. This experiment shows that with the compact informative features, we can learn more accurate classifiers for structured object classes.

Next, we compared the classification accuracy of our method against the state-of-the-art TextonBoost method [Shotton09]. TextonBoost recognizes object classes by combining appearance, layout and context information, which helps it achieve higher classification accuracy than the appearance-only based method [Winn05] in this MSRC 21-class database.

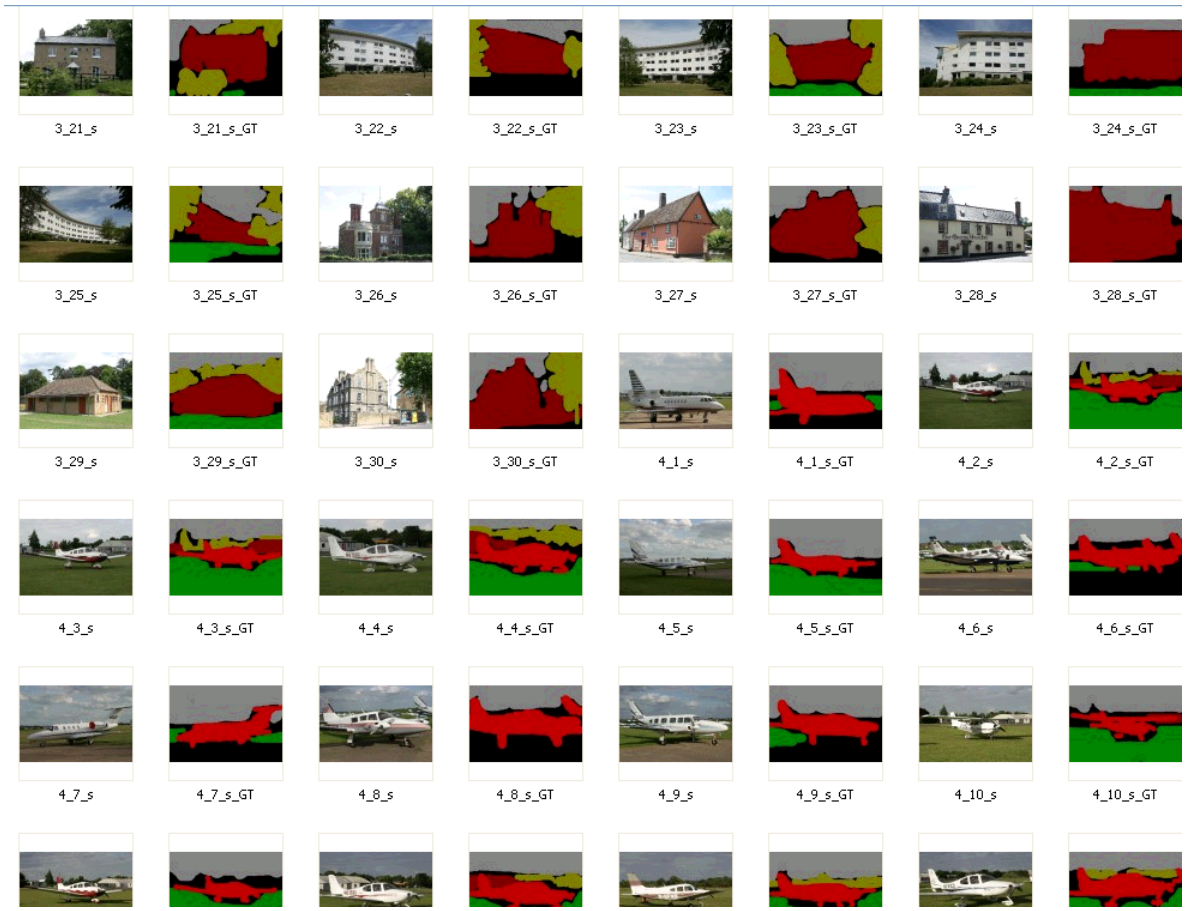


Figure 6.14 Examples of MSRC-21 database [Shotton09].

Table 6.4 Summary of results of background objects classification accuracy.

Accuracy (%)	Sky	Road	Tree	Grass	Water
TextonBoost	87.1	67.4	67.8	93.4	48.7
Our method	**92.3**	**82.9**	**86.1**	**96.3**	**55.3**

Table 6.5 Summary of results obtained by different combination of features on classifying car, building and face. Notice that the highest accuracies are obtained by the use of joint appearance, parts-layout and shape information

	Car	Building	Face
Appearance	63.1	61	60
Appearance + parts-layout	68.4	66.2	73.1
Appearance + shape	80.9	73.8	71.4
Appearance + parts-layout + shape	**85.7**	**76.2**	**82.8**

Table 6.6 Summary of results obtained by different feature selection methods on classifying car, building and face. In this experiment, we use the binary classifier for each class. Notice that the highest accuracies are obtained by the use of our feature selection method.

	Car	Building	Face
Full set	23.8	65.5	73.7
HITON	62.5	52.9	73.7
Our method	**85.7**	**76.2**	**82.8**

Table 6.7 Summary of results obtained by different classification methods on classifying car, building and body.

	Car	Building	Body
TextonBoost	56.3	60	40.9
Our method (based on Ground Truth)	**85.7**	**91.9**	**86.4**
Our method (based on POM)	81.3	83.1	70.8

Following the same setup in [Shotton09], we split the database into 45% training, 10% validation and 45% testing sets. The boost rounds were set as 700. For our method, we split the database into 50% training and 50% testing set. We applied the one-per-class method on the 16 structured binary classifiers to build a multiclass classifier by classifying an instance to the class whose binary classifier returns highest confidence level. In this experiment, we still use the ground truth segments of testing set for feature extraction. Figure 6.16 shows the confusion matrixes of TextonBoost and our method respectively. Notice that our method achieves higher accuracies for almost every structured object class. For building, the classification accuracy of the binary classifier is only 76.2%. The classification accuracy of the multiclass classifier raises up to 91.9%. This means, for many building instances, the building binary classifier alone is not confident enough to classify them as building. But compared to the results of other binary classifiers, the building binary classifier still returns the highest confident levels for these instances. For classes which have low visual variability and many training examples (such as grass, tree, sky, etc.), our method achieves slightly better or comparable results as TextonBoost. Even for the classes that have high visual variability and fewer training examples (like boat, chair, bird, dog), our method achieves higher accuracy than TextonBoost. As pointed out in [Shotton09], with more training examples, the accuracies for these classes are expected to be improved.

All the experiments above use the ground truth segments of the testing images for feature extraction. The experiment results show that under ideal case where highly accurate segments of the objects are given, our object classification method can achieve high classification accuracy. In practical cases, however, the highly accurate segments of the objects can not be obtained automatically. To evaluate the performance of our method in practice, finally, we test our object classification method on our POM segmentations, which can only provide fairly accurate segments for some objects. We select three structured object classes – building, car and people (body) for the experiment. The reason is that the tree object classes often appear in outdoor streets. Recognizing these object classes is helpful for object-based place recognition. For each test image, we apply our POM segmentation method to segment the image. Our POM method can identify five background objects (sky, road, grass, tree and water) and generate fairly accurate segments for the remaining structured objects. We then use the multi-class classifier to classify the segments of the structured objects.

Table 6.7 reports the classification accuracies on the three selected object classes with different object classification methods. It can be observed that our object classification method still achieves good accuracy on the fairly accurate segments of the three object

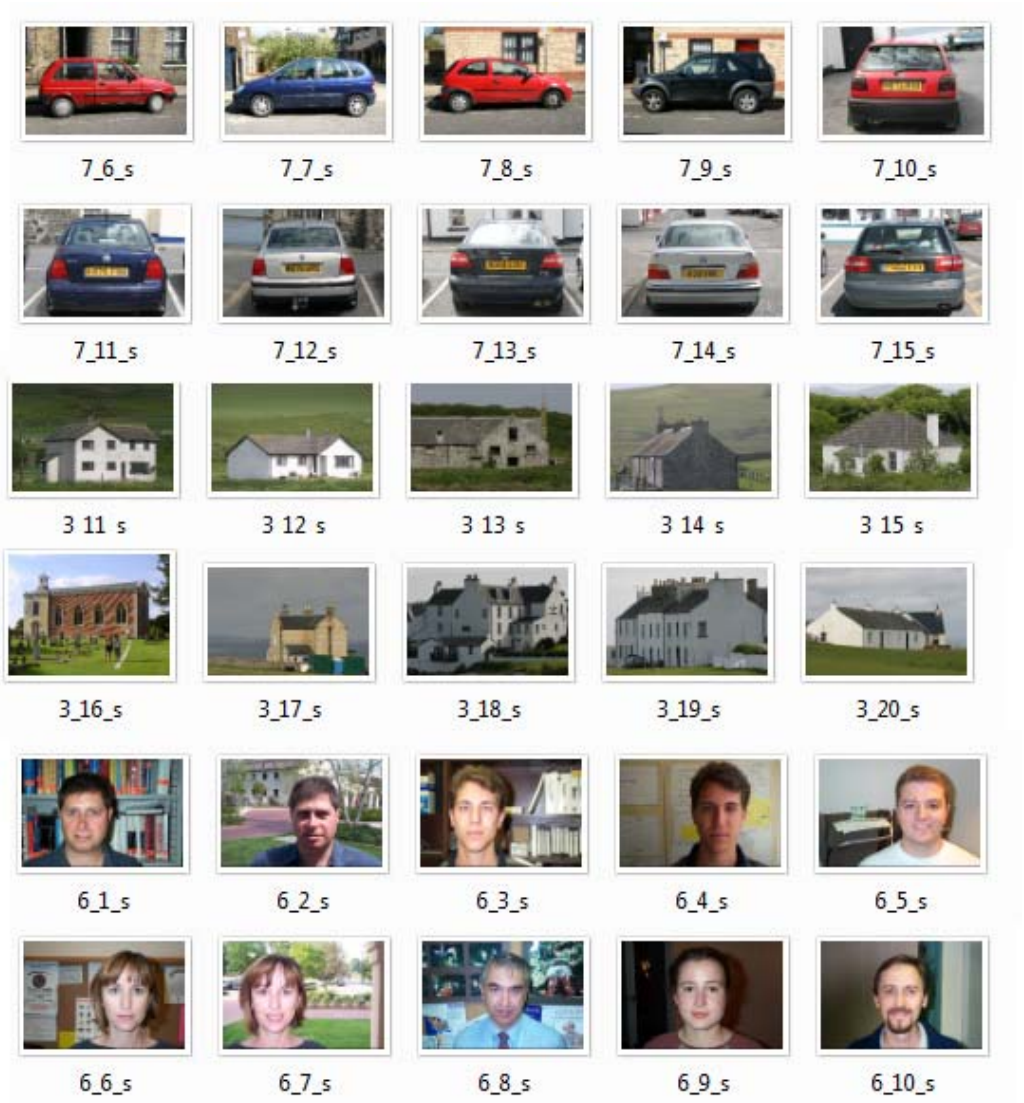


Figure 6.15 Examples of car, building and face images in MSRC-21 database [Shotton09].

Inferred class / True class	building	grass	tree	cow	sheep	sky	aeroplane	water	face	car	bike	flower	sign	bird	book	chair	road	cat	dog	body	boat
building	60		10		1.7	3.3	1.7	5		1.7	1.7		1.7		1.7	1.7	6.7		3.3		
grass	1.1	93.4	1.1		1.1												2.2			1.1	
tree	14.3	6.1	67.4			8.2	2								2						
cow		52.9	5.9	41.2																	
sheep		20			60									13.3			6.7				
sky	1.4	2.9	2.9			87.1		4.3									1.4				
aeroplane	16.7	16.7					58.3	8.3													
water	5.4	5.4	13.5	2.7		13.5		48.7									11				
face	17.2	6.9	6.9		3.5				27.6						24.1				3.5	10.3	
car	6.3		6.3		18.8					56.3							6.3			6.3	
bike	12.5		12.5		6.3					6.3	56.3		6.3								
flower			23.5									64.7					11.8				
sign	33.3							8.3					50		8.3						
bird	5	15	10		5	10		10						10	10	5	15				5
book	5.9														94.1						
chair	17.7	17.7	35.3	5.9												58.9	17.7				
road	3.4	6.8	1.7	1.7		1.7		8.5			3.4				5.1		67.8				
cat	30.8								7.7				7.7		7.7		7.7	38.5			
dog	7.7		7.7	7.7	7.7			7.7	7.7								15.4		15.4	23.1	
body	6.5	9.7		9.7		3.3									25.8					41.9	3.2
boat	12.5					6.3		43.7		25					6.3						6.3

Inferred class / True class	building	grass	tree	cow	sheep	sky	aeroplane	water	face	car	bike	flower	sign	bird	book	chair	road	cat	dog	body	boat
building	**91.9**			1.2							2.3			1.2		1.2				2.3	1.2
grass		**96.3**																			
tree			**86.1**														2.3				
cow	6.3			**84.4**					3.1			6.3									
sheep	25.0			5.0	50.0									10.0		5.0			5.0		
sky						**92.3**															
aeroplane	6.3						**87.5**														6.3
water								**55.3**													
face	5.3								**94.7**												
car	14.3									**85.7**											
bike	12.5										**81.3**									6.3	
flower	21.7											**78.3**									
sign	7.7								15.4				**76.9**								
bird	38.7													**58.1**							3.2
book										15.8					78.9					5.3	
chair	50.0			6.3					12.5							31.3					
road																	**82.9**				
cat	58.3																	**41.7**			
dog	68.8								6.3										**25.0**		
body	11.4								2.3											**86.4**	
boat	23.1										3.9										**73.1**

Figure 6.16 Classification accuracy confusion matrixs. Upper: TextonBoost [Shotton09], down: Ours.

classes generated by our POM method. Especially for the rigid objects like car and building, there are only small performance drops on POM segmentation compared to the performance on the highly accurate ground truth segmentation. For the non-rigid object like body, in some cases our POM can not piece the whole body together due to the large pose deformation. This affects the performance of our classification method.

6.3 Place recognition and loop closing

6.3.1 Experimental setup

We tested the place recognition and loop detection algorithm described in Section 5.5 using imagery collected from a mobile robot. The test environment is shown in Figure 6.17. The robot first traveled over an outdoor trajectory of 1.6km. In this trip, the robot took an image of resolution 640×480 around every four meters. For each image, we first extracted SIFT features from that image using Lowe's implementation [Lowe04]. Then, we run our image segmentation algorithm to segment the image into its constituent objects. Any object that contains more than 8 SIFT features were selected as salient objects. By comparing this set of salient objects with those salient objects detected in several previous locations, we were able to detect a set of landmark objects that were visible from multiple locations. These landmark objects were used to label the place and were stored in a range tree database with color and number of SIFT features inside as index information. Each landmark object was represented with a list of SIFT descriptors contained in the object. Only SIFT descriptors contained in these landmark objects were retained. All other SIFT descriptors were discarded. By doing so, we significantly compressed the database size.

6.3.2 Results

As seen in Table 6.7, during this trip, the robot collected total 374 images. Together, 702,118 SIFT features were extracted from these images. We detected 621 landmark objects from this environment. Total 91,908 SIFT features were contained in these landmark objects. Thus, only less than 14% of total 702,118 SIFT descriptors extracted from 374 images were stored in the range tree database. More importantly, the stored SIFT descriptors are grouped with good index information. The combination of database compression and index information enables us to detect a loop without a costly search.

After closing the loop, the robot traveled along the loop again. In this trip, it randomly took images from 50 places. For each place, it took two images from two different locations so that we could detect some landmark objects in the scene. We then used the detected landmark objects at each place to test if our algorithm can detect the loop. Our algorithm correctly detected the loop at all the 50 places. Depending on the numbers of objects appearing in scenes, instead of detecting a single location where the set of query objects were observed

from in the first trip, our algorithm might list multiple continuous locations where the set of query objects were observed from in the first trip at some places. This is because at these places, only several objects appeared in the scenes. As a result, the robot observed a same set of objects at multiple continuous locations in the first trip. In such cases, our algorithm could not accurately judge which location the set of query objects were observed from in the first trip, but instead found that the set of query objects were observed at multiple (2~4) locations in the first trip.

Figure 6.18 shows an example of loop detection. The building and the tree in the image taken from the second trip are matched with the corresponding objects in an image taken from the first trip. Therefore our algorithm recognizes the images as originating from the same place. Figure 6.19 highlights the robustness of our algorithm to scene change. It can be observed that there exist big scene difference between each pair of images due to different background vegetations and different illumination conditions. This may cause some problems for place recognition methods based on comparison on the global scene image properties like [Cummins07]. Different from these methods, our algorithm recognizes a place based on matching the widely visible objects in the scene to those objects stored in a database. Thus, as long as the same set of objects recorded before appear in a scene, our algorithm can correctly recognize the place despite the big scene difference in background. This shows that place recognition based on object level are more robust to scene changes.

In most cases, our algorithm can detect a loop between 0.2~3.7 seconds. We believe that this performance is relatively efficient compared with existing image-based loop detection methods. Since to our best knowledge, most existing image-based method only applied to small environments. The work [Cummins07] that was tested in a similar environment like ours took maximum 6 seconds to detect a loop. At four places where were totally covered by vegetations, however, the loop detection efficiency dropped down. Unlike man-made objects that have various surface characteristics (e.g. colors and textures), vegetations are not appearance-distinguishable landmarks. Many vegetations have similar green color and contain several hundreds of SIFT features. In such cases, the index information cannot give us too much help. We had to compare the vegetations detected at the four places to many vegetations stored in the range tree database. Thanks to the distinctive SIFT features, our algorithm still correctly recognized the four places. But this was achieved at high price – it took around 5.5~6.3 seconds to detect a loop at the four places. These kinds of places are common challenge for any appearance-based place recognition methods. Even human beings may have trouble distinguishing some places without special characteristic like these. In general, for any place that contains some man-made objects, the loop detection can be finished within 3.7 seconds (excluding the time for extracting SIFT features and image segmentation). Therefore, our method works efficiently in most cases in this complex large outdoor environment.

Figure 6.20 presents the query efficiency for the SIFT database and the range tree database. Six sets of images (ranging from 60 to 160) were selected from the image collections. For each set of images, two databases were built. One was a SIFT database formed by all the SIFT features extracted from these images. The other one was a range tree database formed by all the landmark objects detected from these images. For each database, two queries were executed. One query image was selected from the set of images the database was built on and the other one was not. As expected, the query efficiency was much higher in the range tree

database than in the SIFT database. One can see that for the SIFT database, the query performance quickly dropped down when the database size increased. Especially for the unmatched query, all the SIFT features in the database had to be compared to decide that there was no match. This implies that the works based on affine covariant visual features [Newman05],[Se02],[Se05] are not applied to large complex outdoor environments since these methods require costly searches for loop detection in such environments. For the range tree database, there was no big difference between the matched and unmatched queries because it is an object-based database. For any query object, only a small range of objects in the database needs to be compared to (in this experiment, the query objects are salient objects detected from a single image). Therefore, we claim that the range tree database structure is efficient for large environments.

6.4 Summary

In this chapter, we present the segmentation algorithm, object classification and place recognition experimental results. We first selected 100 typical street scene images from the challenging LabelMe dataset as test images. The test images cover various environments like urban areas, suburban areas, residence areas, roads, airports, etc. We benchmarked three methods of reference: the recent version of *Ncuts* [Cour05], Martin's [Martin04] boundary detector and Hoiem's [Hoiem07] boundary detector. We compared their segmentation results with ours. The experimental results show that our segmentation method outperforms the other three methods. Especially for the structural-rich objects like buildings and vehicles, our method achieve significantly better results. We then evaluate our segmentation method on a recently released challenging dataset (Gould09) [Gould09], which contain wide variety of physical and biological objects like buildings, sign, cars, people, cows, sheep, etc. We compared our algorithm with a baseline outdoor segmentation method Gould09 [Gould09]. Our algorithm outperforms Gould09 by 16% on region-based segmentation accuracy. The experimental result shows that our segmentation method can handle various physical and biological objects well. To further test the performance of our segmentation algorithm on different outdoor scenes, we evaluate our image segmentation method on the Geometric Context dataset (GC) [Hoiem05], which consists of a wide variety of outdoor scenes including beaches, forest, hills, suburbs and urban streets etc. We compared our algorithm with a baseline outdoor segmentation method HBD [Hoiem07]. Our algorithm outperforms HBD by 16% on region-based segmentation accuracy. The experimental result shows that our segmentation can perform well under wide variety of outdoor scenes.

We then evaluate our object classification method on the MSRC-21 dataset. The experimental results show that our method significantly increases classification accuracy for many object classes in the dataset.

At last, we test the place recognition method on a dataset collected from a 1.6km outdoor loop. The experimental result shows that our place recognition method can accurately and efficiently recognize various places that have been visited before, even under situation where there exist big scene differences caused by different background vegetations and different illumination conditions.

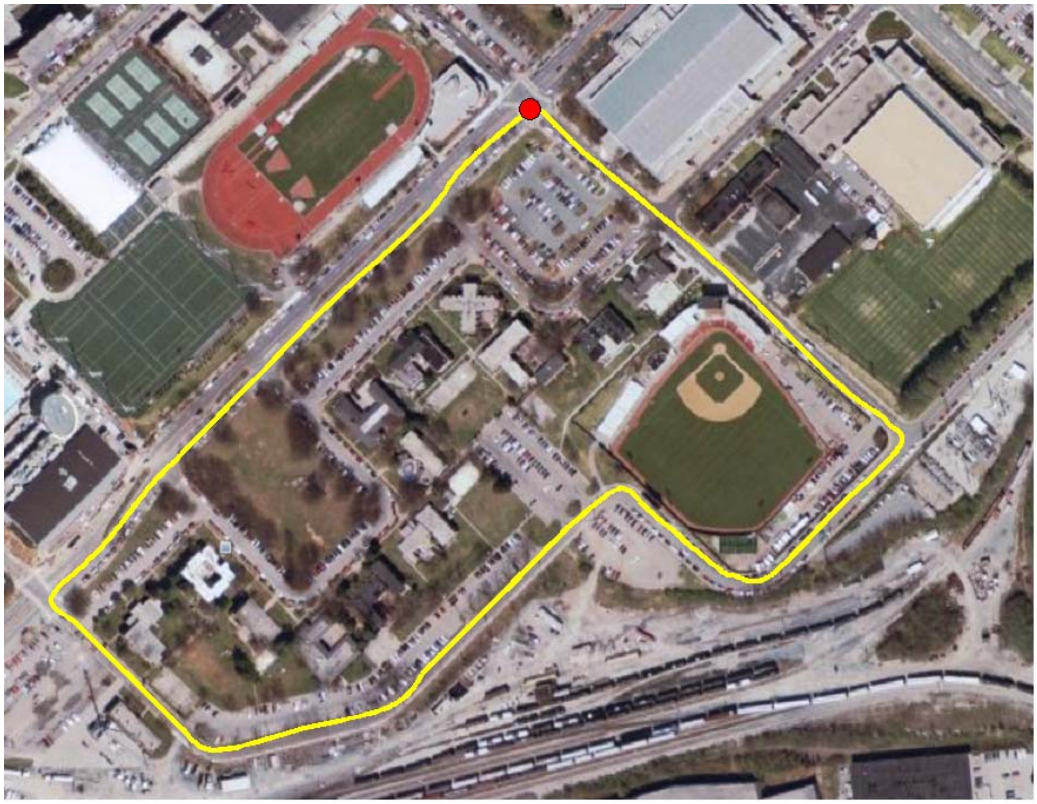


Figure 6.17 Testing environment. The mobile robot travels clockwise around a large loop with total path length 1.6km. The red dot in top marks the starting point.

Table 6.8 Storage efficiency

Total number of images	374
Average SIFT per image	1,878
Total detected SIFT features	702,118
Total detected landmark objects	621
Average SIFT per object	148
Total SIFT descriptors stored in database	91,908

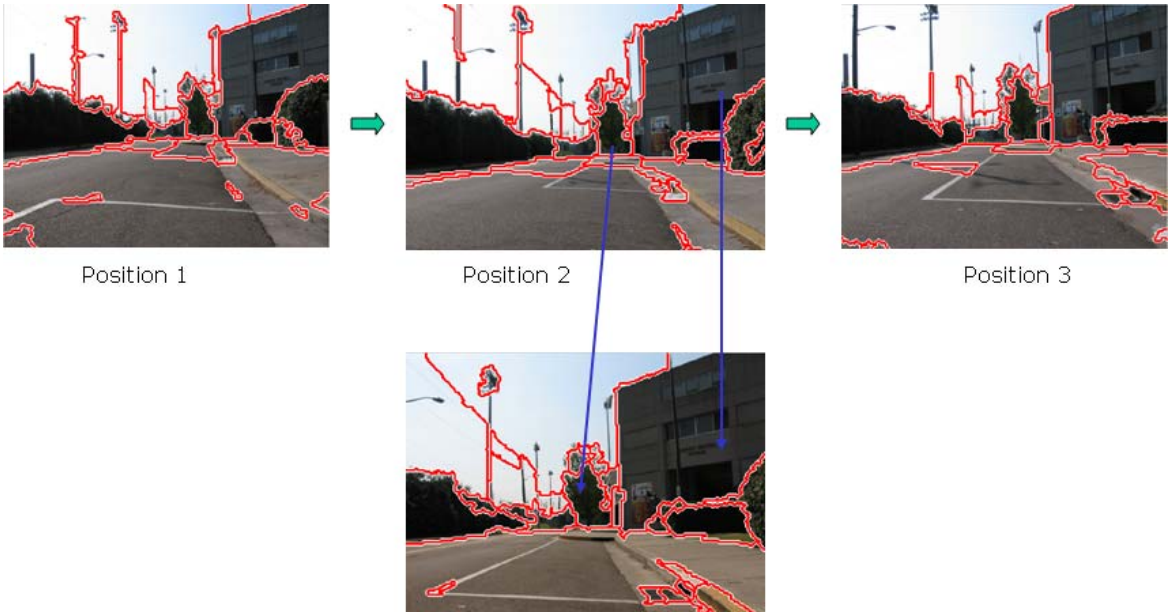


Figure 6.18 An Example of loop closing. Upper: a sequence of images taken from the first trip. Down: An image taken from the second trip. A loop is detected because the building and the tree in the bottom image are matched with the corresponding objects in top images (This figure is best viewed in color).

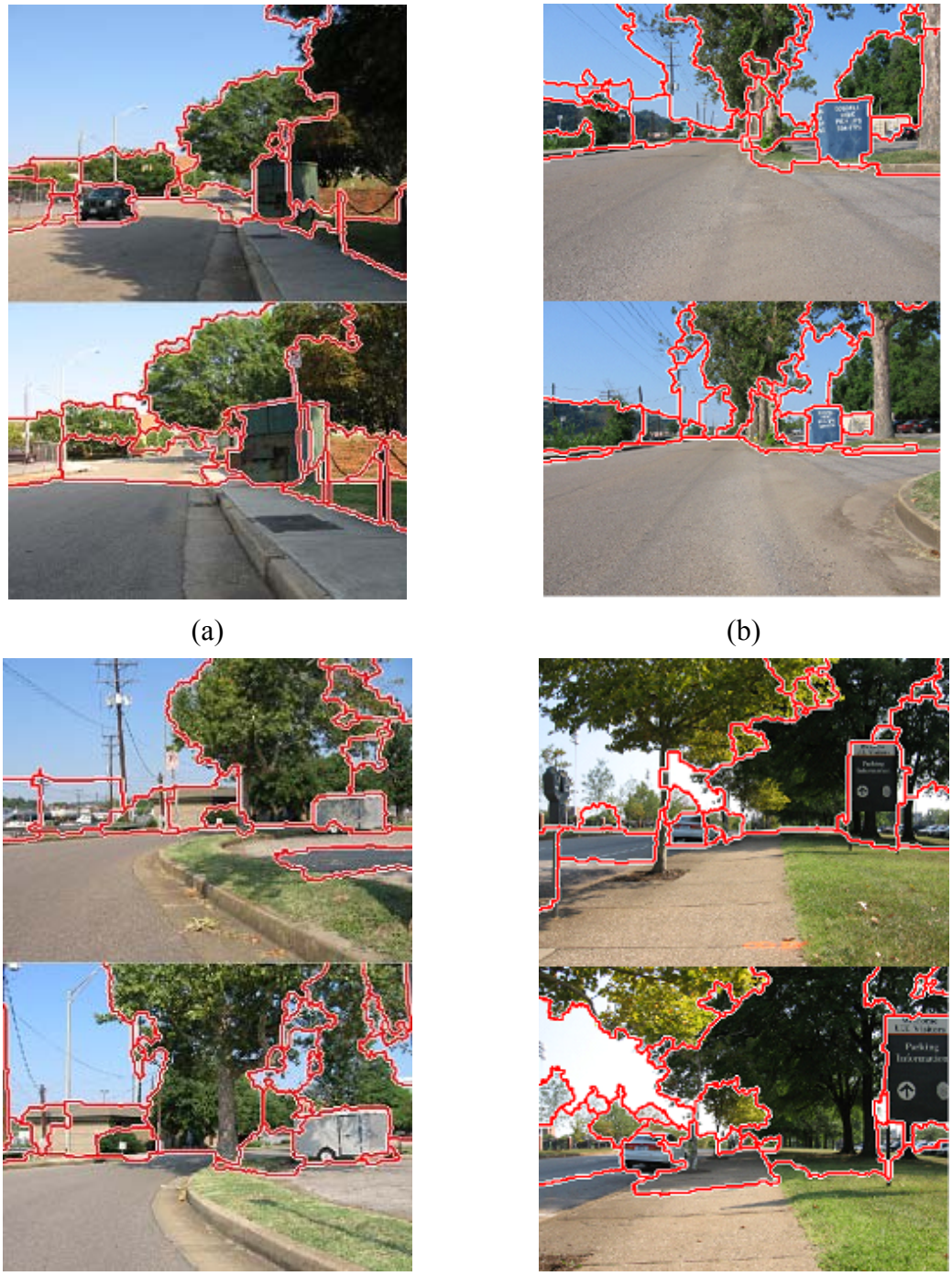


Figure 6.19. Examples of loop detection under scene changes. (a), (b), (c), (d) present a pair of images taken from a same place respectively. For each pair of images, the top image was taken in the first trip along the loop. The bottom image was randomly taken from 50 places in the second trip along the loop. Notice that there exist some scene changes between each pair of images caused by different background vegetation and different illumination. (This figure is best viewed in color.)

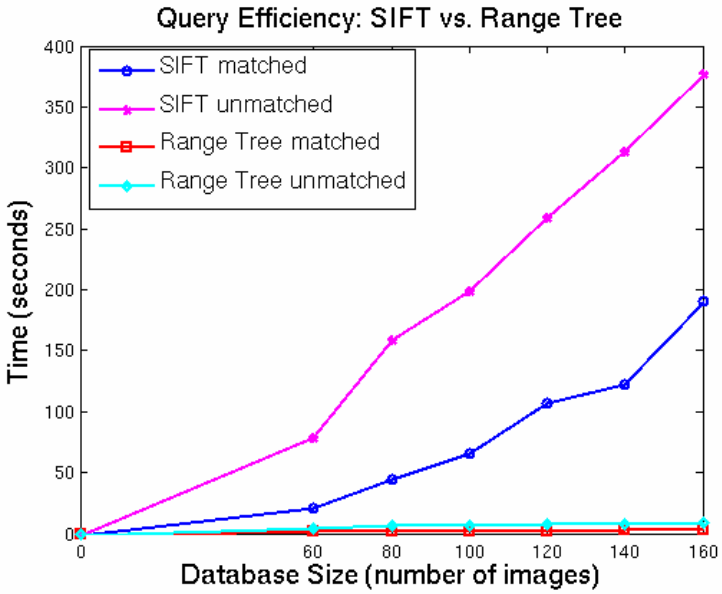


Figure 6.20 Query efficiency: SIFT database vs. range tree

7 Conclusion

7.1 Summary of contribution

This dissertation is motivated by the need of reliable place recognition in large complex environments for autonomous mobile robotics. We notice that human is capable of recognizing places with ease even in large complex environments. Many psychological works support that human perceives a scene based on the perception of objects. Instead of creating a detailed representation of all the objects in a scene, human visual system builds an economic scene representation by putting emphasis on the extraction of a few key 'aspects' of the scene information, such as an inventory of salient objects and the layout of these objects, etc. This economic representation results in an enormous saving of processing and memory resources, which plays a key role for the success of human visual system on place recognition. Therefore, we proposed an object-based place recognition and loop closing method which works in a similar way to human visual system.

In chapter 3, we first developed a novel image segmentation algorithm. The image segmentation algorithm is based on a Perceptual Organization model. The Perceptual Organization model can 'perceive' the special structural relations among the constituent parts of an unknown object and hence can group them together without object-specific knowledge. The image segmentation algorithm allows us to 'perceive' the salient objects in a place.

In chapter 4, we present a novel method for object classification. Our main contributions are two-fold. Firstly, we build an informative object description, which consists of not only appearance, but also certain structural information like parts layout and shape of objects. All these information are combined into a high-dimensional vector. Secondly, we develop a new information-based wrapper feature selection method. Feature selection refers to search algorithms that select a subset of most characterizing features from an initial larger set of features. For many pattern recognition applications, feature selection is critic to minimize the classification error, especially when the original feature set is very large. With this feature selection method, for each object class, we can find a small subset of features that well characterize the object class.

To achieve efficient scene-matching, in chapter 5, we first selected a subset of the salient objects in a scene as landmark objects to label the place. The landmark objects are highly distinctive and widely visible. Each landmark object is represented by a list of SIFT descriptors extracted from the object surface. This object representation allows us to reliably

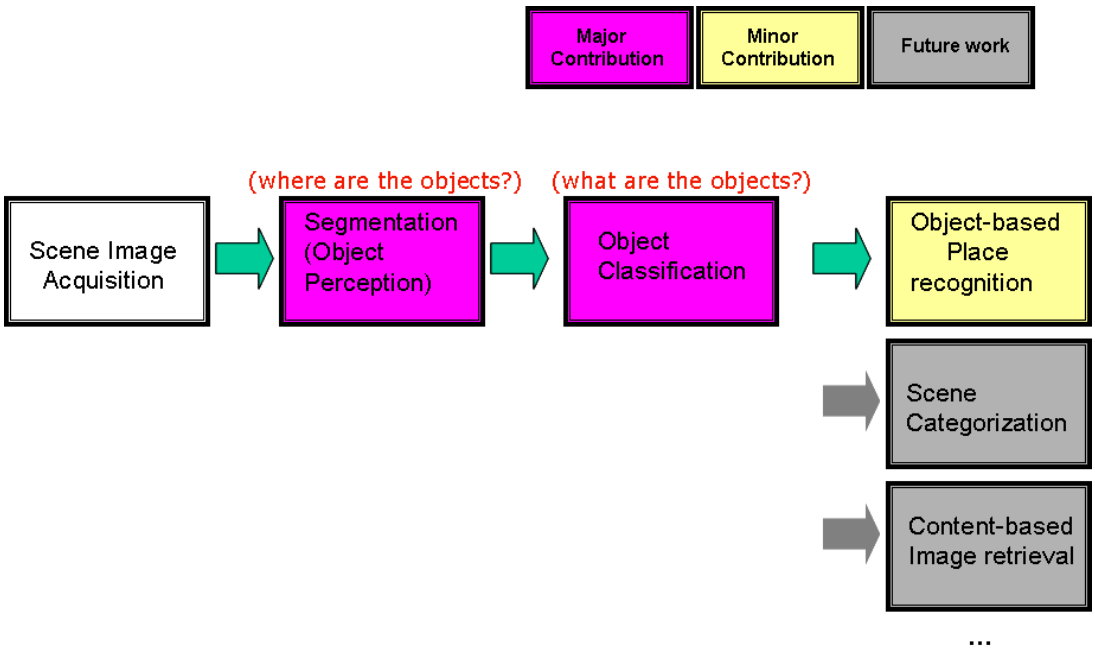


Figure 7.1 Summary of contributions.

recognize an object even under certain viewpoint changes. We then developed an indexing structure. We use both texture feature and color feature of objects as indexing features. Both the texture feature and the color feature are viewpoint-invariant and hence can be used to effectively find the candidate objects with similar surface characteristics to a query object.

The experimental results are presented in chapter 6. The experimental results show that our segmentation method outperforms the other state-of-art reference methods. Especially for the structured objects like buildings and vehicles, our method achieves significantly better results. This shows that our segmentation method can stably segment variant salient objects in different outdoor scenes. We then evaluate our object classification method on the MSRC-21 dataset. The experimental result shows that our method significantly increase the classification accuracy compared to state-of-art methods. This proves that good segmentations can provide more valuable information about object classes to overcome the challenging intra-class variation problem and therefore is the key for object classification. At last, we test the place recognition method on a dataset collected from a 1.6km loop. The experimental result shows that our place recognition method can accurately and efficiently recognize various places in a large complex outdoor environment, even under situation where there exist big scene differences caused by different background vegetations and different illumination conditions.

The pipeline of our contributions is illustrated in Figure 7.1. With this summary of the contributions, we now turn into the future directions for this research.

7.2 Directions for future research

7.2.1 Improvement on the Perceptual Organization model

Image segmentation is one of the fundamental problems for computer vision. Due to the wide variety of object classes, one challenge for outdoor scene segmentation is that one may need to segment an object without recognizing what kind of objects it is. For this reason, we have developed a novel image segmentation method based on Perceptual Organization. We encoded five Gestalt laws into a boundary energy model to form a Perceptual Organization model. Obviously, our implantation of Perceptual Organization does not completely capture the perceptual power of the human mind. Besides the five Gestalt laws we encoded in our Perceptual Organization model, there are many other Gestalt laws discovered in the literature such as *Closure*, *Collinear*, etc. Occlusion is a common problem in outdoor scene segmentation. For an instance, a car or a building may stand behind a tree or people and therefore be divided into separated two parts. In these cases, the *Closure* laws can be applied to handle the occlusion problem. Besides, the implantation of some Gestalt laws in our model can be improved. For instance, we only consider the symmetry between two vertically attached parts in our model. The symmetry along other directions is also needed to be considered in practice. For the *proximity* law, we only consider two vertically attached neighboring parts. There are also many other forms of *proximity* in practice. Therefore, the first major area for future research is to incorporate more Gestalt laws into our Perceptual Organization model meanwhile improve the implementation of some existing Gestalt laws. By doing so, our Perceptual Organization model can be expected to have more 'perceptual power' for handling real world objects.

7.2.2 Object recognition based on semantic parts

We have shown that certain structural information like parts layout and shape of objects can provide valuable information to overcome the challenging intra-class variations for object classification. However, acquiring the structural information (especially for shape) requires object to be well segmented. In practice, this may not be achievable. Although our Perceptual Organization model can piece many fairly attached objects together only based on general geometric statistic knowledge (Gestalt laws), for many objects with complex structures like bicycles, motorcycles, some buildings, etc, the high-level object-specific knowledge is still required to generate perfect segmentations for these objects. For these objects, our simple Perceptual Organization model may not be able to piece the whole object together. Instead, it may only piece the semantic parts like wheels, windows, doors together. Although semantic parts do not contain the same amount of information as the whole object has, they still provide some valuable cues for object classification. For instances, if we recognize a wheel, we may infer that the object may be classified as a bicycle or a motorcycle. If we identify a door, then the object may belong to a building, etc. Since perfect segmentation for the whole objects may not be expected but good segmentation for semantic parts of the objects is achievable, object recognition based on semantic parts may be a more

practical solution. Hence, another avenue for future research is to develop object classification methods based on semantic parts. This will make object classification more robust and reliable in practice.

7.2.3 Place recognition for dynamic environments

A potential problem we found in the initial place recognition and loop closing experiment is that sometimes our method selects some vehicles as landmark objects to label a place. Since the vehicles may move out of a scene in any moment, this may confuse our method and lower the performance of place recognition. So far our method recognizes a place solely based on low-level features of objects, like the colors and textures of object surfaces. To handle dynamic environments where there may exist scene changes, such as some objects may move in a scene and some objects may move out a scene, some high level knowledge is necessary. Therefore a component that we would like to add into the system is the object recognition method developed in Chapter 4. With an object recognition method, we may be able to classify the selected landmark objects into different categories such as buildings, vegetations, vehicles, etc. With the high-level knowledge about the landmark objects, we can assign high weight to permanent landmark objects like buildings and assign low weight to mobile objects like vehicles. This may enhance the strength of our method handling scene changes and make our method more robust.

7.2.4 Other applications

One application of the techniques developed in this dissertation is for the perimeter patrol problem, which usually refers to patrolling around a closed area by a team of robots. The perimeter patrol problem requires robots to visit a target area repeatedly in order to monitor some changes in the state of that area. A scenario is that a team of robots patrol in a large closed area where localization devices like GPS might not receive stable signals at many places, which is often the case in many urban areas. In this scenario, the robots need a reliable localization solution for them to decide which region they are monitoring. Our method can provide the robot with this kind of capability. Assuming that before patrolling, each robot has traveled around the close area once and built a landmark objects database like the one in Chapter 5. Then during patrolling process, at any place, a robot can accurately decide which region it is monitoring solely based on the objects it detects in the scene. This relieves the difficulty of unreliable localization in large environments, which is a common problem for any mobile robotic system. Besides, our method can enhance the capability of robots for scene changes detection. Since it is a common knowledge that surveillance cameras can only detect moving objects in a scene, one strategy an intruder might take is stopping moving when he find a patrolling robot is approaching. This will make a robot difficult to tell if the intruder belongs to a static background object or an intruded object. This strategy cannot work when the patrolling robot incorporates our object-based place recognition system – During place recognition process, if a robot detects an object that has never been seen before at the place, it can signal a scene change alarm. Thus, our object-

based place recognition method can greatly enhance the effectiveness and robustness of the perimeter patrol system.

Other applications include scene categorization, content-based image retrieval, etc. All these applications require well-understanding of an image. Given an image, one needs to know what kind of objects appear in the image and where these objects are in the image to make the decision like if the image contain the query object or the image belong a specific scene category. The image segmentation method and the object classification approach developed in this dissertation can answer the above two questions and therefore make the solution for these applications straightforward.

7.3 Discussion with closing remarks

Object perception forms a foundation for many computer vision tasks. In the first chapter of this dissertation, we illustrate that psychologists believe that people perceive a scene based on the perception of the salient objects in the scene. Firstly the gist and layout are used to prioritize attention, directing it to the objects that are most important in the context. Once an object gets attended, the constituent parts of the object are grouped together and a summary description of the object is formed, like its size, overall shape, dominant colors, etc. In other words, object perception involves a Perceptual Organization process and a following recognition step. The work of this dissertation tries to simulate the object perception procedure. We have developed a Perceptual Organization model by encoding a list of Gestalt laws. By doing so, the Perceptual Organization model can group some constituent parts of objects together without recognizing it. Further, we have developed an object recognition method based on the segments generated by the Perceptual Organization model. Obviously, our implementation of Perceptual Organization does not completely capture the perceptual power of the human mind. But we believe that object perception based on the Perceptual Organization points to a very promising direction. We hope that the concepts presented in this dissertation can build a step towards extending the state of art in computer vision.

Publications

[1] Chang Cheng, A. Koschan, D. L. Page, and M. A. Abidi. “Scene Image Segmentation based on Perceptual Organization,” in Proc. IEEE International Conference on Image Processing ICIP 2009, pp. 1801-1804, November 2009.

[2] Chang Cheng, A. Koschan, and M. A. Abidi, "Object-based Place Recognition and Scene Change Detection for Perimeter Patrol," in *Transactions of the American Nuclear Society*, Vol. 101, pp. 823-824, 2009.

[3] Chang Cheng, D. L. Page, and M. A. Abidi. “Object-based place recognition and loop closing with jigsaw puzzle image segmentation algorithm,” *Proceedings of the 2008 IEEE, International Conference on Robotics and Automation (ICRA 2008)*, pp. 557-562, May 2008.

[4] D. L. Page, Andreas Koschan, Chung-Hao Chen, Chang Cheng, Marcus. Jackson, and Mongi Abidi, “Modular Sensor Bricks and Unmanned Systems for Persistent Large Area Surveillance,” *ANS 2st International Joint Topical Meeting on Emergency Preparedness & Response and Robotics & Remote Systems,* March 2008.

[5] Chung-Hao Chen, Chang Cheng, David Page, Andreas Koschan, and Mongi A. Abidi, “Tracking a Moving Object with Real Time Obstacle Avoidance Capacity,” *International Journal of Industrial Robot, Special Issue on Robot Control and Programming,* Vol. 33, No. 6, pp. 460-468, November 2006.

[6] Chung-Hao Chen, Chang Cheng, David Page, Andreas Koschan, and Mongi A. Abidi, “A moving object tracked by a mobile robot with real-time obstacle avoidance capability,” *The 18th International Conference on Pattern Recognition (ICPR 2006),* Hong Kong, August 2006.

[7] Chung-Hao Chen, Chang Cheng, David Page, Andreas Koschan, and Mongi A. Abidi, “Modular Robotics and Intelligent Imaging for Unmanned Systems,” *Proceedings of SPIE Unmanned Systems Technology VIII,* Vol. 6230, pp. 43-52, USA, April 2006.

[8] Chang Cheng, Chung-Hao Chen, David Page, Andreas Koschan, and Mongi A. Abidi, “Modular Sensor Processing for Robotics-Based Security in Hazardous Environments,” *ANS 1st International Joint Topical Meeting on Emergency Preparedness & Response and Robotics & Remote Systems,* February 2006.

Bibliography

Bibliography

[Abbadeni00] N. Abbadeni, D. Ziou, and S. Wang, "Autocovariance-based Perceptual Textural Features Corresponding to Human Visual Perception," Proceedings of the 15th IAPR/IEEE International Conference on Pattern Recognition, Vol. 3, pp. 901-904, 2000.

[Aliferis03] C.F. Aliferis, I. Tsamardinos, and A. Statnikov. HITON: A novel markov blanket algorithm for optimal variable selection. In *AMIA 2003 Symposium Proceedings*, pp. 21-25, 2003.

[Amit07] Y. Amit and A. Trouve, "POP: Patchwork of parts models for object recognition," *IJCV*, Vol 75, No. 2, pp. 267-282, 2007.

[Bailey00] T. Bailey, E.M. Nebot, J.K. Rosenblatt "Data Association for Mobile Robot Navigation: A Graph Theoretic Approach", Proceeding of the 2000 *IEEE International Conference on Robotics & Automation*, Vol. 3, pp. 2512 - 2517, 2000.

[Bailey01] T. Bailey and E. Nebot, "Localization in large-scale environments," *Robotics and Autonomous Systems*, Vol. 37, no. 4, pp. 261-281, 2001.

[Bailey02] T. Bailey, "Mobile robot localization and mapping in extensive outdoor environments," *Ph.D dissertation*, the University of Sydney, 2002.

[Bimbo99] D. Bimbo, "Visual Information Retrieval," Morgan Kaufmann Publishers, San Francisco, CA, 1999.

[Borenstein85] I. Biederman, "Human image understanding: Recent research and a theory," Computer Graphics, Vision and Image Processing, Vol. 32, pp. 29-73, 1985

[Borenstein91] J. Borenstein and Y. Koren, "The vector field histogram – fast obstacle avoidance for mobile robots," *IEEE Journal of Robotics and Automation*, Vol. 7, pp.278–288, June 1991.

[Borenstein04] E. Borenstein and E. Sharon, "Combining top-down and bottom-up segmentation," In Proc. *CVPR* 2004.

[Bosse04] M. Bosse, P. Newman, J. J. Leonard, and S. Teller. "SLAM in large scale cyclic environments using Altlas framework," *International Journal of Robotics Research*, 23(12): 1113-1139, Dec 2004.

[Brinkhoff95] T. Brinkhoff, H.P. Kriegel, R. Schneider, A. Braun. "Measuring the complexity of polygonal objects." *Proc. Of the Third ACM International Workshop on Advances in Geographical Information System*, pp.109-117, 1995.

[Bruce90] V. Bruce and P. Green "Visual perception: Physiology, Psychology and Ecology," Lawrence Erlbaum Associates Ltd., 1990.

[Buhmann95] J. Buhmann, W. Burgard, A.B. Cremers, D. Fox, T. Hofmann, F. Schneider, J. Strikos, and S. Thrun, "The mobile robot Rhino," *AI Magazine*, Vol. 16, 1995.

[Buker00] U. Buker, G. Hartmann, “Object representation: on combining viewer-centered and object-centered elements,” Pattern recognition, Vol. 1, pp. 956-959, 2000.

[Burgard96] W. Burgard, D. Fox, D. Hennig and T. Schmidt, “Estimating the absolute position of a mobile robot using position probability grids”, In Proceedings of the *Thirteenth National Conference on Artificial Intelligence*, 1996.

[Burgard99] W. Burgard, A.B. Cremers, D. Fox, D. H¨ahnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, “Experiences with an interactive museum tour-guide robot,” *Artificial Intelligence*, Vol. 114(1-2), pp. 3–55, 1999.

[Castellanos99] J. A. Castellanos, J. M. M. Montiel, J. Neira, and J. D. Tardos, “The SPmap: A probabilistic framework for simultaneous localization and map building,” *IEEE Trans. Robot. Automat*., Vol. 15, pp. 948-953, Oct. 1999.

[Castellanos01] J.A. Castellanos, J. Neira and J.D. Tardos, “Multisensor Fusion for Simultaneous Localization and Map Building”, *IEEE Trans. Robot. Automat*, December 2001.

[Chan01] T.F. Chan and L.A. Vese, “Active Contours without Edges,” IEEE Trans. On Image processing, 10(2), pp. 266-277, 2001.

[Cheng08] C. Cheng, D. L. Page, and M. A. Abidi. “Object-based place recognition and loop closing with jigsaw puzzle image segmentation algorithm,” *Proceedings of the 2008 IEEE, International Conference on Robotics and Automation*, pp. 557-562, May 2008

[Cheng09] C. Cheng, A. Koschan, D.L. Page, and M.A. Abidi, "Scene Image Segmentation Based on Perception Organization," In *Proc*. *ICIP*, pp. 1801-1804, 2009.

[Choset96] H. Choset and J.W. Burdick, “Sensor Based Planning: The Hierarhical Generalized Voronoi Graph,” In *Proc. Workshop on Algorithmic Foundations of Robotics*, Toulouse, France, 1996.

[Choset01] H. Choset and K. Nagatani, “Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization,” IEEE Trans. Robot. Autom., vol. 17, no. 2, pp. 125-137, Apr., 2001.

[Comaniciu02] D. Comaniciu and M. Peter, "Mean Shift**:** A Robust Approach Toward Feature Space Analysis". *PAMI,* 24(5), pp. 603-619, *2002.*

[Cortelazzo94] G. Cortelazzo, “Trademark shapes description by string-matching techniques,” Pattern Recognition, Vol. 27 pp. 1005–1018, 1994.

[Crandall05] D. Crandall, P. Felzenszwalb, and D. Huttenlocher, “Spatial priors for part-based recognition using statistical models,” In *CVPR*, pp. 10-17, 2005.

[Csorba97] M. Csorba, “Simultaneous Localisation and Map Building,” PhD thesis, University of Oxford, 1997.

[Csurka04] G. Csurka , C. Dance, J. Willamowski, L. Fan and C. Bray, "Visual categorization with bags of keypoints," in ECCV International Workshop on Statistical Learning in Computer Vision, 2004.

[Cour05] T. Cour, F. Benezit, and J. Shi. “Spectral segmentation with multiscale graph decomposition.” CVPR, 2005.

[Cummins07] M. Cummins and P. Newman, “Probabilistic appearance based navigation and loop closing,” *IEEE international conference on Robotics and Automation*, April 2007.

[Dellaert99] F. Dellaert, D. Fox, W. Burgard and S. Thrun, "Monte Carlo Localization for Mobile Robots," Proc. of the *IEEE International Conference on Robotics and Automation* , 1999.

[Desolneux08] A. Desolneux, L. Moisan, J. M. Morel, "From Gestalt theory to image analysis – a probabilistic method," Springer, 2008.

[Dissanayake01] G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions of Robotics and Automation*, 2001.

[Do02] M. N. Do, and M. Vetterli, "Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance," *IEEE Trans Image Process* Vol. 11, no. 2, pp. 146-158, 2002.

[Dollar06] P. Dollar, Z.W. Tu, and S. Belongie, " Supervised learning of edges and object boundaries," In Proc. CVPR, Vol 2, pp. 1964-1971, 2006.

[Durrant-Whyte01] H. Durrant-Whyte, S. Majumder, S. Thrun, M. de Battista, and S. Scheding, "A Bayesian algorithm for simultaneous localization and map building," In *Proceedings of the 10th International Symposium of Robotics Research (ISRR'01)*, 2001.

[Engelson92] S. Engelson and D. McDermott, "Error correction in mobile robot map learning," In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2555–2560, May 1992.

[Enns92] J.T. Enns and R.A. Rensink, "An object completion process in early vision," *Vision Research*, Vol 33, pp. 1263, 1992.

[Elder93] J.H. Elder and S.W. Zucker, "The effect of contour closure on the rapid discrimination of tow-dimensional shapes," *Vision Research*, Vol 33, pp. 981-991, 1993.

[Epshtein07] B. Epshtein and S. Ullman, "Semantic hierarchies for recognizing objects and parts," In *CVPR*, 2007.

[Estrada05] F.J. Estrada and A.D. Jepson, "Quantitative evaluation of a novel image segmentation algorithm," In Proc. CVPR, Vol 2, pp. 1132-1139, 2005.

[Everingham07] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman., "The PASCAL visual object classes challenge 2007 (VOC2007) results," 2007.

[Feder99] H. J. S. Feder, J. J. Leonard, and C. M. Smith, "Adaptive mobile robot navigation and mapping," *Int. J. Robot. Res.*, vol. 18, no.7, pp. 650-668, 1999.

[Feldman85] J. A. Feldman, "Four frames suffice: A provisional model of vision and space," *Behavioral andBrain Sciences.*, vol. 8, pp. 265-289, 1985.

[Felzenszwalb04] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation,", *IJCV*, vol. 59, no. 2, 2004.

[Felzenszwalb05] P. Felzenszwalb and D. Huttenlocher, "Pictorial structures for object recognition,", *IJCV*, vol. 61, no. 1, pp. 55-79, 2005.

[Felzenszwalb08] P. Felzenszwalb, D. Mcallester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *IEEE International Conference on Computer Vision and Pattern Recognition* (CVPR), June 2008.

[Fergus03] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. CVPR*, pp. 264-271, 2003.

[Friedman00] J. Friedman, T. Hastie and R. Tibshirani, Additive logistic regression: A statistical view of boosting. *The Annals of Statistics,* 38(2):337-374, 2000.

[Fulkerson09] B. Fulkerson, A. Vedaldi, and S. Soatto, "Class Segmentation and Object Localization with Superpixel Neighborhoods," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2009.

[Geusebroek00] J.-M. Geusebroek, R. van den Boomgaard, A.W.M. Smeulders, F. Cornelissen, and H. Geerts, "Color and scale: the spatial structure of color images," *Porc. Sixth European Conference Computer vision*, Vol. 1, pp. 331-341, 2000.

[Geusebroek01] J.-M. Geusebroek, R. van den Boomgaard, A.W.M. Smeulders, H. Geerts, "Color invariance ," *IEEE Transaction on pattern analysis and machine intelligence*, Vol. 23, pp. 1338-1350, 2001.

[Gevers99] T. Gevers and A. W. M. Smeulders, "Color based object recognition," *Pattern Recognition*, Vol 32, pp. 453-464, 1999.

[Gonzalez87] R.C. Gonzalez, P. Wintz, "Digital Image Processing, second ed.," Addison-Wesley, Reading, MA,1987.

[Gorelick05] L. Gorelick, M. Galun, E. Sharon, R. Basri, and A. Brandt, Shape Representation and Classification Using the Poisson Equation. *PAMI,* 28(12):1991-2005, 2005

[Gould08] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller, "Multi-class segmentation with relative location prior," *IJCV*, 2008.

[Gould09] S. Gould, R. Fulton and D. Koller, "Decomposing a scene into Geometric and sematically consistent regions," in *Proc. ICCV*, 2009.

[Gould09b] S. Gould, O. Russakovsky, I. Goodfellow, P. Baumstarck, A. Y. Ng and D. Koller, "The STAIR Vision Library (v2.3)," http://ai.stanford.edu/~sgould/svl, 2009.

[Grady06] L. Grady. "Random Walks for Image Segmentation," *IEEE Transaction on pattern analysis and machine intelligence*, vol. 28, No. 11, Nov 2006.

[Grauman05] K. Grauman andT. Darrell, "Efficient image matching with distributions of local invariant features," In *CVPR*, Vol. 2, pp. 627-634, 2005.

[Grauman05] K. Grauman andT. Darrell, "Pyramid match kernels: Discriminative classification with sets of image features," In *IJCV*, Vol. 2, pp. 1458-1465, 2005.

[Griffin07] G. Griffin, A. Holub and P. Perona, "Caltech-256 object category dataset," Technical report, Caltech, 2007.

[Guivant01] J. Guivant and E. Nebot, "Optimization of the simultaneous localization and map building algorithm for real time implementation," *IEEE Transaction of Robotic and Automation*, May 2001.

[Guruswami99] V. Guruswami and A. Sahai. Multiclass learning, boosting, and error-correcting codes. In Proc. 12th Annual Conf. Computational Learning Theory, pp. 145-155, 1999.

[Gutmann99] J. S. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," In *International Symposium on computational Intelligence in Robotics and Automation*, 1999.

[Guzzoni97] D. Guzzoni, A. Cheyer, L. Julia, and K. Konolige, " Many robots make short work," *AI Magazine*, Vol. 18, pp.55-64, 1997.

[Hafner95] J. Hafner, H.S. Sawhney, W. Equitz, M. Flickner, W. Niblack, "Efficient color histogram indexing for quadratic form distance functions," IEEE Trans. PAMI. Vol. 17, no. 7, pp. 729–736, 1995.

[He04] X. He, R. Zemel, and M. Carreira-Perpinan, "Multiscale CRFs for image labeling," in *Proc. CVPR*, pp. 695-702, 2004.

[Hebert95] M. Hebert, J. Ponce, T. Boult, A. Gross, "Object Representation in Computer Vision," Berlin (Springer), 1995.

[Hoffman97] D.D Hoffman, and M. Singh, " Salience of Visual Parts," Cognition Vol. 63, pp. 29-78, 1997.

[Hoiem05] D. Hoiem, "Geometric context from a single image", In Proc. *ICCV*, 2005.

[Hoiem07] D. Hoiem, A.A. Efros, and M. Hebert. "Recovering surface layout from an image," *IJCV*, 75(1):151-172, 2007.

[Hoiem07] D. Hoiem, A.N. Stein, A.A. Efros, and M. Hebert. "Recovering Occlusion Boundaries from a Single Image," *ICCV* 2007.

[Huang97] J. Huang, S. R. Kumar, M. Mitra, W.J. Zhu, and R. Zabih , "Image indexing using color correlograms," *CVPR*, pp. 762–768, 1997.

[Irwin96] D.E. Irwin. "Integrating information across saccadic eye movements," *Currrent Directions in Psychological Science,* Vol. 5, pp. 94-100, 1996.

[Jacobs96] D.W. Jacobs, "Robust and efficient detection of salient convex groups," IEEE Trans. PAMI, 18(1), pp 23-37, 1996.

[Jacobs03] D.W. Jacobs, "What makes viewpoint-invariant properties perceptually salient?" Journal of the optical society of America, A, Optics, image, science and vision, 20(7), pp 1304-1320, 2003.

[Jain00] A.K. Jain, and R.P.W. Duin, and J. Mao, "Statistical Pattern Recognition: A Review," PAMI, vol. 22, no. 1, pp. 4-37, 2000.

[Jefferies05] M. E. Jefferies, W. Yeap, M. C. Cosgrove and J. T. Baker "Using absolute metric maps to close cycles in a topological map," Journal of Intelligent Manufacturing, Vol.16, pp. 693-702, 2005.

[Jermyn01] I.H. Jermyn, and H. Ishikawa, "Globally optimal regions and boundaries as minimum ratio weight cycles," IEEE Trans. PAMI, 23(10):1075-1088, 2001.

[Kadir01] T. Kadir and M. Brady. "Saliency, scale and image description," *International Journal of Computer Vision*, 45(2):83-105, 2001.

[Kalman60] R. E. Kalman. A new approach to linear filtering and prediction problems. *Trans. ASME, Journal of Basic Engineering*, Vol. 82, pp. 35–45, 1960.

[Kortenkamp94] D. Kortenkamp and T.Weymouth, "Topological mapping for mobile robots using a combination of sonar and vision sensing," In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pp. 979-984, 1994.

[Kuipers91] B. Kuipers and Y.T. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," *Journal of Robotics and Autonomous Systems*, Vol. 8, pp.47–63, 1991.

[Lamon01] P. Lamon, I. Nourbakhsh, B Jensen and R Siegwart, "deriving and matching image fingerprint sequences for mobile robot localization," *Robotics and Automation*, 2001. Proceedings 2001 ICRA.

[Lazebnik04] S. Lazebnik, C. Schmid, and J. Ponce. "Semi-local affine parts for object recognition," In *British Machine Vision Conference,* Vol. 2, pp. 959-968, 2004.

[Lazebnik05] S. Lazebnik, C. Schmid, and J. Ponce, "A maximum entropy framework for part-based texture and object recognition," In *Proc. ICCV*, pp. 832-838, 2005.

[Lazebnik06] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," In *Proc. CVPR*, Volume 2, 2006 pp. 2169 – 2178, 2006.

[Leonard92] J.J. Leonard, H.F. Durrant-Whyte, and I.J. Cox, "Dynamic map building for an autonomous mobile robot," *International Journal of Robotics Research*, Vol. 11, pp. 89–96, 1992.

[Leonard99] J.J. Leonard and H.J.S. Feder, "A computationally efficient method for large-scale concurrent mapping and localization," *Proceedings of the Ninth International Symposium on Robotics Research*, 1999.

[Leonard 03] J. J. Leonard, Paul M. Newman and Richard J. Rikoski, "Towards Robust Data Association and Feature Modeling for Concurrent Mapping and Localization", *Robotics Research*, Vol. 6, pp. 7-20, 2003.

[Leung01] T. Leung , and J. Malik, "Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons," *Int J Comput Vis,* Vol. 43*,* n. 1, pp. 29-44, 2001.

[Lowe85] D. Lowe, "Perceptual organization and visual recognition," The Netherlands: Kluwer Academic Publishers, 1985.

[Lowe04] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, No. 2, pp. 91-110, Nov.2004.

[Liu99] Z.L. Liu, D. W. Jacobs, R. Basri, "The role of convexity in perceptual completion: beyond good continuation," Vision research, Vol. 39, pp. 4244-4257, 1999.

[Lu99] G. Lu, A. Sajjanhar, "Region-based shape representation and similarity measure suitable for content-based image retrieval," *Multimedia Systems,* Vol.7 , pp. 165–174, 1999.

[Lyu05] S. Lyu, "Mercer kernels for object recognition with local features," In *CVPR*, Vol. 2, pp. 223-229, 2005.

[Mahamud03] S. Mahamud, L.R. Williams, K.K. Thornber and K. Xu, "Segmentation of Multiple Salient Closed Contours from Real Images," *IEEE Trans. PAMI*, 25(4), pp. 433-444, 2003.

[Malik01] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *IJCV*, 43(1):7-27, 2001.

[Malisiewicz07] T. Malisiewicz and A.A. Efros. "Improving Spatial Support for Objects via Multiple Segmentations," BMVC, 2007.

[Manjunath01] B. S. Manjunath*,* J.-R. Ohm*,* V. V. Vasudevan*,* and A. Yamada, "Color and texture descriptor," *IEEE Trans. ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, Vol. 11, No. 6, pp. 703-715, 2001.

[Matas02] J. Matas, O.Chum, M.Urban, and T. Pajdla. "Robust wide baseline stereo from maximally stable extremal regions,",*Proceedings of the British Machine Vision Conference*, 2002.

[Margaritis99] D. Margaritis and S. Thrun, "Bayesian network induction via local neighborhoods," In *Advances in Neural Information Processing Systems 12 (NIPS)*, 1999.

[Marr82] D. Marr, "Vision," W.H. Freeman and Co., San Francisco, 1982.

[Martin02] D.R. Martin, "An empirical approach to grouping and segmentation," Ph.D dissertation, U.C. Berkeley, 2002.
[Margaritis99] Margaritis and S. Thrun, "Bayesian network induction via local neighborhoods," In proc. Of NIPS, 1999.

[Martin04] D.R. Martin, C.C. Fowlkes, and J. Malik, " Learning to detect natural image boundaries using local brightness, color, and texture cues," IEEE Trans. PAMI, 26(5) pp. 530-549, 2004.

[Maybeck79] P. Maybeck, "Stochastic Models, Estimation, and Control, Volume 1," Academic Press, Inc, 1979.

[McCafferty90] J.D. McCafferty, "Human and machine vision: computing perceptual organization," Ellis Horwood: West Sussex England, 1990.

[Micusik09] B. Micusik, and J. Kosecka, "Semantic segmentation of street scenes by superpixel co-occurrence and 3D geometry," *IEEE Workshop on Video-Oriented Object and Event Classification* (VOEC), 2009.

[Mikolajczyk03] K. Mikolajczyk and C. Schmid. "A Performance Evaluation of Local Descriptors,"*In IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 257-264, 2003.

[Mikolajczyk05] K. Mikolajczyk, T. Tuytelaars, C. Schmid and A. Zisserman, "A Comparison of Affine Region Detectors," *International Journal of Computer Vision*, Volume 65, Number 1/2 – 2005.

[Mohan92] R. Mohan, and R. Nevatia, R. " Perceptual organization for scene segmentation and description," IEEE Trans. PAMI, Vol 14, No. 6, pp. 616-635, 1992.

[Montemerlo01] M. Montemerlo, S. Thrum, D. Koller and B. Wegbreit, "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem," Proceedings of the AAAI National Conference on Artificial Intelligence, 2001.

[Moore98] C.M. Moore, S. Yantis, and B. Vaughan, "Object-based visual slection: evidence from perceptual completion," *Psychological Science*, Vol 9, No. 2, pp. 104-110, 1998.

[Mundy92] J.L. Mundy, A. Zisserman, "Geometric Invariance in Computer Vision", MIT Press, Cambridge, MA, 1992.

[Neria01] J. Neria and J. Tardos, "Data association in stochastic mapping using the joint compatibility test," IEEE *Trans. Robot. Automat*., Vol. 17, pp. 890-897, Dec. 2001.

[Newman00] P. Newman, "On the Structure and Solution of the Simultaneous Localisation and Map Building Problem," PhD thesis, University of Sydney, 2000.

[Newman05] P. Newman and K. Ho, "SLAM-Loop Closing with Visually Salient Features," *Proceedings of the 2005 IEEE, International Conference on Robotics and Automation*, April 2005.

[Opelt04] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer, "weak hypotheses and boosting for generic object detection and recognition," In *ECCV*, pp. 727-734, 2004.

[Pantofaru08] C. Pantofaru, C. Schmid, and M. Hebert, "Object recognition by integrating multiple image segmentations," In *proc. ECCV*, 2008.

[Pierce94] D. Pierce and B. Kuipers. Learning to explore and build maps. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pp. 1264-1271, 1994.

[Peng05] H.C. Peng, F.H. Long, and C. Ding. Feature selection based on mutual information: criteria of Max-dependency, Max-relevance and Min-redundancy. *PAMI* 27(8):pp. 1226-1238, 2005.

[Perez99] J. A. Perez, J. A. Castellanos and J. D. Tardos, "Continuous Mobile Robot Localization: Vision vs. Laser", Proceedings of the *IEEE international Conference on Robotics & Automation*, Vol. 4, pp. 10-15, 1999.

[Plataniotis00] K. Plataniotis and A. Venetsanopoulos, "Color image processing and applications," Springer, Ch.1 pp268-269, 2000.

[Prasad04] B. G. Prasad, K. K. Biswas, S. K. Gupta, "Region-based image retrieval using integrated color, shape, and location index," *Computer Vision and Image Understanding*, Vol 94, Issues 1-3, pp. 193-233, 2004.

[Ravishankar99] K.C. Ravishankar, B.G. Prasad, S.K. Gupta, K.K. Biswas, "Dominant color region based indexing technique for CBIR," Proc. Internat. Conf. on Image Analysis and Processing, pp. 887–892, 1999.

[Ren03] X. Ren, "Learning a classification model for segmentation", In Proc. *ICCV*, 2003.

[Ren08] X.F. Ren, C.C. Fowlkes, J. Malik, "Learning Probabilistic Models for Contour Completion in Natural Images," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 47-63, 2008.

[Ronald00] A.R. Ronald, "The dynamic representation of scenes," *Visual Cognition*, vol. 7, pp. 17-42, 2000.

[Rufin03] V. Rufin and C. Koch, "Competition and selection during visual processing of natural scenes and objects," *Journal of Visoin*, vol. 3, pp. 75-85, 2003.

[Russell06] B.C. Russell, "Using Multiple Segmentations to discover objects and their extent in image collections," In Proc. *CVPR*, 2006.

[Russell05] B.C. Russell, A. Torralba, K.P. Murphy, and W.T. Freeman. "Labelme: a database and web-based tool for image annotation." *Technical report*, MIT, 2005.

[Rutishauser04] U. Rutishauser, D. Walther. "Is bottom-up attention useful for object recognition?" in *Proc. CVPR*, 2004.

[Sala06] P. Sala, R. Sim, A. Shokoufandeh, and S. Dickinson, "Landmark selection for vision-based navigation", *IEEE Trans. On Robotics*, Vol. 22, pp. 334-349, April 2006.

[Samet06] H. Samet. "Foundations of multidimensional and metric data structures," Morgan Kaufmann Publishers, 2006.

[Savarese06] S. Savarese, J. Winn and A. Criminisi, "Discriminative Object class models of appearance and shape by correlatons," In *CVPR*, pp. 2033-2040, 2006.

[Schneiderman04] H. Scheiderman and T. Kanade, "Object detection using the statistics of parts," *IJCV*, Vol. 56, No. 3, pp. 151-177, 2004.

[Schumitsch05] B. Schumitsch, S. Thrun, G. Bradski, and K. Olukotun. "The information-form data association filter," In *Proceedings of Conference on Neural Information Processing Systems (NIPS)*, Cambridge, MA, 2005. MIT Press.

[Se02] S. Se, D. Lowe and J. Little, "Mobile Robot Localization And Mapping with Uncertainty using Scale-Invariant Visual Landmarks," *The International Journal of Robotics Research*, 2002.

[Se05] S. Se, and D. G. Lowe, "Vision-based global localization and mapping for mobile robots," *IEEE Transactions on robotics*, vol 21. No. 3, June 2005.

[Shatkay97] H Shatkay and L. Kaelbling, "Learning topological maps with weak local odometric information," In *Proceedings of IJCAI-97*. IJCAI, Inc., 1997.

[Shotton08] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *proc. CVPR*, pp. 1-8, 2008.

[Shotton09] J. Shotton, J.Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 81(1):2–23, 2009.

[Shi00] J.B. Shi, and J. Malik, " Normalized cuts and image segmentation," IEEE. Trans. PAMI, 22(8), pp. 888-905, 2000.

[Shishir08] K.S. Shishir, "Performance modeling and algorithm characterization for robust image segmentation," IJCV, doi: 10.1007/s11263-008-0130-z, 2008.

[Simons96] D. J. Simons and D. T. Levin, "Change blindness," *Trends in Cognitive Sciences*, vol. 1, pp. 261-267. 1996.

[Simons03] D. J. Simons and D. T. Levin, "What makes change blindness interesting," In *The Psychology of Learning and Motivation*, Eds. D. E. Irwin and B. H. Ross, Academic Press, San Diego, CA, vol. 42, pp. 295-322. 2003.

[Smith88] R. Smith, M. Self and P. Cheeseman, "A stochastic map for uncertain spatial relationships," in *4th Int. Symp. Robotics Research*, O. Faugeras and G. Giralt, Eds., pp. 467-474, 1988.

[Smith90] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," *Autonomous Robot Vehnicles*, pp. 167–193, 1990.

[Stricker96] M. Stricker, A. Dimai, "Color indexing with weak spatial constraints," Proc. SPIE, Storage Retrieval Still Image Video Databases IV 2670 pp.29–40, 1996.

[Su06] H. Su, A. Bouridane, and D. Crookes. "Scale Adaptive Complexity Measure of 2D shapes." *ICPR*, 2006.

[Sujan05] V. A. Sujan and S. Dubowsky, "Efficient Information-based Visual Robotic Mapping in Unstructured Environments", The *International Journal of Robotics Research*, April 2005

[Theoharatos06] C. Theoharatos, V.K. Pothos, N. A. Laskaris, G. Economou, and S. Fotopoulos, "Multivariate Image Similarity in the Compressed Domain using Statistical Graph Matching," *Pattern Recognit,* Vol. 39*,* pp. 1892-1904, 2006.

[Thrun98] S. Thrun, D. Fox, and W. Burgard, "A probabilistic approach to concurrent mapping and localization for mobile robots," *Machine Learning*, Vol. 31, pp. 29–53, 1998.

[Thrun98] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

[Thrun00] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. H¨ahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "Probabilistic algorithms and the interactive museum tour-guide robot Minerva," *International Journal of Robotics Research*, Vol. 19, pp. 972–999, 2000.

[Thrun00] S. Thrun, W. Burgard, and D. Fox, "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping," In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2000.

[Thrun01] S. Thrun, "An probabilistic online mapping algorithm for teams of mobile robots," *Int. J. Robotics Research*, Vol 20. pp. 335-363, May 2001.

[Thrun02] S. Thrun, "Robotic Mapping: A Survey", In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.

[Thrun05] S. Thrun, W. Burgard, and D. Fox." Probabilistic Robotics," MIT Press, Cambridge, MA, 2005.

[Tomatis03] N. Tomatis, I. Nourbakhsh and R. Siegwart, "Hybrid Simultaneous Localization and Map Building: A Natural integration of Topological and Metric," *Robotics and Autonomous Systems* 44 (2003) 3–14.

[Trehub91] A. Trehub, "The cognitive brain," *Cambridge, MA: MIT PRESS.* 1991.

[Trehub94] A. Trehub, "What does calibration solve?" *Behavioral and Brain Sciences,* Vol.17, pp. 279-280, 1994.

[Tsamardinos03] I. Tsamardinos, CF. Aliferis. Towards principled feature selection: relevancy, filters and wrappers. *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics* AI & Stats, 2003.

[Ullman96] S. Ullman, "High-level Vision," MIT Press, Cambridge, MA, 1996.

[Ullman01] S. Ullman, E. Sali, and M. Vidal-Naquet. A fragment-based approach to object representation and classification. In *Proc. 4th Intl. Workshop on visual Form. IWVF4*, 2001

[Unnikrishnan05] R. Unnikrishnan, C. Pantofaru, and M. Hebert, (2005), "A measure for objective evaluation of image segmentation algorithm," In Proc. CVRP, Vol 3, pp 34-41, 2005.

[Varma05] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *IJCV*, 62(1-1):81-81, 2005.

[Vasselle93] B. Vasselle and G. Giraudon, "2-D digital curve analysis: a regularity measure," Proc. Of IEEE ICCV, pp.556-561, 1993.

[Wan98] X. Wan, C.J. Kuo, "A multiresolution color clustering approach to image indexing and retrieval," Proc. ICASSP, 1998.

[Wertheimer38] M. Wertheimer, "Laws of organization in perceptual forms (partial translation)," A Sourcebook of Gestalt Psychology, W.B. Ellis, ed., pp. 71-88, Harcourt, Brace, 1938.

[Willamowski04] J. Willamowski, D. Arregui, G. Csurka, C.R. Dance and L. Fan, "Categorizing nine visual classes using local appearance descriptors," In *ICPR* workshop on learning for adaptable visual system, 2004.

[Williams01] S.Williams, G. Dissanayake, and H.F. Durrant-Whyte, "Towards terrain-aided navigation for underwater robotics," *Advanced Robotics*, Vol. 15, 2001.

[Wilson97] R. C. Wilson and Edwin R. Hancock, “Structural Matching by Discrete Relaxation”, IEEE Tran. PAMI, Vol. 19, pp. 634-648, June, 1997.

[Winn05] J. Winn, A. Criminisi, and T. Minkal. Categorization by learned Universal visual dictionary. In *Proc. ICCV*, pp. 1800-1807, 2005.

[Witkin83] A. Witkin and J. Tenenbaum, “On the role of structure in vision,” Human and machine vision, Beck, Hope, and Rosenfeld, eds. New York: Academic Press, 1983.

[Wu93] Z. Wu and R. Leahy, “An optimal graph theoretic approach to data clustering: theory and its application to image segmentation,” IEEE Trans. PAMI, 15(11), pp. 1,101-1,113, 1993.

[Xing01] E.P. Xing, M.I. Jordan and R.M. Karp, “Feature selection for high-dimensional genomic microarray data,” In proc. Machine learning, pp. 601-608, 2001.

[Yamauchi96] B. Yamauchi and R. Beer, “Spatial learning for navigation in dynamic environments,” *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, Special Issue on Learning Autonomous Robots, 1996.

[Yang07] L. Yang, P. Meer, and D. J. Foran, “Multiple class segmentation using a unified framework over man-shift patches,” in *Proc. CVPR*, pp. 1-8, 2007.

[Yaramakala05] S. Yaramakala and D. Margaritis. Speculative Markov blanket discovery for optimal feature selection. *Fifth IEEE International Conference on Data Mining*, 2005.

[Zhang06] H. Zhang, S. Cholleti, S.A. Goldman, and J.E. Fritts, “Meta-evaluation of image segmentation using machine learning,” In CVPR, Vol 1, pp. 1138-1145, 2006.

[Zhang03] J. Zhanga, X. Zhanga, H. Krimb, G.G. Walter, “Object representation and recognition in shape spaces,” *Pattern Recognition,* Vol. 36, pp. 1143 – 1154, 2003.

[Zhang07] J. Zhang, Marcin Marszałek, Svetlana Lazebnik, Cordelia 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.

[Zhong00] Y. Zhong, and A.K. Jain, “Object localization using color, texture and shape,” *Pattern Recognit 33,* pp. 671-684, 2000.

[Zhu96] S.C. Zhu and A. Yuille, “Region competition: Unifying snakes, region growing and Bayes/MDL for multi-band image segmentation,” IEEE Trans. PAMI, 18(9), pp. 884-900, 1996.

[Zimmer96] U.R. Zimmer, “Robust world-modeling and navigation in a real world,” *Neurocomputing*, 13(2–4), 1996.

Vita

Chang Cheng was born in Shenyang, Liaoning Province, P.R. China. He moved to Nanjing, Jiangsu Province with his family when he was 7. He attended Hangzhou Dianzi University in Hangzhou where he received a Bachelor of Engineering degree, major in electrical precision machinery in 1995. After graduation, He entered the workforce as a mechanical engineer at the printer company, zijing Group in Nanjing. In 1998, he attended the Southeast University in Nanjing where he received a Master of Engineering degree from the computer science department in 2001. During his graduate studies, he worked as an intern at the Nanjing Institute, Huawei Corp. for 6 months. After graduated from the Southeast University, he worked as a software engineer at the Nanjing Institute, Huawei Corp for one year. In 2002, he attended the University of Tennessee as a doctor student in computer science. During the summer of 2004, he joined the Imaging, Robotics, and Intelligent Systems Laboratory where he completed his Doctor of Philosophy degree in Electrical Engineering in 2010. His research interests are imaging processing, computer vision and robotics.