

VISUAL PLACE CATEGORIZATION

A Thesis
Presented to
The Academic Faculty

by

Jianxin Wu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
College of Computing

Georgia Institute of Technology
August, 2009

VISUAL PLACE CATEGORIZATION

Approved by:

Professor James M. Rehg, Advisor
College of Computing
Georgia Institute of Technology

Professor Henrik Christensen
College of Computing
Georgia Institute of Technology

Professor Frank Dellaert
College of Computing
Georgia Institute of Technology

Professor Irfan Essa
College of Computing
Georgia Institute of Technology

Professor Jitendra Malik
Department of Electrical Engineering
and Computer Sciences
University of California, Berkeley

Date Approved: July 1, 2009

To my parents

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor, Dr. James Matthew Rehg, for his advice and guidance during the past years. I have always been inspired by Jim's vision about the future of computer vision. I am especially grateful for his patience and guidance during the period when I was looking for a thesis topic (for a long period).

I would also like to thank my committee members, Dr. Henrik Christensen, Dr. Irfan Essa, Dr. Frank Dellaert, and Dr. Jitendra Malik. Jim and Henrik guided me into the field of place categorization. We performed data collection together, using the camera Irfan kindly lent us. I learned a lot from my committee's suggestions and critiques, which urged me to understand and explain better the techniques I proposed in this work.

It is a pleasure to work with those wonderful people here in Georgia Tech during my study. In particular, I thank Dr. Aaron Bobick for insightful discussions and Matt Mullin for sharing his mathematical skills. Thanks also to my fellow students Charlie Brubaker, Matt Flagg, Dongshin Kim, Kai Ni, Ping Wang, Pei Yin, and Howard Zhou. My graduate student life is more enjoyable with you guys.

Finally and most of all, I would like to thank my family. Your love, patience, and encouragement supported me throughout the entire Ph.D. study period.

TABLE OF CONTENTS

DEDICATION	iii
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	viii
LIST OF FIGURES	ix
SUMMARY	xi
I INTRODUCTION	1
1.1 The visual place categorization problem	2
1.2 Definition on place category	5
1.3 CT histogram feature descriptor and HIK visual codebooks	7
1.4 The VPC system	9
1.5 Relationship to previous work	9
II RELATED WORK	10
2.1 Closely related research	10
2.2 Robot Mapping	12
2.3 Representation of scene images	13
2.4 Incorporating Spatial Information	18
2.5 Summary of the chapter	20
III CT HISTOGRAM: A DESCRIPTOR FOR PLACE CATEGORIZATION	21
3.1 Shape matters!	22
3.2 Census Transform histograms	23
3.2.1 Constraints among CT histogram components	26
3.3 CT histogram encodes image structures	27
3.4 Reconstructing patches from CT histograms	29
3.5 Comparing CT histogram, SIFT, and Gist	32
3.6 Limitations of the CT histogram	36

3.7	Summary of the chapter	37
IV	HIK VISUAL CODEBOOK	39
4.1	The need for HIK based codebook	39
4.2	Generating visual codebooks using HIK	40
4.3	Fast Evaluation	41
4.4	One-class SVM codeword generation	45
4.5	K-median codebook generation	47
4.6	Validation of HIK codebooks	48
4.6.1	HIK Visual Codebook (Algorithm 1) greatly improves classification accuracy	51
4.6.2	The HIK codebook evaluates quickly (Algorithm 2)	52
4.6.3	K-median is a compromise between k-means and HIK codebooks.	52
4.6.4	One-class SVM improves histogram intersection kernel code words (Algorithm 3)	52
4.6.5	One-Class SVM degrades usual k-means code words	52
4.6.6	Use the right feature for different tasks	54
4.6.7	Comparison with previously published results	55
4.7	Summary of the chapter	56
V	THE VPC DATASET AND SYSTEM	57
5.1	The Visual Place Categorization Dataset	57
5.1.1	Example frames from the VPC dataset	60
5.2	The Visual Place Categorization system	62
5.2.1	Image Representation	63
5.2.2	Bayesian filtering	64
5.3	Experimental setup and evaluation methodology	66
5.4	Results and discussions	68
5.4.1	Baseline system	68
5.4.2	CT histogram is suitable for visual place categorization	69

5.4.3	Bayesian filtering improves system accuracy	70
5.4.4	HIK codebooks further improves VPC	73
5.4.5	Effect of One-class SVM code words	75
5.5	Discussions	75
5.6	Summary of the chapter	77
VI	SUMMARY AND DISCUSSIONS	79
6.1	Summary of contributions	79
6.2	Discussions	79
6.2.1	Attentional mechanism	80
6.2.2	Broader field of view	80
	REFERENCES	81

LIST OF TABLES

1	Results of HIK, k-median and k-means codebooks and one-class SVM code words. (a), (b), and(c) are results for the Caltech 101, 15 class scene, and 8 class sports datasets, respectively. κ_{HI} and κ_{LIN} means that a histogram intersection or linear kernel is used, respectively. oc_{svm} and $\neg oc_{\text{svm}}$ indicate whether one-class SVM is used in generating code words. B and $\neg B$ indicate whether Sobel images are concatenated or not. And $s = 4$ or $s = 8$ is the grid step size when densely sampling features. The number of training/testing images in each category are indicated in the sub-table captions. The best result in each <i>column</i> is shown in boldface	53
2	Results when SIFT descriptors are used in stead of CT histograms. We use $\neg B$, $s = 8$, κ_{HI} , and oc_{SVM} . Note that the second row contains the top right corner numbers in Table 1.	55
3	The 11 semantic categories in the VPC dataset, plus a special category named <i>transition</i>	59
4	Baseline categorization accuracy (in percentages) of all homes and categories when SIFT and k-means are used. The Bayesian filtering is not used.	69
5	Linear k-means codebook categorization accuracy (in percentages) of all homes and categories using CT histogram as the feature descriptor. The Bayesian filtering is not used.	69
6	Linear k-means codebook categorization accuracy (in percentages) of all homes and categories using CT histogram as the feature descriptor. The Bayesian filtering is used.	70
7	Baseline categorization accuracy (in percentages) of all homes and categories when SIFT and k-means are used. The Bayesian filtering is used.	72
8	HIK codebook categorization accuracy (in percentages) of all homes and categories when CT histogram and Bayesian filtering are used.	73
9	HIK codebook and one-class SVM code words categorization accuracy (in percentages) of all homes and categories when the Bayesian filtering is used.	75

LIST OF FIGURES

1	An example kitchen image and its Sobel image	23
2	An example <i>Census Transformed image</i>	24
3	Illustration of constraints between CT values of pixels	25
4	Illustration of Census transforms	28
5	Census Transform encodes shape in 1-D	29
6	Reconstruct images from CT histograms. In each group of images, the left image is the input image. The image in the middle is the initial starting point that is generated by randomly exchanging pairs of pixels in the input. The reconstruction result is shown in the right of each group, which has the same CT histogram as the input image.	31
7	Distribution of similarity values on the 15 class scene recognition dataset. The red curve shows within-class similarity distribution, i.e. similarity of two feature descriptors computed from images in the same categories. The green curve shows inter-class similarity distribution, i.e. similarity of two descriptors computed from images in different categories. (This figure is best viewed in color.)	33
8	Distribution of similarity values on the Caltech 101 object recognition dataset. (This figure is best viewed in color.)	34
9	Histogram comparing similarity values of best in-category nearest neighbor with best out-of-category nearest neighbor of an image. (This figure is best viewed in color.)	35
10	Visualization of images mapped to visual code words. In each row, the first image is an input image, with the second and third being visualization for CT histogram and SIFT codebooks, respectively. (This picture needs to be viewed in color.)	35
11	Images from 15 different scene categories	49
12	Images from 8 different sports event categories	50
13	Illustration of the level 2, 1, and 0 split of an image.	51
14	Effects of one-class SVM.	54
15	VPC data collection hardware.	58
16	Example frames from the bedroom category.	61
17	Example frames from the living room category.	62

18	Diagram of the VPC system.	63
19	Example results of the visual place categorization system, in which Bayesian filtering is used.	67
20	Example results of the Visual Place Categorization system, in which Bayesian filtering is not used.	71
21	Effect of using the Bayesian filtering.	72
22	Example of frames in the living room and dining room category.	74

SUMMARY

Knowing the semantic category of a robot’s current position not only facilitates the robot’s navigation, but also greatly improves its ability to serve human needs and to interpret the scene. Visual Place Categorization (VPC) is addressed in this dissertation, which refers to the problem of predicting the semantic category of a place using visual information collected from an autonomous robot platform.

Census Transform (CT) histogram and Histogram Intersection Kernel (HIK) based visual codebooks are proposed to represent an image. CT histogram encodes the stable spatial structure of an image that reflects the functionality of a location. It is suitable for categorizing places and has shown better performance than commonly used descriptors such as SIFT or Gist in the VPC task.

HIK has been shown to work better than the Euclidean distance in classifying histograms. We extend it in an unsupervised manner to generate visual codebooks for the CT histogram descriptor. HIK codebooks help CT histogram to deal with the huge variations in VPC and improve system accuracy. A computational method is also proposed to generate HIK codebooks in an efficient way.

The first significant VPC dataset in home environments is collected and is made publicly available, which is also used to evaluate the VPC system based on the proposed techniques. The VPC system achieves promising results for this challenging problem, especially for important categories such as bedroom, bathroom, and kitchen. The proposed techniques achieved higher accuracies than competing descriptors and visual codebook generation methods.

CHAPTER I

INTRODUCTION

In this dissertation we describe the problem of Visual Place Categorization (VPC), introduce the first significant VPC dataset collected in home environments, and present a solution approach which forms a first VPC system in home environments. Visual place categorization refers to the problem of predicting the semantic category of a place using visual information collected from an autonomous robot platform. Our VPC system is built using the Census Transform (CT) histogram descriptor, a Histogram Intersection Kernel (HIK) based visual codebook generation method, and a standard Bayesian filtering technique. We collected a VPC dataset from home environments. Experimental results using the VPC dataset demonstrate that the proposed representation (i.e. the CT histogram descriptor and the HIK visual codebook) achieve higher categorization accuracy over standard feature descriptors and visual codebook generation methods. Together with Bayesian filtering, the proposed system shows promising results for the VPC problem.

The thesis of this dissertation is the following:

Census transform histogram and histogram intersection kernel based visual codebook can provide a suitable representation for solving the visual place categorization problem.

Three contributions are made in this dissertation, which include

1. **Problem and dataset.** We introduce the VPC problem and clearly explain its relationship to existing research questions. We collected a VPC dataset and make it publicly available at <http://categorizingplaces.com/dataset.html>.

2. **Representation.** We propose the CT histogram descriptor for VPC, and show that this is the suitable descriptor for recognizing semantic categories of places and scenes. We also propose a HIK based visual codebook generation method which shows better performance for histogram feature vectors. HIK codebooks not only improves the VPC system, but also other scene and object recognition methods when histogram features are involved.
3. **System.** We build a first VPC system in home environments using the proposed techniques, and achieved promising results on the VPC dataset.

In the rest of this chapter these contributions will be further discussed.

1.1 The visual place categorization problem

Visual place categorization refers to the problem of predicting the semantic category of a place using visual information collected from an autonomous robot platform. Canonical examples of places are types of rooms in a home (e.g. kitchen, family room, etc.) or a business (e.g. reception area, loading dock, etc.).

Place categorization is related to, but also quite different from, existing research such as place recognition in topological SLAM and scene categorization in image retrieval. We first describe the related research problems before we distinguish place categorization from them:

- *Place recognition*, or global localization, which identifies the current position and orientation of a robot [31, 63], seeks to find the exact parameterization of a robot’s pose in a global reference frame. Place recognition is an inherent part of a Simultaneous Localization and Map Building (SLAM) system [15, 59].
- *Topological place recognition* answers the same question “Where am I?”, but at a coarser granularity [67]. In topological robot mapping, a robot is not required to determine its 3D location from the landmarks. It is enough to determine a

rough location, e.g. corridor or office 113. A place in topological maps does not necessarily coincide with the human concept of rooms or regions [10]. Places in a topological map are usually generated by a discretization of the robot’s environment based on certain distinctive features or events in the environment.

- *Scene recognition*, or scene categorization, is a term that is usually used to refer to the problem of recognizing the semantic label (e.g. bedroom, mountain, or coast) of a single image [7, 18, 33, 49, 54]. The input images in scene recognition are usually captured by a person, and are ensured to be representative or characteristic of the underlying scene category. It is usually easy for a person to look at an input image in scene recognition, and determine its category label. The learned scene recognizer is generalizable, i.e. it is able to recognize the category of scene images acquired in places that are not present in the training set.

We now focus more specifically on the relationships between place categorization, place recognition, and scene categorization. Place categorization differs from place recognition in that the goal is to predict the category of a place when it is seen for the first time, rather than trying to recognize a specific place when the robot has returned to it.

Place categorization differs from scene recognition in the type of image data that is utilized. Scene recognition typically uses images from the web or image libraries which have been taken by humans for human consumption. As a result, these images tend to be quite representative of the scene category and often frame important scene elements (for example, a strip of beach with sea and sun in a picture of a beach scene). A photographer would be unlikely to take a picture of the sidewalk and then upload it to Flickr with the label “Trevi Fountain”.

In contrast, the images used in place categorization will be captured by a robot agent without the advantage of a human attention mechanism. As a consequence, a

majority of the frames available for place categorization may not be particularly representative of the category, and it is a priori difficult to distinguish the non-representative and representative frames. On the other hand, a robot can capture frames continuously and the temporal continuity of the category labels can be exploited to constrain the place categorization problem.¹

There have been several previous works on place categorization which are related to VPC [53, 56, 65, 78]. These previous efforts also attempted to predict the category label for a place using sensor data acquired autonomously. Like these works, we will leverage a corpus of manually-labeled training examples to learn categorical concepts. Our work is distinguished in two main ways from these previous efforts. First, we employ only visual appearance features for categorization. Second, we emphasize the generalization performance of the method by focusing on a diverse set of categories with significant intra-class variation in appearance. A more detailed comparison can be found in Chapter 2. As an integrated component of the VPC problem definition, we present the first significant dataset for the VPC problem, consisting of image sequences with ground truth labels captured from a variety of different home interiors. Our datasets consists of high resolution images from 6 homes, with 12 categories being manually labeled. Details of this dataset is presented in Chapter 5.

The motivation for this work is the challenging problem of semantic mapping, by which we mean the autonomous recovery of semantic as well as structural properties of the robot’s environment to facilitate it’s execution of tasks. A categorization of the robot’s current location is the natural choice of important and useful semantic information. For example, a delivery robot is more useful if it can distinguish a loading dock from the front reception desk in various businesses. Similarly, a cleaning robot can be more effective if it has the ability to recognize room type (bedroom, bathroom,

¹The difference in input images is clearly demonstrated by the scene recognition input images in Figure 11 of page 49 and randomly chosen frames from the VPC dataset in Figure 16 of page 61.

kitchen etc.), for example by selecting cleaning strategies based upon the type of room.

Solving the place categorization problem will lead to advancement of the state-of-the-art of robot navigation and computer vision:

1. Our method for learning to recognize semantic categories can provide new sensing capabilities for autonomous mobile robot applications. For example, if a home service robot has built-in concept of categories such as kitchen, bedroom, etc., it then can automatically navigate through a new home environment and generate a topological map that encodes (with attributed knowledge) the functionality of each room. No human input is required in this semantic map building process so long as the robot has obstacle avoidance and map building capacity. In the long term we plan to integrate place categorization with autonomous robot mapping capabilities, such as topological SLAM;
2. Knowing the semantic category of an environment exerts strong priors on the objects that may appear in it [65]. Thus successful place category recognition helps object recognition. Our hope is that the recognition of place category and the objects contained in the place should behave synergistically, i.e. both place and object recognition will behave better than considering either problem in isolation. With added object recognition capability, an out-of-the-box robot can immediately perform tasks such as “grab the coffee cup in the kitchen and bring it to the living room” as soon as it arrives at a new home.

1.2 Definition on place category

The word “place” is often interpreted in many different ways. Its meaning is usually varying, depending on the context in which places are mentioned. In this dissertation, we use a supervised learning strategy. Place categories are defined by human, through manually provided category label for every video frame. Different categorization of places can be provided in different VPC applications.

We work within indoor home environments and the place categories we deal with correspond to room types, for example, bedroom and family room. These room types form natural semantic place categories mainly from the functionality that they provide to people. Although architecture and interior design evolve along different paths in various cultures, major room types have more or less similar meanings across the globe.

Visual place categorization, however, is not confined to recognizing room types in a home environment. Different categorization of spaces are possible even within home environments. For example, places can be categorized at a sub-room scale which share the same utility. Categories such as *sitting area* may include the couch and chairs in the living room, chairs in the dining area, or even the area with a reclining chair in a balcony. Depending on the purpose of the robotic platform in which VPC is running, different designations of semantic place categories can be made. We do not work with sub-room categories in this thesis. However, we would expect that the proposed techniques will extend to this line of research. The proposed representation is designed to capture the spatial structure of an image (refer to Chapter 3), while we expect sub-room categories that share similar functions also share similar spatial structures. A supervised machine learning method would be able to learn these new category concepts.

We will also briefly discuss how outdoor places could be categorized, but we do not address that application in this dissertation. In outdoor environments, the function of a place is not as clear as that of indoor places. Fortunately categories can also be defined in a less restrictive sense for outdoor places. For example, in the hybrid Spatial Semantic Hierarchy model [30], large scale spaces are defined as spaces whose structure are beyond the sensory horizon and used as *decision points*. Outdoor places are usually beyond sensory horizon and we want to make our outdoor place categories useful for making decisions. In other words, outdoor place categories will

be supplied by human labeler. The human labeler is responsible for choosing outdoor place category labels that are useful for the robot’s application.

Depending on what is desired for a robot, outdoor visual place categories can be defined in different granularities. Suppose we are categorizing places for a delivery robot. Visual place categories such as street, sidewalk, plaza, parking space, business loading dock etc. will help the robot. It is safe to assume that an autonomous delivery robot (in the future) is equipped with maps and global positioning systems. However, the ability to categorizing places such as streets and sidewalks will help it abide by traffic rules and cope with unexpected situations (e.g. road work). Categories such as plaza, parking space or loading dock will provide way points and help our robot to move to its desired destination (instead of breaking into the front desk of the business). The ability to distinguish the type of business (e.g. restaurant vs. clothing store) can be helpful in identifying the exact delivery location.

In larger areas, outdoor place categories can be defined to facilitate a robot’s large scale navigation ability, e.g. bridge, ford, gully, clearing path, path intersection, etc. In even larger spaces, e.g. forest or mountain, place categorization can be performed at a correspondingly higher abstract level. For example, a robot may be interested in categories like water surface, forest, bushes, grass, rocks, etc. It can even distinguish between only two categories, traversable or not. The proposed techniques have shown state-of-the-art performance on scene recognition tasks, which involve many outdoor places. Thus we expect they will be helpful for outdoor visual place categorization too in future research.

1.3 CT histogram feature descriptor and HIK visual code-books

We believe that an appropriate representation (or, more precisely, feature descriptor) is key to the success of a place or scene recognition task, including VPC. In the literature, SIFT and Gist are probably the most popular feature descriptors in place

and scene recognition [8, 18, 23, 28, 33, 34, 36, 49, 54, 59]. The SIFT descriptor is originally designed for recognizing the same object appearing under different conditions, and has strong discriminative power. Recognizing place categories, however, poses different requirements for the feature descriptors. Images taken from the same place category may look very different, i.e. with huge intra-class variations. Similarly, images taken from different part or view point of the same topological location (e.g. office 113) will also contain huge variations. Rather than capturing the detailed textural information of objects in the scene, we would like to capture the stable spatial structure within images that reflects the functionality of the location [49].

Oliva and Torralba [49] proposed the Gist descriptor to represent such spatial structures. Gist achieved high accuracy in recognizing natural scene categories, e.g. mountain and coast. However, when categories of indoor environments are added, its performance drops dramatically (c.f. Chapter 3).

In Chapter 3, we propose the Census Transform histogram as a novel descriptor. Unlike the histogram of pixel intensities which totally ignores spatial information, histogram of Census Transform values encodes the spatial structure in an image patch through the strong correlation among neighboring CT values. We believe that CT histogram captures the structural properties of an image, instead of detailed textural information. CT histogram outperforms both the SIFT and Gist descriptors, in the context of VPC and scene recognition.

In Chapter 4, we propose to use a Histogram Intersection Kernel (HIK) based visual codebook for using CT histogram in VPC. Unlike existing codebooks generated by using the Euclidean distance, the proposed method uses histogram intersection kernel to compare two histograms. HIK has been repeatedly proven a more suitable similarity measure for comparing histograms in supervised learning tasks. We show that the proposed method apply HIK in unsupervised learning and improves the codebook quality, which in turn yields higher VPC accuracy.

1.4 The VPC system

A first VPC system for home interiors is presented and evaluated using the proposed techniques and the new VPC dataset. Our experiments show that for recognizing place categories, including both VPC and scene recognition, CT histogram is the suitable representation, which yields higher recognition accuracies than the SIFT descriptor. In addition, the Histogram Intersection Kernel based visual codebooks consistently acquire higher system accuracies than the usual Euclidean distance based k-means codebooks. In our VPC system, the standard Bayesian filtering technique not only improves overall system accuracy, but also greatly reduces the fragmentation of predicted category labels. Details about the VPC system is described in Chapter 5.

In summary, the contributions of this paper include the VPC problem and an associated dataset of home interiors, and two proposed techniques for a first VPC system that shows promising results. After reviewing related works in Chapter 2, the contributions are detailed in the remaining chapters.

1.5 Relationship to previous work

The following papers describe part of the research presented in Chapter 3, Chapter 4, and Chapter 5, respectively.

WU, J. and REHG, J. M., “Where am I: Place instance and category recognition using spatial PACT,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

WU, J. and REHG, J. M., “Beyond the euclidean distance: Creating effective visual codebooks using the histogram intersection kernel,” in *The IEEE Conf. on Computer Vision*, 2009.

WU, J., CHRISTENSEN, H. I., and REHG, J. M., “Visual Place Categorization: Problem, Dataset, and Algorithm,” in *Proc. IEEE/RSJ Intl. Conf. Intelligent Robots and Systems*, 2009.

CHAPTER II

RELATED WORK

There is a large body of literature in both place instance and category recognition, and a huge, continuously growing object recognition literature. In this chapter we first review some research efforts that are close to our place categorization problem. We then review the current state of the literature in both place instance and category recognition problems, with a focus on the representation issue.

2.1 Closely related research

Problems that have similar formulation to place categorization have been previously presented. Places in office environment are categorized in the robotics system of [56, 44, 45]. A topological map was also built in this system. The map is classified into place categories including office, laboratory, doorway, corridor, kitchen, and seminar room. This system used both laser range sensors and cameras as their input sensors. They achieved reasonable recognition accuracies. The categories in these systems, however, are intuitively distinguishable by the geometry shape of objects contained in the scene (e.g. ceiling in the corridor, or door frame in the doorway). Originally this system was not generalized to environments that are unseen during the training phase. In [56] Rottmann et al. also tested the AdaBoost based categorical place recognizer on images taken in new environments.

In visual place categorization we are interested in recognizing more complex semantic categories based on their functionality to humans (e.g. bedroom vs. living room). Images from such categories will have much larger intra-class variation than the categories studied in [56, 44, 45]. Thus we need a flexible representation and a larger and more diverse dataset to evaluate place categorization methods.

In [65], which is probably the closest work to this thesis in computer vision research, Torralba et al. recognized both place instances (e.g. Jason corridor vs. Kevin corridor) and categories (e.g. corridor vs. conference rooms) using data collected from a mobile system. Since no human guidance was explicitly exerted in their data collection to capture representative view of a place, their problem formulation is similar to place categorization. They achieved high accuracy in recognizing place instances. In the category level recognition, they achieved reasonable accuracy in 3 categories (conference rooms, corridor, and office), but failed to recognize other categories (kitchen, elevator, lobby, etc). Our conjecture is that objects in the 3 successful classes have a specific geometric shape, which helps in recognition.

It is worth noting that the data in [65] were collected by a camera that was mounted on a person’s head while the person was walking around. By observing the videos we find that the data collection person tend to pay more attention to areas that are representative of the category (e.g. computer desk in an office) than other areas. Thus, the videos in [65] were not autonomously collected.

As mentioned above, in visual place categorization we will recognize categories that are defined by their functionality. Images from the same category may have diverse visual patterns. Our VPC dataset of home interiors is a more complex problem than that reflected in the dataset used in [65].

Pronobis et al. [53] also recognized place categories (offices, corridor, printer area, and kitchen). The classifiers were designed to recognize place categories under various changes: weather conditions, moving persons and furniture, etc. However, they do not apply the learned category concepts to new environments. Instead, they tested the learned classifiers in the same part of the building where the training data were collected.

Besides these closely related research, there are other related works, which we will briefly review in the rest of this chapter.

2.2 *Robot Mapping*

Place recognition is an inherent part of any robot mapping system. Depending on the type of map that is going to be generated (metric, topological, or semantic), various place recognition tasks need to be performed.

In metric maps all information is maintained in a global reference frame. Thus exact localization is necessary with the help of distinct landmarks. For example, Se, Lowe, and Little [59] used SIFT features to detect and track 3D landmarks, to estimate the robot pose, and to build a precise 3D map simultaneously.

Simultaneous Localization and Map Building (SLAM) is an important research topic in robotic science, and has been used to produce both metric and topological maps [3, 12, 14, 15, 21, 42]. Topological maps use a graph to represent the connectivity of an environment. Some easily distinguishable places are placed as nodes in a topological map, and the edges in the graph represent traversability from one place to another.

Place recognition is obviously very important because it is required to recognize a node in the map graph which corresponds to the robot’s current location. However, the efforts in these mapping systems are usually not focused on visual recognition using the image properties of places. Rather, weak correspondences are computed from range or visual sensors, and geometric and statistical methods are used to refine the correspondences, and to close gaps in the loop, etc. (for example, in [64]).

It is hard to give an exact definition for either “semantic knowledge” or “semantic mapping”. However, it is important to possess such information in a robot map in order for a robot to interact with a complex and non-static environment. Kuipers [29] defined Spatial Semantic Hierarchy (SSH), a hierarchical structure that encoded spatial knowledge at various abstraction levels. Early attempts tried to detect easy concepts (flat surfaces) such as walls, doors, ceilings, doorways, etc. For example, Liu et al. [37] built 3D models that consisted of these simple concepts using both the

range sensor and a panoramic camera. Most of these methods used laser scanner data (or with vision sensors as an addition).

There are also systems and datasets that use vision as the only input modality. The KTH INDECS (INdoor Environment under Changing conditionS [52]) and IDOL (Image Database for rObot Localization 1 & 2 [39]) datasets are captured in a five-room office environment, including a one-person office, a two-person office, a kitchen, a corridor, and a printer area. Both datasets include images captured under various illumination and weather conditions, and contain significant variations in the environment. The SVM classifier was utilized to classify which room the image was taken in based on a single input image [53].

It is worth noting that all the aforementioned efforts on place recognition in the robotics community deal with place instances or non-generalizable categories. That is, the learned place concept is not easily generalized to new, unseen environment. For example, the room models learned by these methods using data from the KTH environment will not necessarily be useful in recognizing other office environments (e.g. the kitchen or printer area in Georgia Tech).

2.3 Representation of scene images

Computer vision researchers, however, put their research emphasis on recognizing place categories, or, recognizing *scenes*. Among the scene recognition methods, representation has always being the focus of attention.

Histograms of various image properties (e.g. color [53, 62, 67], or image derivatives [53]) have been widely used in scene recognition. However, after the SIFT [38] feature and descriptor are popularized in the vision community, it nearly dominates the feature choice in place recognition systems [7, 18, 28, 33, 34, 36, 54, 59, 78]. SIFT features are invariant to scale and is robust to orientation changes. The 128 dimensional SIFT descriptors have high discriminative power, while at the same time are

robust to local variations [41]. It has been shown that SIFT features significantly outperform edge points [33], pixel intensities [7, 18], and steerable pyramids [28] in recognizing places and scenes.

Researchers have tried to recognize place categories using important objects in the place, e.g. Ranganathan and Dellaert [55] extended the constellation model into 3D and used the model to detect objects such as computer monitors, printers, chairs, cupboards, and drawers. A graphical model was then used to recognize places from objects.

However, holistic approaches seem to be more popular. It is suggested in [49] that recognition of scenes could be accomplished by using *global configurations*, without detailed object information. Oliva and Torralba argued for the use of *Shape of the Scene*, an underlying similar and stable spatial structure that presumably exists within scene images coming from the same *functional* category, to recognize scene categories. They proposed the Gist descriptor to represent such spatial structures. Gist computes the spectral information in an image through Discrete Fourier Transform (DFT). The spectral signals are then compressed by the Karhunen-Loeve Transform (KLT), a continuous counterpart of the discrete Principal Component Analysis (PCA) method. They showed that many scene signatures such as the degree of *naturalness* and *openness* were reliably estimated from such spectral signals, which in consequence resulted in satisfactory scene recognition results. Since spectral signals were computed from the global image, Oliva and Torralba suggested recognizing scenes without segmentation or recognizing local objects beforehand.

Gist achieved high accuracy in recognizing natural scene categories, e.g. mountain and coast. However, when categories of indoor environments are added, the Gist descriptor’s performance drops dramatically. We will show in Section 3.5 that in a 15 class scene recognition dataset [33], which includes the data used in [49] and several other categories (mainly indoor categories), the accuracy of Gist descriptor is much

worse than its performance on outdoor images, and is significantly lower than the proposed spatial CT histogram descriptor. Our conjecture is that those properties that the Gist descriptor is modeling are not effective discriminators in an indoor environments. For example, almost all indoor images have low degree of naturalness. Similarly, other spatial structure properties modeled by Gist such as the degree of *openness, roughness, and ruggedness* [49] do not apply to indoor scene either.

However, the global configuration argument itself is accepted by many other researchers, whom used the SIFT descriptor to describe the global configuration. Since the SIFT descriptor is designed to recognize the same object instance, statistical analysis of the distribution of SIFT descriptors are popular in scene recognition. Statistics of SIFT descriptors are more tolerant to the huge variations in scene images. SIFT descriptors are first vector quantized to form the *visual codebook* or *visterms*, e.g. by the k-means clustering algorithm. The hope here is that the cluster centers will be meaningful and representative common sub-structures, similar to the codebook in a communication system.

It is always important to find the right balance between the discriminative power and invariance property of the feature descriptor for a specific task. We will show that the proposed CT histogram descriptor is suitable for the place and scene recognition task. It captures the shape of the scene while it is not sensitive to irrelevant textural details as SIFT.

Visual codebooks are usually used in place and scene recognition systems. Visual codebooks (or, vector quantization methods) are helpful in dealing with the huge variations in place category recognition. A visual codebook is a method that divides the feature descriptor space into several regions. Features in one region correspond to the same *visual code word*, which is indexed by an integer between 1 and size of the codebook. An image or image window can then be encoded as a histogram of code words.

K-means is the most widely used method for codebook generation [60]. However, several alternative strategies have been explored. K-means usually positions its clusters almost exclusively around the densest regions. Jurie and Triggs used a mean-shift type clustering method to overcome this drawback [27]. There are also information theoretic methods that try to capture the *semantic* common visual components by minimizing information loss [36, 32]. An extreme method was presented in [66] that divided the space into regular lattice instead of learning a division from data. There are also efforts to build hash functions (multiple binary functions / hash bits) in order to accelerate distance computations [74].

In k-means based methods, a code word is represented by the cluster center (average of all features that belongs to this code word), which is simple and fast to compute. It was discovered that assigning a feature to multiple code words (i.e. soft-assignment) may improve codebook quality [51, 68]. Within a probabilistic framework, codewords can be represented by the Gaussian Mixture Model (GMM) [50, 75]. GMM has better representation power than a single cluster center. However, it requires more computational power. Another interesting representation is hyperfeature [1], which considers the mapped code word indexes as a type of image feature and repeatedly generates new codebooks and code words into a hierarchy.

Methods have been proposed to accelerate the space division and code word mapping. Nistér and Stewénus [46] used a tree structure to divide the space hierarchically and Moosmann et al. [43] used ensembles of randomly created cluster trees. Both methods map visual features to code words much faster than k-means.

We will also learn visual codebooks for our visual place categorization task. However, it is worth noting that all of these previous methods used the l_2 distance metric, i.e. Euclidean distance, to form a visual codebook. For the case of supervised classification, it has been shown that l_2 is not the most effective method for comparing two histograms [40]. In particular, the Histogram Intersection Kernel (HIK) was

demonstrated to give significantly improved performance. In this thesis I propose a new alternative to the simple Euclidean distance based k-means algorithm, using the histogram intersection kernel. Since CT histogram, the proposed feature descriptor for VPC, is also a histogram of image statistics, a visual codebook could therefore in principle be improved through the use of HIK.

Different views are held on whether the vector quantized SIFT features form semantically meaningful visual codebooks. Some researchers believe that the codebook represents meaningful semantic aspects of natural scenes. Liu and Shah [36] used Maximization of Mutual Information co-clustering to cluster SIFT features to form intermediate semantic concepts. Probabilistic Latent Semantic Analysis (pLSA) was also used to unsupervisedly detect latent semantic topics [7, 54]. Quelhas et al. showed that in a 3 class classification task pLSA generated compact representation and improved recognition [54]. However, Lazebnik, Schmid and Ponce showed that pLSA lowered recognition rates by about 9% [33] in their rich 15 class scene recognition dataset. The k-means algorithm was used to cluster SIFT features, and the cluster centers were used as the codebook in [33]. In scene recognition, SIFT features are usually densely sampled on a regular grid, instead of only being sampled at sparse interest points [7].

The distribution properties of densely sampled SIFT features were studied by Tuytelaars and Schmid [66]. Their observations provides insights for evaluating the visual codebook for scene recognition. Tuytelaars and Schmid observed that the vast amount of SIFT features represent simple shapes (homogeneous patches or simple edge/line structures). Complex shapes (those that are useful for recognition) are much less frequent. They also showed that clusters learned in a class-specific way are more useful in recognition tasks.¹ Although it is not totally clear whether clustered SIFT centers will successfully play the roll of a semantically meaningful codebook,

¹The clusters in [66] were fixed size histogram bins.

it seems that a good visual codebook of SIFT features should be constructed in a supervised manner.

A different representation was proposed by Vogel and Schiele [72]. They split each image into 10 by 10 cells. Each cell was given a semantic label from 9 categories (sky, water, grass, etc.). An SVM classifier (“concept classifier”) is then trained to assign labels to new cells. In other words, instead of generating intermediate concepts from data without supervision, they specify a small set of intermediate concepts and learn them in a supervised manner. Category of an image was determined from the concept labels of its 100 cells. Their experiments corroborated the observation that using intermediate concepts gave better performance than using crude image features. However, the concept classifier’s accuracy was lower than 50% in 5 out of the 9 intermediate concepts in [72]. It is not totally clear how the intermediate concepts help recognizing the category of an image.

2.4 Incorporating Spatial Information

SIFT based models usually represent images as *bag of features*, i.e. spatial arrangement information among multiple features are completely ignored. However, it is long recognized that spatial arrangements are essential for recognizing scenes. For example, Szummer and Picard divided images into 4×4 sub-blocks. The K-nearest neighbor classifier was applied to these sub-blocks. The final indoor-outdoor decision was then made based on classification results from the 16 sub-blocks [62]. Their experiments showed that a simple strategy for the second phase classification (majority vote, i.e. assigning the image label to the most common class label among the sub-blocks) significantly improved recognition accuracy (approximately 10% higher compared to the sub-block accuracy). They also tried two other strategies (a one-layer neural network and a Mixture of Experts classifiers [26]), which only gave slightly better results than the majority vote.

Advocating the global configuration approach, Oliva and Torralba [49] also implicitly used spatial information. In their WDST (Windowed Discriminant Spectral Template, part of the Gist descriptor), spectral information was calculated for 8×8 local patches, with a diameter of 64 pixels for each patch. The spatial envelop computed from WDST usually outperformed that computed with DST (global Discriminant Spectral Template), sometimes with a large margin. It is natural to conjecture that the spatial arrangement information (implicitly coded in the local WDST ordering) elicited such performance improvements.

Grauman and Darrell [22] proposed the Pyramid Match Kernel (PMK) to deal with classification problems in which the features were set of features. The feature sets usually are unordered, and have different sizes. Thus, no exact correspondence is available between the features. The bag of features models produce feature sets that possess these properties, and pose difficulty for classical machine learning methods. The basic idea in pyramid match kernels is to build an approximate correspondence between two feature sets, by a hierarchical quantization of the feature space.

The pyramid match kernel successfully find an approximate correspondence in terms of proximity in the feature space. However, the spatial correspondence among features is probably more important in vision applications. Lazebnik, Schmid, and Ponce proposed the Spatial Pyramid Matching to systematically incorporate spatial information [33]. Features are quantized into M discrete types using the k-means clustering with M centroids. They assume that only features of the same type can be matched. Instead of dividing the feature space, the image is divided in a hierarchical fashion (of level L). The image is divided into $2^l \times 2^l$ sub-blocks in level l , with each dimension (horizontal or vertical) being divided into 2^l evenly sized segments. For a feature type m , X_m and Y_m are sets of the coordinates of type m features. The pyramid matching kernel can be used to compute a matching score $\kappa^L(X_m, Y_m)$ for feature type m . Note that the grid division in the pyramid match kernel is now a

division of the spatial dimensions. Note that a SPM kernel with $L = 0$ reduces to the standard bag of features model with the pyramid matching kernel.

2.5 Summary of the chapter

In this chapter we reviewed the related research in both robotics and computer vision. We are interested in recognizing the semantic category of a place, which is different from most of the existing robotics research. On the vision side, we focused on the representation / descriptor issue. Various computer vision findings support the idea of recognizing scene category from global shape characteristics of an image without first detecting objects in the scene. In most methods, a bag of words model with a visual codebook is used. This visual codebook approach has been repeatedly proved to be useful in recognizing scene categories. Incorporating spatial information (relative position among different features or image patches) recently attracted many research efforts, and greatly improved scene category recognition.

We also analyzed SIFT and Gist, two popular descriptors used in place and scene recognition. The SIFT descriptor is originally designed for matching the same object (or object part) under different conditions, and might not be optimal in presence of the huge variations in scene images or visual place categorization. The visual codebook generated by clustering SIFT descriptors does not appear to cluster visually semantically similar patches together.

The Gist descriptor has shown high accuracy in recognizing outdoor scenes. However, it is not suitable for indoor environments. More details will be presented in Chapter 3.

We will propose CT histogram, a feature descriptor that suits place categorization. A histogram intersection kernel based codebook generation method is proposed to generate high quality codebooks for CT histograms in Chapter 4.

CHAPTER III

CT HISTOGRAM: A DESCRIPTOR FOR PLACE CATEGORIZATION

We believe that the central problem in place categorization is a feature descriptor that meets the requirements of this domain:

1. **Flexible visual features.** One property of the place categorization problem is that no obvious structure exists in place images, which is different from object recognition. The intra-class variation is significant within a place category. As a consequence, we can not search for a template pattern using the approaches applied in most object recognition tasks [71]. For example, a bed will likely appear in an image taken from a bedroom. However, the picture can be taken from any viewpoint, and the bed can appear in many variations (mattress, sofa bed, water bed mattress, etc.). Bedrooms can be decorated in any possible color or style, with very different furniture and illumination conditions. These variations pose difficulties for the local patch based representations (e.g. SIFT features [38]). To make the problem even more difficult, since we are interested in autonomous data collection, the images may not be representative of the place category. In the bedroom example, a bed could possibly be invisible in many frames. These difficulties suggest that we need a feature descriptor that is more flexible. In other words, we want the descriptor to (implicitly or explicitly) capture more general structural properties such as *The sky has less variations than a mountain* or *The office environment has many horizontal and vertical structures*. Thus a global configuration of the image is preferred, as suggested in [49]. We seek features reflecting spatial structures, not detailed

textural patterns. We also want the descriptor to have the ability to distinguish between the subtle differences in indoor environments, e.g. bedroom vs. living room;

2. **Rough geometry.** Similarly, strong geometrical constraints (e.g. the constellation model [20]) are not applicable due to the large variations. Also, different objects can be arranged in any spatial configuration in a place image. However, rough geometry constraints are very helpful in recognizing place categories [33]. Constraints such as *The sky should be on top of the ground* will help reduce ambiguity, even when the images are taken from random viewpoints;
3. **Generalizability.** The learned category concepts will be applied to new images. An ideal situation is that the feature descriptors are compact within a category, and are far apart when they belong to different categories.

3.1 *Shape matters!*

Taking into account the desired properties listed above, we believe that shape is an essential part of a suitable representation for place categorization. By shape, we mean “shape of the scene” [49], i.e. the spatial structure property of an image.

Since the global shape of an image focuses on structural properties, detailed textural information needs to be suppressed. In recognizing place categories, these fine-scaled textures will distract the classifier. They can be noisy and harmful if the feature extraction method is not carefully designed. Figure 1 further illustrates this idea. As shown in Figure 1b, the spatial structure is more prominent in the Sobel image, e.g. the shape that reflects the sink and dishwasher. It is possible to recognize the kitchen category from the Sobel image alone.

It is worth noting that most of the perceptual properties used for scene recognition in [49] are well preserved in Sobel images too. For example, the *degree of naturalness*,



(a) An example kitchen image

(b) Corresponding Sobel image

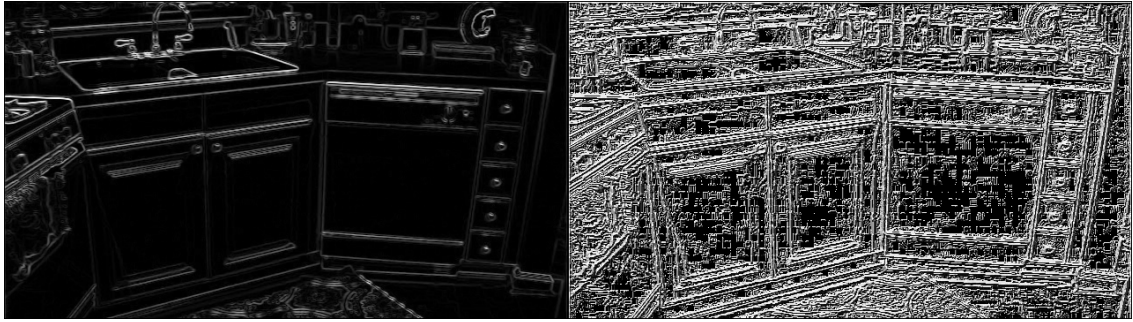
Figure 1: Figure 1a shows an example kitchen image. Figure 1b shows its corresponding Sobel gradients. The Sobel gradients are normalized to $[0 \ 255]$.

defined by the distribution of edges, was used to recognize scenes in [49]. In comparison to the original images, it is easy to read out from the Sobel images that man-made environments have more horizontal and vertical edges, thus they have lower degree of naturalness. Similarly, other spatial structure properties such as the degree of *openness*, *roughness*, and *ruggedness* are also easy to capture in Sobel images,

We are, however, not proposing to use Sobel images directly as a descriptor. On the one hand, structural properties (mainly boundaries/edges) are crucial for recognizing place categories. On the other hand, we need a better descriptor that summarizes such information efficiently. We propose to use Census Transform (CT) histograms as our feature descriptor for the place category recognition task, which is flexible, generalizable, and captures rough geometrical information. We will show that spatial CT histograms efficiently capture the structural properties in an image.

3.2 *Census Transform histograms*

Census Transform (CT) is a non-parametric local transform originally designed for establishing correspondence between local patches [77]. Census transform compares the intensity value of a pixel with its eight neighboring pixels, as illustrated in Eqn. 1. If the center pixel is bigger than (or equal to) one of its neighbors, a bit 1 is set in



(a) kitchen Sobel gradient image

(b) Transformed version

Figure 2: An example *Census Transformed image*.

the corresponding location. Otherwise a bit 0 is set.

$$\begin{array}{c|c|c}
 32 & 64 & 96 \\
 \hline
 32 & \mathbf{64} & 96 \\
 \hline
 32 & 32 & 96
 \end{array}
 \Rightarrow
 \begin{array}{ccc}
 1 & 1 & 0 \\
 1 & & 0 \\
 1 & 1 & 0
 \end{array}
 \Rightarrow (11010110)_2 \Rightarrow \text{CT} = 214 \quad (1)$$

The eight bits generated from intensity comparisons can be put together in any order (we collect bits from top to bottom, and from left to right), which is consequently converted to a base-10 number in $[0\ 255]$. This value is the Census Transform value (CT value) for this center pixel.

Similar to other non-parametric local transforms which are based on intensity comparisons (e.g. ordinal measures [4]), Census Transform is robust to illumination changes, gamma variations, etc. Note that the Census Transform is equivalent (modulo a slight difference in bit ordering) to the *local binary pattern* code $LBP_{8,1}$ [48].

As a visualization method, we create a *Census Transformed image* by replacing a pixel with its CT value. Shown by the example in Figure 2, the Census Transform retains global structures of the picture (especially discontinuities) besides capturing the local structures as it is designed for. Note that Fig. 2a is the same image as Fig. 1b.

Another important property of the transform is that CT values of neighboring pixels are highly correlated. In the example of Figure 3, we examine the direct

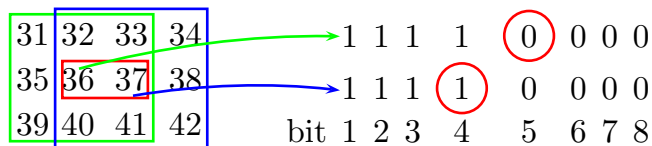


Figure 3: Illustration of constraints between CT values of neighboring pixels. (This picture is best viewed in color.)

constraint posed by the two center pixels. The Census Transform for pixels valued 36 and 37 are depicted in right, and the two circled bits are both comparing the two center pixels (but in different orders). Thus the two circled bits are constrained to be strictly complementary to each other if the two pixels are not equal. More generally, bit 5 of $CT(x, y)$ and bit 4 of $CT(x, y + 1)$ must always be complementary to each other, since they both compare the pixels at (x, y) and $(x, y + 1)$ if these two pixels are not equal. There exist many other such constraints. In fact, there are eight such constraints between one pixel and its eight neighboring pixels.

Besides these deterministic constraints, there also exist indirect constraints that are more complex. For example, in Figure 3, the pixel valued 32 compares with both center pixels when computing their CT values (bit 2 of $CT(x, y)$ and bit 1 of $CT(x, y + 1)$). Depending on the comparison results between the center pixels, there are probabilistic relationships between these bits.

The transitive property of such constraints also make them propagate to pixels that are far apart. For example, in Figure 3, the pixels valued 31 and 42 can be compared using various paths of comparisons, e.g. $31 < 35 < 39 < 40 < 41 < 42$. Similarly, although no deterministic comparisons can be deduced between some pixels (e.g. 34 and 39), probabilistic relationships still can be obtained. The propagated constraints make Census Transform values and Census Transform histograms implicitly contain information for describing global structures, unlike the histogram of pixel values.

Finally, the Census Transform operation transforms any 3 by 3 image region into

one of 256 cases, each corresponding to a special type of local structure of pixel intensities. The CT value acts as an index to these different local structures. No total ordering or partial ordering exists among the CT values. It is important to refrain from comparing two CT values as comparing two integers (like what we do when comparing two pixel intensity values).

A histogram of CT values for an image or image patch¹ can be easily computed, and we use this CT histogram as our visual descriptor. CT histogram can be computed very efficiently. It only involves 16 operations to compute the CT value for a center pixel (8 comparisons and 8 additional operations to set bits to 0 or 1). The cost to compute the CT histogram is linear in the number of pixels of the region we are interested in. There is also potential for further acceleration to the computation of CT histogram, by using special hardware (e.g. FPGA), because it mainly involves integer arithmetic that are highly parallel.

3.2.1 Constraints among CT histogram components

Usually there is not obvious constraints among the components of a histogram. For example, we would often treat the R, G, and B components of a color histogram independent to each other. The CT histogram, however, exhibits strong constraints or dependencies among its components.

Take as example the direct constraint shown in Figure 3, bit 5 of $CT(x, y)$ and bit 4 of $CT(x, y + 1)$ must be complementary to each other if they are not equal. Both bits are 1 if they are equal. If we apply this constraint to all pixels in an image, we get to the conclusion that *the number of pixels whose CT value's bit 5 is 1 must be equal to or greater than² the number of pixels whose CT value's bit 4 is 0*, if we ignore border pixels where such constraints break. Let \mathbf{h} be the Census Transform histogram

¹In fact, CT histogram can be computed for an image region of arbitrary shape.

²These extra 1's are caused by the special case when two neighboring pixels are equal to each other.

of any image. The above statement is translated into the following equation:

$$\sum_{i \& 0x08 = 0x08} \mathbf{h}(i) \geq \sum_{i \& 0x10 = 0} \mathbf{h}(i), \quad (2)$$

where $\&$ is *bitwise and*, $0x08$ is the number 08 in hexadecimal format, and $0 \leq i \leq 255$. Thus the left hand side of Eqn. 2 counts the number of pixels whose CT value's bit 5 is 1. By switching 1 and 0, we get another equation:

$$\sum_{i \& 0x08 = 0} \mathbf{h}(i) \leq \sum_{i \& 0x10 = 0x10} \mathbf{h}(i). \quad (3)$$

Similarly, 6 other linear inequalities can be specified by comparing $CT(x, y)$ with $CT(x - 1, y - 1)$, $CT(x - 1, y)$, and $CT(x - 1, y + 1)$. Thus, any CT histogram resides in a subspace that is defined by these linear inequalities.

We can not write down explicit equations for the indirect or transitive constraints in a CT histogram. However, we expect these constraints will further reduce the dimension of the subspace of CT histograms. A CT histogram, although having 256 bins, is living in a subspace whose dimension is much lower than 256.

3.3 *CT histogram encodes image structures*

In order to understand why CT histogram efficiently captures the essence of a place image, it is worthwhile to further examine the distribution of CT values and CT histograms. Using images from the 15 class scene dataset [33], we find that the 6 CT values with highest frequencies are $CT = 31, 248, 240, 232, 15, 23$ (excluding 0 and 255). As shown in Figure 4b-4g, these CT values correspond to local 3×3 neighborhoods that have either horizontal or various close-to-diagonal edge structures. It is counter-intuitive that vertical edge structures are not among the top candidates. A possible explanation is that vertical structures are usually appearing to be inclined in pictures because of the perspective nature of cameras.

CT histogram of the example ellipse image (Figure 4a) is shown in Figure 4h. It summarizes the distribution of various local structures in the image. Because of the

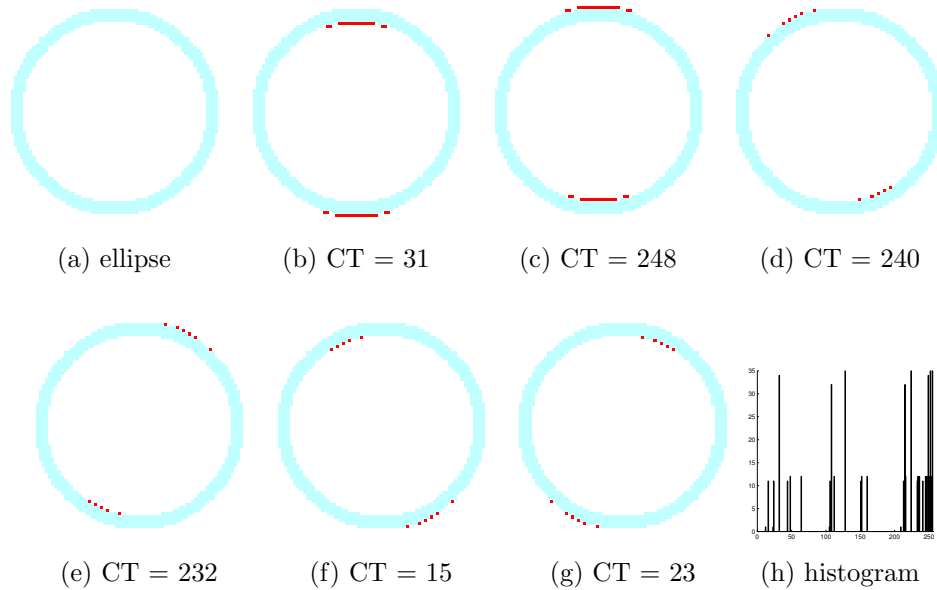


Figure 4: Illustration of Census transforms. 4a is an example image of ellipse. 4b-4g show pixels having the 6 highest frequency CT values (shown in red). 4h is the CT histogram of 4a. (This image is best viewed in color.)

strong correlation of neighboring CT values, the histogram cells are not independent of each other. On the contrary, a CT histogram implicitly encodes strong constraints of the global structure of the image. For example, if an image has a CT distribution close to that of Figure 4h, we would well expect the image to exhibit ellipse shape with a high probability (c.f. Section 3.4 for more evidence.)

A simplification to the one dimensional case better explains the intuition behind our statement. In 1-D there are only 4 possible CT values, and the semantic interpretation of these CT values are obvious. As shown in Figure 5a, the four CT values are $CT = 0$ (valley), $CT = 1$ (downhill), $CT = 2$ (uphill), and $CT = 3$ (peak). For simple shapes in 1-D, the CT histograms encode shape information and constraints. Downhill shapes and uphill shapes can only be connected by a valley, and uphill shapes require a peak to transit to downhill shapes. Because of these constraints, the only other shapes that has the same CT histogram as that of Figure 5a is those shapes that cut a small portion of the left part of Figure 5a and move it to the right.

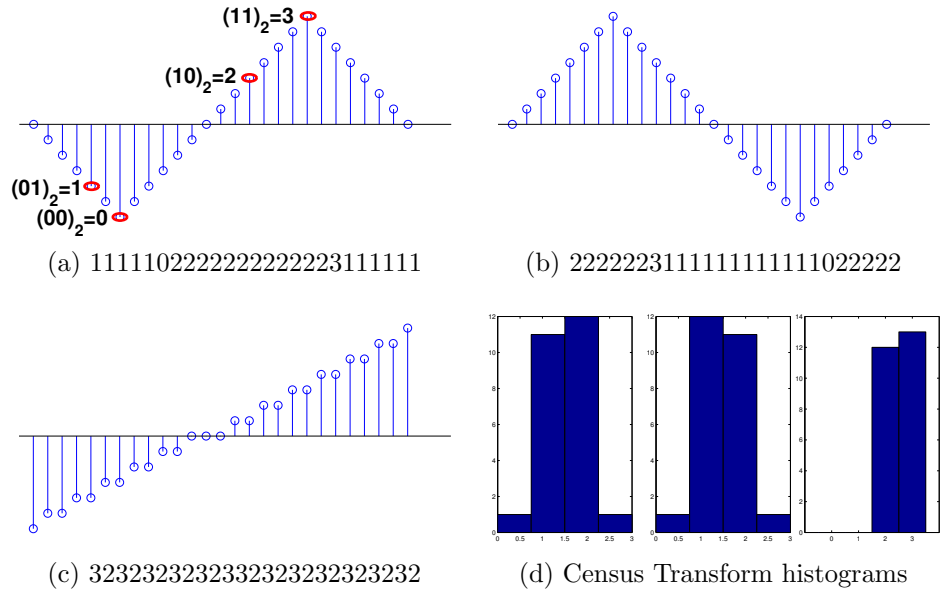


Figure 5: Census Transform encodes shape in 1-D. The Census Transform values of (a)-(c) are shown in the caption. Sub-figure (d) shows the CT histograms of figures (a)-(c), respectively. Both end points are ignored in compute CT. (This image is best viewed in color.)

Images that are different but keep the shapes (e.g. Figure 5b) also are similar in their CT histograms (Figure 5d). On the contrary, a huge number of possible curves have the same intensity histogram as that of Figure 5a. Even if we impose smoothness constraints between neighboring pixel intensities, the shape ambiguity is still large (e.g. Figure 5c is smooth and has the same intensity histogram as that of Figures. 5a and 5b, but it has different shape and a very different CT histogram).

3.4 Reconstructing patches from CT histograms

It is well known that spatial information is totally lost in the histogram of pixel intensities. The CT histogram, however, implicitly retains the global structure of an image patch through the constraints we have discussed. We performed some reconstruction experiments to further illustrate this idea. When we randomly shuffle the pixels of an input image, the original structure of the image is completely lost. Using the shuffled image as an initial state, we repeatedly change two pixels at one

time, until the current state has the same CT histogram as the input image. This optimization is guided by the Simulated Annealing algorithm, and the algorithm terminates when the current state has the same CT histogram as the input image. If structure of the original image is observed in the reconstruction result (i.e. the termination state), this is an evidence that structure of an image is (at least partially) encoded in its CT histogram.

In the reconstruction results in Figure 6, the left image in each subfigure is the input image. A pair of pixels in the input images are randomly chosen and exchanged. The exchange operation is repeated multiple times (equal to the number of pixels in the input image), which gives the initial state for the reconstruction. Our goal is to find an image that has the same CT histogram as the input. Thus the cost function is set to the Euclidean distance between CT histograms of the current state and the input. Simulated Annealing is used to minimize the cost function to 0. The terminating state is output of the reconstruction (right image in each subfigure of Figure 6).

Although the initial states look like random collection of pixels, many of the reconstruction results perfectly match the input images (subfigure (a)-(g) in Fig. 6). More examples are reconstructed with minor discrepancies (subfigure (h)-(p) in Fig. 6). Large scale structures of the input digits and characters are successfully reconstructed in these images, with small errors. In the rest examples, e.g. ‘2’ and ‘e’, major structures of the original input images are still partially revealed. These results empirically validated that CT histograms have the ability to encode the shape of an image.

An analogy to these results is the jigsaw puzzle. The CT value in each pixel location is analogous to a puzzle piece of certain type. Pieces can be put next to each other only if their shapes satisfy certain constraints (similar to the constraints between neighboring CT values). After breaking a puzzle into pieces, there are only a very limited number of ways to assemble the pieces together, and we would expect

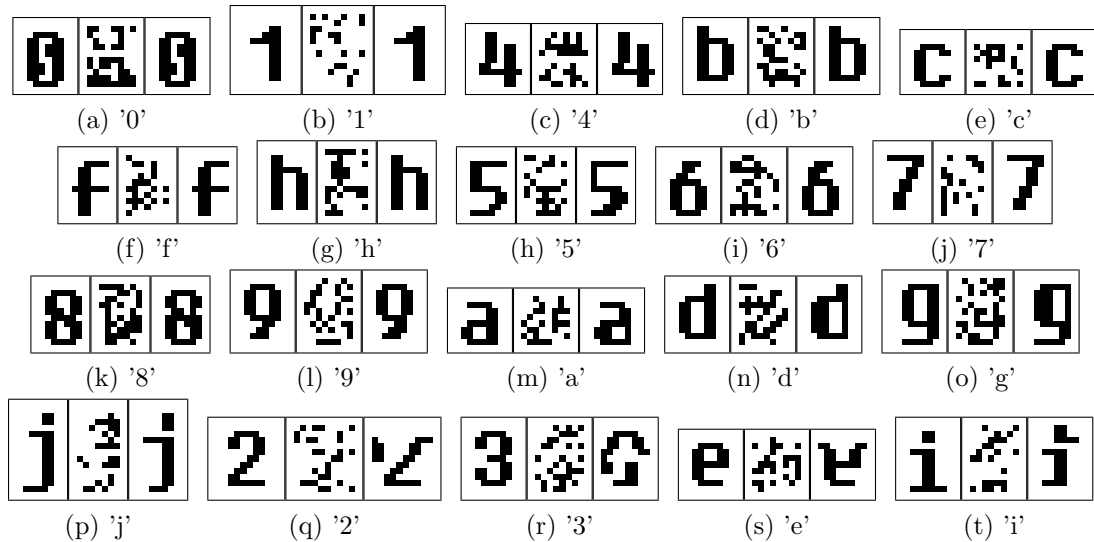


Figure 6: Reconstruct images from CT histograms. In each group of images, the left image is the input image. The image in the middle is the initial starting point that is generated by randomly exchanging pairs of pixels in the input. The reconstruction result is shown in the right of each group, which has the same CT histogram as the input image.

the assembled version to resemble the original one with a high probability.

Similarly, there are a huge number of ways to shuffle pixels of an input image (possibly exponential in the number of pixels). However, if we add an additional constraint that the CT histogram should be same as the input image, there is only a small number of possibilities. As shown in Fig. 6, these remaining reconstructions have a large chance to share same or similar structure as the input image.

Two points are worth pointing out about the reconstruction results. First, in larger images a CT histogram is not enough to reconstruct the original image.³ However, as a feature descriptor, it has the ability to distinguish between images with different structural properties. Second, it is essentially impossible to reconstruct a small image using other descriptors (e.g. SIFT or Gist).

³Note that the images in Figure 6 are black-and-white images instead of gray-scale ones.

3.5 Comparing CT histogram, SIFT, and Gist

In this section we further compare the CT histogram descriptor to the SIFT and Gist descriptors.

As mentioned in Section 2, we observe that the perceptual properties Gist is modeling are mainly valid for outdoor environments. Our experiments on the 8 outdoor scene categories [49] and the 15 scene categories (which is a super set of the 8 category dataset) further corroborated this observation. Using the Gist descriptor⁴ and SVM classifier, the recognition accuracy was $82.60 \pm 0.86\%$ on the 8 outdoor categories, which is lower than $85.65 \pm 0.73\%$, the accuracy using CT histogram on this dataset.

However, on the 15 class dataset which include several indoor categories, the accuracy using Gist dramatically dropped to $73.28 \pm 0.67\%$, which is significantly lower than CT histogram’s accuracy, $83.10 \pm 0.60\%$. Our conjecture is that the frequency domain features in the Gist descriptor are not discriminative enough to distinguish between the subtle differences between indoor categories, e.g. bedroom vs. living room.

On the contrary, SIFT is originally designed to have high discriminative power. Thus it may not be able to cope with the huge intra-class variation in place images. For any two feature vectors, we can compute their Histogram Intersection Kernel (HIK) value [61] as a simple measure for the similarity between them. Please refer to Chapter 4 for the exact definition of HIK in Eqn. 4. By observing the similarity distribution between- and within- categories, which are shown in Figure 7 for both SIFT and CT histograms, we can have an estimate of their capability in place and scene recognition. Note that we scaled both feature descriptors so that the similarity score will be between 0 and 1024.

⁴<http://people.csail.mit.edu/torralba/code/spatialenvelope/>

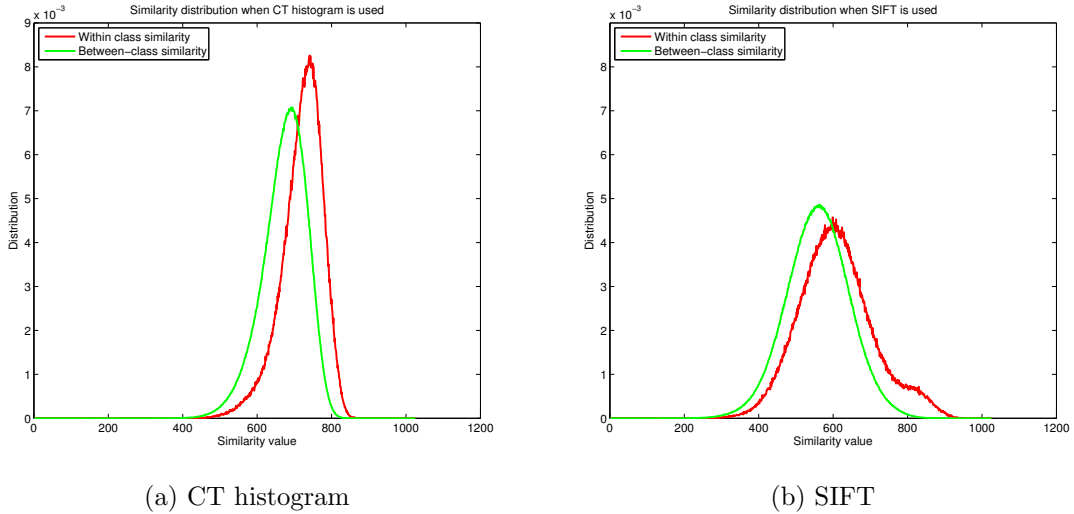


Figure 7: Distribution of similarity values on the 15 class scene recognition dataset. The red curve shows within-class similarity distribution, i.e. similarity of two feature descriptors computed from images in the same categories. The green curve shows inter-class similarity distribution, i.e. similarity of two descriptors computed from images in different categories. (This figure is best viewed in color.)

Figure 7a shows that on the scene recognition dataset, the within-class distribution of CT histogram is well separated from the inter-class distribution. The area of the region when the within-class curve is on top of the inter-class curve is 0.34. However, in Figure 7b, the area is only 0.20 for the SIFT descriptor, and the two curves are only separable when the similarity score is very high. Figure 7 indicates that CT histogram is better suited to scene recognition than SIFT. However, we observe different trend in the object recognition task, as shown in Figure 8. The area for CT histogram and SIFT are 0.19 and 0.25 respectively. Thus, while CT histogram is suitable for place and scene recognition, for tasks requiring high discriminative power such as object recognition, SIFT is a better choice.

The information in Figures 7 and 8, although indicative, is not a direct measure of classification accuracy. Figure 9 shows more directly how accurate these two feature descriptors will be using a baseline nearest neighbor classification rule. For any image, we can find its nearest neighbor in the same category and the nearest neighbor in

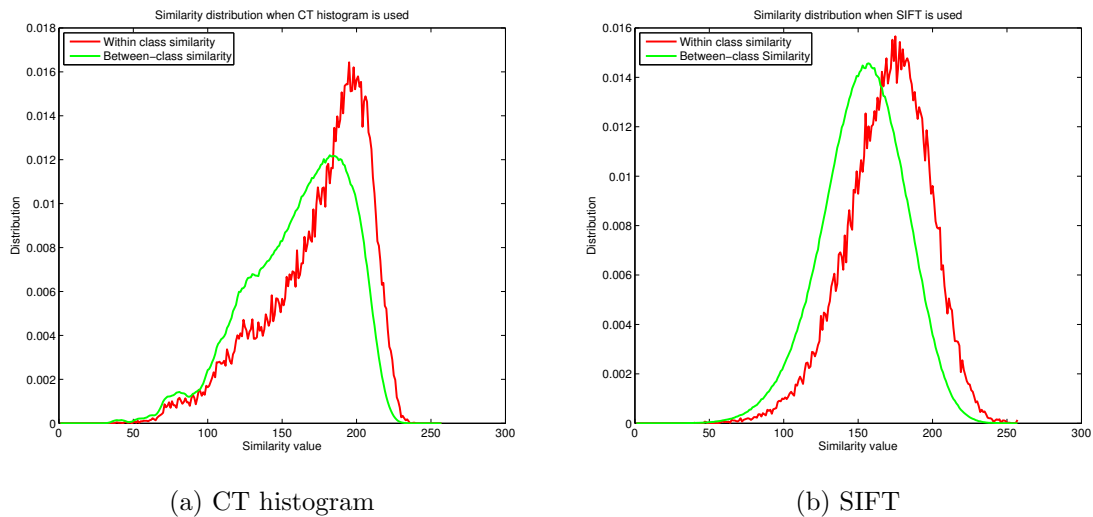


Figure 8: Distribution of similarity values on the Caltech 101 object recognition dataset. (This figure is best viewed in color.)

a different category. If the out-of-category nearest neighbor has a higher similarity value than the in-category nearest neighbor, the simple nearest neighbor classifier will make a wrong decision for this image. In Figure 9 the x-axis shows the difference of these two similarity values. In other words, a value in the left hand side of 0 (the black line) means an error. For any given curve, if we find area of the part that is at the left hand side of the black dashed line, and divide it by area of the entire curve, we get the leave one out estimation of the classification error of a nearest neighbor rule. Thus Figure 9 is an indication of the discriminative power of the descriptors. We observe the same trend as what is shown in Figures 7 and 8. CT histogram has an advantage in recognizing place and scene images (35.83% error, compared to 57.24% for SIFT), while SIFT is suitable for object recognition (67.39% error, compared to 83.80% for CT histogram).

Further intuitions are illustrated in Figure 10. We build a visual codebook with 256 visual code words using the 15 class scene recognition dataset. Details of visual codebook will be provided in Chapter 4. Given an input image, an image patch with coordinates $[x - 8, x + 8) \times [y - 8, y + 8)$ can be mapped to a single integer by the

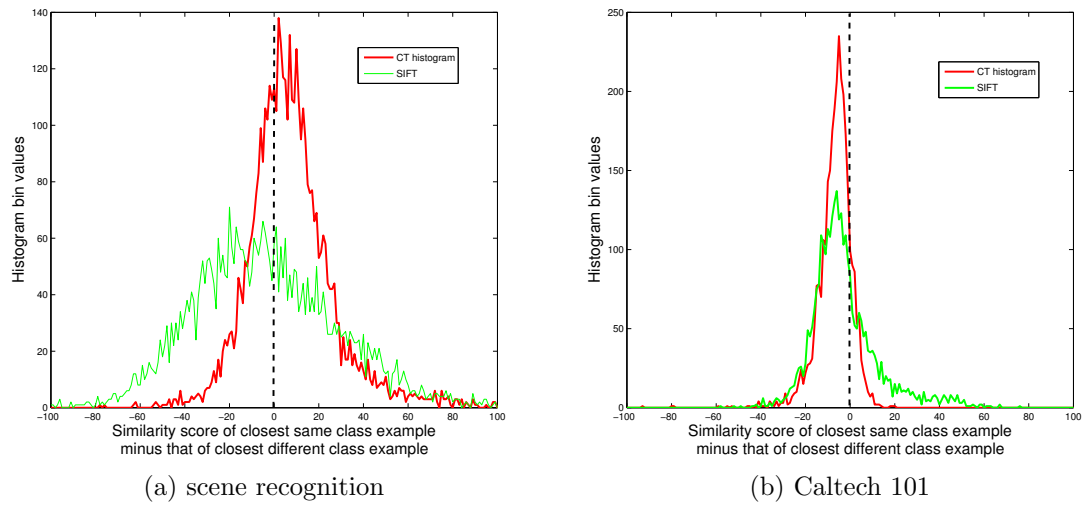


Figure 9: Histogram comparing similarity values of best in-category nearest neighbor with best out-of-category nearest neighbor of an image. (This figure is best viewed in color.)

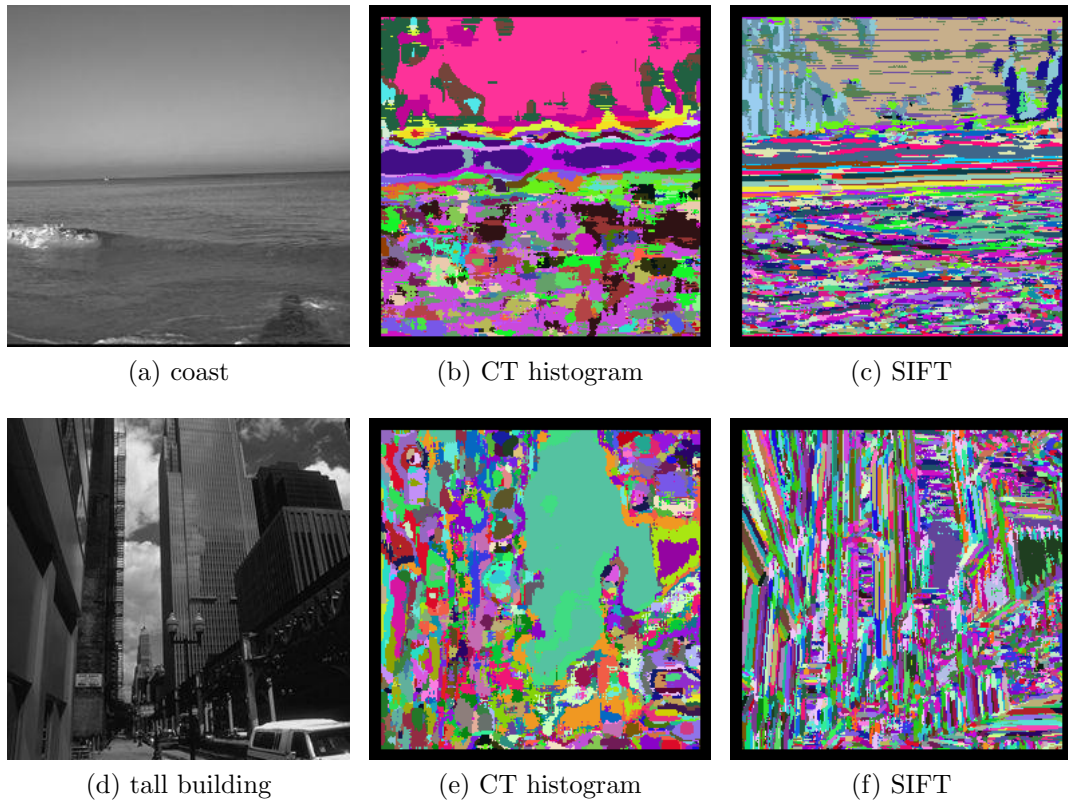


Figure 10: Visualization of images mapped to visual code words. In each row, the first image is an input image, with the second and third being visualization for CT histogram and SIFT codebooks, respectively. (This picture needs to be viewed in color.)

following procedure. We first extract the CT histogram from this window (whose size is 16 by 16). This CT histogram vector is compared to all codewords, and the index of the nearest neighbor is the mapping result for pixel position (x, y) . By choosing a random RGB tuple for each codeword index, a gray scale image can be transformed into a visualization of corresponding code word indexes.

Figure 10 are examples of the codebook visualization results for a coast and a tall building image. The SIFT code words tend to emphasize discontinuities in the images. Edges (especially straight lines) usually are mapped to the same codeword (*i.e.*, displayed in the same color in the visualization). The visualization also suggests that SIFT pays more attention to detailed textural information, because the visualization is fragmented (connected component of the same color is small). Image patches with similar visual structure and semantics are mapped to different visual code words, e.g. the tall building in the right half of Fig. 10d.

Instead, CT histogram visualizations tend to group image regions with similar visual structure into the same code word. The connected component in CT histogram visualizations are larger than those in the SIFT visualizations. For example, the sky in the coast image share similar semantics and visual structures. This region is mostly mapped to the same color (*i.e.* same code word) using CT histogram, which is desirable for the scene category recognition task. Instead, the SIFT descriptor maps this region to different colors.

3.6 Limitations of the CT histogram

As we have stated from the very beginning, CT histogram is designed to be a representation that suits place recognition and categorization problems, *i.e.* capturing shape of the scene. This design choice renders limitations that prevent it from being applied in some applications. We list the limitations below, and explain how these limitations affect the place category recognition performance.

- CT histogram is sensitive to rotations. Thus it is not suitable for 3-D or multiview object recognition, e.g. the Caltech 101 dataset [17] or the Kentucky recognition benchmark dataset [46].⁵ However, in place recognition images are always taken in the upright view and we usually pay attention to the overall shape of the scene, which is not prone to rotational variances. Furthermore, CT histogram is invariant to translation and robust against scale changes;
- CT histogram is not a precise shape descriptor. It is designed to recognize shape categories, but not for exact shape registration applications, e.g. the shape retrieval task in [19, 35].
- CT histogram ignores color information.

3.7 Summary of the chapter

We analyzed a few high level requirements for a place categorization representation / descriptor. We want a descriptor which is flexible to accommodate the huge variations. We want it to incorporate rough geometrical information and has relatively high discriminative power to work in indoor environments. We also want it to be a category level descriptor, i.e. not confined to any specific instance in a category, but generalizable to new instances.

We then proposed the Census Transform histogram as our descriptor. A CT histogram encodes local shapes in 3×3 local neighborhoods through the Census Transform. Unlike the histogram of pixel intensities which totally ignores spatial information, histogram of Census Transform values encodes the shape in an image patch through the strong correlation among neighboring CT values. As argued in this chapter, we believe that CT histogram captures the structural properties of an image, instead of detailed textural information. The emphasis of this chapter is to

⁵<http://vis.uky.edu/~stewe/ukbench/>

analyze this descriptor and provide intuition for understanding why CT histogram encodes spatial structures in an image. We also compare the CT histogram with both SIFT and Gist descriptors, in the context of place and scene recognition.

CHAPTER IV

HIK VISUAL CODEBOOK

As we have pointed out in Chapter 3, although the CT histogram descriptor does not contain enough information to reconstruct a large image patch, it encodes useful information about global image structure for categorizing place categories. Similar to many scene category and object category recognition systems, we apply visual codebook to vector quantize CT histograms in visual place categorization.

A visual codebook is a method that divides the feature descriptor space into several regions. Features in one region correspond to the same *visual code word*, which is indexed by an integer between 1 and size of the codebook. Visual codebook are usually generated by clustering methods.

The vector quantization operation has at least two benefits. First, if an appropriate feature descriptor and a suitable distance metric are used, we expect that semantically similar image patches are mapped to the same visual code word, while ignoring their fine scale textural differences. Second, vector quantization significantly reduces dimensionality of our feature descriptor. For example, if we divide an image into $4 \times 4 = 16$ blocks and extract CT histogram for each block, the overall feature descriptor will be $16 \times 256 = 4096$ dimensional. After applying a visual codebook method, each CT histogram is mapped into a single integer and we only need to deal with a new 16 dimensional feature vector.

4.1 The need for HIK based codebook

No matter whether a visual codebook is generated by a supervised or unsupervised method, comparing the similarity (or, distance or dissimilarity) is always a crucial component in these methods. As discussed in Chapter 2, existing codebook generation

methods all apply the Euclidean distance to compare two vectors. However, our feature descriptor is census transform histogram, a histogram feature. For the case of supervised classification, it has been repeatedly shown that l_2 distance is not the most effective method for comparing two histograms (e.g. in [40]). In particular, the Histogram Intersection Kernel (HIK) was demonstrated to give improved performance than linear kernel (which corresponds to the Euclidean distance).

HIK was introduced by Swain and Ballard [61] for color-based object recognition. [47] demonstrated that HIK forms a positive definite kernel, facilitating its use in SVM classifiers. Simultaneously, works such as [38, 11] demonstrated the value of histogram features for a variety of tasks. However, the high computational cost of HIK at run-time remained a barrier to its use in practice. This barrier was removed for the case of SVM classifiers by the work of Maji, Berg and Malik, who presented a technique to accelerate the kernel evaluations [40].

It is natural to extend this idea to the unsupervised learning paradigm and visual codebook generation. The main point of this chapter is that when histogram features are employed, the histogram intersection kernel (HIK) should be used to compare them, including the visual codebook generation task. It is worth noting that the proposed Algorithm 1 is not confined to generating codebooks for the CT histogram descriptor. Instead, it is readily applicable to popular feature descriptors such as SIFT [38] and HOG [11].

4.2 *Generating visual codebooks using HIK*

Let $\mathbf{h} = (h_1, \dots, h_d) \in R_+^d$ be a histogram. \mathbf{h} could represent an image (e.g. histogram of code words) or a sub-window (e.g. SIFT feature descriptor). The histogram intersection kernel κ_{HI} is defined as follows

$$\kappa_{\text{HI}}(\mathbf{h}_1, \mathbf{h}_2) = \sum_{i=1}^d \min(h_{1i}, h_{2i}). \quad (4)$$

It is proved that κ_{HI} is a valid positive definite kernel [47]. Thus there exists a mapping ϕ that maps any histogram \mathbf{h} to a corresponding vector $\phi(\mathbf{h})$ in a high dimensional (possibly infinite dimensional) feature space Φ , such that $\kappa_{\text{HI}}(\mathbf{h}_1, \mathbf{h}_2) = \phi(\mathbf{h}_1) \cdot \phi(\mathbf{h}_2)$. Through the nonlinear mapping ϕ , histogram similarity is equivalent to an inner product in the feature space Φ .

This kernel trick makes it possible to use the histogram intersection kernel in creating codebooks, while keeping the simplicity of k-means clustering. We propose to use a histogram kernel k-means algorithm to generate visual codebooks. In Algorithm 1 (page 42), by kernel k-means in the feature space spanned by ϕ , histograms are compared using HIK instead of the inappropriate Euclidean distance.

Note that since *k-means++* used in Algorithm 1 is a randomized algorithm, two runs of Algorithm 1 with the same input will possibly generate different results.

4.3 Fast Evaluation

The major component of Algorithm 1 is a kernel k-means algorithm [58] using κ_{HI} , the Histogram Intersection Kernel. Since the centers \mathbf{m}_i are vectors in the unrealized, high dimensional space Φ , the key computation is carried out in the following way (using the usual kernel trick $\phi(\mathbf{x}) \cdot \phi(\mathbf{y}) = \kappa_{\text{HI}}(\mathbf{x}, \mathbf{y})$ such that \mathbf{m}_i does not need to be explicitly generated.):

$$\begin{aligned}
& \|\phi(\mathbf{h}_*) - \mathbf{m}_i\|^2 \\
&= \left\| \phi(\mathbf{h}_*) - \frac{\sum_{j \in \pi_i} \phi(\mathbf{h}_j)}{|\pi_i|} \right\|^2 \\
&= \|\phi(\mathbf{h}_*)\|^2 + \frac{\sum_{j,k \in \pi_i} \kappa_{\text{HI}}(\mathbf{h}_j, \mathbf{h}_k)}{|\pi_i|^2} - \frac{2 \sum_{j \in \pi_i} \kappa_{\text{HI}}(\mathbf{h}_*, \mathbf{h}_j)}{|\pi_i|}. \tag{6}
\end{aligned}$$

The first term in Eqn. 6 does not affect the result in lines 5 and 9 of Algorithm 1, and the second term can be pre-computed. Thus most of the computations are spent in computing the last term $\sum_{j \in \pi_i} \kappa_{\text{HI}}(\mathbf{h}_*, \mathbf{h}_j)$.

Algorithm 1 HIK Visual Codebook Generation

- 1: {Given n histograms $\mathbf{h}_1, \dots, \mathbf{h}_n$ in R_+^d , m (size of the codebook), and ε (tolerance).}
- 2: {The output is a mapping from a histogram to a code word index, $w_1(\mathbf{h}_*) : R_+^d \rightarrow \{1, \dots, m\}$.}
- 3: Use the *k-means++* method [2] to choose m histograms $\bar{\mathbf{h}}_1, \dots, \bar{\mathbf{h}}_m$, and use $\mathbf{m}_i = \phi(\bar{\mathbf{h}}_i)$ as initial centers. ϕ is the mapping associated with κ_{HI} .

4: **repeat**

- 5: For all $1 \leq i \leq n$,

$$l_i \leftarrow \arg \min_{1 \leq j \leq m} \|\phi(\mathbf{h}_i) - \mathbf{m}_j\|^2.$$

(Set l_i to index of the center that is closest to \mathbf{h}_i .)

- 6: For all $1 \leq i \leq m$,

$$\pi_i = \{j | l_j = i, 1 \leq j \leq n\}.$$

(Set π_i to the set of indexes that belong to the center \mathbf{m}_i .)

- 7: For all $1 \leq i \leq m$,

$$\mathbf{m}_i \leftarrow \frac{\sum_{j \in \pi_i} \phi(\mathbf{h}_j)}{|\pi_i|}.$$

(Update the centers.)

8: **until**

$$\max_{1 \leq i \leq m} \|\phi(\mathbf{h}_i) - \mathbf{m}_i\|^2 \leq \varepsilon.$$

(Change of the cost function value is small enough)

- 9: **output:** For any histogram $\mathbf{h}_* \in R_+^d$,

$$w_1(\mathbf{h}_*) = \arg \min_{1 \leq i \leq m} \|\phi(\mathbf{h}_*) - \mathbf{m}_i\|^2. \quad (5)$$

Note that this term is similar to the binary SVM classifier using HIK, which has the following form

$$\text{sgn} \left(\sum_{i \in \pi} \alpha_i y_i \kappa_{\text{HI}}(\mathbf{h}_*, \mathbf{h}_i) + \rho \right) \quad (7)$$

where \mathbf{h}_i , α_i , and y_i are support vectors and their corresponding weights and labels.

A naive method will take $O(|\pi_i|d)$ steps to compute Eqn. 6. In [40], Maji, Berg and Malik proposed fast methods to compute Eqn. 7 (exact answer in $O(d \log |\pi|)$ steps and approximate answer in $O(d)$ steps.) In this chapter we generalize their method and propose a variant that find the exact answer for Eqn. 6 in $O(d)$ steps.

Note that both Eqn. 7 and the last term in Eqn. 6 are special forms of the following expression

$$f(\mathbf{h}_*) = \sum_{i \in \pi} c_i \kappa_{\text{HI}}(\mathbf{h}_*, \mathbf{h}_i), \quad (8)$$

where π indexes a set of histograms (support vectors) and c_i are constant coefficients.

A histogram representing an image has the property that every histogram component is a non-negative integer, i.e. it is a vector in N_+^d . Similarly, a feature descriptor histogram can usually be transformed into the space N_+^d . For example, the SIFT descriptors are stored as vectors in N_+^{128} . In general, a vector in R_+^d can be transformed into N_+^d by first multiplying an integer to the histogram and then rounding its components to nearest integers. Note that the CT histogram contains integers in all dimensions (histogram cells).

In the rest of this chapter we assume that any histogram $\mathbf{h} = (h_1, \dots, h_d)$ satisfies that $h_i \in N_+$ and $h_i < h_{\max}$ for all i . Then the quantity $f(\mathbf{h}_*)$ can be computed as

follows,

$$\begin{aligned}
f(\mathbf{h}_*) &= \sum_{i \in \pi} c_i \kappa_{\text{HI}}(\mathbf{h}_*, \mathbf{h}_i) \\
&= \sum_{i \in \pi} \sum_{1 \leq j \leq d} c_i \min(h_{*j}, h_{ij}) \\
&= \sum_{1 \leq j \leq d} \left(\sum_{i \in \pi} c_i \min(h_{*j}, h_{ij}) \right) \\
&= \sum_{1 \leq j \leq d} \left(\sum_{h_{*j} \geq h_{ij}} c_i h_{ij} + h_{*j} \sum_{h_{*j} < h_{ij}} c_i \right). \tag{9}
\end{aligned}$$

Note that the two summands in Eqn. 9 can both be pre-computed. [40] approximated histogram components by uniformly sampled points in the range of possible values in order to achieve the $O(d)$ speed. However, since we can assume that h_{*j} is an integer in the range $0..h_{\max}$, we have an even faster computing method (less overhead).

Let T be a table of size $d \times h_{\max}$, with $\sum_{k \geq h_{ij}} c_i h_{ij} + k \sum_{k < h_{ij}} c_i$ being assigned to the (j, k) -th entry $T(j, k)$, $1 \leq j \leq d, 1 \leq k \leq h_{\max}$. Then it is clear that

$$f(\mathbf{h}_*) = \sum_{j=1}^d T(j, h_{*j}). \tag{10}$$

This method is summarized in Algorithm 2.

It is obvious that $f(\mathbf{h}_*)$ can be evaluated in $O(d)$ steps after the table T is pre-computed. Because Algorithm 2 only involves table lookup and summation, it is faster (has less overhead) than the approximation scheme in [40], which is also $O(d)$. Depending on the relative size of h_{\max} and the number of approximation bins used in [40], Algorithm 2's storage requirement ($O(h_{\max}d)$) could be larger or smaller than that of [40]. The pre-computation of T requires $O(dn h_{\max})$ steps. Although filling in the table T is computationally expensive, it is done only once.¹ It is also worth noting that under our assumption Algorithm 2's result is precise rather than approximate.

¹A better method is to first bucket sort each dimension of the histograms then fill in the values of T sequentially, which takes only $O(d(n + h_{\max}))$ steps.

Algorithm 2 Fast Computing of HIK Sums

- 1: {Given n histograms $\mathbf{h}_1, \dots, \mathbf{h}_n$ in N_+^d , with $0 \leq h_{ij} \leq h_{\max}$ for $1 \leq i \leq n, 1 \leq j \leq d$ }
- 2: {The output is a fast method to compute $f(\mathbf{h}_*) = \sum_{i=1}^n c_i \kappa_{\text{HI}}(\mathbf{h}_*, \mathbf{h}_i)$, where $\mathbf{h}_* \in N_+^d$.}
- 3: Create T , a $d \times h_{\max}$ table.
- 4: Set

$$T(j, k) \leftarrow \sum_{k \geq h_{ij}} c_i h_{ij} + k \sum_{k < h_{ij}} c_i$$

for $1 \leq j \leq d, 1 \leq k \leq h_{\max}$.

- 5: **output:**

$$f(\mathbf{h}_*) = \sum_{j=1}^d T(j, h_{*j}).$$

Both the pre-computation complexity and storage requirement are linear in h_{\max} , which is a parameter specified by users. Our experiments show that while a small h_{\max} usually produces inferior results, larger h_{\max} does not necessarily improve system performance. We choose $h_{\max} = 128$, which seems to give the best results in our experiments.

Algorithm 1 has the same complexity as a usual linear k-means when generating a visual codebook or mapping from histograms to visual codeword indexes (Eqn. 5 or 6). In practice the proposed method takes about twice the time of k-means. In summary, the proposed method generates a visual codebook that can not only run almost as fast as the k-means method, but also can utilize the non-linear similarity measure κ_{HI} that is suitable for comparing histograms.

4.4 *One-class SVM codeword generation*

A codebook generated by the k-means algorithm first divides the space R_+^d into m regions, then represents each code word (region) by the centroids of those vectors (histogram, feature vectors, etc.) that fall into this region. This approach is optimal if we assume that vectors in all regions follow the Gaussian distributions with the same spherical covariance matrix (only differ in their means).

This assumption rarely holds. Different regions usually have very different densities and covariance structures. Simply dividing the space R_+^d into a Voronoi diagram from the set of region centers is, in many cases, misleading. However, further refinements are usually computationally prohibitive. For example, if we model regions as Gaussian distributions with distinct covariance matrices, the generation and mapping from visual features to code words will require much more storage and computational resources than we can afford.

We propose to use one-class SVM [57] to represent the divided regions in an effective and computationally efficient way. Given a set of histograms in a region $\mathbf{h}_\pi = \{\mathbf{h}_1, \dots, \mathbf{h}_n\}$, we construct a one-class SVM with parameter $\nu \in (0, 1]$,

$$\text{sgn} \left(\sum_{i \in \pi} \alpha_i \kappa_{\text{HI}}(\mathbf{h}, \mathbf{h}_i) - \rho \right) \quad (11)$$

where α_i 's are non-negative, sparse and $\sum_i \alpha_i = 1$. Intuitively, a one-class SVM classifier seeks a simple (compact) subset of \mathbf{h}_π (or the divided region) that retains a large portion of the histograms (or densities). It is proved that ν is the upper bound on the fraction of outliers (i.e. on which Eqn. 11 is less than 0), and at the same time a lower bound on the fraction of support vectors (i.e. $\alpha_i \neq 0$) [57].

The one-class SVM (using HIK) summarizes the distribution of histograms inside a region (i.e. a code word). It takes into consideration the shape and density of the histogram distribution. It seeks to include most of the histograms (at least $(1 - \nu)|\pi|$) in a compact hypersphere in the feature space, while paying less attention to those borderline cases (at most $\nu|\pi|$ examples). We believe that this compact hypersphere better summarizes a visual code word.

At the same time, these new code words can be computed very efficiently. Eqn. 11 is evaluated in $O(d)$ steps because it is again a special case of Algorithm 2. We propose Algorithm 3 to use the one-class SVM to generate visual code words.² We set the

²Note that we use the space R_+^d because Algorithm 3 is not restricted to N_+^d .

Algorithm 3 One-class SVM Code Word Generation

- 1: {Use Algorithm 1 to generate the divisions π_i ($i = 1, \dots, m$) from n histogram $\mathbf{h}_1, \dots, \mathbf{h}_n$ in R_+^d . }
- 2: For each division $1 \leq i \leq m$, train a one-class SVM from its data \mathbf{h}_{π_i} with a parameter ν ,

$$w_2^i(\mathbf{h}_*) = \text{sgn} \left(\sum_{j \in \pi_i} \alpha_j \kappa_{\text{HI}}(\mathbf{h}_*, \mathbf{h}_j) - \rho_i \right) \quad (12)$$

- 3: **output:** For any histogram $\mathbf{h}_* \in R_+^d$,

$$w_2(\mathbf{h}_*) = \arg \max_{1 \leq i \leq m} w_2^i(\mathbf{h}_*). \quad (13)$$

parameter $\nu = 0.2$.

In many applications a histograms $\mathbf{h} = (h_1, \dots, h_d)$ satisfy that $\|\mathbf{h}\|_1 = \sum_{i=1}^d h_i = N$ is a constant. Under this condition, Eqn. 12 is equivalent to

$$w_2^i(\mathbf{h}_*) = \text{sgn} \left(r_i^2 - \|\phi(\mathbf{h}_*) - \mathbf{m}_i\|^2 \right)$$

where $\mathbf{m}_i = \sum_{j \in \pi_i} \alpha_j \mathbf{h}_j$ and $r_i^2 = N + \|\mathbf{m}_i\|^2 - 2\rho_i$. In other words, a histogram is considered as belonging to the i -th visual word if it is inside the sphere (in the feature space Φ) centered at \mathbf{m}_i with radius r_i . A sphere in Φ is different from the a usual k-means sphere (in R_+^d) because it respects the similarity measure κ_{HI} , and its radius r_i may differ with each other.

Although the proposed algorithms are described using the histogram intersection kernel κ_{HI} , they are readily applied to other kernel types. For example, if we use the linear kernel $\kappa_{\text{LIN}}(\mathbf{h}_1, \mathbf{h}_2) = \mathbf{h}_1 \cdot \mathbf{h}_2$, Algorithm 1 will reduce to a usual k-means visual codebook generation method, and Algorithm 3 will perform one-class SVM in the space R_+^d .

4.5 *K-median codebook generation*

Although k-mean (or, equivalently κ_{LIN} or l_2 distance) is the most popular codebook generation method, the histogram intersection kernel has a closer connection to the l_1

distance. For two numbers a and b , it is easy to show that $2 \min(a, b) + |a - b| = a + b$. As a consequence, we have

$$2\kappa_{\text{HI}}(\mathbf{h}_1, \mathbf{h}_2) + \|\mathbf{h}_1 - \mathbf{h}_2\|_1 = \|\mathbf{h}_1\|_1 + \|\mathbf{h}_2\|_1. \quad (14)$$

In cases when $\|\mathbf{h}\|_1$ is constant for any histogram \mathbf{h} , κ_{HI} and the l_1 distance are linearly correlated.

For an array x_1, \dots, x_n , it is well known that the value that minimizes the l_1 error ($x_* = \arg \min_x \sum_{i=1}^n |x - x_i|$) equals the median value of the array. Thus k-median is a natural alternative for codebook generation.³

K-median has been less popular than k-means for the creation of visual codebooks. An online k-median algorithm has been used by Larlus and Jurie to create visual codebooks in the Pascal challenge [16]. In this chapter we will compare the batch version of k-median to k-means and the proposed HIK method.

4.6 Validation of HIK codebooks

In this section we will present brief experimental results that support the proposed algorithms. Experiments in this chapter were tested on scene and object recognition datasets. Since semantic categories are recognized in scene recognition, the success in scene recognition not only hints on the ability of HIK codebooks for the visual place categorization problem. They also, together with results on object recognition, show that HIK codebooks can be applied to broader domains.

We validate the proposed methods using three datasets: the Caltech 101 object recognition dataset [17], the 15 class scene recognition dataset [33], and the 8 class sports events dataset [34]. The 15 class scene recognition dataset was built gradually by Oliva and Torralba ([49], 8 classes), Fei-Fei and Perona ([18], 13 classes), and Lazebnik, Schmid and Ponce ([33], 15 classes). Categories in this dataset include

³The only difference between k-median and k-means is that k-median uses l_1 instead of l_2 as the distance metric.



Figure 11: Images from 15 different scene categories, one image from each of the 15 categories from [33]. The categories are bedroom, coast, forest, highway, industrial, inside city, kitchen, living room, mountain, office, open country, store, street, suburb, and tall building, respectively (from left to right, then from top to bottom).

office, store, coast, etc. Please refer to Figure 11 for example images and category names.

The sports event dataset [34] contains images of eight sports: badminton, bocce, croquet, polo, rock climbing, rowing, sailing, and snowboarding (see Figure 12 for example images from each category). In [34], Li and Fei-Fei used this dataset in their attempt to classify these events by integrating scene and object categorizations (i.e. deduce *what* from *where* and *who*). We use this dataset for scene classification purpose only. That is, we classify events by classifying the scenes, and do not attempt to recognize objects or persons.

In each dataset, the available data are randomly split into a training set and a testing set. The random splitting is repeated 5 times, and the average accuracy is reported. In each train/test splitting, a visual codebook is generated using the training images, and both training and testing images are transformed into histograms



Figure 12: Images from 8 different sports event categories.

of code words using the codebook. During the testing time, accuracy is computed as the mean accuracy of all categories (i.e. average of the diagonal entries in the confusion matrix).

The proposed algorithms can efficiently process huge numbers of histogram features, e.g. approximately 200k to 320k histograms are clustered across the training sets in the two datasets.

We use a Bag of Visual Words model in which features are densely sampled. We use 16x16 image patches and densely sample features over a grid with a spacing of 4 or 8 pixels. We use the 256 dimensional Census Transform histogram. All feature vectors are scaled and rounded such that a histogram only contains non-negative integers that sum to 128 (thus $h_{\max} = 128$ is always valid.)

The first step is to use feature descriptors from the training images to form a visual codebook, in which we use $m = 200$ to generate 200 visual code words. Next every feature is mapped to an integer (code word index) between 1 and m . Thus an image or image sub-window is represented by a histogram of code words in the specified image region. In order to incorporate spatial information, we use the spatial hierarchy in

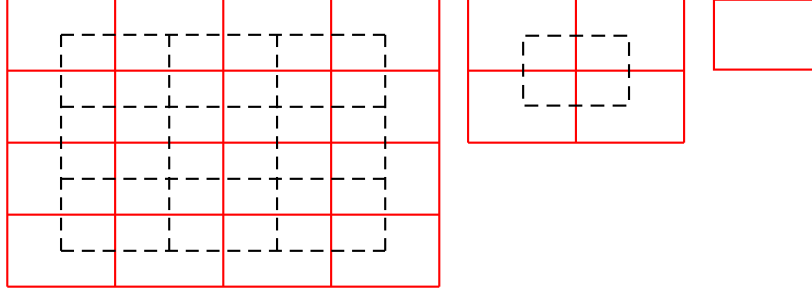


Figure 13: Illustration of the level 2, 1, and 0 split of an image.

our previous work [76], as illustrated in Figure 13. An image is represented by the concatenation of histograms from all the 31 sub-windows in Figure 13, which is a 6200 dimensional histogram. To capture the edge information, we sometimes use Sobel gradients of an input image as an addition input (c.f. Figure 1b in page 23), and concatenate histograms from the original input and Sobel gradient images (which is 12400 dimensional). Following [5], we also sample features at 5 scales.

SVM is used for classification. LIBSVM [9] is used for the scene and sports dataset. It uses the 1-vs-1 strategy, which will produce too many classifiers for the Caltech 101 dataset (more than 5000). Instead we use the Crammer & Singer formulation in BSVM [24]. Since we are classifying histograms, we modified both LIBSVM and BSVM so that they are able to utilize the histogram intersection kernel.

We conducted several sets of experiments to validate the proposed algorithms. These experimental results are organized into sub-sections.

4.6.1 HIK Visual Codebook (Algorithm 1) greatly improves classification accuracy

We compare the classification accuracies of systems that use Algorithm 1 (*i.e.* using κ_{HI}) and the usual k-means algorithm (*i.e.* using κ_{LIN}) and k-median. From the experimental results in Table 1 (page 53), it is obvious that in all three datasets, the classification accuracy with a κ_{HI} -based codebook is consistently higher than that with a k-means or k-median codebook. Using a paired *t*-test with significance level

0.05, the differences are statistically significant in 16 out the 18 cases in Table 1, when comparing κ_{HI} and κ_{LIN} based codebooks.

4.6.2 The HIK codebook evaluates quickly (Algorithm 2)

We have theoretically showed that Algorithm 2 evaluates in $O(d)$ steps, in the same order as k-means. Empirically, the κ_{HI} -based method spent less than 2 times CPU cycles than that of k-means, in both the codebook generation process and the image to histogram translation task.

4.6.3 K-median is a compromise between k-means and HIK codebooks.

HIK codebooks consistently outperformed k-median codebooks. However, k-median generally outperformed the popular k-means codebooks. Furthermore, k-median requires less memory than the proposed method.

4.6.4 One-class SVM improves histogram intersection kernel code words (Algorithm 3)

Algorithm 3 improved classification accuracy of the κ_{HI} -based method in 8 out of 9 cases in Table 1. Five out of the 9 differences are statistically significant according to the paired t -test. However, the t -test is not powerful enough here because we have only 5 paired samples in each test and they are not necessarily normally distributed. The Wilcoxon signed-rank test is more appropriate [13], which showed that using Algorithm 3 is significantly different from not using it (at significance level 0.02).

In short, using histogram intersection kernel visual codebook and one-class SVM code words altogether generated the best results in almost all cases of Table 1 (best results are shown in boldface within each column).

4.6.5 One-Class SVM degrades usual k-means code words

It is interesting to observe a completely reversed trend when κ_{LIN} is used with one-class SVM. Applying Algorithm 3 in a usual k-means method reduced accuracy in all

Table 1: Results of HIK, k-median and k-means codebooks and one-class SVM code words. (a), (b), and(c) are results for the Caltech 101, 15 class scene, and 8 class sports datasets, respectively. κ_{HI} and κ_{LIN} means that a histogram intersection or linear kernel is used, respectively. oc_{svm} and $\neg oc_{\text{svm}}$ indicate whether one-class SVM is used in generating code words. B and $\neg B$ indicate whether Sobel images are concatenated or not. And $s = 4$ or $s = 8$ is the grid step size when densely sampling features. The number of training/testing images in each category are indicated in the sub-table captions. The best result in each *column* is shown in **boldface**.

	$B, s = 4$	$B, s = 8$	$\neg B, s = 8$
$\kappa_{\text{HI}}, oc_{\text{svm}}$	67.44±0.95%	65.20±0.91%	61.00±0.90%
$\kappa_{\text{HI}}, \neg oc_{\text{svm}}$	66.54±0.58%	64.11±0.84%	60.33±0.95%
k-median	66.38±0.79%	63.65±0.94%	59.64±1.03%
$\kappa_{\text{LIN}}, oc_{\text{svm}}$	62.69±0.80%	60.09±0.92%	56.31±1.13%
$\kappa_{\text{LIN}}, \neg oc_{\text{svm}}$	64.39±0.92%	61.20±0.95%	57.74±0.70%

(a) Caltech 101, 15 train, 20 test

	$B, s = 4$	$B, s = 8$	$\neg B, s = 8$
$\kappa_{\text{HI}}, oc_{\text{svm}}$	84.12±0.52%	84.00±0.46%	82.02±0.54%
$\kappa_{\text{HI}}, \neg oc_{\text{svm}}$	83.59±0.45%	83.74±0.42%	81.77±0.49%
k-median	83.04±0.61%	82.70±0.42%	80.98±0.50%
$\kappa_{\text{LIN}}, oc_{\text{svm}}$	79.84±0.78%	79.88±0.41%	77.00±0.80%
$\kappa_{\text{LIN}}, \neg oc_{\text{svm}}$	82.41±0.59%	82.31±0.60%	80.02±0.58%

(b) 15 class scene, 100 train, rest test

	$B, s = 4$	$B, s = 8$	$\neg B, s = 8$
$\kappa_{\text{HI}}, oc_{\text{svm}}$	84.21±0.99%	83.54±1.13%	81.33±1.56%
$\kappa_{\text{HI}}, \neg oc_{\text{svm}}$	83.17±1.01%	83.13±0.85%	81.87±1.14%
k-median	82.13±1.30%	81.71±1.30%	80.25±1.12%
$\kappa_{\text{LIN}}, oc_{\text{svm}}$	80.42±1.44%	79.42±1.51%	77.46±0.83%
$\kappa_{\text{LIN}}, \neg oc_{\text{svm}}$	82.54±0.86%	82.29±1.38%	81.42±0.76%

(c) 8 class sports, 70 train, 60 test

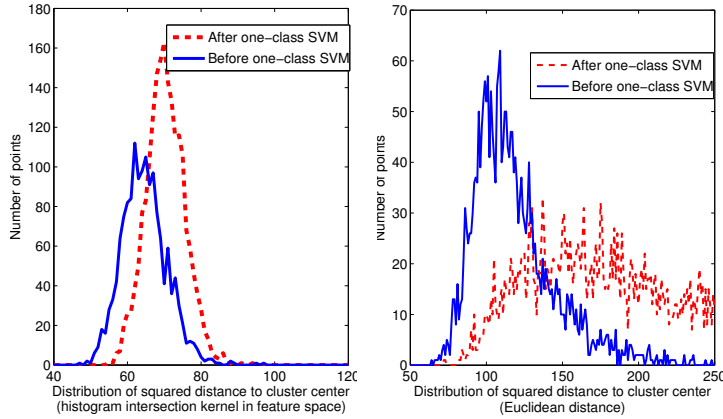


Figure 14: Effects of one-class SVM.

cases. Since a vector in R^d is not an appropriate understanding of a d -dimensional histogram, we conjecture that Algorithm 3 (equipped with κ_{LIN}) produced a better division of the space R^d , but probably a worse one in the space of histograms which is a subset of R^d .

Figure 14 shows the effect of applying Algorithm 3 to example code words. The distribution of squared distance to cluster center becomes more compact in case of κ_{HI} with a minor increase in the average error. However, in the k-means case, the distances spread to larger values.

4.6.6 Use the right feature for different tasks

The SIFT descriptor is widely used in object recognition for its performance. We have shown that it has higher discriminative power than CT histogram in object recognition (refer to Section 3.5). We also showed that CT histogram is more suitable for categorizing scene images, which is supported further by empirical results in Table 2. As shown in Table 2, if we use SIFT for scene recognition, the recognition accuracies are reduced.

Table 2: Results when SIFT descriptors are used in stead of CT histograms. We use $\neg B$, $s = 8$, κ_{HI} , and oc_{SVM} . Note that the second row contains the top right corner numbers in Table 1.

	15 scene	8 sports
SIFT	$78.54 \pm 0.22\%$	$81.17 \pm 0.65\%$
CT Histogram	$82.02 \pm 0.54\%$	$81.33 \pm 1.56\%$

4.6.7 Comparison with previously published results

In this section we will compare our methods with previously published results. The setting we choose to compare is κ_{HI} , oc_{SVM} , and B .

In the scene recognition tasks, the proposed method achieved the highest accuracy to the best of our knowledge. In the 15 class scene recognition task, the proposed method has an accuracy of $84.12 \pm 0.52\%$. The Spatial Pyramid Matching method achieved $81.4 \pm 0.5\%$. SP-pLSA [8], a method combining spatial pyramids and pLSA (probabilistic Latent Semantic Analysis), had 83.7% correct recognitions.

The sports dataset was first published in [34], which achieved a 73.4% accuracy. The method in [34] used manual segmentation and object labels as additional inputs for their training method. Spatial PACT [76], using a spatial hierarchy of PACT (Principal Component analysis of Census Transform histograms), achieved 78.50% correct category predictions, which is still inferior to the proposed methods ($84.21 \pm 0.99\%$) by a large margin.

Results for the Caltech 101 dataset are usually divided into two types: methods that use a single type of features and methods that integrate multiple cues (*e.g.* color, texture, shape, *etc.*). Several methods that use multiple cues outperformed our method, for example [6, 5, 70]. The proposed method (which uses only a single type of feature), however, has much higher accuracy than published single cue methods. With $m = 1000$ and $s = 2$ and 15 training examples per category, its accuracy is $70.74 \pm 0.69\%$. This accuracy is higher than methods such as NBNN (Naive-Bayes Nearest-Neighbor) [5] ($65.0 \pm 1.14\%$). For more single cue results, please refer to [5].

It is expected that when the κ_{HI} -based codebook and one-class SVM code words are used, the proposed method will be integrated into and further improve the multiple cue methods.

4.7 Summary of the chapter

In this chapter we proposed a visual codebook generation method that utilizes the Histogram Intersection Kernel (HIK), which is a better similarity measure for comparing histograms. Since HIK has exhibited superior performances for histogram features in classification tasks, we would expect applying HIK in unsupervised learning will generate better visual codebooks, especially for the proposed CT histogram descriptor. We extended a fast computing algorithm in [40] and the proposed algorithm has the same complexity as the k-means algorithm. We also proposed to use one-class SVM to represent a visual code word.

The proposed algorithms were tested on two scene recognition datasets and an object recognition dataset, and have shown the benefits of HIK based visual codebooks. It's application to the visual place categorization problem is described in the next chapter.

CHAPTER V

THE VPC DATASET AND SYSTEM

To our knowledge there is no publicly available dataset that satisfy our problem definition: visual information collected by autonomous robot platform for place categorization. We collected a Visual Place Categorization Dataset and make it available at

<http://categorizingplaces.com/dataset.html>.

In this chapter we will describe the philosophy and procedures we followed in collecting the VPC dataset. A first VPC system is also built using the proposed CT histogram feature descriptor, the HIK based codebooks, and a standard Bayesian filtering method. The VPC system is evaluated using our VPC dataset.

5.1 The Visual Place Categorization Dataset

We choose to collect data from home environments. Homes provide many place categories that are naturally defined by their function. It is also a trend to deploy intelligent software and hardware systems (including robots) in homes, e.g. to help take care of elderly people. We anticipate numerous applications of VPC in home environments.

Ideally, high resolution images with well-calibrated focus, appropriate viewing angles, even illumination, and white-balanced colors are desired. However it is difficult to achieve such desired settings simultaneously. We balanced these requirements by choosing a high definition camcorder (JVC GR-HD1) which captures 1280x720 images. We used the automatic settings of the camcorder to let it adjust the camera



Figure 15: VPC data collection hardware.

parameters during recording. The camcorder is able to automatically adapt to illumination changes in different rooms and adjust its focus and white balance. The camcorder is mounted on a rolling tripod to mimic a mobile robot platform. Although it is desirable to use a real robot, mobility and speed issues make this an impractical choice for capturing a large dataset in a wide variety of environments. (We experimented with a PeopleBot platform with an attached Prosilica camera in our initial capture sessions. But we found that the tripod+camera solution made it much easier to quickly capture high-quality images and navigate in small spaces.) The data collection hardware are shown in Figure 15.

To date, we have collected data from 6 homes and manually labeled 11 semantic categories. We asked the volunteers who allowed us to collect data in their homes to keep their homes as natural as possible. We made only two modifications to the home environments that we captured: First, we removed objects that could reveal the identity or the address of the occupants (e.g. family pictures or letters). Second, we closed the blinds in each room and relied upon artificial light. This helped to normalize the illumination environment across homes and times of day.

Table 3: The 11 semantic categories in the VPC dataset, plus a special category named *transition*.

bedroom	bathroom	kitchen	storage closet
living room	dining room	family room	workspace
exercise room	media room	corridor	transition

Within each home, we captured two datasets. The first was a continuous run through the entire home, one floor at a time. During the continuous run, the operator mimicked the behavior of a robot following a predefined path through the home environment. He pushed the tripod with the camera facing forward, so that it traveled through all traversable areas in each room. The operator did not look at the captured video during recording, and simply ensured that the tripod followed a smooth path without colliding with any objects in the room. Following this continuous capture, we went room-by-room and captured cylindrical panoramic video at two elevation angles. We mainly use the first series of data for the experiments in this dissertation, but the second series of videos are available for use by interested researchers.

Our protocol for data capture had two consequences for the images that we acquired: First, because the camera viewpoint simply followed the path of the tripod, uninformative views (such as a close-up of a section of wall) are a major portion of the captured video. Second, because the tripod often passed close by major furniture items such as beds and sofas, these objects are typically only partially visible in any specific frame. We believe these are realistic attributes for conventional video data collected by an autonomous platform in a home environment.

The VPC dataset was generated by extracting every third frame from the videos as JPEG (95% quality) images to keep the dataset to a manageable size. Each image is 1280x720 in resolution. Depending upon the size of the home, each home produced images totaling 1 to 2 gigabytes (corresponding to about 6000 to 10000 frames per home).

We provided manual annotations for this dataset. There are 11 categories (see

Table 3 for category names). We used a special category name *transition* to annotate video segments that are either difficult to categorize or those that contain two or more categories. One category label is attached to a segment of the video (i.e. continuous image frames) instead of a single frame. Because of the autonomous image collection process, frames within a short contiguous time span have a high likelihood to share the same category label. This choice reduces the required manual labeling labor, but still retains enough information for learning place categories.

The homes in the VPC dataset span a wide range of styles and sizes, from modern suburban homes to Craftsman-style urban bungalows. The home owners span a variety of age groups, and include families with and without children. The homes vary in size and age and are designed and decorated in a variety of styles. Both single story and two-story homes are included, and some homes had a finished basement.

Note that not all room categories are present in all homes. However, there are five categories that exist in all homes: bedroom, bathroom, kitchen, living room, and dining room. We will use these five categories to evaluate our VPC system.

Along with the dataset, we also provided a baseline evaluation package, which uses leave one out cross validation and is based on per-frame accuracy. More evaluation details are described in Section 5.2.

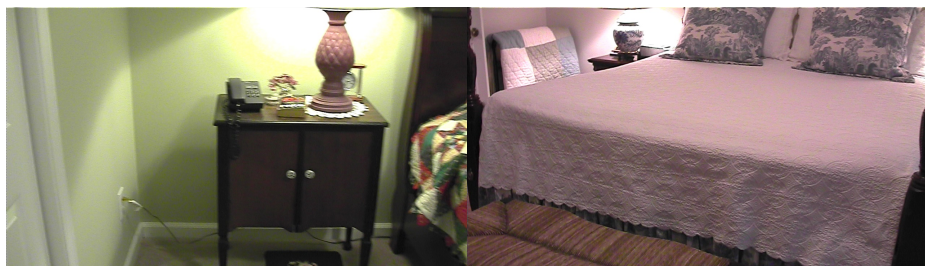
5.1.1 Example frames from the VPC dataset

Figure 16 shows six example frames from the bedroom category, where each subfigure is a randomly chosen frame from the bedroom category of one home. As shown in Figure 16, appearance of these bedrooms are very different. They have varying illumination conditions and the bedrooms are decorated in different styles. The camera is positioned at different viewing angles and all possible areas of the bedrooms are observed. Bed, which is a key object in a bedroom, are also exhibiting large variations. The bed object is not visible in some frames. In the remaining frames, the beds



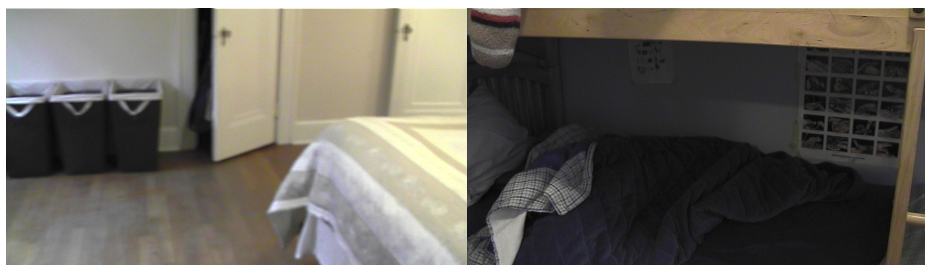
(a)

(b)



(c)

(d)



(e)

(f)

Figure 16: Example frames from the bedroom category.



Figure 17: Example frames from the living room category.

have different structures and sheets. These variations show that VPC is a difficult categorization task.

Similarly, Figure 17 shows randomly chosen example frames from the living room category. These example frames show another property of VPC. Besides huge variations in these images, key objects such as the stove or couches are usually missing, occluded, or only partially shown.

5.2 *The Visual Place Categorization system*

The block diagram for our VPC system is shown in Figure 18. In the following, we will discuss each component in detail. A key aspect of our approach is an image

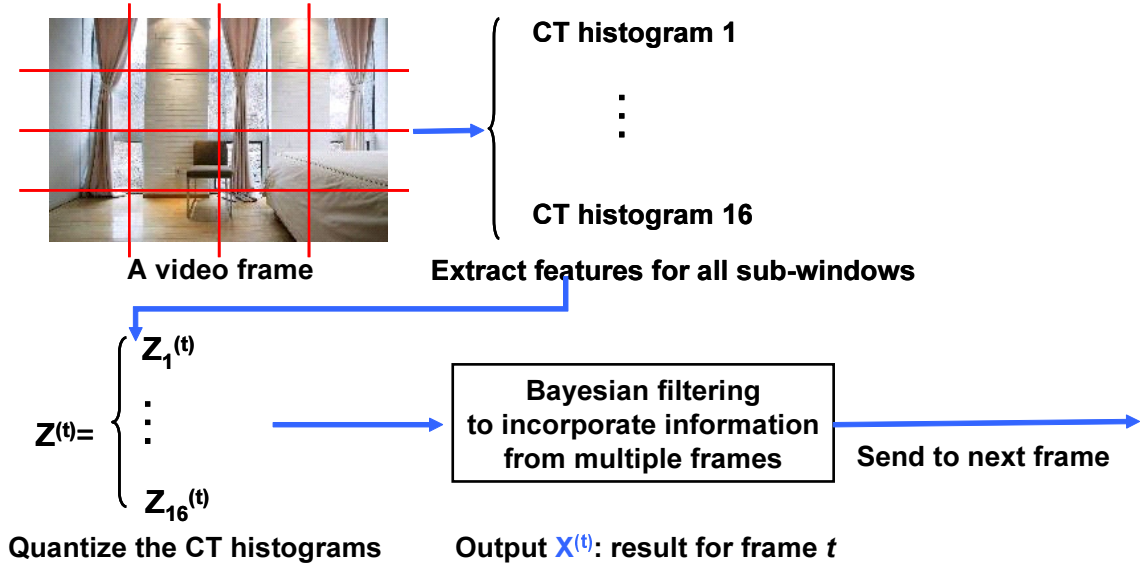


Figure 18: Diagram of the VPC system.

representation based on vector quantized spatial Census Transform histograms and a Bayesian filtering approach [25, 65].

5.2.1 Image Representation

In the VPC system we use CT histogram as our visual descriptor. A 256-bin histogram of CT values can be computed for any rectangular sub-window. As shown in previous scene recognition research, incorporating spatial information greatly improves recognition accuracy. Thus we evenly divide an image into $4 \times 4 = 16$ sub-windows, and extract a CT histogram from each sub-window. An image is represented by the concatenation of the 16 histograms (spatial CT histograms). We also tried to divide an image hierarchically as shown in Figure 13. However, although the division in Figure 13 almost doubles the feature vector length, we did not observe an increase in categorization accuracy. Thus we choose to use the simple 4×4 division method to incorporate spatial information.

We index the sub-windows in an image from 1 to 16 by their position. For each $i, 1 \leq i \leq 16$, we collect together all sub-window i CT histograms across the entire

training set. We then use the HIK-based method to generate a visual codebook with K centers (c.f. Chapter 4). Any CT histogram from the i -th sub-window position will then be mapped to an integer between 1 and K . Thus an image is represented by a 16 dimensional vector

$$Z = (z_1, z_2, \dots, z_{16}), \quad (15)$$

where z_i is the vector quantized index of the CT histogram extracted from sub-window position i .

Let X be the category index of a video frame. Then we use a Naive Bayes approach to estimate $P(X|Z)$

$$P(X|Z) \propto P(Z|X)P(X) \quad (16)$$

$$= \prod_{i=1}^{16} P(z_i|X)P(X), \quad (17)$$

in which $P(z_i|X)$ is easily estimated from the training data (i.e. set to the empirical distribution of the training set).

5.2.2 Bayesian filtering

Given that we are using normal cameras and we are not specifically identifying representative frames, the probability that a robot will both capture a representative frame and recognize the place category from such a frame is small. Thus it is vital to integrate information from many frames. We maintain a belief (probability of the current frame belonging to a certain category) and use a Bayesian filtering approach for updating category beliefs. Specifically, let Z_t be the image observed at time t and $Z_{1:t}$ represent the image history till time t , i.e. the set of images observed from time 1 to t . Correspondingly, let X_t and $X_{1:t}$ be the category label at time t and the history of category labels till time t , respectively. Our purpose is to estimate the distribution $P(X_t|Z_{1:t})$.

The Bayesian filtering process exploits the entire image history to efficiently integrate information from several images. We assume a Markovian property between the category labels X , i.e. $P(X_t|X_{1:t-1}) = P(X_t|X_{t-1})$. Furthermore, we assume that the distribution of the observed image frame Z_t at time t is determined if we know the category label X_t at time t . Thus, the Bayesian filtering process is governed by three distributions [25, 65]:

1. The prior category distribution $P(X_0)$;
2. The category transition distribution $P(X_t|X_{t-1})$; and
3. The observation distribution $P(Z|X)$.

Using the three distributions and our independence assumptions, $P(X_{1:t}, Z_{1:t})$ can be factorized as:

$$P(X_{1:t}, Z_{1:t}) = P(X_0) \prod_{i=1}^t P(X_i|X_{i-1}) \prod_{i=1}^t P(Z_i|X_i). \quad (18)$$

In the VPC system, these distributions are specified as follows.

1. The prior distribution $P(X_0)$ is a discrete uniform distribution since we assume the robot knows nothing about the environment at the beginning;
2. The category transition distribution is specified as $P(X_t|X_{t-1}) = p_e$ if X_t equals X_{t-1} . We set p_e to a large number (e.g. we set $p_e = 0.99$ in our experiments) to reflect the fact that image frames within a consecutive time span have a high likelihood to share the same category label. The rest of the probability mass is shared uniformly among all the other values of X_t that is different from X_{t-1} .
3. The last component, the observation model, is specified by Eq. 17.

After the three distributions are available, the desired quantity can be efficiently

updated at each image frame, as shown in [25]:

$$P(X_t|Z_{1:t}) \propto P(Z_t|X_t)P(X_t|Z_{1:t-1}) \quad (19)$$

$$P(X_t|Z_{1:t-1}) = \sum_i P(X_t|X_{t-1} = i)P(X_{t-1} = i|Z_{1:t-1}). \quad (20)$$

A frame t is then classified as

$$x_t = \arg \max P(X_t|Z_{1:t}). \quad (21)$$

5.3 *Experimental setup and evaluation methodology*

We used $K = 50$ in our experiments, i.e. for each sub-window location, 50 visual codewords are generated. We are interested in the spatial structure property of an image rather than detailed textural information. Thus, instead of extracting CT histograms from input video frames, we first compute the Sobel gradients of the input image, and the CT histograms are extracted from the Sobel gradient images.

Although there are 11 categories (plus a special *transition* category), only 5 categories are present in all homes. Thus we tested the proposed VPC system on these 5 categories: bedroom, bathroom, kitchen, living room, and dining room. Categorization results on frames whose groundtruth label is not within this set are simply ignored. In each home, the accuracy for a category is computed as the number of correct categorizations in this category divided by the total number of video frames in this category. The accuracy of a home is computed as the average accuracy of the five categories inside this home.

We used a leave one out cross validation strategy to evaluate the VPC system. The proposed method was applied 6 times. In each run, one home was reserved for testing and all other 5 homes were combined to form a training set. The overall accuracy of our VPC system is the average of the 6 individual homes.

Our VPC system runs at approximately 20 frames per second.

5.4 Results and discussions

Figure 19a to 19c provides example visualizations for the VPC system results. The gray bar indicates the groundtruth and the red bar is the categorization result. All categories that are not used and the special category *transition* are attributed to the *other* category in Figure 19. The two bars progressed with time and the end of both bars indicated results for the current frame (e.g. around middle in Figure 19a and at the end in Figure 19b and 19c). As shown in Figure 19, the bathroom and bedroom category are predicated well, with minor fragments in results and small periods of errors. However, the living room in Figure 19c are poorly recognized. The sub-figures 19a and 19b show result for the second floor of home 5. Note that this floor only contains a bathroom and a bedroom, which are the best learned categories. Results for other homes and floors are generally inferior to the one shown in these sub-figures, e.g. the living room in Figure 19c.

In the following sections we will provide detailed categorization accuracies when different components (e.g. feature descriptor, visual codebook generation method, etc.) are used in the VPC system.

5.4.1 Baseline system

Our baseline system used the SIFT descriptor (i.e. replacing the CT histogram descriptors in Figure 18 with SIFT descriptors) and a visual codebook that is generated by the *k-means++* clustering algorithm [2]. Bayesian filtering was not used in the baseline system.

Detailed categorization accuracy rates of all homes and categories are presented in Table 4. The baseline system has an overall accuracy of 35.01%. The different performance among the 5 categories is noteworthy. Only bathroom has an accuracy that is higher than 50%, while the dining area is categorized correctly in only 12.83% cases, which is approximately half of the random guess probability (20%).

Table 4: Baseline categorization accuracy (in percentages) of all homes and categories when SIFT and k-means are used. The Bayesian filtering is not used.

	bed	bath	kitchen	living	dining	average
home 1	30.59	67.58	19.09	59.76	17.70	38.94
home 2	30.13	39.16	42.31	15.87	22.61	30.02
home 3	43.26	55.35	32.46	29.94	7.89	33.78
home 4	27.42	58.61	83.41	36.92	9.61	43.19
home 5	30.82	62.52	31.82	15.54	2.70	28.68
home 6	38.62	48.27	37.23	36.78	16.44	35.47
average	33.47	55.25	41.05	32.47	12.83	35.01

Table 5: Linear k-means codebook categorization accuracy (in percentages) of all homes and categories using CT histogram as the feature descriptor. The Bayesian filtering is not used.

	bed	bath	kitchen	living	dining	average
home 1	55.02	70.32	17.63	62.20	18.69	44.77
home 2	49.05	32.30	53.64	12.24	19.43	33.33
home 3	65.98	88.39	39.12	7.77	2.12	40.68
home 4	36.76	53.07	70.85	28.57	27.17	43.28
home 5	53.77	73.39	41.95	33.08	3.29	41.10
home 6	28.19	76.79	56.17	31.19	48.00	48.07
average	48.13	65.71	46.56	29.18	19.78	41.87

5.4.2 CT histogram is suitable for visual place categorization

As illustrated in Figure 18, our VPC system uses CT histogram as a descriptor for each of the 16 sub-windows in a video frame. When CT histogram is used, the categorization accuracy is presented in Table 5. Note that the only difference between Table 5 and Table 4 is whether CT histogram or SIFT is used.

CT histogram has an overall recognition accuracy of 41.87%, which is much higher than the SIFT rate 35.01%. Looking into the tables more closely, the CT histogram based system has higher average accuracies than the baseline system in all homes and almost all categories (except the living room category). Together with the scene recognition results in Chapter 4, we believe that these experiments clearly demonstrate that CT histogram is a more suitable descriptor for categorizing and recognizing semantic place and scene categories.

Table 6: Linear k-means codebook categorization accuracy (in percentages) of all homes and categories using CT histogram as the feature descriptor. The Bayesian filtering is used.

	bed	bath	kitchen	living	dining	average
home 1	75.76	80.04	12.03	43.90	11.15	44.58
home 2	67.10	32.14	64.37	2.04	13.78	35.89
home 3	80.07	95.32	26.14	3.26	0.00	40.96
home 4	49.77	63.92	69.06	30.50	36.41	49.93
home 5	81.47	86.41	45.05	21.30	0.30	46.91
home 6	35.17	90.81	72.77	22.54	56.00	55.46
average	64.89	74.77	48.24	20.59	19.61	45.62

Another important difference between Table 5 and Table 4 is that in our CT histogram based system, accuracies are almost higher than random guess in all categories (the dining area has a 19.78% accuracy, which is very close to 20%). The three categories with higher accuracies (bedroom, bathroom, and kitchen) have recognition rates higher than or close to 50%.

5.4.3 Bayesian filtering improves system accuracy

In this section we examine the effect of applying the Bayesian filtering. Table 6 shows the results when both CT histogram and Bayesian filtering are used. Note that the only difference between Table 6 and Table 5 is whether Bayesian filtering is used.

The VPC system achieves a 45.62% overall accuracy when Bayesian filtering is used with CT histogram. Three categories (bedroom, bathroom, and kitchen) have relatively high accuracy (higher or close to 50%). The bathroom and bedroom categories have the highest accuracies, and are close to being useful in practice. However, the living room and dining room categories exhibit poor performance, close to that of random guessing (which is 20%). In general, the Bayesian filtering technique improves system performance when CT histogram is used as our feature descriptor. However, it is making the rich get richer, and the poor get poorer.

However, the real utility of Bayesian filtering is to reduce fragmentation of the

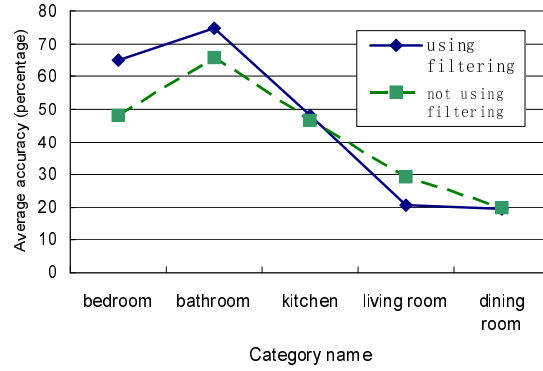


Figure 21: Effect of using the Bayesian filtering.

Table 7: Baseline categorization accuracy (in percentages) of all homes and categories when SIFT and k-means are used. The Bayesian filtering is used.

	bed	bath	kitchen	living	dining	average
home 1	54.36	86.50	14.94	52.44	2.95	42.24
home 2	51.07	37.09	50.81	7.48	20.85	33.46
home 3	64.16	84.62	59.12	11.66	1.52	44.21
home 4	47.66	64.50	86.55	33.71	3.14	47.11
home 5	75.65	68.24	36.95	0.25	0.00	36.15
home 6	57.93	60.44	44.47	16.10	0.00	35.79
average	58.47	66.90	48.75	20.27	4.74	39.83

Table 8: HIK codebook categorization accuracy (in percentages) of all homes and categories when CT histogram and Bayesian filtering are used.

	bed	bath	kitchen	living	dining	average
home 1	75.95	80.40	12.24	69.51	11.48	49.92
home 2	56.30	35.19	45.95	22.22	14.49	34.83
home 3	82.02	96.08	27.89	1.37	0.00	41.47
home 4	54.01	66.04	62.78	15.41	15.71	42.79
home 5	88.91	75.68	43.27	32.08	0.00	47.99
home 6	41.48	92.67	74.26	26.78	83.11	63.66
average	66.44	74.34	44.40	27.89	20.80	46.78

lot in different homes (especially the living room and dining room categories), the average accuracy of homes remain relatively stable.

5.4.4 HIK codebooks further improves VPC

In this section we examine the effect of applying HIK based visual codebooks, compared against codebooks generated by the k-means cluster algorithm (Table 6). Table 8 shows results when HIK codebooks, CT histogram, and Bayesian filtering are used. Note that the only difference between Table 8 and Table 6 is whether a HIK or k-means codebook is used.

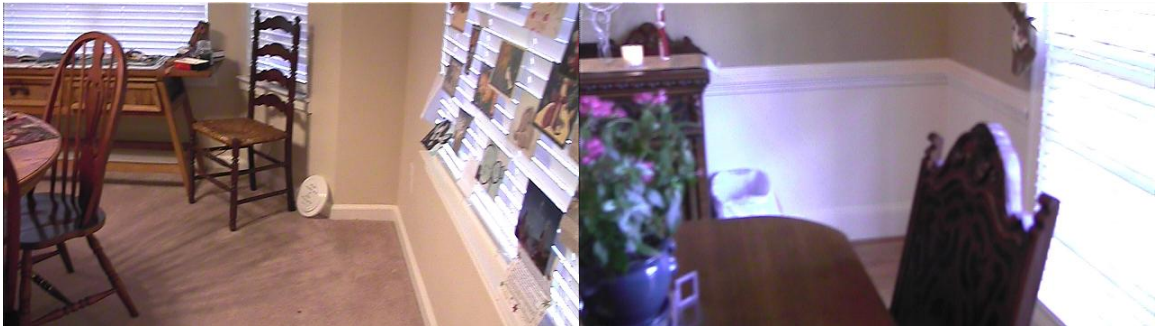
The Histogram Intersection Kernel based visual codebook further improves overall system accuracy. Besides, a new trend is observed. Now the average accuracy of the five categories are all above the random guess probability of 20%, although both living room and dining area still have lower accuracies.

Our conjecture about the low accuracy in these categories are as follows. The key objects in these categories are usually very large and our camera can not capture the entire instance of such objects into a single frame. For example, we can only capture half (or even less) of the dining table in one frame in many frames due to the limited field of view of our camera. Similarly, our camera encounters the same problem with large living room furniture such as sofas. This limitation is clearly illustrated by the example frames in Figure 22. These frames are taken from 4 different homes and in



(a) Living room example frame 1

(b) Living room example frame 2



(c) Dining room example frame 1

(d) Dining room example frame 2

Figure 22: Example of frames in the living room and dining room category.

general most of the frames in these two categories suffer from the same limitation.

We used a global image representation and did not specifically detect any characteristic objects. However, we observed that the VPC system usually recovers from error when such objects (e.g. sink in a kitchen and fireplace in a living room) came into the robot's sight. This observation make us believe that we could use object recognition to help visual place categorization, and vice versa.

HIK codebooks also improves system performance when SIFT is used. We will omit the detailed recognition rates. The overall accuracy is improved from 38.68% to 39.83% when applying Bayesian filtering, and from 35.01% to 36.45% if not applying Bayesian filtering, using SIFT as the feature descriptor.

Table 9: HIK codebook and one-class SVM code words categorization accuracy (in percentages) of all homes and categories when the Bayesian filtering is used.

	bed	bath	kitchen	living	dining	average
home 1	76.04	81.85	13.49	70.73	5.24	49.47
home 2	61.23	36.17	51.82	5.90	0.00	31.02
home 3	85.54	98.94	27.54	1.16	0.00	42.64
home 4	56.72	68.28	60.99	7.06	19.22	42.45
home 5	79.53	84.55	46.60	33.58	0.00	48.85
home 6	38.58	95.00	72.98	31.36	69.78	61.54
average	66.27	77.47	45.57	24.97	15.71	46.00

5.4.5 Effect of One-class SVM code words

Although one-class SVM was shown to generate better visual code words for scene recognition tasks (c.f. Chapter 4), unfortunately this is not the case in VPC. The results using HIK codebooks, one-class SVM code words and Bayesian filtering is shown in Table 9. Note that the only difference between Table 9 and Table 8 is whether one-class SVM code words are used or not.

The three categories with higher accuracies remained at about the same performance level. However, the living room and dining area category accuracies drop by a large percentage.

5.5 Discussions

Besides performance numbers of our VPC system reported in the previous section, there are a few other observations that are worth discussing.

First, we observe that scene recognition systems generally have higher accuracies than that of visual place categorization. There are three categories that exist in both our VPC dataset and the 15 class scene recognition dataset: bedroom, living room, and kitchen. Note that the scene dataset may contain images from hundreds of distinct homes, which exhibits larger variations than our VPC dataset. Furthermore, there are 15 categories in scene recognition, compared to 5 in VPC. In scene recognition these three categories have recognition accuracy 69%, 74%, and 74% respectively.

In contrast, their accuracies in VPC are 66.44%, 44.40%, and 27.89% (from Table 8). We conjecture that this performance gap is caused by the manner in which images are collected. Images taken at canonical views in scene recognition greatly improves recognition accuracy.

A second observation is that room categories have different difficulty levels. Both living room and dining room have much lower accuracies than the other three categories in VPC. One possible explanation is the imbalance property in machine learning. Frames from living room only constitute about 10% of all the frames. Dining room have even less frames than living room. It is known in the machine learning literature that such *minor classes* usually have lower accuracies [73]. However, we also believe that the inferior performance is caused at least in part because these two categories are inherently difficult. In a new experiment when we only recognize these two categories in our VPC dataset (all other categories are ignored), the recognition accuracy is 64.61% for living room and 47.84% for dining room. Under this scenario, these two classes are balanced (i.e. having approximately the same number of frames). However, the accuracy is only around random guess probability (50%).

We also observed limitations of the current VPC dataset from the VPC system output. We have discussed about limitations such as limited viewing angles, which could be alleviated by building local spatial structures (maybe in 3D) from a few consecutive frames. This approach will incorporate information from larger horizontal viewing angles. However, it will not help enlarge the vertical viewing angle. Objects and structures that are too high or too low are missing from our videos. A possible solution is to use multiple cameras to simultaneously capture multiple views of the environment.

We conducted a brief experiment to study the effect of vertical viewing angles. The second series of videos in our VPC dataset (refer to Section 5.1) are taken by rotate the camera 360 degrees slowly in all rooms, at two different vertical angles.

The first angle is same as the angle used to shoot the first set of videos, while the second vertical angle is looking at higher parts of the rooms. Using videos from the second vertical angle, we distinguish between living room and dining room and get 61.64% accuracy for living room and 31.61% for dining room. The accuracies are lower than using the first vertical angle. It seems that objects and structures at different heights provide different information sources. However, none of them alone is enough for reliably distinguish these difficulty categories. We need to either utilize a camera with large vertical viewing angles, or use multiple cameras in the future.

One major limitation that has not yet been discussed is the lack of other sensory inputs, e.g. odometry and laser range sensor readings. If we have access to these sensors, we will be able to further reduce the fragmentation of system predictions. For example, the laser range sensor readings will be able detect that the robot has not recently passed a door so that it should have stayed in the same room. Similarly, the room category prediction must not change if odometry data only contain rotations. Furthermore, since we only need to provide a single category label for all the frames in the same room, we expect the categorization accuracy to improve by a large percentage if we are able to detect when the robot changes into a different room. We also expect the VPC system to work better if it can exchange information with other modules in a robot, e.g. topological mapping.

5.6 Summary of the chapter

In this chapter we introduced a dataset that is specifically designed and captured for visual place categorization. A first VPC system is also presented and evaluated using the new VPC dataset.

Our experiments show that for recognizing place categories, including both VPC and scene recognition, CT histogram is the suitable representation, which yields higher recognition accuracies than the SIFT descriptor. In addition, the Histogram

Intersection Kernel based visual codebooks consistently acquires higher system accuracies than the usual Euclidean distance based k-means codebooks. In VPC systems, the standard Bayesian filtering technique not only improves overall system accuracy, but also greatly reduces the fragmentation of predicted category labels.

CHAPTER VI

SUMMARY AND DISCUSSIONS

6.1 Summary of contributions

The thesis statement of this dissertation is:

Census transform histogram and histogram intersection kernel based visual codebook can provide a suitable representation for solving the visual place categorization problem.

The analysis and experiments presented in previous chapters have explained and supported this statement. We now restate the contributions as follows:

1. We introduced the visual place categorization problem that emphasizes autonomous data collection and collected a VPC dataset of home interiors which is now publicly available.
2. We proposed the CT histogram descriptor and a histogram intersection kernel based visual codebook generation method. Both techniques show superior performance in the visual place categorization problem.
3. We built a first VPC system using the proposed techniques plus a standard Bayesian filtering method, and achieved promising results on the VPC dataset.

6.2 Discussions

A number of improvements to the VPC system can be made to improve its accuracy. We will focus on the two issues that we deem most important. Possible directions to overcome these difficulties are also discussed.

6.2.1 Attentional mechanism

An attentional mechanism is completely missing in our current VPC system. It will not surprise us if an attentional mechanism will further improve the system accuracy. As we are employing a global configuration approach that does not detect any specific object, it is probably suitable to learn such a simulated attention mechanism from observed data. Certain objects may appear multiple times in a home and may be useful for other vision tasks. However, it may not be useful for categorizing places. A window is an example of such an object. Thus, it might be important to “learn to focus” in images using relationship among images plus their place category labels. This strategy might be also helpful to solve the imbalance problem. With an effective attention mechanism, difficult categories such as living room and dining area might have better performance.

6.2.2 Broader field of view

The camera we use have a limited field of view. The consequence is that we only see a small part of the room and large objects are only partially visible in many cases. The Bayesian filtering method alleviated this difficult to some extent. However, if an important structure is separated into two parts in two different frames, Bayesian filtering might not help in this case. A hardware solution is to utilize better cameras. A camera with broader field of view enlarges what we see in a single frame. At the same time it also increases the difficulty because we do not know which part of the larger image we should pay our attention to. We could also reveal the structure of part of a room using a few consecutive frames. A complete 3D reconstruction might be overkill. However, simpler methods such as stitching a few consecutive frames will provide important information that is missing in our Bayesian filtering framework.

REFERENCES

- [1] AGARWAL, A. and TRIGGS, B., “Multilevel image coding with hyperfeatures,” *International Journal of Computer Vision*, vol. 78, no. 1, pp. 15–27, 2008.
- [2] ARTHUR, D. and VASSILVITSKII, S., “k-means++: the advantage of careful seeding,” in *18th Symposium on Discrete Algorithms (SODA)*, pp. 1027–1035, 2007.
- [3] BAILEY, T. and DURRANT-WHYTE, H., “Simultaneous localization and mapping (SLAM): part II,” *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.
- [4] BHAT, D. and NAYAR, S., “Ordinal measures for image correspondence,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 4, pp. 415–423, 1998.
- [5] BOIMAN, O., SHECHTMAN, E., and IRANI, M., “In defense of nearest-neighbor based image classification,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.
- [6] BOSCH, A., MUÑOZ, X., and ZISSERMAN, A., “Image classification using random forests and ferns,” in *The IEEE Conf. on Computer Vision*, 2007.
- [7] BOSCH, A., ZISSERMAN, A., and MUÑOZ, X., “Scene classification via pLSA,” in *European Conf. Computer Vision*, vol. 4, pp. 517–530, 2006.
- [8] BOSCH, A., ZISSERMAN, A., and MUÑOZ, X., “Scene classification using a hybrid generative/discriminative approach,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 30, no. 4, pp. 712–727, 2008.
- [9] CHANG, C.-C. and LIN, C.-J., *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [10] CHOSET, H. and NAGATANI, K., “Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization,” *IEEE Trans. on Robotics and Automation*, vol. 17, no. 2, pp. 125–137, 2001.
- [11] DALAL, N. and TRIGGS, B., “Histograms of oriented gradients for human detection,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893, 2005.
- [12] DELLAERT, F., “Square root SAM,” in *Proceedings of Robotics: Science and Systems*, 2005.

- [13] DEMŠAR, J., “Statistical comparisons of classifiers over multiple data sets,” *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.
- [14] DISSANAYAKE, G., NEWMAN, P., DURRANT WHYTE, H., CLARK, S., and CSORBA, M., “A solution to the simultaneous location and map building (SLAM) problem,” *IEEE Trans. on Robotics and Automation*, vol. 17, no. 2, pp. 229–241, 2001.
- [15] DURRANT-WHYTE, H. and BAILEY, T., “Simultaneous localization and mapping: part I,” *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–108, 2006.
- [16] EVERINGHAM, M., ZISSERMAN, A., WILLIAMS, C., and GOOL, L. V., “The PASCAL visual object classes challenge 2006 (VOC 2006) results,” tech. rep., 2006.
- [17] FEI-FEI, L., FERGUS, R., and PERONA, P., “Learning generative visual models from few training example: an incremental bayesian approach tested on 101 object categories,” in *CVPR 2004, Workshop on Generative-Model Based Vision*, 2004.
- [18] FEI-FEI, L. and PERONA, P., “A bayesian hierarchical model for learning natural scene categories,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. II, pp. 524–531, 2005.
- [19] FELZENSZWALB, P. F. and SCHWARTZ, J. D., “Hierarchical matching of deformable shapes,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.
- [20] FERGUS, R., PERONA, P., and ZISSERMAN, A., “Object class recognition by unsupervised scale-invariant learning,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. II, pp. 264–271, 2003.
- [21] FOLKESSON, J. and CHRISTENSEN, H. I., “Graphical SLAM – a self-correcting map,” in *Proc. IEEE Intl. Conf. Robotics and Automation*, pp. 383–390, 2004.
- [22] GRAUMAN, K. and DARRELL, T., “The pyramid match kernel: Discriminative classification with sets of image features,” in *The IEEE Conf. on Computer Vision*, vol. II, pp. 1458–1465, 2005.
- [23] HAYS, J. and EFROS, A. A., “IM2GPS: estimating geographic information from a single image,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.
- [24] HSU, C.-W. and LIN, C.-J., *BSVM*, 2006. Software available at <http://www.csie.ntu.edu.tw/~cjlin/bsvm>.

- [25] ISARD, M. and BLAKE, A., “CONDENSATION – Conditional Density Propagation for Visual Tracking,” *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.
- [26] JORDAN, M. I. and JACOBS, R. A., “Hierarchical mixtures of experts and the EM algorithm,” *Neural Computations*, vol. 6, pp. 181–214, 1994.
- [27] JURIE, F. and TRIGGS, B., “Creating efficient codebooks for visual recognition,” in *The IEEE Conf. on Computer Vision*, vol. 1, pp. 604–610, 2005.
- [28] KIVINEN, J. J., SUDDERTH, E. B., and JORDAN, M. I., “Learning multiscale representaiton of natural scenes using dirichlet processes,” in *The IEEE Conf. on Computer Vision*, 2007.
- [29] KUIPERS, B., “The spatial semantic hierarchy,” *Artificial Intelligence*, vol. 119, no. 1-2, pp. 191–233, 2000.
- [30] KUIPERS, B., “An intellectual history of Spatial Semantic Hierarchy,” in *Robot and Cognitive Approaches to Spatial Mapping* (JEFFERIES, M. and YEAP, A. W.-K., eds.), vol. 38 of *Springer Tracts in Advanced Robotics*, pp. 243–264, 2008.
- [31] KUIPERS, B. and BEESON, P., “Bootstrap learning for place recognition,” in *AAAI Conference on Artificial Intelligence*, pp. 174–180, 2002.
- [32] LAZEBNIK, S. and RAGINSKY, M., “Supervised learning of quantizer codebooks by information loss minimization,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 7, pp. 1294–1309, 2009.
- [33] LAZEBNIK, S., SCHMID, C., and PONCE, J., “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. II, pp. 2169–2178, 2006.
- [34] LI, L.-J. and FEI-FEI, L., “What, where and who? Classifying events by scene and object recognition,” in *The IEEE Conf. on Computer Vision*, 2007.
- [35] LING, H. and JACOBS, D. W., “Shape classification using the inner-distance,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 286–299, 2007.
- [36] LIU, J. and SHAH, M., “Scene modeling using Co-Clustering,” in *The IEEE Conf. on Computer Vision*, 2007.
- [37] LIU, Y., EMERY, R., CHAKRABARTI, D., BURGARD, W., and THRUN, S., “Using EM to learn 3D models of indoor environments with mobile robots,” in *Int. Conf. on Machine Learning*, pp. 329–336, 2007.
- [38] LOWE, D., “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

- [39] LUO, J., PRONOBIS, A., CAPUTO, B., and JENSFELT, P., “The KTH-IDOL2 database,” Tech. Rep. CVAP304, Kungliga Tekniska Högskolan, CVAP/CAS, October 2006.
- [40] MAJI, S., BERG, A. C., and MALIK, J., “Classification using intersection kernel support vector machines is efficient,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.
- [41] MIKOLAJCZYK, K. and SCHMID, C., “A performance evaluation of local descriptors,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.
- [42] MONTEMERLO, M., THRUN, S., KOLLER, D., and WEGBREIT, B., “Fast-SLAM: A factored solution to the simultaneous localization and mapping problem,” in *AAAI Conference on Artificial Intelligence*, pp. 593–598, 2002.
- [43] MOOSMANN, F., NOWAK, E., and JURIE, F., “Randomized clustering forests for image classification,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 30, no. 9, pp. 1632–1646, 2008.
- [44] MOZOS, Ó. M., STACHNISS, C., and BURGARD, W., “Supervised learning of places from range data using AdaBoost,” in *Proc. IEEE Intl. Conf. Robotics and Automation*, pp. 1742–1747, 2005.
- [45] MOZOS, Ó. M., TRIEBEL, R., JENSFELT, P., ROTTMANN, A., and BURGARD, W., “Supervised semantic labeling of places using information extracted from sensor data,” *Robotics and Autonomous Systems*, vol. 55, no. 5, pp. 391–402, 2007.
- [46] NISTÉR, D. and STEWÉNIUS, H., “Scalable recognition with a vocabulary tree,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 2161–2168, 2006.
- [47] ODONE, F., BARLA, A., and VERRI, A., “Building kernels from binary strings for image matching,” *IEEE Trans. Image Processing*, vol. 14, no. 2, pp. 169–180, 2005.
- [48] OJALA, T., PIETIKÄINEN, M., and MÄENPÄÄ, T., “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.
- [49] OLIVA, A. and TORRALBA, A., “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.
- [50] PERRONNIN, F., “Universal and adapted vocabularies for generic visual categorization,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1243–1256, 2008.

- [51] PHILBIN, J., CHUM, O., ISARD, M., SIVIC, J., and ZISSERMAN, A., “Lost in quantization: Improving particular object retrieval in large scale image databases,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.
- [52] PRONOBIS, A. and CAPUTO, B., “The KTH-INDECS database,” Tech. Rep. CVAP297, Kungliga Tekniska Högskolan, CVAP, September 2005.
- [53] PRONOBIS, A., CAPUTO, B., JENSFELT, P., and CHRISTENSEN, H. I., “A discriminative approach to robust visual place recognition,” in *Proc. IEEE/RSJ Intl. Conf. Intelligent Robots and Systems*, 2006.
- [54] QUELHAS, P., MONAY, F., ODOBEZ, J.-M., GATICA-PEREZ, D., and TUYTELAARS, T., “A thousand words in a scene,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1575–1589, 2007.
- [55] RANGANATHAN, A. and DELLAERT, F., “Semantic modeling of places using objects,” in *Robotics: Science and Systems*, 2007.
- [56] ROTTMANN, A., MOZOS, Ó. M., STACHNISS, C., and BURGARD, W., “Semantic place classification of indoor environments with mobile robots using boosting,” in *AAAI Conference on Artificial Intelligence*, pp. 1306–1311, 2005.
- [57] SCHÖLKOPF, B., PLATT, J. C., SHAWE-TAYLOR, J., SMOLA, A. J., and WILLIAMSON, R. C., “Estimating the support of a high-dimensional distribution,” *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.
- [58] SCHÖLKOPF, B., SMOLA, A., and MÜLLER, K.-R., “Nonlinear component analysis as a kernel eigenvalue problem,” *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.
- [59] SE, S., LOWE, D. G., and LITTLE, J. J., “Vision-based mobile robot localization and mapping using scale-invariant features,” in *Proc. IEEE Intl. Conf. Robotics and Automation*, pp. 2051–2058, 2001.
- [60] SIVIC, J. and ZISSERMAN, A., “Video Google: A text retrieval approach to object matching in videos,” in *The IEEE Conf. on Computer Vision*, vol. 2, pp. 1470–1477, 2003.
- [61] SWAIN, M. J. and BALLARD, D. H., “Color indexing,” *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.
- [62] SZUMMER, M. and PICARD, R. W., “Indoor-outdoor image classification,” in *CAIVD*, pp. 42–51, 1998.
- [63] THRUN, S., FOX, D., BURGARD, W., and DELLAERT, F., “Robust Monte Carlo localization for mobile robots,” *Artificial Intelligence*, vol. 128, no. 1-2, pp. 99–141, 2001.

- [64] THRUN, S., GUTMANN, J.-S., FOX, D., BURGARD, W., and KUIPERS, B., “Integrating topological and metric maps for mobile robot navigation: A statistical approach,” in *AAAI Conference on Artificial Intelligence*, pp. 989–995, 1998.
- [65] TORRALBA, A. B., MURPHY, K. P., FREEMAN, W. T., and RUBIN, M. A., “Context-based vision system for place and object recognition,” in *The IEEE Conf. on Computer Vision*, pp. 273–280, 2003.
- [66] TUYTELAARS, T. and SCHMID, C., “Vector quantizing feature space with a regular lattice,” in *The IEEE Conf. on Computer Vision*, 2007.
- [67] ULRICH, I. and NOURBAKHSI, I. R., “Appearance-based place recognition for topological localization,” in *Proc. IEEE Intl. Conf. Robotics and Automation*, pp. 1023–1029, 2006.
- [68] VAN GEMERT, J. C., GEUSEBROEK, J.-M., VEENMAN, C. J., and SMEULDERS, A. W., “Kernel codebooks for scene categorization,” in *European Conf. Computer Vision*, 2008.
- [69] VAPNIK, V. N., *The Nature of Statistical Learning Theory*. Springer, 1999.
- [70] VARMA, M. and RAY, D., “Learning the discriminative power-invariance trade-off,” in *The IEEE Conf. on Computer Vision*, 2007.
- [71] VIOLA, P. and JONES, M., “Robust real-time face detection,” *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.
- [72] VOGEL, J. and SCHIELE, B., “Semantic modeling of natural scenes for content-based image retrieval,” *International Journal of Computer Vision*, vol. 72, no. 2, pp. 133–157, 2007.
- [73] WEISS, G. M., “Mining with rarity: A unifying framework,” *ACM SIGKDD Explorations*, vol. 6, no. 1, pp. 7–19, 2004.
- [74] WEISS, Y., TORRALBA, A., and FERGUS, R., “Spectral hashing,” in *Advances in Neural Information Processing Systems 21*, pp. 1753–1760, 2009.
- [75] WINN, J., CRIMINISI, A., and MINKA, T., “Object categorization by learned universal visual dictionary,” in *The IEEE Conf. on Computer Vision*, vol. 2, pp. 1800–1807, 2005.
- [76] WU, J. and REHG, J. M., “Where am I: Place instance and category recognition using spatial PACT,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.
- [77] ZABIH, R. and WOODFILL, J., “Non-parametric local transforms for computing visual correspondence,” in *European Conf. Computer Vision*, vol. 2, pp. 151–158, 1994.

- [78] ZIVKOVIC, Z., BOOIJ, O., and KRÖSE, B. J. A., “From images to rooms,” *Robotics and Autonomous Systems*, vol. 55, no. 5, pp. 411–418, 2007.