UAB

Universitat Autònoma
de Barcelona

Compact, Adaptive and Discriminative Spatial Pyramid for Improved Scene and Object Classification

A dissertation submitted by **Noha Elfiky** at Universitat Autònoma de Barcelona to fulfil the degree of **Doctor en Informàtica**.

Bellaterra, March 2012

Director: **Dr. Jordi Gonzàlez i Sabaté**
Centre de Visió per Computador
Dept. de Ciències de la Computació, Universitat Autònoma de Barcelona.
Co-director: **Dr. Xavier Roca i Marvà**
Centre de Visió per Computador
Dept. de Ciències de la Computació, Universitat Autònoma de Barcelona.

This document was typeset by the author using LaTeX 2_ε.

The research described in this book was carried out at the Computer Vision Center, Universitat Autònoma de Barcelona.

ISBN XX–XXXX–XX–X

Printed by Ediciones Gráficas Rey, S.L.

To my parents.

Acknowledgements

There are some people without whom this thesis would not be possible to come to light. First of all, I would like to thank my director: Jordi Gonzalez and my co-director: Xavi Roca for having confidence in me and for their continuous support and courage to me during my Ph.D. and also for giving me the chance of collaboration with other laboratory and professors that i had learned a lot from them. I would also like to thank Theo Gevers, Joost van de Weijer, Arjan Gijsenij and Fahad Shahbaz Khan for the fruitful collaboration, for guiding, encouraging my research, and teaching me lots of stuff that enriched my background knowledge while working with them.

I would also like to thank all the people at the Center of Computer Vision (CVC) by a way or another who had contributed to this thesis. I would like to thank Andrew Bagdanov for setting up the cluster and introducing it to all the students at CVC. I also want to thank my colleagues Jorge Bernal and Sheida Beigbour for helping me with the Spanish language and paperwork. Special thanks to my best friend Wenjuan Gong, with whom i started this long journey and for offering me a great company that i enjoyed a lot. I would also like to thank her for providing me with the feeling of home and for the amazing moments we had shared together (laughter, suffer, traveling, etc.).

I would also like to thank all the people i get to know during my stage at the University of Amsterdam, who made my stay pleasant. I am so grateful to all the members that i had collaborated with them during this stage, especially, Theo Gevers and Arjan Gijsenij.

A very special thanks is dedicated to my family, my lovely parents, and my wonderful sisters “Shaimaa” and “Hagar”, who suffered a lot from being far away from them for such a long time and who were very interested in what i am doing, and for encouraging and supporting me all the time. Specifically, I want to thank my parents for caring a lot about my education since i was a child and for all their patience and support that they had offered me during all my life. Thanks also to my amazing parents who are always reminding me that nothing is achieved without effort. My father who always gave me a helping hand without asking anything in return, and who is always being thinking of what is the best for me before himself. My lovely mother who always asked me to come back to make sure by herself that i am really fine. She is always worrying about everything and wants me to be the best and make sure that i never missed anything. Thanks to “Shaimaa” and “Hagar” for being not only great sisters, but also very closer friends. I would like to thank them deeply because of the great times we had together and that they are always keen on drawing a smile on my face.

Everlasting gratitude is for my wonderful husband “Ahmed Mounir”. I would like to thank him deeply from the bottom of my heart for motivating and encouraging me to achieve this dream. He always believed in me and he always knew how to get the best out of me. He is always pushing me forward to improve myself not only professionally but also as a person. Thanks a lot “Ahmed” for always being next to me in the hard moments and in the good moments and for inspiring and supporting me and not allowing anything to separate us.

Abstract

The release of challenging datasets with a vast number of images, requires the development of efficient image representations and algorithms which are able to manipulate these large-scale datasets efficiently. Nowadays the Bag-of-Words *(BoW)* based image representation is the most successful approach in the context of object and scene classification tasks. However, its main drawback is the absence of the important spatial information. Spatial pyramids *(SP)* have been successfully applied to incorporate spatial information into *BoW-based* image representation. The main SP approach, works by repeatedly sub-dividing the image into increasingly finer sub-regions by doubling the number of divisions on each axis direction, and further computing histograms of features over the resulting sub-regions. Observing the remarkable performance of spatial pyramids, their growing number of applications to a broad range of vision problems, and finally its geometry inclusion, a question can be asked what are the limits of spatial pyramids.

Within the *SP* framework, the optimal way for obtaining an image spatial representation which is able to cope with it's most foremost shortcomings, concretely, it's high dimensionality and the rigidity of the resulting image representation still remains an active research domain. In summary, the main concern of this thesis is to search for the limits of spatial pyramids and try to figure out solutions for them. This thesis explores the problem of obtaining compact, adaptive, yet informative spatial image representations in the context of object and scene classification tasks.

In the first part of this thesis, we first analyze the implications of directly applying the state-of-the-art compression techniques for obtaining compact *BoW-based* image representation within the context of spatial pyramids. We then introduce a novel *SP* compression technique that works on two levels; (i) compressing the least informative spatial pyramid features, followed by, (ii) compressing the least informative *SP* regions for the purpose of obtaining compact, and adaptable *SP*.

We then introduce a new texture descriptor that represents local image texture and its spatial layout. Texture is represented as a compact vector descriptor suitable for use in standard learning algorithms with kernels. Experimental results show that texture information has similar classification performances and sometimes outperforms those methods using only shape or appearance information. The resulting spatial pyramid representation demonstrates significantly improved performance on challenging scene classification tasks.

In the second part of this thesis, we present a novel technique for building adaptive spatial pyramids. In particular, we investigate various approaches for learning adaptive spatial pyramids, which are specially tailored for the task at hand. To this end, we analyze the use of (i) standard generic 3D scene geometries; the geometry of a scene is measured based on image statistics taken from a single image. (ii) discriminative spatial partitionings, which are generated based on an information-theoretic approach. The proposed method is tested on several challenging benchmark object classification datasets. The results clearly demonstrated the effectiveness of using adaptive spatial representations, which are steered by the 3D scene geometry present in images.

In the third part of this thesis, we investigate the problem of obtaining compact spatial pyramid image representations for object and scene classification tasks. We present a novel framework for obtaining compact spatial pyramid image representation up to an order of magnitude without any significant reduction in accuracy. Moreover, we also investigate the optimal combination of multiple features such as color and shape within the context of our novel compact pyramid representation.

Finally, we investigate the importance of using the spatial knowledge within the context of color constancy as an application. To this end, we present a novel framework for estimating the image illuminant based on spatial 3D geometry for learning the most appropriate color constancy algorithm to use for every image region. The final image illuminant is obtained based on a weighted combination of each individual illuminant-estimate obtained per region. We test and compare our performance to that of previous state-of-art methods. We will show that the set of innovations introduced here lead to a significant increase on performance on challenging color constancy datasets.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Computer vision is a scientific discipline that focuses on theory and applications for obtaining information from visual data. Due to the widespread availability of personal and professional imaging devices, low cost of multimedia storage and ease of content transmission and sharing, the need to automatically analyze and organize large amounts of visual data becomes more and more prominent. To automatically search an image, search engines typically make use of the text associated with it. This reliance on the meta-data and associated text, while ignoring the semantics of an image, hampers the search performance. Contrary to modern search engines, humans have an outstanding ability of classifying images based on their visual content. When asked about the content of an image, a person can tell whether there is a car, a building or a coast, etc., in a fraction of a second [118, 130].

One of the main tasks of computer vision is the recognition task, where the goal is to determine whether or not the image data contains some specific property or object or activity. Applications for visual recognition range from automated organization of personal multimedia to large-scale surveillance and security systems. Visual recognition receives a lot of attention from both the scientific community and the industry, but the general problem still remains challenging due to the large variations between images belonging to the same category. Several other factors such as significant fluctuations in viewpoint and scale, illumination changes, partial occlusions and multiple instances also have a significant influence on the final results and that make the problem even more complicated [19, 39, 47].

Different varieties of the recognition problem are described in the literature. An important recognition problem is classification problem, where one has to assign unknown data to a set of predefined categories. In a generic classification setup, the typical visual classification problem is texture classification (see the top row of fig. 1.1), where a system provided with texture exemplars learns to classify unseen samples. Such a visual classification setup can be extended to objects, where systems should predict labels for items depicted in images (see the middle row of fig. 1.1). Even though the problem setup remains the same, in a real-world scenario object classification becomes substantially more difficult due to possible presence of intra-class variations, background clutter, occlusions and object articulations. Finally, the same setup can be used to name activities and events in videos (see the bottom row of fig. 1.1). This brings new challenges like temporal variability of an action or presence of camera motion and shot boundaries. Furthermore, the amount of data to process might require specialized techniques and further complicates the classification problem.

In this dissertation we mainly focus on scene and object classification in images, which is currently one of the most rapidly developing research directions in computer vision. But since we develop generic methods, we also can apply our research to other domains and investigate related problems.

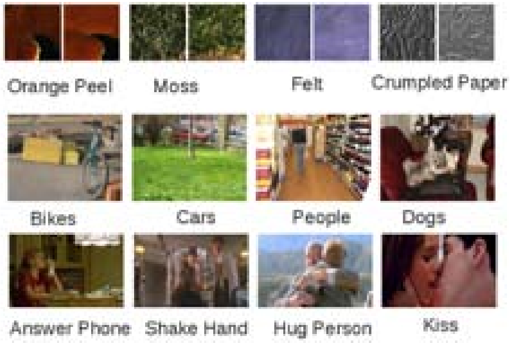


Figure 1.1: Various types of data in the scope of visual recognition: texture samples (top), images with objects (middle), action videos (bottom).

1.1 Motivation

Describing images by their contents can be very useful for organizing and accessing the huge amount of image data generated every day. Moreover there are lots of applications that can benefit from content-based image classification:

Image search. Image search is one of the direct application when people talk about content-based image classification, where the goal is to find all images in a given dataset which have a specific content. For instance, we can think about searching images in the largest database in the world, Internet image search engines, or simply provide applications to search images in a personal computer. Nowadays the poor performance when searching images in Internet is due to their use of the image filename or surrounding HTML rather than the actual image content. However, the natural way to find images is to search visually -as humans do- using computer vision methods. Moreover, many companies have large archives of images which they wish to search them in a more rapid and efficient manner that is similar to what the humans do.

Video search. Lots of adverts and video data have been generated during last years. People working in marketing is often interested in looking for coffee adverts televised in the past years, or adverts filmed in the mountains. Nowadays all these adverts are manually annotated and stored in databases using meta-data information. It would be very useful to provide techniques to access them automatically, by their content. Also producers or film directors would be interested in recovering by an automatic way those shots of movies filmed near a lake, or those shots where a certain actor/actress appears in the middle of the ocean.

Surveillance. Fundamental systems remain relatively unintelligent requiring a person screening the image sequences, looking for suspicious people and unusual events. Advanced systems try to automatically detect these unusual events. A subject of relative importance is that related to understand crowded environments (e.g. a football stadium) and/or detecting risk situations (e.g. fights).

Robotics. Provide eyes to a robot is maybe one of the most ambitious things in the computer vision field. In this way a completely autonomous robot specialized to recognize certain objects of interest will be able to substitute humans in dangerous situations such as underwater exploration, fireman help etc.

Video compression. Due to the very limited bandwidth of a number of important communication channels (e.g. wireless, underwater, low-power camera networks, etc.), video communication over such channels requires substantial compression of the video signal. One of the most promising answers to this challenge is to adopt a new compression paradigm that relies heavily in scene understanding. It would allow the compression of different objects in a scene with specific compression levels in such a way as to adjust the trade-off between space reduction and visual quality on a per-object basis. The basic idea is that important objects such as actors should retain the highest visual quality, while objects in the background can be encoded with lower quality to save bytes. Here, computer vision must help to perform automatically the task of separating a video into the objects of which it is composed.

However, there is not a clear solution for the applications above mentioned and this is due to the difficulties together with the challenges confronting the image classification task. Recently, many studies on image classification have been presented since 2000, and it is not yet a solved problem. This is because it is one of the most challenging and ambitious problems on computer vision. Humans are able to recognize a tree even if it is far away or if it is very close to us. The same tree has different appearances depending on the season of the year: it has no leaves in winter, brown leaves in autumn, green leaves in spring etc. and humans can recognize it in all these situations. There are lots of things that humans can do automatically but are still a challenge in computer vision. Next, we will discuss the major aspects we have to take into account to develop a robust image classification system:

Illumination changes. An important concern to take into account is the illumination changes within images. For example, if we look at fig. 1.2 we can recognize three different coast scenes under different illumination conditions are presented, and we are able to classify them. Although, this is a trivial task for humans, it is not for computers. Hence, we need to develop robust systems which are able to recognize objects and scenes under these different illumination conditions.

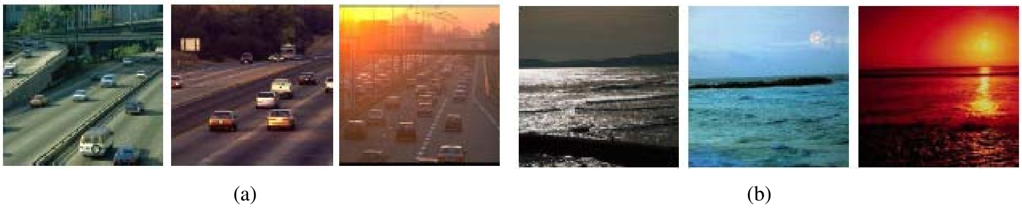


Figure 1.2: Illumination Challenge. (a)Three road scenes affected by different illumination conditions, (b) three coast scenes affected by different illumination conditions.

Intra-class variability. Identifying instances of general object or scene classes is an extremely difficult problem. This is because of the variations among instances of many common object classes, many of which do not afford precise definitions. A coast scene can come in different ways: a coast with mountains, a cliff coast or a coast with just water (see fig. 1.3). This means that we need an approach that can generalize across all possible instances of certain categories.

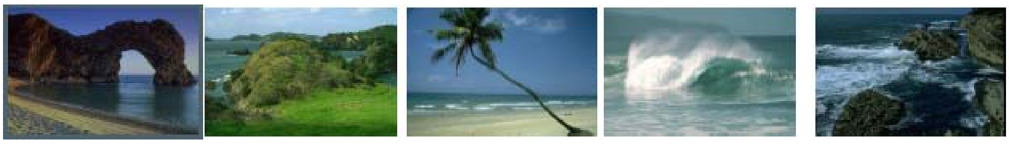
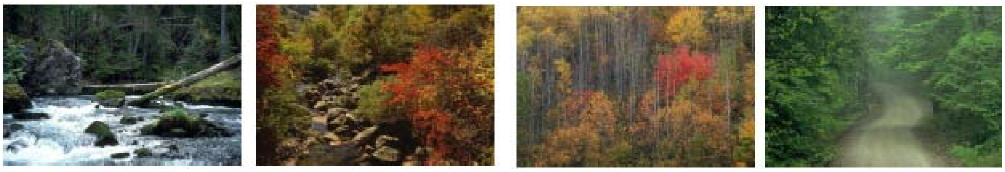


Figure 1.3: Intra-Class Variation Challenge. Different coast scenes which present high intra-class variation.

Inter-class variability. Related to the intra-class variability problem, another major difficulty is the inter-class variability within the model. We do not want to confuse between objects or scenes of different categories that are quite similar. For example, the forest and river scene are not labeled as the same category and we can see in fig. 1.4(a-b) that they could be easily confused, as they have a very similar texture, appearance and shape characteristics.

Scale invariance. Another important aspect to take into account for the task of image classification is the scale variability challenge. For instance, we can have some objects (e.g a person) which appear at different scale in the images. We can also have images with a mountain in front of us, or images with a mountain far away and in both cases it is a mountain scene that the system must classify. Fig. 1.5 shows some examples related to this problem.



(a) (b)

Figure 1.4: Inter-class variation problem. Two forest images (a) that could be easily confused by the two river (b) images.

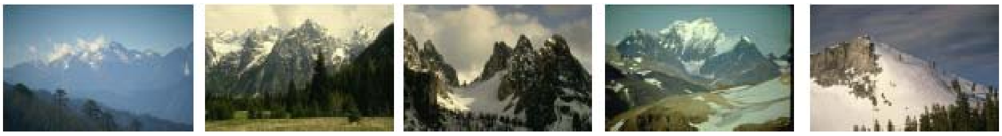


Figure 1.5: Scale variation problem: Mountain scenes from different scales.

Others. Rotations, occlusions and view point variations should also be taken into account. A part from the above mentioned problems, for the scene classification task there are other factors related to the human perception that we would like to comment: the ambiguities and the subjectivity of the viewer. The obtainable classification accuracies depend strongly on the consistency and accuracy of the manual annotations, and sometimes annotation ambiguities are unavoidable. For example, the annotation of mountains and open country is quite challenging. Imagine an image with fields and snow hills in the far distance: is it open country or mountain? Vogel and Schiele [176] analyzed in detail the ambiguities between scene categories, showing that there is a semantic transition between categories. Their experiments with human subjects showed that many images cannot be clearly assigned to one category.

1.2 Objectives

The objective of this work is to develop simple yet efficient image representations for scene and object classification tasks. Concretely, given a set of images our objective is to classify them by the scene or the object category they contain (e.g. coast, forest, bikes, cars, horses). This problem has been the subject of many recent papers [70, 92, 97, 125, 190, 191] using specific scene classification datasets, the Pascal Visual Object Classes datasets, the Caltech datasets, etc. The release of challenging datasets with increasing numbers of object and scene categories, is forcing the development of efficient image representations and algorithms that are efficient both in training and testing and that can cope with large scale image classification datasets with multiple categories.

In recent years the bag-of-words based framework has been demonstrated to be one of the most successful approaches to scene and object classification tasks [19, 96, 136]. The first stage in the pipeline, *feature detection*, involves detecting keypoint regions in an image either by employing dense sampling or through interest point detection. The feature detection step is then followed by the *feature description stage*, where the selected keypoint regions are further descripted using local descriptors. Afterwards, a *visual vocabulary* is constructed by quantizing the local descriptors into fixed-size visual vocabulary. Finally, the *assignment stage* is performed, where the image is represented by a histogram over the visual code-book. These histogram representations are then used to train a classifier to recognize different scene and object categories. This relatively simple image representation was found to obtain superior results on image classification tasks even for difficult cases.

Image representations obtained using the standard bag-of words approach lacks of the important spatial information. The spatial pyramid matching approach proposed in [97] provides a simple way of introducing spatial information within the bag-of-words framework. The technique works by dividing an image into increasingly finer sub-regions and constructing histograms for each region. This results in a multi-resolution histogram that captures the spatial layout of an object or a scene. Although spatial pyramid schemes yield excellent performance, it has fixed rigid split-ups and high dimensional histogram representation. Fig. 1.6 shows an image together with the spatial pyramid scheme applied upon it. The final dimensionality of the spatial pyramid histogram depends on the size of the visual vocabulary and increases by going deeper towards finer pyramid levels. Consequently, a compact spatial pyramid image representation is expected to obtain an efficient image representation that is capable of improving the image classification performance.

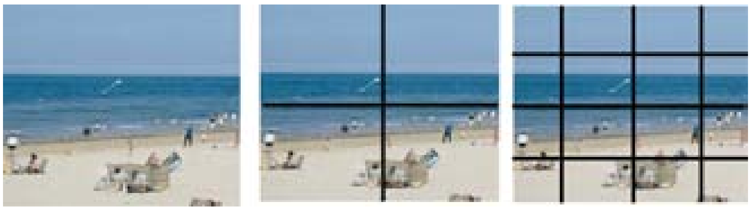


Figure 1.6: An example image with spatial pyramid scheme [97]. An image is divided into finer rigid regions and a histogram is constructed for each region. The histograms from all regions are then concatenated into a single representation. The dimensionality of the final histogram is equal to the number of regions times the size of the visual vocabulary.

The goal of this thesis is to search for the limits of the spatial pyramid image representation within the bag-of-words image classification framework and to further address these limitations by developing a simpler yet efficient spatial representation that is capable of improving the task of object and scene classification. The resulting image classification system would then be capable of reducing the computational cost for learning the classifier, discriminating among the dataset categories that we want to classify, as well as improving the image classification performance required. In particular, we are looking for a trade-off among efficiency and performance. These characteristics are crucial for enabling the classification system to function in real-world applications. To reach this goal, we will address three main problems which we have encountered in typical spatial pyramid setups, namely, (i) the high dimensional image representation, (ii) the lack of flexible spatial pyramid representation, and (iii) the optimality of fusing multiple features within the context of spatial pyramids.

1.3 Thesis scope and Contributions

The conventional spatial pyramid scheme has been demonstrated to significantly improve the performance over the standard bag-of-words approach. However, this performance gain is obtained at a high computational and memory cost due to the high dimensionality of the spatial pyramids. Concretely, the scope of this thesis is to address the spatial pyramid main drawbacks, which we have encountered in typical spatial pyramid setups. These drawbacks are (i) the high dimensional image representation, (ii) the lack of flexible spatial pyramid representation, and (iii) the optimality of fusing multiple features within the context of spatial pyramid. The resulting spatial image representation should reduce the computational cost without deteriorating the classification performance. Moreover, such a compact pyramid representation is also expected to allow for the combination of multiple visual cues efficiently.

The contributions of this thesis are an in-depth study and analysis of the conventional state-of-the-art spatial pyramid image representation framework using a broad set of scene, object classification datasets. We carefully investigate the limitations of such a framework and propose constructive solutions to overcome them. This analysis had lead us to the following approaches which allow us to improve the overall classification accuracy, while resolving the aforementioned drawbacks of the standard spatial pyramids. To this end, the contributions of this thesis can be divided into five main themes, summarized below. Along the thesis we show that these contributions, result in more insights and classification accuracy improvements compared to recent state-of art publications, in all cases using standard datasets and testing protocols. A more detailed account of the contributions and how they affect the final performance will be discussed in section 7.1.

A new spatial texture descriptor. The analysis of incorporating the spatial information within shape and appearance features proposed in [1, 18], demonstrated a significant gain in the performance over the standard bag-of-words cues. However much less attention is devoted for investigating the role of texture features within this context. In Chapter 3, we explore how the spatial distribution of texture information can benefit in classification tasks. In essence, we wish to assess how well an exemplar image matches (the texture of) another image. To this end, we extend the spatial pyramid method of Lazebnik et al. [97] to represent texture in the form of Local binary patterns (LBPs), which have been demonstrated to be one of the best features when working for classification tasks [3, 71, 72, 57, 189] together with the complementary color information [59]. In particular, we propose a novel descriptor which integrates color information with spatial patch-based LBPs features. This descriptor is termed *PC-TPLBP* (for pyramid of Colored Three-Patch Local Binary Patterns).

Compact and Adaptive pyramid shapes. To make use of the spatial pyramid more efficiently, a lower dimensional representation is highly desirable without significant loss of accuracy. An advantage of obtaining compact spatial pyramid representation is that it allows for the combination of visual cues without increasing the classification time. In Chapter 3, having developed the *PC-TPLBP* descriptor we then introduce a two-stage pyramid compression scheme for reducing the pyramid dimensionality for classification tasks. This contribution is obtained based on a two-fold approach, which works on compressing both of the pyramid *features* and *blocks* using the *Agglomerative Information Bottleneck (AIB)* theory. First, when applied for *feature* compression, our approach can be seen as a form of feature combination and selection for eliminating the least informative features within each pyramid level. Second, when used for *block* compression the least informative blocks are eliminated within each level in a class-specific learning fashion. We show that both folds are crucial for obtaining compact spatial pyramid representation, which is suitable for each particular category. For example, a category such as coast is best described using the horizontal-bars pyramid, while for the tall buildings category the vertical-bars pyramid are more suitable for representing it. The resulted representation is referred to as *CASPs* (for Compact Adaptive Spatial Pyramids).

Adaptive 3D Geometry-based Pyramid Shapes. To describe images which are not constrained in pose or that have significant background clutter, with rigid spatial pyramid proposed by lazebnik et al. [97] is not sufficient. Instead, we address this issue by considering a standard generic set of 13 pre-defined shapes, referred to as scene geometries in [119, 175]. These geometries cover broader range of standard generic spatial shapes, which are used for improving the task of image classification. In chapter 4, we propose an improved adaptive spatial pyramids in terms of adaptability and efficiency and complexity based on standard 3D scene geometries proposed in [119, 175]. In particular, we automatically learn whether to select or combine the most suitable shapes for each category among those predefined shapes. These learned shapes are fast to obtain, besides being compact, efficient and informative spatial image representation. The proposed method is tested on several benchmark object classification datasets and the results clearly demonstrate the effectiveness of learning those adaptive spatial shapes (i.e. geometries), which are steered by the standard generic 3D scene geometries.

Discriminative Compact Pyramid Representations. Recent advances in information-theoretic clustering techniques [38, 54, 152] permit us to revisit the problem of constructing compact and discriminative spatial pyramids. In chapter 5, a novel approach based on a *Divisive Information Theoretic feature Clustering (DITC)* algorithm [38] is used to to reduce the dimensionality of the pyramid significantly while preserving its accuracy for object and scene classification tasks. This method allows for reducing the size of a high dimensional pyramid representation up to an order of magnitude without significant loss of accuracy. Comparison with clustering based on *AIB*, shows that *DITC-based* pyramid compression obtains superior results at significantly lower computational costs. Moreover, an evaluation of the optimal way for combining multiple cues in the context of our compact spatial pyramid representation is performed. The experiments clearly demonstrate the effectiveness of the proposed approach at a significantly lower computational cost.

A novel NIS-geometry-based color constancy approach. The last contribution of this thesis is an efficient color constancy application based on the spatial image representation (i.e., 3D scene geometry) together with its natural statistics. In chapter 6, typical $3D$ scene geometries, called *stages* proposed in [119] as well as the images' natural statistics are applied for estimating the image illuminant. In particular, 13 pre-learned stage classifiers are used to assign the most appropriate shape (stage) to the input image. Next, we automatically select the most appropriate color constancy algorithm for each image segment based on the *Natural Image Statistics (NIS)* color constancy algorithm proposed in [62]. Finally, we combine the estimated illuminant for each image segment using various weighting schemes for obtaining the final estimated image illuminant. On standard color constancy benchmark datasets, the proposed approach is shown to improve the state-of-the-art methods.

1.4 Outline of the Thesis

The structure of the thesis is as follows: In chapter 2 we review recent literature of image classification focusing on the most frequent image descriptors. Special attention is given to the spatial-pyramid representation, which forms the basis for this thesis work. In chapter 3 we introduce a new spatial texture descriptor and study different ways for obtaining its compactness using the Agglomerative information bottleneck theory. In chapter 4, we propose a new approach that learns automatically the most appropriate pyramid shape for each category based on a set of pre-defined 3D scene geometries. In chapter 5 we propose a novel method for obtaining compact, efficient yet informative spatial pyramid image representation based on the divisive agglomerative information bottleneck theory. We also investigate how to optimally combine multiple features within the context of spatial pyramids. A novel geometry-based color constancy approach for estimating an image illuminante based on its local natural image statistics is shown in chapter 6. Finally, the dissertation is concluded in chapter 7. We reformulate the key contributions putting particular emphasis on the results we have obtained and discuss the perspectives for future work. The most used terminology and abbreviations are summarized in appendix A.

Chapter 2

Related work

In this chapter we review the most recent and significant work in the literature on image classification. Image classification is the problem of assigning an unknown image to a set of predefined categories. Image classification is a difficult task, due to the large variations between images belonging the same class. Several other constraints such as view point changes, variations in illumination, make image classification an extremely difficult task to accomplish. Moreover, the existence of a wide variety of image categories ranging from man-made categories such as car, bus, piano, etc. to natural categories such as plants, sheep, dolphins, etc. Such diversity further increases the problem. In order to introduce the field, in section 2.1 we initially discuss some basic ideas regarding the organization and representation of images.

The problem of image representation using low-level features has been studied in image and video retrieval for several years, and we review them in section 2.1.1. Then, in section 2.1.2 we pay special attention to the most recent representation methods which use local regions. Concretely, we review the bag-of-words approach which represents an image as a histogram of local features and is currently on of the most successful approaches to obtain very good performances for image classification tasks. The approach works by counting the number of occurrences of each visual word in an image. The histogram is then used as input to the classifier. A model is trained to use a set of training images by projecting the histogram values into a space in order to optimize the gap between examples of the different data set categories. Finally, given a test image the model is used to predict the category label of the image.

In this chapter, we provide a detailed overview of each stage of the bag-of-words pipeline. The bag-of-words framework consists of two main parts namely, image representation and the machine learning. To obtain an image representation, the subsequent stages to follow are feature detection, feature extraction, vocabulary construction and assignment. We provide an overview of each of these stages within the bag-of-words framework. In section 2.6, we provide an overview of existing approaches used to capture the important spatial image information within this framework. In section 2.7, we review the work done on color constancy, and how the spatial knowledge can benefit such applications. Finally, section 2.9 introduces the variety of datasets used in the thesis to assess the proposed methods.

2.1 Image representation

In the 1950s and early 1960s much of the research in visual recognition was focused on $2D$ pattern classification. From 1960s to 1980s early approaches to recognize objects in images were developed. Many techniques have been used to represent the content of an image. Here we classify them into two main approaches: (i) those methods which directly extract low-level features from the images, and (ii) those methods which model the image using local patches as intermediate representation. The philosophies of each approach as well as the main methods are described here below.

2.1.1 Modeling low level image representation

The problem of image classification is often approached by representing the images using low-level features (e.g color histograms). This representation is then used to classify the images into a category (e.g. street, coast). These methods consider that images can directly be described by their low-level properties. For instance, a forest scene presents highly textured regions (trees), the presence of straight horizontal and vertical edges denotes an urban scene, a red color represents a stop sign, and blue color represents a coast image etc. Among these methods we can distinguish two approaches: (i) global representations where the low-level features are computed over the whole image, and (ii) local representations where the image is first partitioned into several blocks, and then features are extracted from each of these blocks. Fig. 2.1a shows an example of a global model representation, and Fig. 2.1b shows an example of a local model representation, both using low-level features. These two models are reviewed here below.

Global models

Vailaya et al. [164, 163, 162] consider the hierarchical classification of vacation images, and show that a global representation can successfully discriminate between many scenes types using a hierarchical structure. Using binary Bayesian classifiers, they attempt to capture the image category from global image features under the constraint that the test image belongs to one of the classes. At the highest hierarchical level, images are classified as indoor or outdoor; outdoor images are further classified as city or landscape. Finally, a subset of landscape images is classified into sunset, forest, and mountain categories. Different qualitative measures, extracted from the whole image, are used at each level depending on the classification problem: indoor/outdoor (using spatial color moments); city/landscape (edge direction coherence vectors), and so on. The classification problem is addressed by using *Bayes* decision theory. The proposal reports an excellent performance over a set of 6931 images.

Chang et al.[30] use a global image representation to produce a set of category labels with a certain belief for each image. They manually label each training image with a category and train k classifiers (one for each category) using Support Vector Machines (SVM). Each test image is classified by the k classifiers and assigned a confidence score for the category that each classifier is attempting to predict. As a result, a k-nary label vector consisting of k-class membership is generated for each image. This approach is especially useful for Content

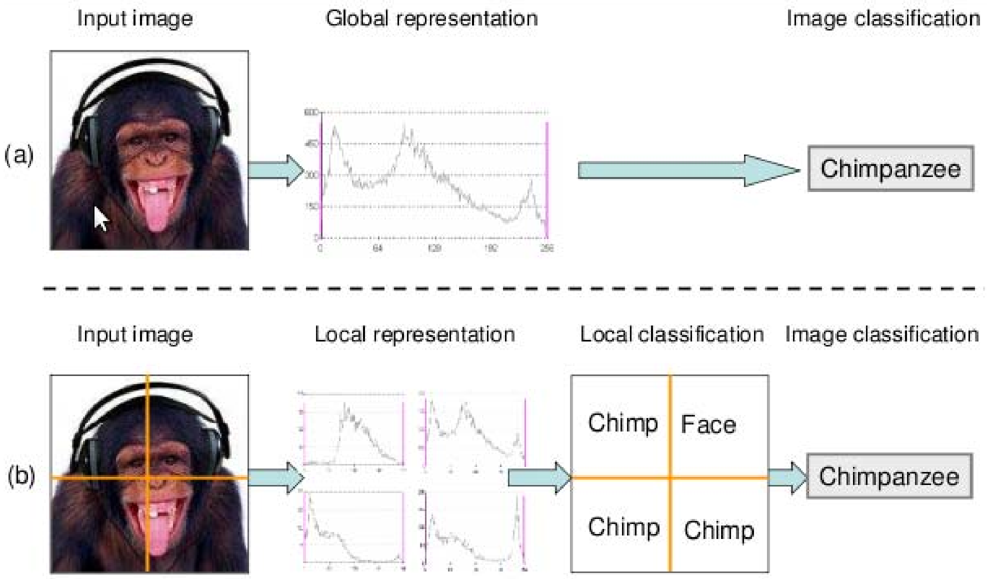


Figure 2.1: Example of global and local image representation. (a) Global image representation by using low-level features (e.g. a color histogram). This representation is the input of the classifier and a final category is given; (b) local image representation by using low-level features (e.g. a color histogram at each sub-block). Each sub-block is independently classified obtaining a category for each one. These results are finally combined to obtain an image category.

Based Image Retrieval (CBIR) and Relevance Feedback (RF) systems. Other authors have followed this global approach, although they have taken other aspects into account. For example, Shen et al. [81] makes emphasis on the type of features that must be used. The authors argue that due to the complexity of the visual content, a classification system cannot be achieved by considering only a single type of feature such as color, texture and shape alone and proposed Combined Multi-Visual Features. It produces a low-dimensional feature vector which is useful for an effective classification. Their method is tested on image classification using three different classifiers: SVM, K-Nearest Neighbors (K-NN) and Gaussian Mixture Models (GMM). Other authors use global edges or orientation histograms [158].

Local models

These approaches are a direct extension of the global low-level approaches described above. The global approaches use low-level features extracted from the whole image, while the local ones first split the image into a set of sub-regions, which are further represented by their low-level properties. Each block is then classified as a certain category and finally the image is categorized from the individual classification of each block.

The origin of this approach is found in 1997, when Szummer and Picard [155] proposed to independently classify image subsections to obtain a final result using a majority voting classifier. The goal of this work was to classify images as indoor or outdoor. The image is first partitioned into 16 sub-blocks, from which Ohta-space color histograms and MSAR texture features are then extracted. K-NN classifiers are employed to classify each sub-block using the histogram intersection norm. Finally the whole image is classified using a majority voting scheme from the sub-block classification results. They demonstrated that performance is improved by computing features on sub-blocks, classifying these sub-blocks, and then combining these results. Similar results were also obtained by Paek and Chang [129]. Moreover, they developed a framework to combine multiple probabilistic classifiers in a belief network. They trained classifiers for indoor/outdoor, sky/no sky and vegetation/no vegetation as secondary cues for the indoor/outdoor problem. The classification results of each one are then feed into a belief network to take the integrated decision.

The proposal of Serrano et al. [144] in 2004 shares this same philosophy, but using SVM for a reduction in feature dimensionality without compromising classification accuracy. Color and texture features are also extracted from image sub-blocks and separately classified. Thus indoor/outdoor labels are obtained for different regions of the scene. The advantage of using SVM instead of K-NN classifier is that the sub-block beliefs can be combined numerically rather than by majority voting, which minimizes the impact of sub-blocks with ambiguous labeling.

Discussion

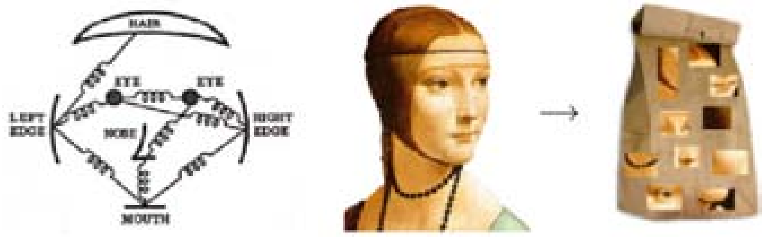
The main advantage of these methods is that they provide a very simple image representation. The main drawback of representing the images by low level features is that if images have a notable background clutter or there is lot of intra-class variability, this representation is not enough to discriminate among different categories. Some authors who use this representation argue that the images they use have low intra-class variation and can be easily separated by using low-level features. Moreover these methods have been used to classify among a few number of image categories (from 2 to 5).

2.1.2 Local patches representation

In this case images are represented by hundreds of local patches. They use a region detector to find a set of interesting parts of the image and then represent them by some kind of descriptor. The use of local descriptors has become popular for object detection and recognition. In recognition a matching between the region descriptors of the new image and those in the database is computed. The new image is classified if sufficient matches occur. These methods can be extended by applying geometric constraints. For example, Fergus et al.[48] model object classes as probabilistic constellations of parts. The appearance of each part, as well as pair-wise relations between parts, are modeled using Gaussian distributions. The model can be nicely visualized as a collection parts connected by springs, so that parts can move with respect to each other. The learning algorithm automatically looks for a configuration of detected image regions consistent over the training data. Recognition proceeds by first detecting potential part locations in an image, and then comparing hypotheses as to whether the observed features are generated by the category model or by the background model. A Bayesian extension of the constellation model of Fergus et al. [48], capable of learning from a small number (3-5) of training images, was presented by Fei-Fei et al. [46]. This work shows that knowledge about other object classes, here in the form of a prior, can help in learning new object class models.

Bag-of-Words Model

The bag-of-words (or sometimes called the bag-of-features) methodology is a popular choice for representing textual data for indexing purposes. Bag-of-words model has a long history of success in document retrieval [140], but it was not until the local features were further developed [111] and the idea of a visual vocabulary emerged [32, 150], that it was possible to bring this model to vision (see fig. 2.2, right). Since then, we can observe a growing popularity of the appearance-based methods. The bag-of-words representation was combined with modern machine learning techniques and has shown excellent performance for texture [33], object [149] and action recognition [142].



(a) Constellation models of M. Fischler

(b) bag-of-words of L. Fei-Fei

Figure 2.2: Current models for visual recognition: geometrical constellation models (left) and orderless bag-of-features (right).

Recent works have shown that local features represented by bags-of-words are suitable for image classification showing impressive levels of performance [47, 97, 136, 190]. Constructing the bag-of-words from the images involves the following steps: (i) Automatically detect regions/points of interest (local patches), (ii) compute local descriptors over these regions/points, (iii) quantize the descriptors into words to form the visual vocabulary, (iv) find the occurrences in the image of each specific word in the vocabulary in order to build the bag-of-words (histogram of words). Fig. 2.3 schematically describes the four steps involved in the definition of the bag-of-words model. The advantage of this method is its simplicity and the relatively small amount of supervision required. Labeling training data only requires indicating the image category. Moreover these methods have been used to classify images into a big number of categories (up to 100).

First works using the "bag-of-words" representation can be found in the literature related to texture classification. The goal of these works is to recognize textures captured from different camera viewpoints, and under varying illumination. Leung and Malik [100] quantized responses of a filter bank applied densely over an entire image. This quantization of appearance descriptors are called "Texton" and textures are represented by distributions of textons.

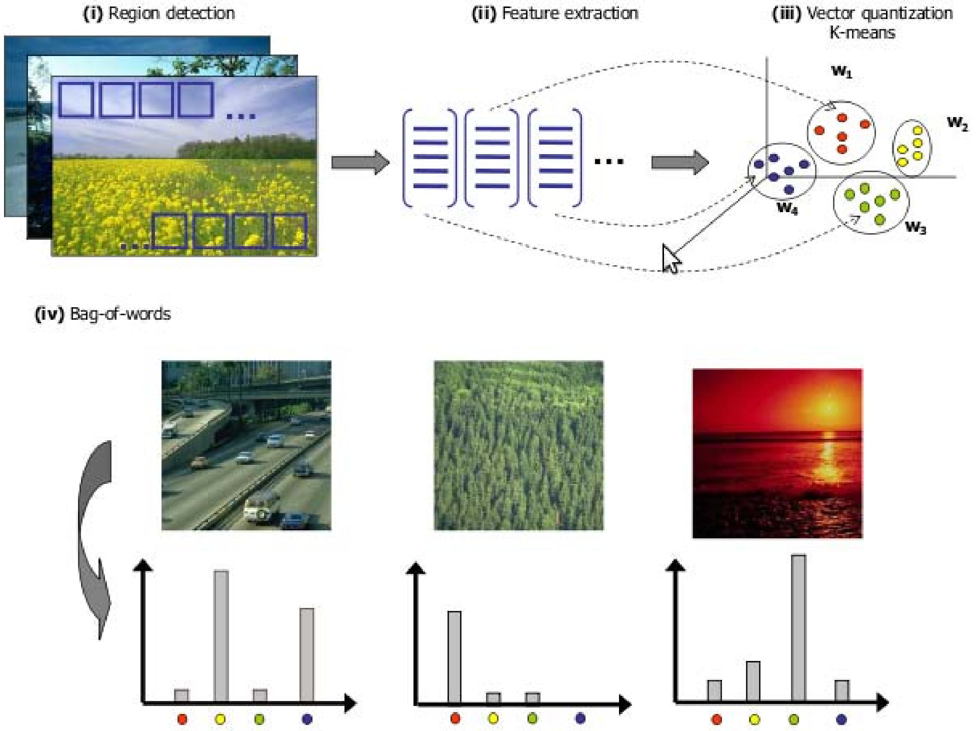


Figure 2.3: Four steps to compute the bag-of-words when working with images. (i-iii) obtain the visual vocabulary by vector quantizing the feature vectors, and (iv) compute the image histograms bag-of-words for images according the obtained vocabulary.

Varma and Zisserman [173] modified this approach by quantizing small image patches rather than filter responses. Lazebnik et al. [95] address texture classification using quantized affine covariant regions.

Recently, these representations have been extended for object recognition and scene classification. Perronnin et al. [133] defined a universal vocabulary, which describes the content of all the considered images, and class visual vocabularies which are obtained through the adaptation of the universal vocabulary using class-specific data. While previous approaches characterize an image with a single histogram, here an image is represented by a set of histograms, one per class. Each histogram describes whether an image is more suitably modeled by the universal vocabulary or the corresponding adapted vocabulary. The Universal vocabulary is trained using maximum likelihood estimation (MLE) and the class vocabularies are adapted using the maximum a posteriori (MAP) criterion. They successfully test the method classifying images like sunset, underwater, cars, bikes.

Traditional bag-of-words techniques, as described above, do not take the spatial information into account. However, in complex natural images, image classification systems can be further improved by using contextual knowledge like common spatial relationships between neighboring local objects [109] or the absolute position of objects in certain scenes [177]. While the above methods have shown to be effective, their neglect of spatial structure ignores valuable information which could be useful to achieve better results for image classification. Lazebnik et al. [97] proposed a method which is based on spatial pyramid matching of Grauman and Darrell [66]. Pyramid matching works by placing a sequence of increasingly coarser grids over the feature space (in this case over the image) and taking a weighted sum of the number of matches that occur at each level of resolution (L). At any fixed resolution, two points are said to match if they fall into the same bin of the grid; matches found at finer resolutions are weighted more highly than matches found at coarser resolutions. The resulting spatial pyramid is an extension of the bag-of-words image representation, it reduces to a standard bag-of-words when L = 0 (see fig. 2.6).

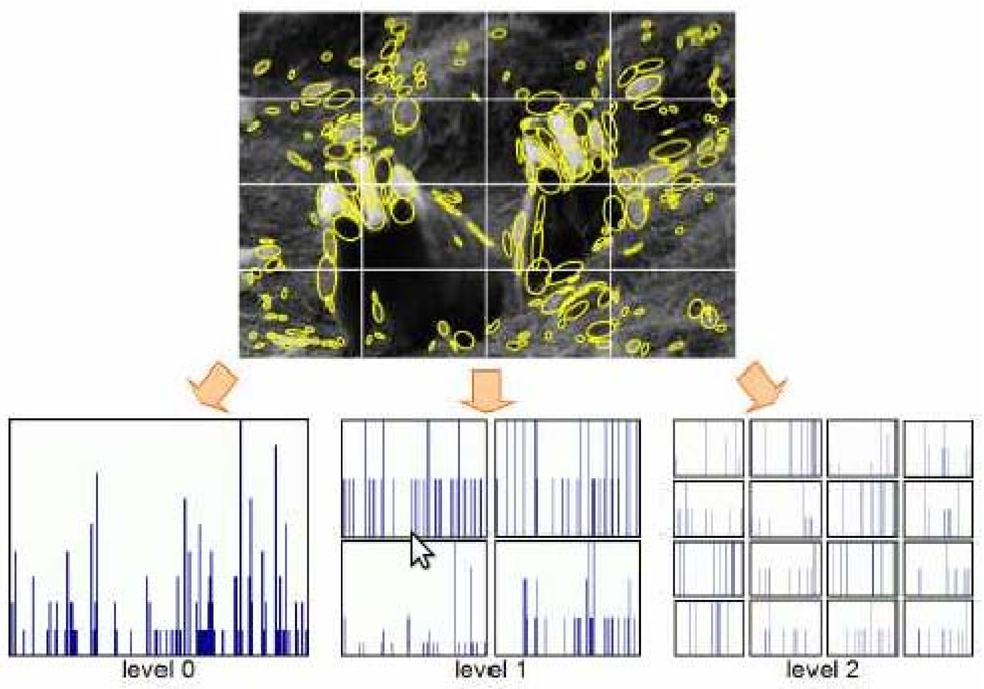


Figure 2.4: The Pyramid matching method of Lazebnik et al. [97]. Set of histograms computed over a multi-level pyramid decomposition of the image.

2.2 Feature Detection

The first stage within the bag-of-words approach involves detecting keypoints or regions in an image. There exist multiple strategies for selecting regions in an image. These strategies can be divided into two broad categories namely: dense sampling and interest point sampling strategies. The dense sampling technique works by scanning the image with either single or multiple scales at fixed locations forming a grid of rectangular windows. Dense sampling scheme is often beneficial for scene classification since all regions in the image provide information for the recognition task.

The second class of sampling strategy employed to find regions is called interest point sampling. Interest point strategy employed to find regions is called interest point sampling. Interest point techniques rely on finding salient points (such as corners, blobs etc.) in an image. Interest point strategies are often helpful for object recognition task as they ignore the homogeneous areas and focus on the object and its surroundings in an image. Several interest point strategies have been proposed in the literature [113, 120]. The Harris-Laplace point detector [113, 170] focuses on locating corners that are scale invariant in an image. The Laplacian operator is used to find the scale of the corner. Other than finding corners in an image, there also exists blob like structures in an image. Laplacian-of-gaussian is a commonly used blob detector where an image is convolved using a guassian kernel at certain scales to obtain a scale space representation. Most of the existing interest point schemes make use of shape saliency as a selection criteria for detection.

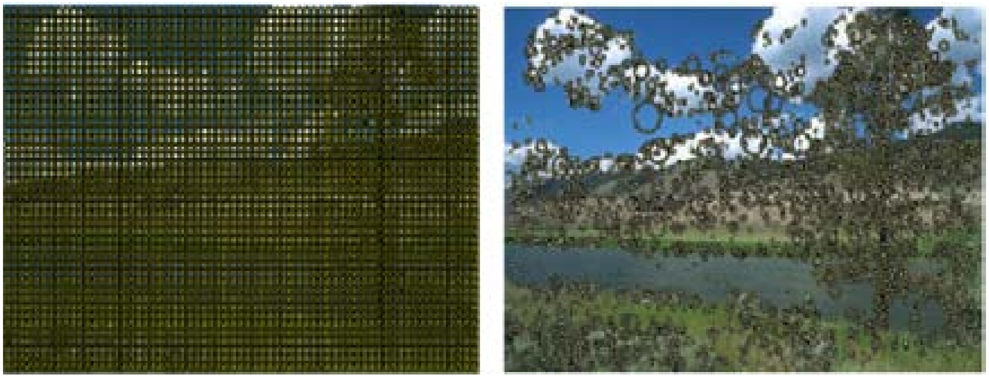


Figure 2.5: Sampling strategies used for selecting regions in an image. The left image shows a dense grid representation. The right image shows an interest point sampling techniques (blob detection).

Among the color-based interest point detectors proposed in the literature, color saliency boosting [170] is the most commonly used approach. The method exploits the saliency of color edges which is computed by applying information theory to the statistics of the color image derivatives. The color boosting approach has been successfully applied for object recognition and retrieval tasks [166]. Fig 2.5 shows example of different point sampling strategies. The dense sampling is covering the whole image while the interest point sampling target salient regions of an image.

2.3 Feature Extraction

Detection of local image regions is only the first part of the feature extraction process. The second part is the computation of descriptors to characterize the appearance of these regions.

A goal descriptor should be distinctive, so as to provide strong consistency constraints for image matching yet robust to illumination changes and other appearance variations. Many features such as color, texture, shape have been used to describe visual information for object and scene classification. In the next paragraphs, we review a variety of the most recent descriptors used for image classification which are based on appearance, shape, and texture information.

2.3.1 Appearance Information

Most of the recently proposed image classification methods use the Scale Invariant Feature Transform (SIFT) descriptor proposed by Lowe [104]. This descriptor takes each region, finds its gradients and then normalizes for orientation by finding the dominant orientation rotating the region so as to make it axis aligned. Then 8-bin orientation histograms are formed of the gradients in each cell of a 4×4 spatial grid overlaid on the region. Each region is described by a $4 \times 4 \times 8 = 128$ dimensional vector (fig. 2.6 shows an overview of this method). The idea is that the loose grid gives a little bit of slop to accommodate minor translation and scale offsets due to inexact feature detection while the gradient based representation makes it less sensitive to illumination changes. It has been demonstrated that these features achieve a higher performance for scene and object classification [112, 19, 39, 179]. Some extensions of the original SIFT features exist. PCA-SIFT [83] and the Gradient location and orientation histogram (Gloh) [112] from which change the location grid and use PCA for dimensionality reduction. A different matching scheme called SURF (Speeded Up Robust Features) was presented by Bay et al. [67]. The standard version of SURF is faster than SIFT and proved to be more robust against different image transformations than SIFT. SURF is based on sums of $2D$ Haar wavelet responses and makes an efficient use of integral images.

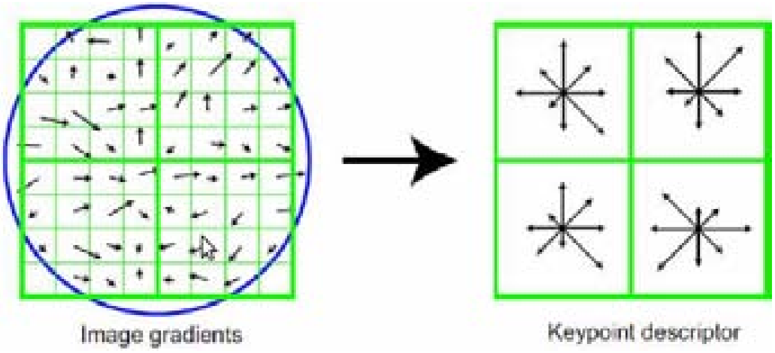


Figure 2.6: The SIFT descriptor of Lowe [104]. On the left are the gradients of an image patch. The blue circle indicates the Gaussian center-weighting. These samples are then accumulated into orientation histograms summarizing the contents over 4×4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. A 2×2 descriptor array computed from an 8×8 set of samples is shown here.

Color Information

Color is an important component of the natural image categories [141]. However, outdoor scenes are especially complex to deal with interms of the lighting conditions and the fact that color-based features suffer from the problem of color constancy [86]. Buluswar and Draper [22] provided a survey detailing analysis and causes of color variation due to illumination effects in outdoor images. Color being undoubtedly one of the most interesting characteristics of the natural world, which can be computationally treated in many different ways. In many cases the basic *RGB* components may provide valuable information about the environment. However, the perceptual models, such as CIE or HSI, are more intuitive and therefore enable the extraction of characteristics according to the model of human perception. Invariance and discriminative power of the color invariants is experimentally investigated in [**?**], showing the invariants to be successful in discounting shadow, illumination, highlights, and noise.

Some authors, like Ohta [122], have proposed their own color space. Celenk [27] proposed operating with the CIE (L*, a*,b*) uniform color coordinate system L*, H* and C* (Luminance, Hue and Chroma). This color space defines approximately a space having uniform characteristics. Campbell et al. [25] also proposed a set of color parameters in order to work with outdoor scenes. Similarly, Mori et al. [115] proposed the use of the *r-b* model (where r and b denote normalized red and blue components respectively) in order to solve the problems of hue shift, due to outdoor conditions and shadows. We can find in the literature some models based on the Hue, Saturation, and Intensity (HSI) for example, the model described by Smith [153] or the alternative proposed by Tenenbaum [157]. Yagi et al. [186] calculates the hue and intensity based on Smith work, and proposed a different way to obtain the saturation. The most typical way to calculate the components HSI is described in [64].

In 2002, Buluswar and Draper [23] developed models for illumination and surface reflectance to be used in outdoor color vision, and in particular to predict the color of surfaces under various outdoor conditions. Moreover, they demonstrated the disadvantages of using the *CIE* model for predicting color in outdoor images. Berwick and Lee [13] presented a framework of logarithmic chromaticities for the interpretation of the image color change due to illumination pose and color. More recently, in 2007, Joost et al. [180] introduced the color names descriptor, which involve the assignment of linguistic color labels to image pixels. The 11 basic color terms of the English language are black, blue, brown, grey, green, orange, pink, purple, red, white, and yellow [12]. Color names display a certain amount of photometric invariance as several shades of a color are mapped to the same color name. It also provides an added advantage of allowing the description of achromatic colors such as black, grey, white, etc.

Some approaches of using color with SIFT features (*ColorSIFT*) have also been proposed recently in [58, 179, 166]. a performance evaluation of color descriptors has been performed by Van de Sande et al. [166]. Among several *ColorSIFT* descriptors evaluated in their study, *OpponentSIFT* is shown to provide superior performance for object recognition task. The *OpponentSIFT* is based on the opponent color space as:

$$\begin{pmatrix} O1 \\ O2 \\ O3 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & -\frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{pmatrix} \times \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

The O3 channel describes the intensity information, whereas the color information is represented in O1 and O2. In *OpponentSIFT*, SIFT is computed on the three opponent channels respectively. The resulting feature vectors are concatenated into a single representation. It is further shown in [166] that *C-SIFT* performs best on the PASCAL VOC 2007 dataset. The *C-SIFT* descriptor is derived from the opponent color space as $\frac{O1}{O3}$, and $\frac{O2}{O3}$. Both *OpponentSIFT* and *C-SIFT* are invariant to light intensity changes.

2.3.2 Shape Information

It has been demonstrated that shape represented by the edges is a very good cue for object recognition [128, 147]. Thus, we would like to give a fast review of some of the techniques used to detect and describe the shape information.

Basic techniques for edge detection include the Sobel/Prewitt edge detector [65]. Canny [26] developed an improved approach to find edges. Since then various others made slight improvements in various directions [37, 146]. Recently Martin et al. [110] have proposed the Berkeley natural boundary detector which has reported excellent results.

Several methods to describe and compare edges have been proposed. One popular method is Chamfer Matching which was introduced by Barrow et al. [9] and extended to hierarchical matching by Borgefors [17]. The algorithm matched edges by minimizing a generalized distance between them. The matching is performed in a series of images depicting the same scene, but in different resolution (e.g. in a resolution pyramid). Olson and Huttenlocher [127] used the Hausdorff-Distance to compare edges. They also showed how such a technique improves if edge orientation is taken into account. Another method for describing edges is the shape context descriptor of Belongie et al. [10]. There each point on an edge is characterized by the histogram of the log-polar coordinates (related to that specific point) of all other points (in a certain radius). Rothwell et al. [138] presented a method where edges are described between bitangent points. This results in a projectively invariant description for near planar curves. This method is an extension of the affine invariant representation of Lamdan et al. [88].

One of the first approaches for detecting objects of a category which was useful for real world images and based on shape was introduced by Gavrila and Philomin [55]. Sets of significant contour examples were used and then the whole contours were applied to test images. Berg et al. [11] perform generic object recognition on the basis of deformable shape matching using a new correspondence finding algorithm. Their algorithm is formulated as an integer quadratic program, where the cost function is a combination of geometric blur descriptors and geometric distortion between feature points. The recognition procedure is incorporated in a nearest neighbor framework. Dalal and Triggs [34] use shape information in the form of grids of Histograms of Oriented Gradients *(HOG)*. Studying influences of the binning of scale, orientation, and position they yield excellent categorization by a SVM-based classifier. Leibe et al. [99] include shape information to detect pedestrians. They use a verification step that uses chamfer matching of a representation of the whole object contour. Recently, Shotton et al. [147] and Opelt et al. [128] presented a method based on local boundary fragments.

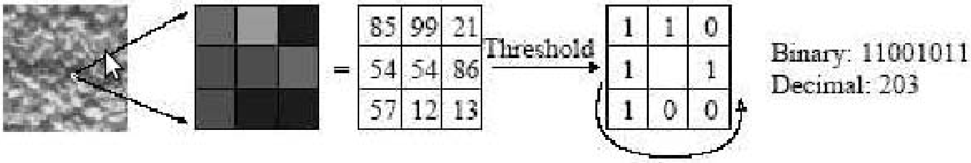
2.3.3 Texture Information

It has been demonstrated that texture is a very good cue for recognition tasks [57, 135, 189]. Thus, we would like to give a fast review of some of the techniques used to detect and describe the texture information. The basic idea of these methods is to compute a texton histogram based on a universal representative texton vocabulary. First, Leung and Malik [100] constructed a 3D texton representation for classifying a “stack” of registered images of a test material with known imaging parameters. The special requirement of calibrated cameras limits the usage of this method in most practical situations. This limitation was removed by the work of Cula and Dana [33], who used single-image histograms of $2D$ textons. Varma and Zisserman [172, 173] have further improved 2D texton-based representations, achieving very high levels of accuracy on the Columbia-Utrecht reflectance and texture (CUReT) dataset [35]. The descriptors used in their work are filter bank outputs [172] and raw pixel values [173]. Hayman et al. [69] extend this method by using support vector machine classifiers with a kernel based on χ^2 histogram distance.

Recently, Local Binary Patterns *(LBP)* has been shown to be one of the best performing texture descriptors. It has been widely used in various computer vision applications, such as face and object recognition, yielding in outstanding results [3, 72, 57, 71]. It has been proven to be highly discriminative and its key advantages, namely, its invariance to monotonic gray-scale changes, it has also shown to discriminate a large range of rotated textures efficiently. Moreover its computational simplicity and efficiency as the operator can be realized with a few operations in a small neighborhood and a lookup table, make it suitable for demanding image analysis tasks. Thus, we would like to give a fast review of some of the techniques used to detect and describe LBPs.

The LBP operator was originally designed for texture description. The LBP descriptor and its variants use short binary strings to encode properties of the local micro-texture around each pixel. The simplest form of LBP is created at a particular pixel location by threshold the 3×3 neighborhood surrounding the pixel with the central pixel's intensity value, and treating the subsequent pattern of 8 bits as a binary number, as shown in fig. 2.7. A histogram of these binary numbers in a predefined region is then used to encode the texture of that region. To be able to deal with textures at different scales, the LBP operator was later extended to use neighborhood of different sizes [123].

Recently, several methods have been proposed [72, 182] to describe and compare texture features based on patch-based LBP descriptors. These descriptors are variants of LBP, which are able to improve the performance of the pixel-based LBP descriptor, which are based on patch statistics. One approach is the Center-Symmetric Local Binary Patterns *(CSLBP)*,

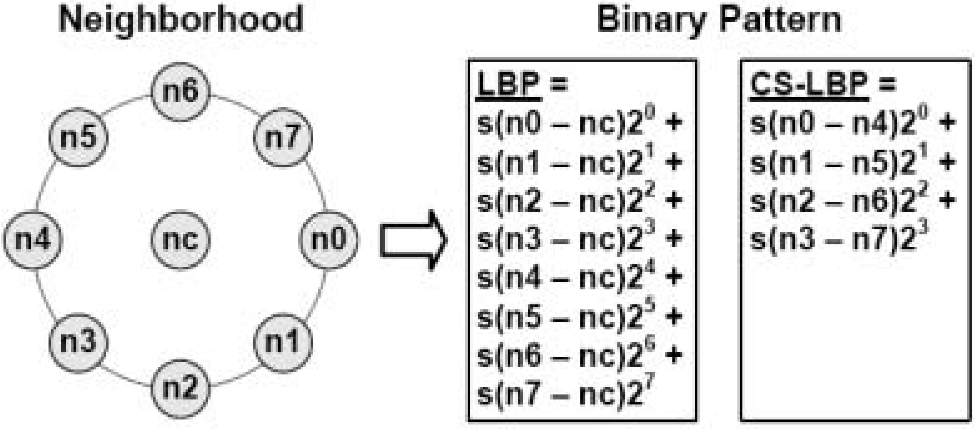


85	99	21
54	54	86
57	12	13

1	1	0
1		1
1	0	0

Figure 2.7: The Basic LBP Operator: The LBP image-texture descriptor is computed locally at each pixel location. It considers a small neighborhood of a pixel, and thresholds all values by the central pixel's value. The bits which represent the comparison results are then transformed into a binary number. The histogram of these numbers is used as a signature describing the texture of the image.

which was introduced by Heikkilaa et al. [72]. The technique encodes at each pixel the gradient signs at the pixel at four different angles; eight intensities around a central point are measured. The binary vector encoding the local appearance at the central point consists of four bits which contain the comparison of intensities to intensities on the symmetric position. Fig. 2.8 shows an example of the *CS-LBP* features.

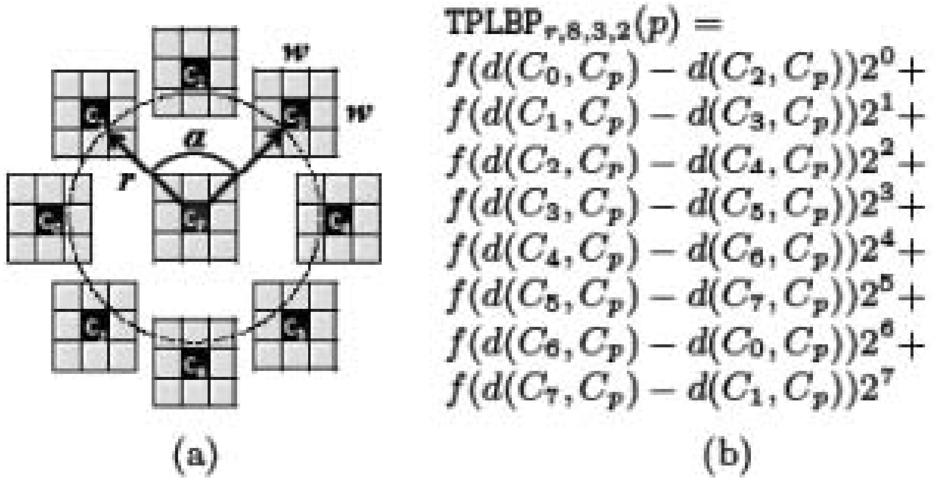


LBP =
$s(n0 - nc)2^0 +$
$s(n1 - nc)2^1 +$
$s(n2 - nc)2^2 +$
$s(n3 - nc)2^3 +$
$s(n4 - nc)2^4 +$
$s(n5 - nc)2^5 +$
$s(n6 - nc)2^6 +$
$s(n7 - nc)2^7$

CS-LBP =
$s(n0 - n4)2^0 +$
$s(n1 - n5)2^1 +$
$s(n2 - n6)2^2 +$
$s(n3 - n7)2^3$

Figure 2.8: LBP and CS-LBP Features for a Neighborhood of 8 Pixels. CSLBP encodes in each pixel the signs at the pixel at four different angles; eight intensities around a central point are measured. The binary vector encoding the local appearance at the central point consists of four bits, which contain the comparison between intensities on the symmetric position.

Another approach for describing patch-based LBP is the family of related descriptors, namely, Three-Patch LBPs (TPLBP) and Four-Patch LBP (FPLBP) proposed by Wolf et al. in [182]. There each descriptor is designed to encode additional types of local texture information by using different ways of bit strings to encode the similarities between neighboring patches of pixels. Hence, capturing information which is complementary to pixel based ones. For each pixel in the image, a $w \times w$ patch centered on the pixel, and S additional patches distributed uniformly in a ring of radius r around it is considered. Concretely, pairs of patches, which are alpha patches apart along the circle are taken, and then their values are compared with those of the central patch. The value of a single bit is set according to which of the two patches is more similar to the central patch. The resulting code has S bits per pixel, as shown in fig. 2.9.



$$
\begin{aligned}
\mathrm{TPLBP}_{r,8,3,2}(p) = {} & f(d(C_0,C_p) - d(C_2,C_p))2^0 + \\
& f(d(C_1,C_p) - d(C_3,C_p))2^1 + \\
& f(d(C_2,C_p) - d(C_4,C_p))2^2 + \\
& f(d(C_3,C_p) - d(C_5,C_p))2^3 + \\
& f(d(C_4,C_p) - d(C_6,C_p))2^4 + \\
& f(d(C_5,C_p) - d(C_7,C_p))2^5 + \\
& f(d(C_6,C_p) - d(C_0,C_p))2^6 + \\
& f(d(C_7,C_p) - d(C_1,C_p))2^7
\end{aligned}
$$

Figure 2.9: (a) The Three-Patch LBP (TPLBP) Code with alpha= 2 and S = 8. (b) TPLBP Code Computed with Parameters S = 8, w = 3, and alpha= 2. For each pixel in the image, a $w \times w$ patch centered on the pixel, and S additional patches distributed uniformly in a ring of radius r around it is considered.

2.4 Modeling compact BoW-based image representations

Feature extraction is followed by visual vocabulary construction stage within the bag-of-words framework. Typically, a visual vocabulary is constructed using K-means algorithm. The algorithm is a simple iterative approach where the number of clusters (vocabulary size) are predefined. Initially the cluster centers (visual words) are initialized by randomly selecting descriptor points. The distance is calculated for each sample point to the cluster centers (visual words), and the point is assigned to the cluster having the closest center (visual word). After assigning all the points, the cluster centers are updated by averaging all the points in a cluster. The procedure is repeated for a fixed amount of iterations. After constructing the visual vocabulary, each descriptor is assigned to a single visual-word in the codebook. Consequently, a histogram is constructed by counting the number of occurrences of each visual-word in an image.

The quality of the visual word depends on the size of the vocabulary. Generally, improved results are obtained using larger visual vocabularies. Although larger visual vocabulary improve the performance, yet this improvement comes at the cost of high dimensionality, and consequently high classification time. Towards this direction, several approaches in the literature [47, 149, 19] have addressed the problem of obtaining compact image representation.

One approach is to use some Bayesian models used for text document classification, such as Latent Dirichlet Analysis *(LDA)* and probabilistic Latent Semantic Analysis *(pLSA)*, which work over the bag-of-words model and have been adapted and used to model image categories. These models are generative models from the statistical text literature [76]. In text analysis they are used to discover topics in a document using the bag-of-words document representation. In this case, there are "images" as "documents" and they discover "topics" as "object categories" (e.g. grass, houses, bikes, planes), so that an image containing instances of several objects is modeled as a mixture of topics. Quelhas et al. [136] provided an approach to model visual scenes in image collections, based on local invariant features and pLSA. pLSA was also used in [149, 19] for scene and object recognition, see fig. 2.10.

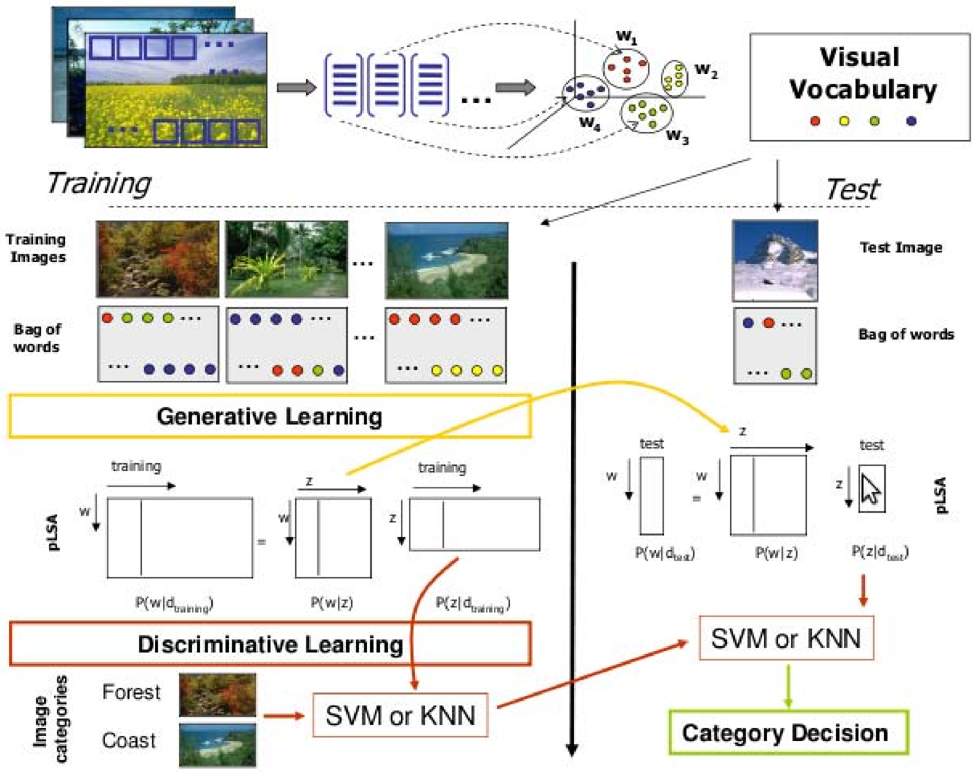


Figure 2.10: Overview of visual vocabulary formation, learning and classification stages.

Li and Perona [47] independently proposed two variations of LDA firstly proposed by Blei et al. [15, 156] which was designed to represent and learn document models. In this framework, local regions are first clustered into different intermediate themes, and then into categories. Probability distributions of the local regions as well as the intermediate themes are both learned in an automatic way, bypassing any human annotation. No supervision is needed apart from a single category label to the training image.

More recently, many works have addressed the problem of modeling compact image representation by obtaining compact vocabulary construction [181, 54, 94] based on the information-theoretic clustering techniques [152, 38]. Slonim and Tishby [152] proposed a compression technique, denoted as Agglomerative Information Bottleneck (AIB), that constructs small and informative dictionaries by compressing larger vocabularies following the information bottleneck principle. Winn et al. [181] do discriminative compression in a similar fashion. Similarly, Fulkerson et al. [54] proposed a fast implementation of the AIB algorithm and showed good performance for the construction of visual vocabularies.

Agarwal et al. [2] cluster features to create a whole image descriptor called a "hyper-feature" stack, which repeatedly quantizes the data in fixed pyramid. Leibe et al. [98] also perform compression, but not in a discriminative sense. Liu et al. [103] proposed recently a co-clustering scheme maximizing mutual information (MMI) for scene recognition. Lazebnik et al. [151, 94] also perform discriminative learning to optimize k-means, but are limited to small dictionaries and visual words which are Voronoi cells.

Last but not least, Dhillon et al. [38] proposed the usage of the Divisive Information Theoretic Clustering (DITC) technique for text categorization. DITC, is an unsupervised probabilistic model for the collection of discrete data, has the dual ability to generate low-dimensional image representation, and to automatically capture meaningful image aspects. DITC immediately clusters the words into the desired number of clusters (during initialization) after which it iteratively improves the quality of these clusters as shown in fig 2.11. This approaches has recently shown a significant success for obtaining compact visual vocabularies, which is capable of maintaining the discriminative power of the original vocabulary.

An important advantage for the compression of large visual vocabularies, is that it allows us to incorporate more features, which helps in improving the classification performance as we will further investigate in this thesis.

2.5 Combining Multiple Features

Generally, the local description is performed by extracting low-level texture or shape or etc. features in an image. There exist two main approaches to incorporate multiple features within the bag-of-words framework. The first approach, "early fusion", combines features at the local features level. This combination at the feature level results in constructing a joint visual vocabulary. A weight vector β is introduced to tune the relative weight between the different features within the combined vocabulary (V_{ts}).

$$V_{ts} = (\beta V_t, (1 - \beta) V_s) \tag{2.1}$$

where, V_t are the texture features and V_s are the shape features. The weight vector β is learned through cross-validation on the training data.

The second approach, "late fusion", fuses multiple features at the histogram level by concatenating the features histograms obtained independently. Here the different vocabularies

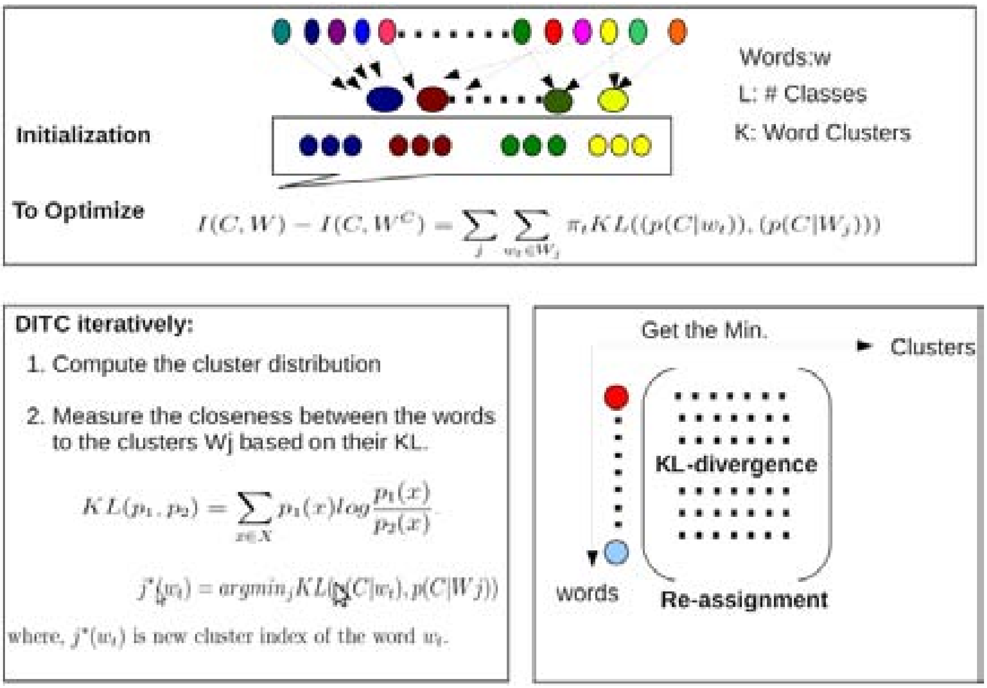

Words:w
L: # Classes
K: Word Clusters
Initialization
To Optimize
$I(C,W) - I(C,W^C) = \sum_j \sum_{w_t \in W_j} \pi_t KL((p(C|w_t)), (p(C|W_j)))$
DITC iteratively:
1. Compute the cluster distribution
2. Measure the closeness between the words to the clusters Wj based on their KL.
$KL(p_1, p_2) = \sum_{x \in X} p_1(x) log \frac{p_1(x)}{p_2(x)}$
$j^*(w_t) = argmin_j KL(p(C|w_t), p(C|Wj))$
where, $j^*(w_t)$ is new cluster index of the word w_t.
Get the Min.
Clusters
KL-divergence
words
Re-assignment

Figure 2.11: Overview of visual vocabulary formation, DITC-based initialization and learning stages.

are concatenated after quantization. A weight vector α is introduced to obtain a combined histogram $\mathbf{F}(\omega|\mathrm{I})$ of multiple vocabularies for an image *I*.

$$\mathbf{F}(\omega_{t\&s}|I) = \begin{bmatrix} \alpha\mathbf{F}(\omega_t|I) \\ (1-\alpha)\mathbf{F}(\omega_s|I) \end{bmatrix} \tag{2.2}$$

where ω is the number of the total vocabulary words, ω_t are texture words, and ω_s are shape words. The weight vector α is learned through cross-validation on training data.

The two approaches, early and late fusion, have their own advantages and drawbacks as well. Early fusion provides a more discriminative visual vocabulary, since the texture and shape words are constructed by quantizing the local texture and shape cues combined at the feature level. This helps in recognizing object categories having consistent texture and shape features, which is commonly the case in many natural categories like plants and lions. On the other hand, late fusion provides a more compact representation of both texture and shape as separate visual vocabularies are constructed for individual cues. This is especially important for man made categories such as cars and chairs which vary considerably in texture.

2.6 Spatial Pyramid-based Image Representation

The spatial pyramid scheme proposed by [97] is a simple and computationally efficient extension of an order-less bag-of-words image representation. This approach represents an image by using weighted multi-resolution histograms which are obtained by repeatedly subdividing an image into increasingly finer sub-regions. Histograms are computed over the resulting sub-regions. For each resolution level, the image is subdivided into the cells of a grid. At resolution l, the grid has 2^{2l} cells. The number of points in each grid cell is then recorded.

Matches within each grid cell are then determined. A match is found when two points are in the same grid. Matches found at finer resolutions are closer to each other in the image space and are therefore more heavily weighted. To accomplish this, each level is weighted to $1/2^{L-l}$. The matches found at resolution level $l+1$ are included in the matches found at the larger resolution l. Therefore, matches at the finest level were first computed. After this, matches at the coarser levels were determined by summing the matches found within each cell for level $l+1$ that is contained within the corresponding cell at level l. For each cell, a histogram of the matches is created. When histograms for all cells at all levels have been created, these histograms are concatenated to form the final image representation.

Marszalek et al. [107] evaluate both regular and irregular grids. Further, they consider a broader set of coarse subdivisions for each dimension, such as a 1×1 grid corresponding to the standard representation of the bag-of-words, a 2×2 grid (i.e. four blocks), a horizontal 3×1 grid as well as a vertical 1×3 one. They show that dividing the image plane in three horizontal (i.e. 3×1 grid) regions, provides the highest recognition performance. Further, this approach reduces the dimensionality of the conventional 4×4 (i.e. sixteen blocks) structure; from $vocabularysize \times 21$ to $vocabularysize \times 8$.

2.7 Color Constancy Review

The ability to correct for different illumination colors is called color constancy. An example is given in fig. 2.12, where the same cube is rendered under two different light sources. When seen in reality, the two patches indicated by the green circle are generally termed red patches by human observers, while the reflected colors in fact are very different. Similarly, the patches indicated by the red circle are generally termed yellow and blue patches, respectively, while in fact the color is exactly the same.

The ability of color constancy allows for the interpretation of colors within the context of their surroundings. The human visual system is equipped with this ability, to some extent, but computer vision systems often perform quite poor. On digital cameras, this ability is usually implemented as white balancing in such a simplistic way that results often are not satisfactory. This can cause images to appear very different from the actual scenes, e.g. an unnatural yellowish/orange or blue cast to all colors in the scene. Consequently, extensive calibration of the camera or manual post-processing is necessary to make the output realistic and appealing. Other computer vision applications, like object and scene recognition, image

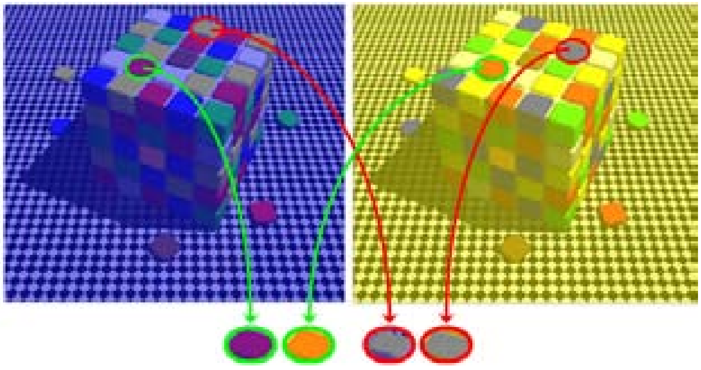


Figure 2.12: Image adapted from [134], showing the same colored cube rendered under two different light sources. The color of the patches indicated with the green circle is usually termed red, while the actual color of the two patches differs greatly. On the other hand, the colors of the patches indicated with the red circle are generally termed yellow and blue, while the color of these two patches is exactly the same.

and video retrieval and object tracking, could also benefit from accurate color constancy, to increase the robustness of color features.

2.7.1 Color Constancy

From a computational point of view, color constancy is defined as the transformation of an input image, taken under an unknown light source, so that it appears to be taken under a known canonical, often white, light source. The process is illustrated in figure 1.2. First, images are gathered that are taken under an unknown light source. Then, the color of the light source is estimated for each input image. These estimates are used to transform the input image, resulting in an output image. These output images depict the same scene as the input image, but now appear to be taken under a known (white) light source.

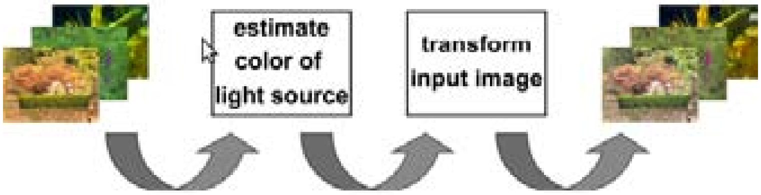


Figure 2.13: Overview of the computational approach to color constancy. First, the illuminants of the input images are estimated. Then, using this estimate, the input images are transformed, constructing output images that contain the same scene as the input image, but now appear to be taken under a white light source.

2.7.2 Illuminant Estimation

Images are composed of a combination of object reflectance and light source properties. This causes the problem of illuminant estimation to be ill-posed, i.e. the color of the light source cannot be retrieved unambiguously given only the colors in images, and consequently all algorithms make use of simplifying assumptions. One of the most well-known and often used assumptions is the Grey-World assumption [21]: the assumption that the average reflectance in a scene, under a white light source, is achromatic. Another well-known algorithm is based on the White-Patch assumption, i.e. the assumption that the maximum response in the RGB-channels is caused by a perfect reflectance [91]. Other methods that are based on simple statistics of images include the Shades-of-Grey algorithm [52] and Local Space Averaging algorithm [41]. Rather than only using statistics of pixel values for estimating the illuminant, more complex methods are developed that use information acquired in a learning phase. Possible light sources, distributions of possible reflectance colors and prior probabilities on the combination of colors are learned and used for estimating the color of the light source. One of the first algorithms of this type is the gamut mapping algorithm by Forsyth [53]. This algorithm is based on the assumption that in real-world images, for a given illuminant, only a limited number of colors can be observed. Using this assumption, the illuminant can be estimated by comparing the distribution of colors in the current image to a pre-learned distribution of colors (called the canonical gamut). Many algorithms have been derived from the original algorithm including Color-by-Correlation [50] and the Gamut constrained illuminant estimation [51]. However, none of these extensions consider the extension to higher-order statistics, discarding much information that is present in images. Gijsenij et al.[62], propose recently the use of natural image statistics to identify the most important characteristics of color images. Then, based on these image characteristics, the proper color constancy algorithm is selected for a specific image. To capture the image characteristics, the Weibull parameterization (e.g. grain size and contrast) is used. It is shown that the Weibull parameterization is related to the image attributes to which the used color constancy methods are sensitive to. A MoG-classifier is used to learn the correlation and weighting between the Weibull-parameters and the image attributes (number of edges, amount of texture and SNR). The output of the classifier is the selection of the best performing color constancy method for a certain image.

2.8 Conclusion

Observing the remarkable performance of spatial pyramids, their growing number of applications to a broad range of vision problems, and finally its geometry inclusion, a question can be asked what are the limits of spatial pyramids. In summary, the main concern of this thesis is to search for the limits of spatial pyramids and try to figure out solutions for them.

In the second part of this thesis, we propose the usage of spatial knowledge (i.e., 3D scene geometries) to determine which color constancy method to use for the different geometrical regions found in images using the natural image statistics. To this end, images are first classified into rough 3D geometry models called stages. According to the stage models, images are divided into different regions using hard or soft segmentation. After that, the best color constancy algorithm is selected for each geometry segment. As a result, light source estima-

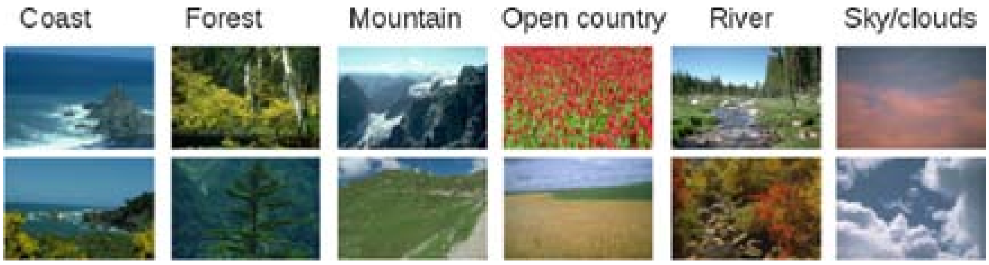


Figure 2.14: Some images from VS data set.

tion is tuned to the global scene geometry. Nevertheless, the question is how to select the method that performs best for a specific image region. Furthermore, the next question is how to combine the algorithms in a proper way. These issues are addressed in chapter 6.

2.9 Data sets

We will evaluate our classification algorithms on different data sets recently used in the literature. We can divide theses data sets in two groups: (i) Oliva and Torralba [125], Vogel and Schiele [176], Sports Events [101], Lazebnik et al. [97] and Quattoni and Torralba [135] are the data sets containing natural scene images; while (ii) Butterflies, Pascal2007 [43] and Caltech 101 [45] are the data sets with objects images. In this chapter we will describe these data sets in more detail.

2.9.1 Scene classification

Vogel & Schiele (VS) data set

Vogel and Schiele [176] data set (called as VS in this work): includes 702 natural scenes consisting of 6 categories: 144 coasts, 103 forests, 179 mountains, 131 open country, 111 river and 34 sky/clouds. Figure 2.14 shows some images from this data set. The size of the images is 720×480 (landscape format) or 480×720 (portrait format). Every scene category is characterised by a high degree of diversity and potential ambiguities since it depends strongly on the subjective perception of the viewer. For example river and forest are considered as two kind of different scenes. However most of the river images also contain a forest and can easily be confused. There is a low inter-class variability and a high intra-class variability making the scene classification problem a bit more difficult when working with this data set.

Oliva & Torralba (OT) data set

Oliva and Torralba [125] data set (called as OT in this work) includes 2688 images classified as 8 categories: 360 coasts, 328 forest, 374 mountain, 410 open country, 260 highway, 308 inside of cities, 356 tall buildings and 292 streets. Figure 2.15 shows some images from this data set. Note that now river and forest scenes are all considered as forest, moreover there is not an specific sky scene since almost all of the images contain the sky object. These

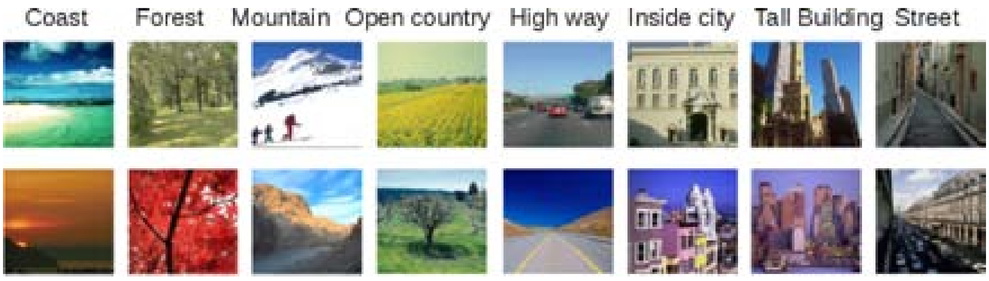


Figure 2.15: Some images from OT data set.

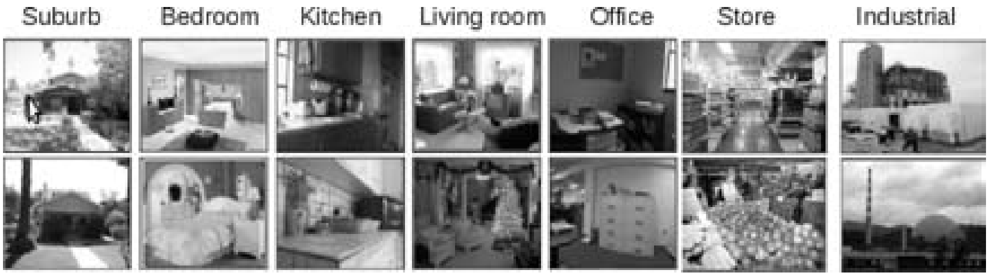


Figure 2.16: Some images from LSP data set.

annotations make a higher inter-class variability. Most of the scenes present a large intra-class variability. The average size of each image is 250 × 250 pixels.

Lazebnik, Schmid & Ponce (LSP) data set

Lazebnik et al. [97] data set (referred to it as LSP in this work) contains 15 categories and is only available in greyscale. This data set consists of the 2688 images (8 categories) of the OT data set plus: 241 suburb residence, 174 bedroom, 151 kitchen, 289 living room, 216 office, 315 store and 311 industrial. Figure 2.16 shows some images from this data set. The average size of each image is approximately 250 × 300 pixels.

Events Sports data set

The Sports Events data set [101] contains 8 sports event categories collected from the Internet namely: bocce, croquet, polo, rowing, snowboarding, badminton, sailing, and rock climbing. The number of images in each category varies from 137 (bocce) to 250 (rowing). For each event class, 70 randomly selected images are used for training and 60 are chosen for testing. Figure 2.17 shows some images from this data set.

Quattoni and Torralba (QuT) data set

The QuT data set [135] is a recent indoor scene data set. It is characterized by 67 indoor categories with high intra-class variations, since the classification of indoor scenes are very challenging.

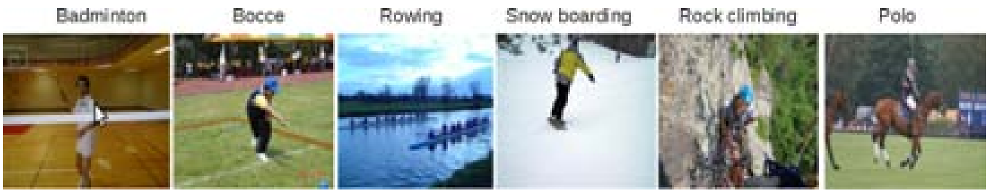
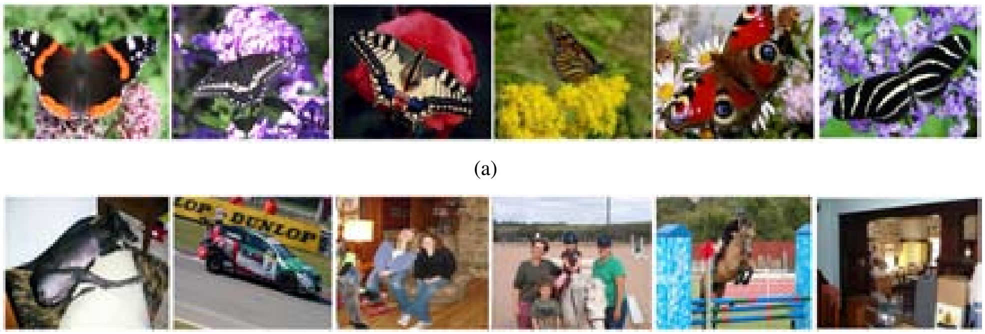


Figure 2.17: Some images from Sports Events data set.


(a)

(b)

Figure 2.18: Some images from: (a) Butterflies and (b) PASCAL VOC data sets.

2.9.2 Object classification

Butterflies data set

The Butterflies data set consists of 619 images of seven classes of butterflies namely: Admiral, Swallowtail, Machaon, Monarch 1, Monarch 2, Peacock and Zebra. Figure 2.18a shows some images from this data set.

Pascal Visual Object Classes (VOC) data set

The experiments are also performed on PASCAL VOC Challenge [43]. Figure 2.18b shows some images from this data set. The PASCAL VOC 2007 data set consists of 9963 images of 20 different classes with 5011 training images and 4952 test images. The Pascal 2009 data set contains 13704 images of 20 different object categories with 7054 training images and 6650 test images.

Caltech-101 data set

The Caltech-101 data set (collected by Fei-Fei et al. [45]) consists of images from 101 object categories. This database contains from 40 to 800 images per category however most categories have about 50 images. Most images are medium resolution, about 300 × 300 pixels. The significance of this database is its large inter-class variability. Moreover most images have little or no clutter, objects tend to be cantered in each image and most objects are presented in a stereotypical pose. So, as was noted by Lazebnik et al. [97], Caltech-101

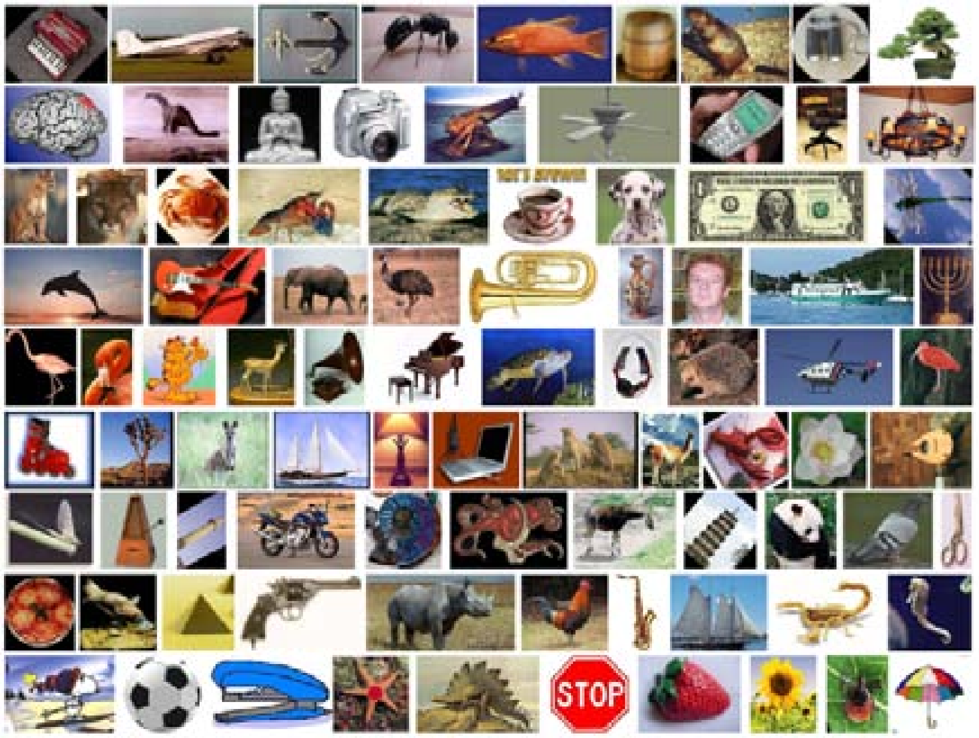


Figure 2.19: Some images corresponding to scenes/objects from Caltech-101 data set.

is essentially a scene classification data set as the objects are well aligned within each class (i.e. rotated, scaled and centred) with little clutter. An image from each category is shown in figure 2.19.

Chapter 3

Compact and Adaptive Spatial Pyramids for Scene Recognition

Most successful approaches on scene recognition tend to efficiently combine global image features with spatial local appearance and shape cues. On the other hand, less attention has been devoted for studying spatial texture features within scenes. Our method is based on the insight that scenes can be seen as a composition of micro-texture patterns. This chapter analyzes the role of texture along with its spatial layout for scene recognition. However, one main drawback of the resulting spatial representation is its huge dimensionality. Hence, we propose a technique that addresses this problem by presenting a compact Spatial Pyramid (SP) representation. The basis of our compact representation, namely, Compact Adaptive Spatial Pyramid (CASP) consists of a two-stages compression strategy. This strategy is based on the Agglomerative Information Bottleneck (AIB) theory for (i) compressing the least informative SP features, and, (ii) automatically learning the most appropriate shape for each category. Our method exceeds the state-of-the-art results on several challenging scene recognition data sets.

3.1 Introduction

Scene recognition is one of the most appealing, yet challenging problems in computer vision. Fig. 3.1 shows such kind of challenges, namely, illumination changes, intra-class variabilities, scale variabilities, and inter-class similarities. The goal is to identify an image as belonging to one of several scene classes such as mountains, beaches, indoor-offices, etc.. Effective solutions to this problem can be useful in many other applications, such as detection [4, 160], action recognition [79], and content based image retrieval [121]. Approaches to scene recognition can be divided into two main categories. First, methods that use low-level features such as color, texture, etc. [162, 155]. Despite the good performance obtained using these approaches, they lack an intermediate image description (such as the presence of the sky, grass, or other semantic concepts) that can be extremely valuable in determining scene types [191, 112, 177]. On the other hand, other techniques make use of intermediate representa-

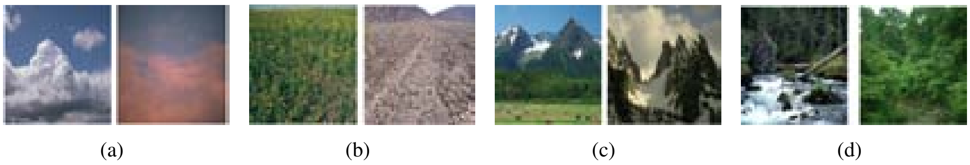


Figure 3.1: Scene recognition challenges are: (a) illumination changes, (b) intra-class variabilities, (c) scale variabilities, and (d) inter-class similarities (in the example, river can be easily confused with forest).

tions [47, 102, 108]. Towards this direction, the Bag-of-Words (BoW) approach has been used to model scenes [20, 185, 131]. However, its foremost shortcoming is the lack of spatial information. Recently, several approaches considered the immense success of SP [97, 1], due to its inclusion of important spatial information. For example, Bosch et. al. [1] showed how spatial appearance features benefit the scene recognition task. Besides, the work in [18, 135], which demonstrated the significance of fusing complementary spatial shape, and appearance features along with global cues. However, much less attention has been devoted to studying the texture features within this context. To this end, we propose three contributions over the standard SP:

- The first contribution is concerned with, the exploration of the spatial texture features along with the shape, and appearance cues for scene recognition. Our novel descriptor is mainly inspired by two sources: (i) the Pyramid of Histograms of Orientation Gradients (PHOG) descriptor [18], and (ii) the Histogram of Three Patch Local Binary Patterns (TPLBP) [182, 189], which has been recently proposed to encode texture data in both static images and videos.
- The second concern is addressing the huge dimensional histograms generated using the standard SP scheme, while going towards the finest level of representation. We address this problem by finding a more compact SP representations that maintain or even improve their original counterparts.
- The third contribution is regarding the rigid SP assumption proposed by Lazebnik et. al. [97], that suits each category. We propose a method for learning the best partition for each category. The resulting SP shapes have the advantages of being compact, while improving the original SP performance.

We denote the resulting representation of combining the last two contributions as *Compact Adaptive Spatial Pyramid* (CASP).

Finally, we propose fusing multiple aspects of complementary information using our *CASP* strategy. This powerful representation helps in overcoming the common scene recognition challenges, shown in Fig. 3.1, and, consequently, improving the scene recognition performance.

Outline This chapter is organized as follows: next section proposes our novel spatial texture descriptor. Section 3.3 briefly explains the basic idea of SPs, and discusses how AIB theory is extended for building our novel CASPs. Section 3.4 describes the datasets used in the experiments. Section 3.5 shows, and compares the experimental results with the state-of-the-art. Finally, section 3.6 presents the main conclusions of this work, and shows the most important avenues of future research.

3.2 Pyramids of Colored Three-Patch Local Binary Patterns

Our first contribution exploits the spatial texture features for the task of scene recognition. We propose using our texture representation that retains both local image texture, and its spatial layout. This novel representation is able to capture the large illumination variabilities illustrated in Fig.3.1(a).

Our descriptor is an extension of the Local Binary Patterns (LBP), which has been shown to be one of the best performing texture descriptors [124, 3, 73, 189]. *LBP* has been successfully used in various applications, such as face recognition [3], background subtraction [74], object recognition [57], interest regions description [73], and action recognition [189]. It also has various properties that favor its usage such as, its tolerance against illumination changes, and it's ability to discriminate a large range of rotated textures efficiently. Moreover, its computational simplicity, and efficiency makes it suitable for the scene recognition task.

The next sections describe our novel spatial texture descriptor: First, we give a brief survey on *LBP*. Subsequently, we examine the incorporation of color [116, 1, 166], and spatial information to our final texture representation.

3.2.1 Local Binary Patterns

The *LBP* descriptor [124], and its variants use short binary strings to encode properties of the local micro-texture around each pixel. The *LBP* simplest form works as follows: For a 3×3 neighborhood, (i) the value of each pixel is compared with the central pixel's intensity value, and (ii) the result from each pixel is then concatenated to form an 8 bits binary descriptor.

Recently, significant works introduce variants of *LBP* descriptors, which are based on patch statistics, namely: Center-Symmetric Local Binary Patterns *(CSLBP)* [73], Three-Patch LBP *(TPLBP)* [182], and Four-Patch LBP *(FPLBP)* [182]. *CSLBP* encodes at each pixel the gradient signs at the pixel at four different angles. *TPLBP* and *FPLBP* encode the similarities between neighboring patches of pixels, thus capturing information which is complementary to pixel-based descriptors.

3.2.2 Pyramids of Colored TPLBP

In this section, we propose to fuse *TPLBP* with complementary color information. We refer to it as *Colored TPLBP* (C-TPLBP). To compute *C-TPLBP*, we extract the *TPLBP* descriptor for each channel of the examined color spaces. Consequently, a dictionary is created for each color-texture channel. The generated histograms of each color-texture channel are then

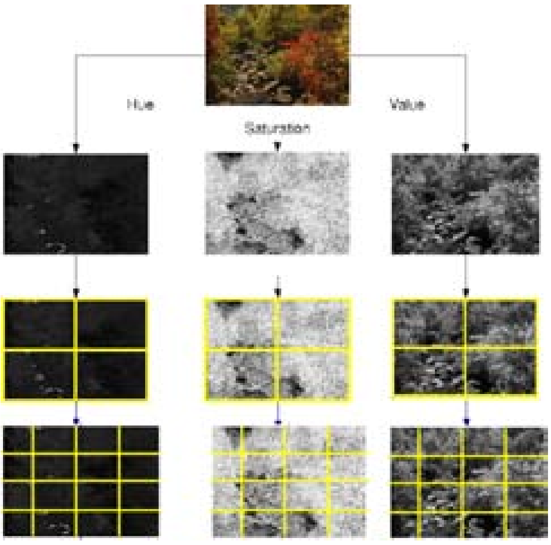


Figure 3.2: Given the dictionaries built for each colored-texture channel, they are concatenated using a $(1 \times 1) + (2 \times 2) + (4 \times 4) = 21$ image representation. This representation is denoted to as *PC-TPLBP* resulting in a 21 × $(vocsize)$ dimensional histogram.

concatenated. This results in a $(vocabularysize \times 3)$ dimensional histogram for the standard *BoW* representation.

In particular, we examine two standard color spaces, namely, Opponent Color (*(OppC)* [166, 168]) and *HSV*. *OppC* is defined as:

$$O1 = \frac{R-G}{\sqrt{2}}, O2 = \frac{R+G-2B}{\sqrt{6}}, O3 = \frac{R+G+B}{\sqrt{3}} \tag{3.1}$$

So the fusion of *OppC* with *TPLBP* is done as follows[1]:

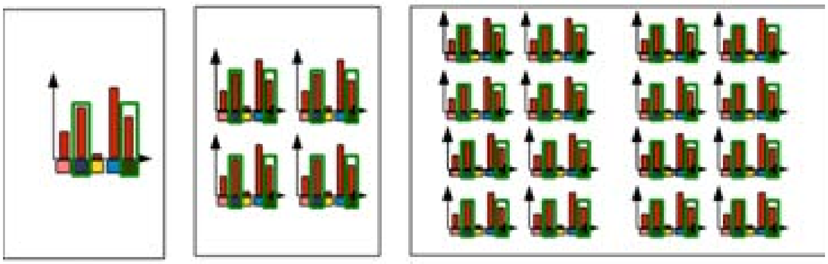
$$\text{TPLBP(OppC)} = \text{TPLBP(O1)} + \text{TPLBP(O2)} + \text{TPLBP(O3)}. \tag{3.2}$$

Similarly, fusing *TPLBP* with *HSV* is done as:

$$\text{TPLBP(HSV)} = \text{TPLBP(H)} + \text{TPLBP(S)} + \text{TPLBP(V)}. \tag{3.3}$$

Finally, in order to include spatial information, we follow the scheme proposed by [97], as shown in Fig. 3.2. The resulting descriptor is referred to as *Pyramids of C-TPLBP* (PC-TPLBP). However, the main drawback of using this texture- and colored-based SP, is its high dimensionality. So in the following section we will introduce our *CASP* approach for reducing SP the dimensionality of the final histogram, while preserving its original performance.

[1] The $+$ operator indicates that the histograms of each colored-texture channel are concatenated.



(a) L0 (b) L1 (c) L2

Figure 3.3: Example of SP high dimensionality problem.

3.3 Compact and Adaptive Spatial Pyramids

SP proposed by [97] is a simple, and computationally efficient extension of the order-less *BoW*. This technique works by representing an image using weighted multi-resolution histograms. These histograms are obtained by repeatedly sub-dividing the image into increasingly finer sub-regions by doubling the number of divisions on each axis direction and computing histograms of features over the resulting sub-regions.

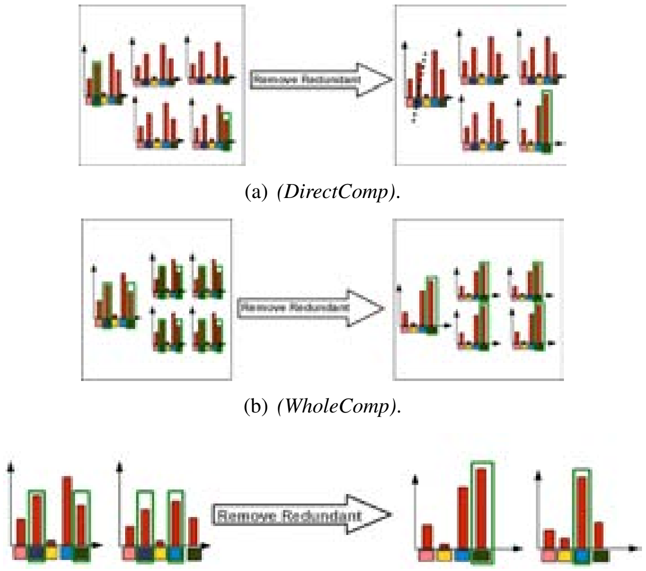
Matches within each sub-region are then determined. Matches found at finer resolutions are closer to each other in the image space, and are therefore more heavily weighted. For each sub-region, a histogram of the matches is created. When histograms for all regions at all levels are created, they are concatenated to form the final image representation. Fig. 3.3, shows an example of a three level SP. This results in a $21 \times (vocabularysize)$ dimensional histogram. This clarifies the SP high dimensional problem, and hence, the huge memory usage during the classification stage.

In our work, we aim at tailoring these high dimensional *SPs* histograms to discriminate between different categories. Concerning this objective, several works address the problem of compact vocabulary construction [181, 54, 94]. In particular, the agglomerative information bottleneck *(AIB)* algorithm [54, 152] provides a guideline for the compression of vocabularies. The common strategy starts with a large vocabulary, and subsequently merges these words together while maintaining the discriminative power of the original vocabulary.

In the next section, we give a brief explanation about the *AIB* theory. We then explain our two-stages SP compression technique, namely, *feature*, and *block* compression, respectively.

3.3.1 Agglomerative Information Bottleneck Theory

The main goal of *AIB* is reducing the dictionary of visual words X required for representing the categories Y. Using terms from information theory, this means generating a compact set of words $\hat{X}$ from the original dictionary X so that the loss of mutual information:



(a) *(DirectComp)*.

(b) *(WholeComp)*.

(c) *(LevelComp)*.

Figure 3.4: Two highlighted words are the most similar ones to be merged (see text).

$$I(X;Y) = D_{KL}\left[p(x,y)\|p(x)p(y)\right] \tag{3.4}$$

to the categories Y is minimal [152, 54]. The functional $D_{KL}\left[p\|q\right]$ is the *Kullback-Leiber* divergence, and the joint distribution $p(x,y)$ is estimated from the training set by counting the number of occurrences of each visual word in each category.

The information about x captured by y can be measured by the mutual information,

$$I(X,Y) = \sum_i \sum_t p(x_i, y_t) log \frac{p(x_i, y_t)}{p(x_i)p(y_t)}, \tag{3.5}$$

which measures the amount of information (discriminative power) I that one random variable carries about the other.

The merging of visual words is achieved by applying the *AIB* method [152]. In essence, *AIB* is applied by iteratively merging the two visual words x_i and x_j into $\hat{x}$ that causes the smallest decrease $I(\hat{X};Y)$ in the original mutual information $I(X;Y)$:

$$\delta I_y(X;\hat{X}) = I(X;Y) - I(\hat{X};Y). \tag{3.6}$$

At each step *AIB* performs the best possible merge $argmin_{\hat{X}} = \delta I_y(X;\hat{X})$.

3.3.2 Feature Compression

The usage of non-optimized dictionaries for building *SP* results in its huge dimensional histograms. Our first aim is to optimize the dictionaries in a way that maintains the original *SP* performance. For this purpose, we investigate the direct usage of the original *AIB* algorithm as proposed in [54] for the task of SP compression. We refer to this scheme as *(DirectComp)*.

In *DirectComp* strategy, two features in any SP region can be suggested to be fused. Since, these candidate features can be located at different SP regions, then different vocabularies for each SP region can be obtained, as shown in Fig. 3.4(a). Hence, discarding the important spatial property of *SP*.

Consequently, we propose two alternative spatial feature compression strategies, which we refer to them as *WholeComp* and *LevelComp*. In *WholeComp* strategy, we propose to remove the spatially least informative features within the whole *SP* levels simultaneously. Hence, the occurrence of a spatial word (x_{sp}) all over the *SP* levels $l0$, $l1$, and $l2$ is measured as:

$$p(x_{sp}) = \sum_{l=0}^{L} \sum_{i=1}^{2^{2l}} p(x_i), \tag{3.7}$$

$$\tag{3.8}$$

where, L is the number of SP levels. Fig. 3.4(b) shows an example of applying our *WholeComp* compression scheme on a two-level *SP*: where the original 5 features per region are reduced to 4 features for all the levels of the pyramid.

on the other hand, in *LevelComp* strategy, we propose to eliminate the spatially least informative features within each *specific* SP level l, as follows:

$$p(x_{sp}) = \sum_{i=1}^{2^{2l}} p(x_i) \tag{3.9}$$

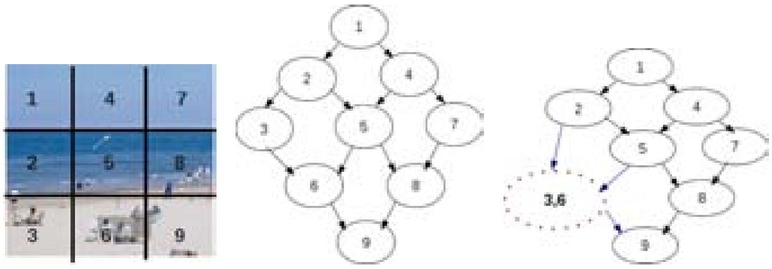
Then, the information content of each x_{sp} word w.r.t. each category y_t is computed as:

$$I(X,Y) = \sum_{i} \sum_{t} p(x_{sp_i}, y_t) log \frac{p(x_{sp_i}, y_t)}{p(x_{sp_i}) p(y_t)}, \tag{3.10}$$

Lastly, for *LevelComp* strategy, we propose to learn the most compact vocabulary $\hat{x}_{sp}$ that best suits each level, see Fig. 3.4(c).

3.3.3 Block Compression

Our last contribution, is concerned with the fact that the hierarchical approach proposed by [97] with a set of regular grids of increasing density is rather inappropriate for scenes. Be-



(a) (b) (c)

Figure 3.5: Block Compression Example. (a) Given an input (3 × 3) block image. (b) Each block is represented by a node in a decision tree. (c) We calculate the discriminative power of each possible merging. See text for details.

sides, this approach enforces that the number of bins in each dimension grows exponentially, which significantly reduces the number of useful grid configurations.

In this section, we propose a method for learning a proper spatial split-up that best suits each category. Hence, we adopt the original AIB such that, we relate each block (b_k) in level (l) with the data set categories. The probabilities $p(b_k)$ of each block (b_k) is formualted as:

$$p(b_k) = \sum_{k=1}^{2^{2l}} \sum_{v=(k-1)s+1}^{ks} p(\hat{x}_{sp_v}), \tag{3.11}$$

where s is the size of the compact vocabulary, k indicates the current block index within level l and v refers to the current vocabulary index within block k. In essence, $p(b_k)$ is calculated by summing up the probabilities of the compact vocabularies they contain. In order to fuse the least informative blocks, we evaluate the discriminative power of each block:

$$I(X,Y) = \sum_k \sum_t p(b_k, y_t) log \frac{p(b_k, y_t)}{p(b_k)p(y_t)}, \tag{3.12}$$

Fig. 3.5 visualizes our block fusion approach. The Decision Tree *(DT)* shown in Fig. 3.5(b) represents our input image in Fig. 3.5(a). Each node in *DT* is equivalent to an image block, while arrows indicate neighboring blocks that are candidates for fusion. Each block can be merged with either its right or its bottom neighboring block (if any). Initially, all possible block fusions indicated by arrows are considered for fusion. However, the actual fusion occurred is between b_3, and b_6 as it caused the minimum loss in discriminative power, see Fig. 3.5(c). As a result, b_3 is updated to $b_{(3,6)}$, and the neighbors of both b_3 and b_6 are inherited (i.e. b_2, b_5, b_9). Thus, a dimensionality reduction is achieved by removing b_6 from *DT*.

This iterative procedure results in generating adaptive shapes, and it terminates when all blocks are merged. Hence, it converges when it reaches the standard *BoW* representation.

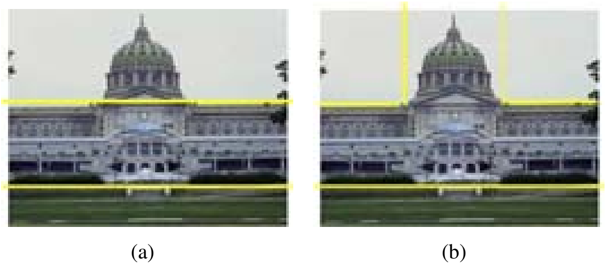


Figure 3.6: (a) Our Global Compact, Adaptive Spatial Pyramid (CASP) learning scheme converges to the Ad-Hoc SP proposed in [107], vs. (b) Our Class-Specific (CASP).

We also considered two methods for optimally learning adaptive shapes, namely *Global Pyramid Shapes (GPS)* and *Class-specific Pyramid Shapes (CPS)*:

- In *GPS*: Instead of having a fixed rigid shape for representing any task at hand as in [97], we learn those shapes which give the best classification performance over all the categories.
- In *CPS*: Instead of learning the shapes across all classes, we learn those shapes for each class separately by optimizing classification performance for that class vs. the rest.

Fig. 3.6 shows an example of learning the most suitable shape using the proposed approaches. Fig. 3.6(a), shows the globally learned shape that suits all the categories. On the other hand, Fig. 3.6(b) shows the learned shape specifically tailored for the *inside city* category. This shows the importance of developing specific image representations for the task at hand.

3.4 Experimental Setup

The *CASP* approach has the following key advantages: (i) it generates a variety of efficient SP shapes that outperform the original SP, and (ii) it is fast and scalable, which makes it possible to operate on much larger dictionaries and data sets. These merits are next evaluated on three standard scene recognition data sets:

- Vogel and Schiele (**VS**) data set [177] includes 7 natural scenes consisting of 6 categories: 142 coasts, 111 rivers/lakes, 103 forests, 131 plains, 179 mountains, and 34 sky/clouds. Every scene category is characterized by a high degree of diversity, and potential ambiguities.
- Oliva and Torralba (**OT**) data set [126] includes 2688 images classified as 8 categories: 360 coats, 328 forest, 374 mountain, 410 open country, 260 high way, 308 inside of cities, 356 tall buildings, 292 streets.

- Quattoni and Torralba (**QuT**) indoor scene data set [135] is a recent data set characterized by 67 indoor categories with high intra-class variations, since the classification of indoor scenes are very challenging.

3.4.1 Implementation Details

In this section, we briefly discuss the implementation details. For the purpose of classification, we use a multiple-scale grid detector. In the feature extraction step, we use various state of the art features. We use GIST descriptor [126] to represent the scene semantics. We use two standard color spaces, namely, HSV, and opponent *(OppC)* [166, 168] for obtaining color information. For capturing texture aspects, we use *TPLBP* descriptor [182]. We use both Opponent SIFT [166], and PHOG [18] descriptors for capturing both appearance, and shape aspects, respectively. For the vocabulary creation, we use a standard K-means for constructing vocabularies of size $1.5k$ as in [1]. We use a three-level SP, which results in $(1.5k \times 21 = 31.5k)$ dimensional histogram. Finally, we use a non-linear Support Vector Machine (SVM) classifier with χ^2 kernel.

To evaluate the classification performance, we use the mean of the diagonal values of the confusion matrix. This score is averaged over 10 trials, where training and testing samples are replaced randomly. For **VS**, and **OT** data sets, we follow the same learning protocol proposed in [19]. Hence, the data sets are split randomly into two separate sets of images, half for training and half for testing. From the training set, we randomly select 100 images to form a validation set.

3.5 Experiments

In this section, we provide experimental results to validate our proposed contributions.

3.5.1 Evaluation of PC-TPLBP

In this section, we investigate the effect of texture features for scene recognition. We first compare the different LBP descriptors discussed in Sec. 3.2.1 over OT, and VS data sets. We also report the performance score of the *Gabor* descriptor, as a baseline. As shown in Table 3.1 *TPLBP* outperforms the results significantly.

Table 3.1, also shows the importance of fusing color information with *TPLBP*. For **OT**, the combination of *TPLBP* with *oppC* yields to a minor improvement (83.8%). We refer to it as *(C-TPLBP in OppC)* in Table 3.1. However, the combination of *TPLBP* with *HSV* yields a performance increase up to (85.0%). We refer to it as *(C-TPLBP in HSV)*. For **VS**, similar behavior is obtained. *C-TPLBP in HSV* results in improving the performance from (82.0%) to (84.5%). However, less gain (82.6%) is obtained using *C-TPLBP in OppC*. This demonstrates the importance of *C-TPLBP* over *TPLBP*.

Finally, we examine the effect of incorporating spatial information to *C-TPLBP*. For **OT**, this gives an increase up to (87.0%). For **VS**, *PC-TPLBP* results in increasing the score up to (86.6%). As depicted from the results, *PC-TPLBP* improves the performance of *C-TPLBP*. We conclude that both color, and spatial information cues play an important role for scene recognition.

Method	OT	VS
Gabor	66.0	65.2
CSLBP	73.0	71.4
FPLBP	76.0	74.5
TPLBP	83.0	82.0
C-TPLBP in OppC	83.8	82.6
C-TPLBP in HSV	85.0	84.5
PC-TPLBP	**87.0**	**86.6**

Table 3.1: Classification Score using different LBP operators.

3.5.2 Evaluation of CASP with PC-TPLBP

In this section, we demonstrate the benefits of our two-stages compression strategy. Based on the empirical evaluation, we determined the best compression scheme over *OT* data set, and continued the rest of the experiments using the same configuration.

Evaluation of Feature Compression with PC-TPLBP

As a baseline, we directly apply the original *AIB* on our *PC-TPLBP* (denoted as *DirectComp*). We then examine our feature compression approaches described in Sec. 3.2.2), namely: *WholeComp*, and *LevelComp*.

Table 3.2 shows a major loss in the performance using the *DirectComp* scheme. We attribute this performance degradation to the fact that *AIB* is greedy in its nature. When *AIB* suggests fusing two features, it just looks greedily all over the SP features which minimize the overall loss in its discriminative power. As explained earlier, these two candidate features can be from different SP regions, which in turn leads to obtaining different vocabularies within SP region. Hence, discarding the important spatial property of SPs. Subsequently, this results in dropping the final SP performance.

The quantitative results in Table 3.2 also shows that both of our spatially enhanced feature compression schemes outperform the *DirectComp* method. Moreover, *LevelComp* is the best performing scheme: as it preserves the original SP performance, while reducing the dimensionality significantly.

For *LevelComp*, we show in Fig. (3.7) that there is a strong relation between the vocabulary size, and the SP level of concern. In other words, the finer the *SP* level, the fewer

Method	Size	Score
$Original$	$31.5k$	87.0
$DirectComp$ [54]	$14k$	82.0
$WholeComp$	$14k$	85.2
$LevelComp$	$14k$	87.0
GPS	$6k$	89.5
CPS	$< 6k$	**90.6**

Table 3.2: Classification Score on OT data set to compress a SP of size $31.5k$ to a $14k$ one using $PC - TPLBP$, see text for details.

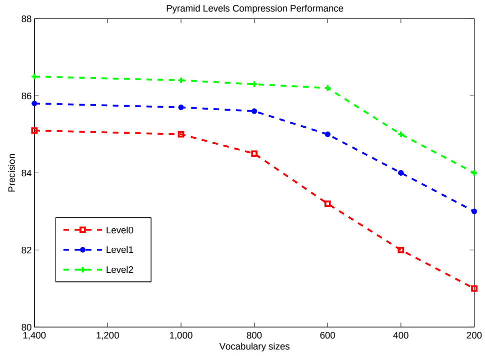


Figure 3.7: Learning specific vocabulary compression per SP level. See text for details.

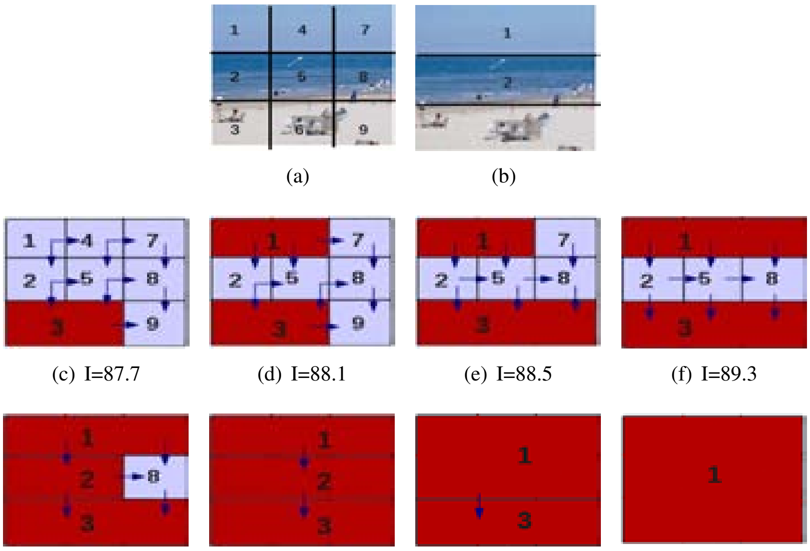
the number of features required to represent it, while maintaining its accuracy. Hence, for a three-level SP a vocabulary of size $0.6k$ is sufficient. While, for a two-level SP a vocabulary of size $0.8k$ is needed. However, for coarser *BoW* representation, a vocabulary of size $1k$ is required.

Evaluation of Block Compression with PC-TPLBP

In this section, we examine the usage of our block compression scheme proposed in Sec. 3.3.3. The quantitative results reported in Table 3.2 show the successfulness of our proposed scheme, in terms of accuracy and dimensionality reduction.

GPS reduces the SP dimensionality up to $6k$ ($1k + 0.8k \times 4 + 0.6k \times 3$), while improving the original SP performance. Fig. (3.8) shows an example of the adaptive shapes obtained after each iteration. Interestingly, our approach recommends the usage of the horizontal 3×1 shape for scene recognition. Thus, justifying theoretically the better performance obtained using the ad-hoc horizontal 3×1 SP proposed by Marszalek et al. [**?**] over the standard SP proposed by Lazebnik et al. [97]. We also demonstrate in Fig. 3.8(j) the convergence of our approach.

The last column in Table 3.2 shows that the performance of *CPS* scheme improves over



(a) (b)

(c) I=87.7 (d) I=88.1 (e) I=88.5 (f) I=89.3

(g) I=89.5 (h) **I=91.1** (i) I=89.0 (j) I=86.4

Figure 3.8: Top left grid in blue represents the (3×3) grid for an input image. Our *GPS* fuses the most similar blocks (depicted as red). The scores (I) are optimized over all the categories. Fig. 3.8(h) shows the successfulness of the 3×1 shape for scene recognition.

GPS. For dimensionality comparison, we use the notion $< 6k$ to indicate the upper bound of *CPS* dimensionality, since it varies per category, see Fig. 3.9. For instance, for the *coast* category, we obtain a $6k$-dimensional histogram $(1k + [0.8k \times 4] + [0.6k \times 3])$. While, for the *forest* category, a $5.4k$-dimensional histogram $(1k + [0.8k \times 4] + [0.6k \times 2])$ is obtained. Fig. (3.9) shows the learned shapes obtained for each category using our *CPS* scheme.

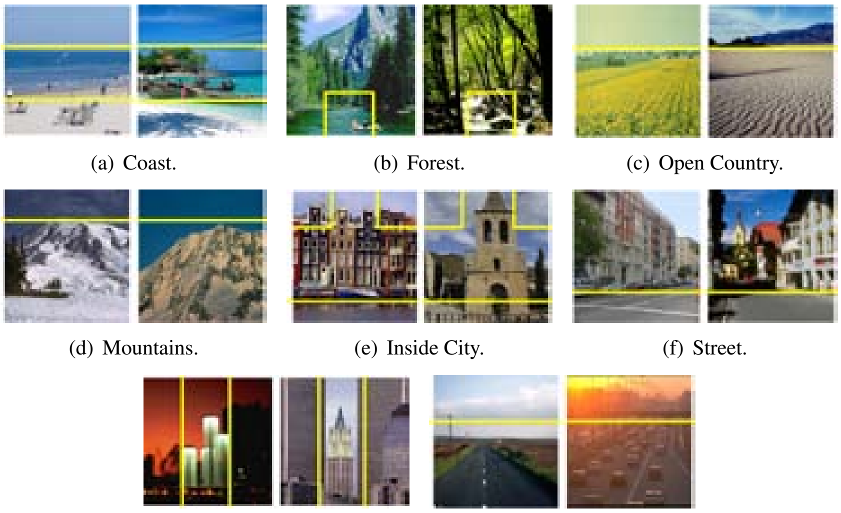
In conclusion, our feature, and block compression stages, which we refer to them as *CASP* are both necessary for obtaining compact, yet efficient SP.

3.5.3 Combining Multiple Cues using CASP

In this section, we investigate the importance of fusing spatial texture, shape, and appearance features besides the global image cues using our *CASP* representation.

To this end, we used CASP of *PC-TPLBP* for capturing texture aspects. For appearance, we used CASP of Opponent-SIFT features (denoted as PC-SIFT). For shape, we extended the PHOG descriptor to incorporate color information motivated by [1, 166, 143, 80]. Table 3.3 demonstrates that coloring *PHOG* is beneficial for our task. Better results are obtained by fusing PHOG with HSV, which we further refer to it as *(PC-HOG)*. Table 3.3 also shows the importance of fusing *CASPs* of shape, and appearance cues.

In Table 3.4, we use the notion Local Descriptors (LD) to refer to CASP, which are based on pixel-based statistics features (PC-HOG, and PC-SIFT). While, we use the notion Re-



(a) Coast. (b) Forest. (c) Open Country.

(d) Mountains. (e) Inside City. (f) Street.

(g) Tall Building. (h) Highway.

Figure 3.9: Examples of our *CPS* adaptive shapes learned over OT data set.

gional Descriptors (RD) to refer to CASP, which are based on patch-based statistics features (PC-TPLBP). Lastly, we use the notion Global Descriptors (GD) for features that capture the global image semantics (GIST).

The quantitative results in Table 3.4 illustrate the importance of combining (i) GD with LD as demonstrated in [135], (ii) RD with GD, (iii) RD with LD using our *CASPs* representation, and, finally (iv) GD, RD, and LD. Table 3.4 also shows that the *CPS* learning scheme improves the performance over the *GPS* by around 2%, while reducing the dimensionality to less than $< 18.4k$.

Finally, we compare the performance of our approach with previous work on OT, VS, and QuT data sets. Table 3.5 summarizes, and compares our results with state-of-the-art methods. For **OT**, our best score using our approach that exploits the CASPs of *(LD + RD + GD)* along with the *CPS* learning scheme is 97.4%. The results obtained excel the state-of-the-art for this dataset [131, 97, 1, 126]. For **VS** data sets, we achieved a score of 96.2%. These results also outperformed the best reported ones on this data set [97, 1, 177]. In **Indoor**67, our best score is 48.9%, which exceeds the state-of-the-art score 45.5% in [117].

3.6 Discussion and Conclusion

In this work, we proposed a novel and efficient texture descriptor based on patch-based texture features *TPLBP*. For this, we incorporated both color information *C-TPLBP*, and spatial information *PC-TPLBP* to *TPLBP*. Furthermore, we addressed the high dimensionality problem of the generated SP histograms.

Method	OT		VS	
	Size	Score	Size	Score
PHOG [18]	31.5k	79.5	31.5k	78.2
PHOG + OppC	31.5k	81.7	31.5k	80.2
PHOG + HSV *(PC-HOG)*	31.5k	83.2	31.5k	82.6
PC-HOG + LevelComp	14k	83.0	14k	82.5
CASP-CHOG	6k	85.2	6k	84.6
PC-SIFT [166]	31.5k	88.4	31.5k	87.7
PC-SIFT + LevelComp	14k	88.2	14k	87.5
CASP-CSIFT	6k	90.2	6k	89.5
CASP-CHOG&CSIFT	12k	**91.6**	12k	**90.8**

Table 3.3: Classification scores of (i) Fusing PHOG with HSV (denoted as PC-HOG) outperforms that of OppC. (ii) Combining CASPs of PC-HOG, and PC-SIFT.

Features for CASP	OT		VS	
	Size	Score	Size	Score
GD	0.4k	83.7	0.4k	82.9
RD	6k	89.5	6k	88.8
LD	12k	91.6	12k	90.8
GD + RD	6.4k	91.0	6.4k	90.2
GD + LD	12.4k	92.8	12.4k	92.0
LD + RD	18k	93.5	18k	92.5
GPS with LD+RD+GD	18.4k	95.2	18.4k	94.2
CPS with LD+RD+GD	$< 18.4k$	**97.4**	$< 18.4k$	**96.2**

Table 3.4: Experimental results with *CASP* demonstrate that combining shape, appearance (*LD*), texture (*RD*) with global cues (*GD*) improves the performance significantly. See text for details.

Method	OT	VS	QuT
Vogel et al.[177]	-	75.1	-
Oliva et al. [126]	83.7	-	-
Bosch et al. [1]	86.6	85.7	-
Perina et al. [131]	92.8	90.3	-
Quattoni et al. [135]	-	-	25.0
Nakayama at al. [117]	-	-	45.5
CPS with LD+RD+GD	**97.4**	**96.2**	**48.9**

Table 3.5: Comparison with state-of-the-art.

For this purpose, we introduced a novel SP compression approach, which works on two stages. The first compression stage is done within the SP features. We eliminated the spatially least informative features for each SP level. We also showed that there is a strong relation between the compact vocabulary size, and the SP level in concern: the finer the level, the fewer the required words for representing it. The second compression stage is done within the blocks of each level. We further introduced two alternative approaches, namely, *GPS* and *CPS* for learning the best SP block partitioning. Considering the *GPS* scheme, we justified theoretically the better performance of the ad-hoc horizontal h3 $\times$ 1 pyramid [**?**] over the traditional one [97] for the task of scene recognition. When the *CPS* scheme is considered, the resulting *CASP* representation maintains the performance of their original counterparts, while reducing the dimensionality significantly.

Finally, we showed the importance of combining the complementary patch-based texture features *(regional)* with the pixel-based shape and appearance ones *(local)*. In addition, we investigated the effect of fusing *global* image cues along with *regional*, and *local* ones, which resulted in improving the overall performance. Consequently, we conclude that *CASP-based* complementary descriptors, together with class-specific learning are all important for obtaining good performance. We evaluated the proposed framework on scene recognition task, and obtained state-of-the-art results on several scene recognition benchmark data sets.

Chapter 4

Spatial Pyramids derived from 3D Scene Geometry for Improved Object Recognition

The Bag-of-Words (BoW) approach has been successfully applied in the context of category-level image classification. To incorporate spatial image information in the BoW model, in general, spatial pyramids are used. However, spatial pyramids are rigid in nature and are based on pre-defined grid configurations. As a consequence, they often fail to coincide with the underlying spatial structure of images from different categories which may negatively affect the classification accuracy.

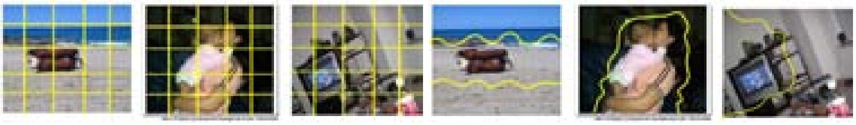
The aim of the paper is to use the 3D scene geometry to steer the layout of spatial pyramids for category-level image classification (object recognition). Our approach provides an image representation by inferring the constituent geometrical parts of a scene. As a result, the image representation contains the descriptive spatial information to yield a structural description of the image.

From large scale experiments on the Pascal VOC 2007 and Caltech101, it can be derived that SPs which are obtained by search selective outperforms the standard SPs with 12.4% and 14.0% for Pascal VOC 2007 and Caltech101 respectively. The use of 3D scene geometry, to select the proper SP configuration, provides an even higher improvement of 16.0% and 19.7% respectively.

4.1 Introduction

For category-level image classification and object recognition, the Bag-of-Words *(BoW)* approach has been successfully applied [32, 112, 47, 191]. The *BoW* is based on the occurrences of image features. Hence, it treats the image as an order-less collection of local features completely ignoring the spatial image layout. This is generally considered as the foremost shortcoming of the standard *BoW* approach.

To extend the *BoW* with spatial information has therefore received considerable attention. Recently, several approaches consider the success of the Spatial Pyramid *SP* approach proposed by Lazebnik et al. [97]. It is shown that the use of *SP* outperforms the 1×1 image representation on challenging image classification tasks [97], due to the inclusion of image-to-image geometric correspondences. However, in general, SPs are based on rigid image subdivisions (e.g., grids). These rigid spatial configurations are not suited for freely formed objects and scenes. In Fig. 4.1(a), some examples are shown taken from different image categories together with their standard SP sub-division. Sub-regions divide objects into two separate parts increasing the probability of dissimilar image features within cells and similar image features across cells. Hence, a rigid division may result in a negative equivalence-class configuration of image features.



(a) Rigid SP.

(b) Flexible SP.

Figure 4.1: Show example images from different categories. (a) shows the standard SP proposed by Lazebnik et al. [97]. (b) shows the proposed flexible "spatial partitionings" which best suit each category. Images are from the Pascal dataset [43]

Our aim is to use the 3D scene geometry to steer the layout of spatial pyramids for category-level image classification. Images within a category share similar scene geometries. We exploit correspondences between categories by scene geometry matching scheme. For example, Fig. 4.1(b) shows the geometrical (depth) layers for some example images. The *cow example* corresponds to scene geometry style consisting of 3 segments: (1) ground, (2) background and (3) sky. The ground part depicts different objects than the background and sky. Each segment contains similar features and features across segments are more dissimilar. Therefore, the *BoW* should be applied separately to each geometrical scene sub-region.

In this paper, we propose a method to obtain a holistic image representation by inferring the constituent geometrical parts of a scene. The method steers the image layout on the basis of 3D scene geometry (i.e., "Stages") computed from a single image. We propose two alternative approaches to obtain structural image representations from 3D scene geometries:

(1) *Generic SPs:* by exploiting the 3D scene geometry of images. The 13 stages of [175] are used as 3D priors. After the image scene geometry is estimated, the most appropriate stage per object category is selected as the SP.

(2) *Selective SPs:* by obtaining the spatial subdivision representation based on selective search guided by the *Agglomerative Information Bottleneck (AIB)* theory [152, 54]. The models are used to select the spatial geometry that best suits each category at hand.

The two methods to generate SPs will be compared to existing rigid SP for object recognition tasks. To this end, two benchmark data sets are used in the experiments: Pascal VOC 2007 [43] and Caltech-101 [45]. Besides, a large data set is provided *(denoted as "stage data*

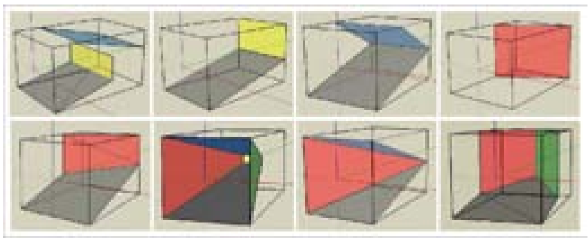


Figure 4.2: Stage models and their corresponding instantiations. Top row, from left to right: “sky+backgnd+gnd”, “backgnd+gnd”, “sky + gnd”, “gnd + diagalBackgndLR”. Bottom row: “diagalBackgndLR”, “box”, “1side-wallLR”, “corner”

set”) to learn each stage [1].

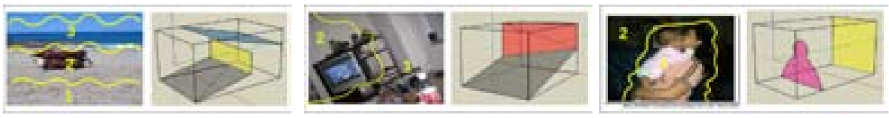
This paper is organized as follows. First, in section 4.2, we give the motivation of our approach. In section 4.3, the method is proposed to generate dynamic spatial pyramids. In section 4.4, the experimental setup is discussed and the results are given. Finally, we give the conclusion in Sec. 4.5.

4.2 Motivation

Although, SPs divide images in an efficient way, the question still remains what the right subdivision scheme is. The subdivision scheme should consider the trade-off between two important design properties *invariance* and *descriptiveness*. *Larger* sub-regions are preferred to gain invariance to viewpoint changes (translation, orientation and scale) and object occlusion. Sub-regions should cover the range of possible positions of occurring objects. For example, the entire image is invariant to all possible object positions. *Smaller* sub-regions are required to obtain more descriptive regions and spatial layout. Sub-regions should depict similar object/background augmenting the descriptive ability of the SP. Finally, regions should not be constrained in shape allowing for a natural division of the image into its constituent parts.

In this paper, we propose a strategy to divide the image into its constituent *scene geometry* parts to obtain an *invariant* and *descriptive* image representation. The aim is to split the image into sub-regions corresponding to generic scene (depth) layers. These layers provide a middle ground between low-level features and high-level object categories. A number of methods have been proposed to estimate the rough scene geometry from single images [77, 36, 154]. We use the scheme which derives scene information for a wider range of generic scene categories by using *stages* [175]. Stages are defined as a set of prototypes of often recurring scene configurations. They can be seen as discrete classes of scene geometries. Typical classes of discrete $3D$ *scene geometries* include single-side backgrounds (e.g. walls and buildings) or three sides (e.g. corridor and narrow streets). A number of stage models are shown in Figure 4.2. These models are dependent on the inherent geometrical structure

[1] The dataset will be made publicly available.



(a) gnd+backgnd+sky (b) gnd+diagonalLR. (c) Person.

Figure 4.3: Example depth images with their corresponding 3D geometries

of images. In this paper, 13 different stages are used excluding *noDepth* or *tab+pers+bkg*, as these stages are specific characteristics of the data set used in [175].

As shown in Fig. 4.2, the scene structures of the stage models are shown in different colors. The stage models are used to determine how the image is divided in sub-regions. For instance, images of stage *sky+backgnd+gnd* are divided into three layers: sky (in blue), background (in yellow) and ground (in brown). In Fig. 4.3(a), it is shown that the example images from Fig. 4.1 are instantiations of the *"sky+backgnd+gnd"*, *"person"* and *"gnd+diagonal"* stages, respectively. Each scene (depth) layer is equivalent to an image segment. Hence, SPs are constructed based on 3D scene geometries in which each geometry layer (e.g., ground, background and sky) is represented by a different sub-region.

4.3 Spatial Layouts derived from 3D Scene Geometry

We aim to exploit similarities in geometric scene style for object recognition. To incorporate correspondences between geometries, we use techniques inspired by SPs. However, instead of using rigid SPs, we propose the use of stage correspondences. In this section, we briefly review the SP, after which we present a generalization resulting in our stage pyramid.

The SP scheme proposed by [97] represents an image by using weighted multi-resolution histograms which are obtained by repeatedly sub-dividing an image into increasingly finer sub-regions, where the spatial pyramid at level $l \in \{0, \ldots, L\}$ has $R(l) = 2^{2l}$ sub-regions. For image X, all features are assigned to their best visual word v selected from a vocabulary V. The frequency of v inside sub-region i of image X is given by the histogram bin $H_X^i(v)$. The similarity or matching rate between images X and Y at level l, is given by the histogram intersection function [66]:

$$I^l(X, Y) = \sum_{i=1}^{R(l)} \sum_{v=1}^{|V|} \min(H_X^i(v), H_Y^i(v)). \tag{4.1}$$

Matches found at finer resolutions are closer to each other in the image space and are therefore more heavily weighted. To accomplish this, each level is weighted to $1/2^{L-l}$ which results in the final SP:

$$\kappa^L(X, Y) = \frac{1}{2^L} I^0(X, Y) + \sum_{l=1}^{L} \frac{1}{2^{L-l+1}} I^l(X, Y). \tag{4.2}$$

For the geometry-driven pyramid we use the same approach, only for geometric equivalent classes.

Formally, for each scene geometry or stage $s \in S_1, \ldots, S_n$ of n stage types, let $R(s)$ denote the number of sub-regions of s. Instead of using a fixed pyramid, we propose that a spatial stage pyramid is created by computing the similarity between images X and Y for stage s by

$$I^s(X, Y) = \sum_{i_s=1}^{R(s)} \sum_{v=1}^{|V|} \min(H_X^{i_s}(v), H_Y^{i_s}(v)). \tag{4.3}$$

Where the different sub-regions i_s for stage s correspond to a scene part at a certain depth (layer). For each stage s there are $R(s)$ sub-regions.

We propose two alternative approaches for selecting the appropriate spatial image representation for each category. This is achieved by learning a proper class-specific spatial model. These models encode the proper spatial partitioning for each category; which is used further to obtain class-specific spatial models. To this end, we first exploit the use of the standard 3D geometry model as a prior for learning the best candidate template for each category. Then, we propose a learning approach for learning the most suitable category-model based on information theory.

4.3.1 Generic Geometry Pyramids

In this section, instead of using rigid SPs, a more natural way to construct SPs based on scene geometries is proposed. The 13 different scene geometries $s \in \{S_1, , S_{13}\}$ proposed by [175] are used. It has been shown that these geometries cover most of the image partitionings encountered in real-world scenarios. Hence, 3D geometry structures are used to determine how the image should be divided; where each geometry depth corresponds to a pyramid region i_s. Therefore, the 13 prior stages are considered as *generic models*. Images of each category are classified into one of these stages in order to select the most appropriate spatial representation. More formally, the proposed method consists of the following steps: first, training images are spatially represented according to each of the 13 binary *mask maps*. We use the *stage* data set to obtain the binary mask maps. The training set is manually annotated and divided into scene geometries as in [139]. The parameters of each geometry (horizon, vanishing points) are computed to fit the underlying data. Image segmentation is based on the occurrence probability in the training set, see [139] for details. For each category, we then train 13 geometry models and learn on the validation set which geometry or even the combination of geometries that best suits each category. The whole process is demonstrated by the block diagram in Fig. 4.4. For a new image, it is represented using the 13 binary maps (off-line step). We evaluate the learned geometry model of each category w.r.t. its appropriate representation. Consequently, the test image will have a score towards each category and is assigned to the category with the highest score. We summarize the whole procedure in Algorithm 1.

4.3.2 Selective Spatial Pyramids

In this section, we propose an approach for learning the spatial partitioning using AIB [152, 54]. In fact, the main goal of *AIB* is to reduce the dictionary of visual words V required for

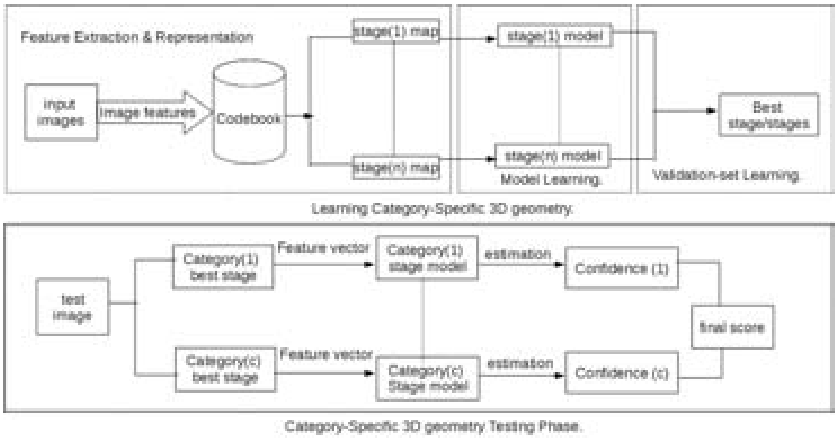


Figure 4.4: Outline of the spatial representation using 3D scene geometry. Note that the codebook models and the stage models are obtained off-line. For each category, the proper stage model is obtained

Algorithm 1 Generic Geometry Pyramids

```
1. Require: Binary Map (BM) of the 13 scene geometries $s \in \{S_1, , S_{13}\}$). Each BM has a number of subregions $R^s$.
2. Training from 2-6: Construct a histogram $H_X^{i_s}$, for each BM subregion $i_s$ of image $X$.
3. Train a Geometry Model (GM) with each BM representation.
4. Evaluate the performance score on the validation set.
5. Repeat steps 2 to 5 for each category.
6. Set the category geometry to GM with the highest score.
7. Testing from 7-8: For a test image, evaluate its performance for each category using its GM & BM.
8. Assign the test image to category label with the highest score.
9. The matching between images $X$ and $Y$ for a stage $s$ pyramid is given by:
$I^s(X, Y) = \sum_{i_s=1}^{R(s)} \sum_{v=1}^{|V|} \min(H_X^{i_s}(v), H_Y^{i_s}(v))$.
```

representing the categories Z. This means generating a compact set of words $\hat{V}$ from the original dictionary V so that the loss of mutual information to the categories Z is minimal:

$$I(V; Z) = D_{KL}\left[p(v, z) \| p(v)p(z)\right] \tag{4.4}$$

The functional $D_{KL}\left[p\|q\right]$ is the *Kullback-Leiber* divergence. The joint distribution $p(v, z)$ is estimated from the training set by counting the number of occurrences of each visual word in each category. The information about v captured by z can be measured by the mutual information,

$$I(V, Z) = \sum_k \sum_c p(v_k, z_c) log \frac{p(v_k, z_c)}{p(v_k)p(z_c)}, \tag{4.5}$$

measuring the discriminative power I that one random variable carries about the other. The merging of visual words is achieved by iteratively applying *AIB* for fusing those two visual words v_k and v_j into $\hat{v}$ that causes the smallest decrease $I(\hat{V}; Z)$ in the original mutual information $I(V; Z)$:

$$\delta I_z(V; \hat{V}) = I(V; Z) - I(\hat{V}; Z). \tag{4.6}$$

At each step *AIB* performs the best possible merge $argmin_{\hat{V}} = \delta I_z(V; \hat{V})$. We extend the original *AIB* to learn the most appropriate class-specific spatial structure. Concretely, we first over-segment the images as [114]. For each category, we learn on the validation set the most suitable split-up by pruning the least informative image segments based on *AIB*. Hence, we relate each image segment (i_s) with the data set categories Z, such that the probabilities of each segment $p(i_s)$ is calculated by summing up the probabilities of the vocabularies (v_u) it contains as:

$$p(i_s) = \sum_{i_s=1}^{R} \sum_{u=(i_s-1)|V|+1}^{k|V|} p(v_u), \tag{4.7}$$

where i_s is the segment index, R is the total number of image segments, u is vocabulary index within segment i_s. To evaluate the Discriminative Power *(DP)* of the generated segments w.r.t. the categories, we use the information content criteria:

$$I(S, Z) = \sum_{i_s=1}^{R} \sum_{c=1}^{|C|} p(i_s, z_c) log \frac{p(i_s, z_c)}{p(i_s)p(z_c)}, \tag{4.8}$$

where c is the category index, and C is the total number of data set categories. Finally, the loss in *DP* is then obtained by:

$$\delta I_z(S; \hat{S}) = I(S; Z) - I(\hat{S}; Z). \tag{4.9}$$

Formally, for each category the best spatial split-up (partitioning) $s \in S$ is learned. Each split-up s has a number of subregions $R(s)$. We propose that that the spatial partitioning matching function is created by computing the similarity between images X and Y for partitioning s by:

$$I^s(X, Y) = \sum_{i_s=1}^{R(s)} \sum_{v=1}^{|V|} \min(H_X^{i_s}(v), H_Y^{i_s}(v)). \tag{4.10}$$

Where the different sub-regions i_s for the partitioning s correspond to a different image segment, such as sky, ground, etc. The whole process is demonstrated by the block diagram in Fig. 4.4. We summarize the procedure of the selective search in Algorithm 2.

4.4 Experiments

Three independent data sets are used in the experiments. The first data set is a large dataset consisting of 3589 images classified as 15 different categories representing the standard scene geometries. We refer to it as *"stages data set"*. This data set is used for the sake of generating the binary mask maps used in the *Generic Pyramids* approach. Some example images in this data set are shown in Fig. 6.5. We also use Caltech-101 [45] and Pascal 2007 [43] data sets as benchmark for evaluating our approach.

Algorithm 2 Selective Spatial Pyramids

```
1. Require: a neighborhood matrix (N); defines the neighbors of each segment $i_s$.
2. Compute DP of the current over-segmented shape, using Eq. (4.8).
3. Calculate the DP loss to merge segment $i_s$ with its neighbors by Eq. (4.9).
4. Perform segment merge leading to the minimum DP loss; generate split-up $(S)$.
5. Evaluate S score on the validation set.
6. Set the current shape to S for next iteration.
7. Update the N matrix to reflect the NS.
8. Repeat steps 3 to 7 until convergence (i.e., N is empty).
9. Pick split-up $s \in S$ (has $R(s)$ segments) of highest score as the category model.
10. The similarity between images $X$ and $Y$ for same spatial split-up $s$ is given by: $I^s(X,Y) = \sum_{i_s=1}^{R(s)} \sum_{v=1}^{|V|} \min(H_X^{i_s}(v), H_Y^{i_s}(v))$.
```

4.4.1 Experimental Setup

To compare the different spatial pyramids, a standard BoW approach with SIFT features of 16×16 pixel patches are used. For Caltech-101, we use 30 images per category for training and 50 for testing. The general architecture follows [97]. The SIFT descriptors are extracted on a dense grid and we use a codebook of size 300. Experiments are conducted over 10 random splits of the data, and the average per-class recognition rates are recorded for each run. The final result is reported as the mean accuracy and its standard deviation from the individual runs. For Pascal 2007, a standard multi-scale grid detector is used together with a Harris-Laplace point detector [112], and a blob detector. SIFT descriptors are computed for all regions in the feature descriptor step, which are then quantized to a codebook of size 1000. Mean average precision *(MAP)* is used to evaluate the performance of the features over all the data set categories. We compare our approach with the standard *SP* proposed by Lazebnik et al.[97] as a baseline.

4.4.2 Generic Spatial Pyramids-*(Generic SP)*

In this section, the "Generic SP" approach proposed in Sec. 4.3.1 is evaluated using $3D$ scene geometries. For each category, the geometry with the highest score is selected for representing it. In table 4.1, we show the obtained scores compared with the standard SP. It is demonstrated that the *Generic SP* improves the results by 8.5% and 9.0% (relative to the baseline) on the Pascal and Caltech data sets, respectively. We attribute this to, (i) the generic $3D$ geometries contains a wide range of spatial partitionings which cover most of the "real-world" object categories; (ii) the obtained representations of our approach, are tailored for each category, and therefore can efficiently capture the variabilities that exists within each category.

In Fig. 4.6, we show some example images for various data set categories together with their most appropriate $3D$ scene geometries. These quantitative results illustrate that different $3D$ scene geometries are selected for representing the various data set categories. For instance, the *plane* category is instantiated from the *sky+gnd* scene geometry. While, the

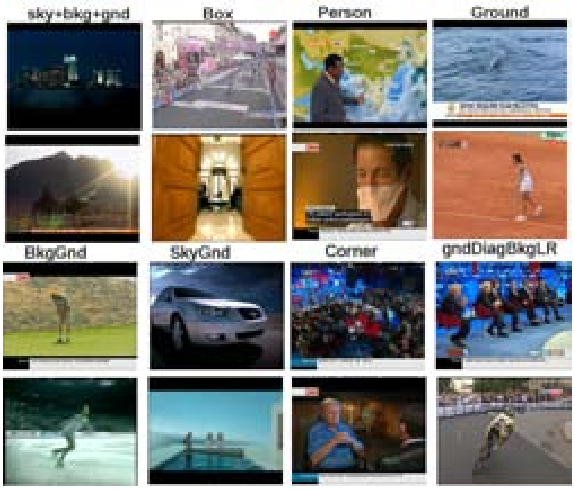
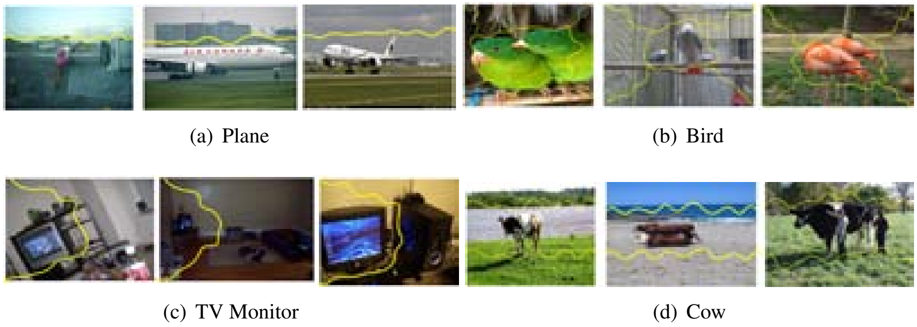


Figure 4.5: Example images of “stages data set”

Method	Pascal	Caltech
Lazebnik et al. [97], linear kernel	48.3	51.4 ± 0.9
Lazebnik et al. [97], intersection kernel	51.5	64.6 ± 0.8
Generic SP, linear kernel	52.5 (+8.7%)	55.5 ± 1.6 (+8.0%)
Generic SP, intersection kernel	**55.9** (+8.5%)	$\mathbf{70.5 \pm 1.3}$ (+9.0%)
Selective SP, linear kernel	54.2 (+12.2%)	57.9 ± 1.5 (+12.6%)
Selective SP, intersection kernel	**57.9** (+12.4%)	$\mathbf{73.6 \pm 1.5}$ (+14.0%)
Generic SP + MKL	**59.7** (+16.0%)	$\mathbf{77.3 \pm 1.1}$ (+19.7%)

Table 4.1: Results obtained on Pascal (MAP score) and Caltech (Average per-class recognition rates). Our approach improves over the standard SP proposed by Lazebnik et al.[97]. The best geometric split-up learned over multiple kernels (*MKL*) improves the performance significantly (see text)



(a) Plane

(b) Bird

(c) TV Monitor

(d) Cow

Figure 4.6: Pascal data set examples with their learned geometries. Plane instantiated from "sky+gnd", Bird from "box" geometry, TV monitor from "gnd+DiagBkgLR" and Cow from "sky+backgnd+gnd"

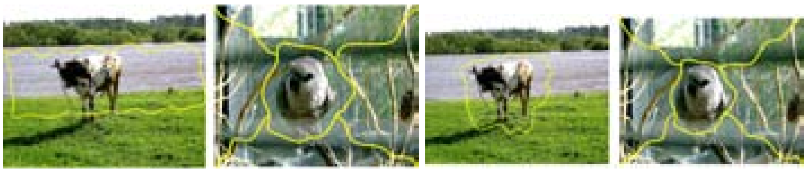
bird category is instantiated from the *box* geometry. Hence, using $3D$ scene geometries is important for efficiently capturing the spatial layout per category.

Another advantage of the "Generic SP" scheme is its ability of reducing the dimensionality of the generated histograms. The maximum number of partitions that exist for representing a category is 6 (i.e., "box geometry" + *BoW*). This leads to a final representation of size $6 \times |V|$, where $|V|$ is the size of the vocabulary. On the other hand, the number of partitions of the standard "SP" is 21. This leads to a final representation of size $21 \times |V|$. We now investigate the use of "Multiple Kernels Learning *(MKL)*" proposed by Gehler et al. [57], for the selection of the most appropriate geometry or the combination of geometries per category among multiple kernels. The results, in table 4.1, demonstrate the importance of using multiple kernels for our approach. This improves the performance significantly by 16.0% and 19.7% (relative to the baseline) on Pascal and Caltech data sets, respectively.

4.4.3 Selective Pyramid- *Selective SP*

In this experiment, we evaluate the *Selective SP* proposed in Sec. 4.3.2, where we learn the most appropriate "partitioning" by eliminating the least informative partitions. The quantitative results, in table 4.1, show an improvement of 12.4% and 14% on Pascal and Caltech data sets, respectively. The main advantage of this method, is the ability of its learning procedure to learn compact yet discriminative partitioning (from its initially over-segmented images) which can efficiently fits each category.

In Fig. 4.7, we show some examples with their learned spatial partitionings using the *selective* and the *generative* SPs. The quantitative results demonstrate that the *selective SP* learns compact spatial split-ups, and are instantiations of the generic $3D$ scene geometries. For instance, the *selective-SP* partitionings learned for "cow" and "bird" examples are instantiations of the "skyBkgGnd" and "box" geometries, respectively. This explains the performance improvement obtained based on the *selective-SP* approach over the *generative-SP* approaches.



(a) Generic.

(b) Selective.

Figure 4.7: Pascal data set examples for the spatial split-up learned using (a) *Generic* and (b) *Selective* SP approaches for "Cow " and "Bird" examples. The *Selective SP* generates compact representations and are instantiated from $3D$ scene geometries (see text)

To summarize, we compare the computational complexity between the proposed methods with respect to the standard SP in terms the SP levels, dimensionality and speed. The standard Lazebnik SP has 3 levels (i.e., $BoW + 4 + 16$) and a dimensionality of $21 \times V$. On the other hand, both the Generic and Selective SPs have an average of 2 levels (i.e., $BoW + 2$). This leads to a dimensionality of $3 \times V$ and $6 \times V$ for the average and worst case, respectively. Regarding the speed, the Lazebnik SP, *Selective SP* and *Generic SP* need almost same time for the testing scenario. For the training phase, the *selective SP* has the highest computational complexity, although it generates compact yet efficient spatial partitioning for each category. On the other hand, *Generic SP* approach balances between accuracy and complexity. Hence, we conclude that there is a trade-off between the required accuracy and the complexity of the method. Therefore, we consider *Generic SP* as our baseline approach.

4.4.4 Comparison with State of the Art

In this section, we investigate the *Generic SP* performance under varying vocabulary sizes. Finally, we compare our approach to state-of-the-art methods based on a single type of descriptor.

Comparison under Multiple Vocabulary Sizes.

In Figure 4.8, we compare the performance of our approach to the state-of-art SP using various vocabulary sizes (i.e., $1k$, $2k$, $4k$ and $6k$) on the Pascal data set. The results show that our approach outperforms the standard state-of-art SP over all the examined vocabularies. In addition, our results confirm the experimental findings in the work of [57], where the use of *MKL* in our approach improves the overall performance for the various vocabulary sizes.

Comparison to State-of-the-Art.

Comparison with previously published results obtained using one type of descriptor are shown in Table 4.2 and Table 4.3 for the Caltech and the Pascal data sets, respectively.Zhang et al.

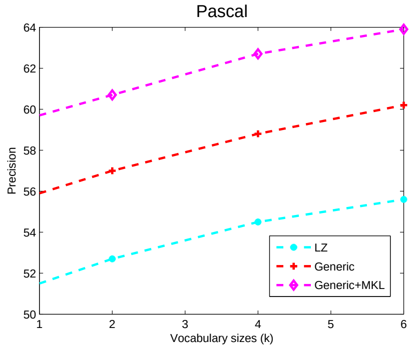


Figure 4.8: Comparison between our $GenericSP$ and $GenericSP + MKL$ approaches with standard SP (denoted as *LZ*) using the SIFT descriptor under different vocabulary sizes $1k$, $2k$, $4k$ and $6k$ on Pascal data set (see text)

	Method	Caltech
Boureau et al. [20]	SP + hard quantization [256]	64.2 ± 1.0
Boureau et al. [20]	SP + hard quantization + max pooling [256]	64.3 ± 0.9
Lazebnik et al. [97]	SP + hard quantization + kernel SVM	64.6 ± 0.8
Boureau et al. [20]	SP + soft quantization [512]	66.1 ± 1.2
Boureau et al. [20]	SP + sparse codes [1024]	70.3 ± 1.3
Boiman et al. [16]	Nearest neighbor + spatial correspondences	**70.4**
This paper	Selective SP	$\mathbf{73.6 \pm 1.5}$ $(+4.5\%)$
This paper	Generic SP + MKL	$\mathbf{77.3 \pm 1.1}$ $(+9.8\%)$

Table 4.2: Results obtained by several recognition schemes using a single type descriptor and intersection kernel on Caltech data set, see text for details. The numbers shown inside brackets in [20] are the codebook sizes used in this work.

For Caltech. A performance improvement of 4.5% is obtained based on our *Selective SP* approach w.r.t. the best performing method as shown in table 4.3. Moreover, an improvement of 9.8% is obtained based on our *Generic SP + MKL* approach. Note that better performance has been reported with subcategory learning 83% [159] or multiple descriptor types (e.g., methods using multiple kernel learning achieved $77.7\% \pm 0.3$ [57] and $78.0\% \pm 0.3$ [174]). Similarly, the final performance of our approach will increase by using multiple descriptors, better coding (i.e., soft quantization, sparse codes) schemes, etc.

For Pascal. Compared to the work of Van de Sande et al. [166] using the SIFT descriptor and a vocabulary of size $4k$, an improvement of 7.9% is obtained based on our *Generic SP*. We obtain an improvement of 10.4% based on the *Selective SP* approach. An improvement of 15.0% is obtained based on our *Generic SP* + *MKL* approach using the SIFT features.

	Method	MAP
Lazebnik et al.[97]	SP + SIFT, intersection kernel	54.5
Van de Sande et al. [166]	SP + SIFT, χ^2 kernel	55.8
Van de Sande et al. [166]	SP + C-SIFT, χ^2 kernel	56.6
This paper	Generic SP + SIFT, intersection kernel	58.8% (+7.9%)
This paper	Generic SP + C-SIFT, intersection kernel	**59.7**% (+5.5%)
This paper	Selective SP + SIFT, intersection kernel	60.2% (+10.4%)
This paper	Selective SP + C-SIFT, intersection kernel	**61.3**% (+8.3%)
This paper	Generic SP + *MKL* + SIFT	62.7% (+15.0%)
This paper	Generic SP + *MKL* + C-SIFT	**63.6%** (+12.4%)

Table 4.3: Comparison of our approach with state-of-art methods reported in literature on PASCAL VOC 2007 using a vocabulary of size 4000 (see text)

Compared with the best performing method using the C-SIFT descriptor (i.e., with a χ^2 kernel) in [166], we obtain a performance improvement of 5.5% based on our *Generic SP* approach. We also obtain an improvement of 12.4% based on the *Selective SP* approach. Finally, an improvement of 12.4% is obtained based on our *Generic SP* + *MKL* approach. Note that again better performance has been reported with multiple descriptor types (e.g., SIFT, opponentSIFT, rgSIFT, C-SIFT, RGB-SIFT with χ^2 kernel achieved 60.5% [166]). It should be noted that the final performance of our approach will benefit from using multiple descriptors as demonstrated in [166]. In summary, we demonstrated that our proposed approach outperforms the state-of-the-art methods on both Caltech-101 and Pascal 2007 data sets using only one type of feature.

4.5 Conclusion

Spatial Pyramids have been proposed which are steered by the 3D scene geometry present in images. The geometry of a scene is measured based on image statistics taken from a single image. After the estimation of the scene geometry of an image, the corresponding SP is selected as the geometrical representation.

From large scale experiments on the Pascal VOC 2007 and Caltech101, it can be derived that SPs which are obtained by search selective outperforms the standard SPs with 12.4% and 14.0% for Pascal VOC 2007 and Caltech101 respectively. The use of 3D scene geometry, to select the proper SP configuration, provides an even higher improvement of 16.0% and 19.7% respectively.

Chapter 5

Discriminative Compact Pyramids for Object and Scene Recognition

Spatial pyramids have been successfully applied to incorporating spatial information into bag-of-words based image representation. However, a major drawback is that it leads to high dimensional image representations.

In this work, we present a novel framework for obtaining compact pyramid representation. Firstly, we investigate the usage of the divisive information theoretic feature clustering (DITC) algorithm in creating a compact pyramid representation. In many cases this method allows to reduce the size of a high dimensional pyramid representation up to an order of magnitude with little or no loss in accuracy. Furthermore, comparison to clustering based on agglomerative information bottleneck (AIB) shows that our method obtains superior results at significantly lower computational costs. Moreover, we investigate the optimal combination of multiple features in the context of our compact pyramid representation. Finally, experiments show that the method can obtain state-of-the-art results on several challenging datasets.

5.1 Introduction

Bag-of-words based image representation is one of the most successful approaches for object and scene recognition [191, 32, 19, 1, 39, 47, 96, 112, 136, 166]. The first stage in the method involves selecting key points or regions followed by a suitable representation of these key points using robust local descriptors, like SIFT [104]. The descriptors are then vector quantized into a visual vocabulary, after which an image is represented as a histogram over visual words. The final representation lacks any spatial information since the location of the local features is ignored. This is generally considered as the foremost shortcoming of the standard bag-of-words representation.

Including spatial information into bag-of-words has therefore received considerable attention. The spatial pyramid scheme proposed by [97] is a simple and computationally efficient extension of an order-less bag-of-words image representation, as it captures the spatial information in such a way that traditional histogram-based image representations do not. This technique works by representing an image using multi-resolution histograms, which are obtained by repeatedly sub-dividing an image into increasingly finer sub-regions. The final representation is a concatenation of the histograms of all the regions. Many applications, such as classification and detection, [89, 44, 43, 184, 183] benefit form the spatial pyramid representation.

However, spatial pyramids have a major drawback due to the high dimensionality of the generated histograms while going towards the finest level of representation. This drawback is especially apparent for challenging data sets such as Pascal VOC where it is found that large size visual vocabularies generally improve the overall results. The combination of large vocabularies with spatial pyramids can easily lead to image representations as big as 160K words (e.g. [166]). If these large pyramid representation could be optimized for discrimination between different categories, a more compact representation would be sufficient. This will lead to compact yet efficient pyramid representations that have the advantages of the original pyramid representation [97] while avoiding their computational burden. This is precisely what we aim at, keeping in mind the constraint of reducing the size of the spatial pyramid representations while maintaining or even improving the performance.

Many recent works addressed the problem of compact vocabulary construction [181, 54, 94]. One popular strategy starts with a large vocabulary (e.g. generated by hierarchical k-means) and subsequently clusters these words together while intending to maintain the discriminative power of the original vocabulary [152, 38]. Slonim and Tishby [152] proposed a compression technique, denoted as Agglomerative Information Bottleneck (AIB), that constructs small and informative dictionaries by compressing larger vocabularies following the information bottleneck principle. Similarly, [54] proposed a fast implementation of the AIB algorithm and showed good performance for the construction of visual vocabularies. In this work, we will apply the theory and algorithms developed in these works, for the construction of compact discriminative spatial pyramids. These methods are especially appropriate due to the high dimensionality of the pyramid representation.

An additional advantage of compact pyramid representations is that it allows us to combine more features at the same memory usage for image representation. Combining multiple features especially color and shape has recently shown to provide excellent results [19, 1, 166, 57, 24, 179] on standard image classification data sets. The two main most common approaches to combine multiple features are early and late fusion. Early fusion based schemes combine features before the vocabulary construction phase. In case of late fusion separate visual vocabularies are constructed for each features. Subsequently, the bag-of-word representations (histograms) over the different vocabularies are concatenated. Both fusion approaches have been investigated within the context of standard bag-of-words. However, in the context of spatial pyramids, it is still uncertain which of the two fusion approaches is more beneficial. Therefore, in this work we investigate which fusion approach is more appropriate within the spatial pyramids framework.

In summary, the objective of this work is twofold. Firstly, we show that the AIB approach used to compress the vocabulary size significantly degrades accuracy when applied at spatial pyramids. To overcome this problem, we propose to use the Divisive Information

Theoretic Clustering (DITC) technique [38] that preserves the overall accuracy while reducing the dimensionality of the pyramid histogram significantly. Our results clearly suggest that pyramid compression based on the DITC approach provides superior results. Furthermore, DITC is computationally superior to AIB. Secondly, we evaluate the two existing fusion approaches for combining multiple features at the spatial pyramids level. We conclude that late fusion significantly outperforms early fusion based approaches in spatial pyramids. Finally, we combine both proposed contributions and obtain promising results on challenging data sets.

This chapter is organized as follows: next section describes the datasets used in the experiments. Section 5.3 discusses how AIB and DITC can be used for building compact pyramids. Subsequently, section 5.4 proposes both an early and a latefusion strategies for combining multiple features in the context of spatial pyramids. Section 5.5 compares our results with current state-of-the-art performance results. Finally, section 5.6 concludes this work and describes the most important lines of future research.

5.2 Datasets and Implementation Details

In this section we provide details about the datasets which will be used throughout this work, followed by the experimental setup employed to validate the two main contributions of our approach, namely the use of DITC for vocabulary compression and the use of early and late fusion in spatial pyramids. Fig. 5.1 shows some example images from the five data sets.

5.2.1 Data sets

For scene classification, the experiments are performed on Sports Events data set and 15 category Scenes data set. The Sports Events data set[101] contains 8 sports event categories collected from the Internet namely: bocce, croquet, polo, rowing, snowboarding, badminton, sailing, and rock climbing. The number of images in each category varies from 137 (bocce) to 250 (rowing). For each event class, 70 randomly selected images are used for training and 60 are chosen for testing.

The 15 class Scenes recognition data set [97] is composed of fifteen scene categories. Each category has 200 to 400 images. The major sources of the pictures in the data set include the COREL collection, personal photographs, and Google image search.

For object classification, the experiments are performed on Butterflies and Pascal VOC 2007 and 2009 data sets. The Butterflies data set consists of 619 images of seven classes of butterflies namely: Admiral, Swallowtail, Machaon, Monarch 1, Monarch 2, Peacock and Zebra. Finally, the experiments are also performed on PASCAL Visual Object Classes Challenge [43]. The PASCAL VOC Challenge 2007 data set consists of 9963 images of 20 different classes with 5011 training images and 4952 test images. The Pascal 2009 data set contains 13704 images of 20 different object categories with 7054 training images and 6650 test images.

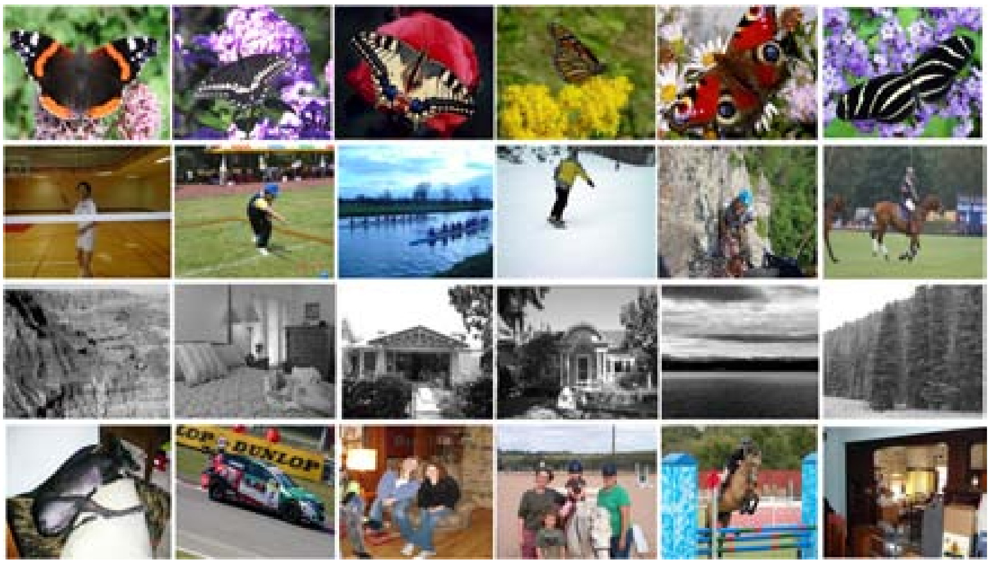


Figure 5.1: Example images from the data sets. From top to down: Butterflies, Sports Events, 15 class Scenes and PASCAL VOC data sets.

5.2.2 Implementation Details

We shortly discuss the implementation details we use for the bag-of-words based image classification. We apply a standard multiple-scale grid detector along with interest point detectors (Harris-Laplace and blob detector). In the feature extraction step, we use SIFT descriptor [104] for shape features, Color Names [169] descriptor for color features and the SelfSimilarity descriptor [145] to measure similarity based on matching the internal self-similarity. We use a standard K-means for constructing visual vocabularies. Finally we use a non-linear SVM with intersection kernel for classification as in [106].

5.2.3 Image Representation using Spatial Pyramids

Spatial pyramid scheme proposed by [97] have recently proven very successful results. These are formed by representing an image using weighted multi-resolution histograms, which are obtained by repeatedly sub-dividing an image into increasingly finer sub-regions by doubling the number of divisions in each axis direction and computing histograms of features over the resulting sub-regions. Resemblances found at finer resolutions are closer to each other in image space and are therefore more heavily weighted. To accomplish this, each level l is weighted to $l/2^{L-l}$, where L is the total number of pyramid levels considered. When histograms for all sub-regions at all levels have been created, these histograms are concatenated to form the final image representation. For example, a level 2 spatial pyramid is constructed by concatenating a total of $1 + 4 + 16 = 21$ histograms.

Although a notable performance gain is achieved by using the spatial pyramid method, the resulting histogram is often a magnitude higher in dimensionality over its standard bag-of-words based counterpart [1].

5.3 Compact Pyramid Representation

As discussed in the introduction, one of the main drawbacks of the spatial pyramid representation is its memory usage. We will discuss two existing approaches, namely AIB and DITC, which were shown to be successful for compact text document representation [152, 38]. Only AIB has been applied for compact image representation [54], and none of them has been studied in the context of spatial pyramids. We will give experimental results on the Sports Events [101] and 15 class scenes [97] data sets that will show that the constructed compact pyramid representation maintain the performance of their larger counterparts.

In practice the final size of the pyramid is dependent on the application, where users have to balance compactness versus classification accuracy. Depending on the task a smaller representation could be preferred over larger at the cost of performance (e.g. real-time object detection based on ESS [89, 90], or large scale image retrieval [132]. In the case that users do not want a drop in accuracy but do want to compress their representation, cross validation could be used to select the optimal cluster size. Throughout this work we consider that the final representation size is an input parameter to the compression algorithm.

5.3.1 Highly Informative Compact Spatial Pyramids

Let C be a discrete random variable that takes on values from the set of classes C=$\{c_1$,...,$c_l\}$ and let W be the random variable that ranges over the set of words W=$\{w_1$,...,$w_m\}$. It is important to note that we consider the number of words for the spatial pyramid representation to be equal to the number of words used for the visual vocabulary times the number of subregions in the spatial pyramid. For a level two pyramid constructed from a 1000 word vocabulary, this will lead to a final representation of $(1+4+16)\times 1000 = 21000$ words. We will consider clustering these 21000 words into a smaller set where each cluster represents words with similar discriminative power.

The joint distribution $p(C, W)$ is estimated from the training set by counting the number of occurrences of each visual word in each category. The information about C captured by W can be measured by the mutual information,

$$I(C,W) = \sum_i \sum_t p(c_i, w_t) log \frac{p(c_i, w_t)}{p(c_i)p(w_t)}, \tag{5.1}$$

which measures the amount of information that one random variable contains about the other. Ideally, in forming word clusters we aim at preserving the mutual information; however usu-

[1] The winners of Pascal VOC 2007 [107] showed that dividing an image horizontally 3×1 yields better performance than a conventional 4×4 structure. The resulting histogram is therefore reduced from vocabulary size $\times 21$ to vocabulary size $\times 8$

ally clustering lowers mutual information. Thus, we aim at finding word clusters that minimize the decrease in the mutual information:

$$I(C, W) - I(C, W^C). \tag{5.2}$$

where W^C are the word clusters $\{W_1, \ldots, W_k\}$. This can be rewritten as

$$\sum_i \sum_t \pi_t p(c_i|w_t) log \frac{p(c_i|w_t)}{p(c_i)} - \sum_i \sum_j \sum_{w_t \in W_j} \pi_t p(c_i|w_t) log \frac{p(c_i|W_j)}{p(c_i)} \tag{5.3}$$

where π_t is the prior of word, and is given by $\pi_t = p(w_t)$.

In the seminal work [38], Dhillon et al. prove that this is equal to

$$I(C, W) - I(C, W^C) = \sum_j \sum_{w_t \in W_j} \pi_t KL((p(C|w_t)), (p(C|W_j))) \tag{5.4}$$

where the Kullback-Leibler(KL) divergence is defined by

$$KL(p_1, p_2) = \sum_{x \in X} p_1(x) log \frac{p_1(x)}{p_2(x)}. \tag{5.5}$$

Eq. (5.4) is a global objective function that can be applied to measure the quality of word clustering. This object function states that we should group words w_t into clusters W_j, in such a way that the summed KL-divergence between the word distributions $p(C|w_t)$ and their cluster distributions $p(C|W_j)$ is as low as possible. Since the KL-divergence is a measure of similarity between distributions, we are clustering words together which contain similar information with respect to the classes as described in $p(C|w_t)$. Next we discuss two existing algorithms which aim to find the optimal clusters W_j as defined by Eq. (5.4).

AIB Compression [152]: AIB iteratively compresses the dictionary W by merging the visual words w_i and w_j that cause the smallest decrease in the mutual information given by Eq. (5.1). The decrease in the mutual information is monotonically reduced after each merge. Merging is iterated until one obtains the desired number of words. AIB is greedy in nature as it optimizes the merging of just two word clusters at every step (a local optimization) and thus the resulting algorithm does not directly optimize the global criteria defined in Eq. (5.4).

DITC Compression [38]: Other than AIB which iteratively reduces the number of words until then desired number of clusters is reached, DITC immediately clusters the words into the desired number of clusters (during initialization) after which it iteratively improves the quality of these clusters. Each iteration monotonically reduces the decline in mutual information as given by Eq. (5.4), therefore the algorithm is guaranteed to terminate at a local minimum in a finite number of iterations.

To optimize the global objective function of Eq. (5.4), DITC iteratively performs the following steps:

1. Compute the cluster distribution $p(C|W_j)$ according to:

$$p(C|W_j) = \sum_{w_t \in W_j} \frac{\pi_t}{\pi(W_j)} p(C|w_t), \tag{5.6}$$

where, $\pi(W_j) = \sum_{w_t \in W_j} \pi_t$.

2. Re-assign the words w_t to the clusters W_j based on their closeness in KL-divergence:

$$j^*(w_t) = argmin_j KL(p(C|w_t), p(C|Wj)) \tag{5.7}$$

where, $j^*(w_t)$ is new cluster index of the word w_t.

The initialization of the k clusters is obtained by first clustering the words into l clusters, where l is the number of classes. Every word w_t is then assigned to cluster W_j such that $p(c_j|w_t) = max_i\ p(c_i|w_t)$. This strategy guarantees that every word w_t is part of one of the clusters W_j. Subsequently we split each cluster arbitrarily into $\lfloor k/l \rfloor$ clusters. In the case that $l > k$ we further merge the l clusters to obtain k final clusters.

The basic implementation of the DITC algorithm [38] can result in a large number of empty clusters, especially for large vocabularies. To overcome this problem we propose a modified version of the basic DITC algorithm. At each iteration our algorithm retrieves the index e of the empty word clusters c_e, where $e \subset j$. Subsequently we assign at least one word w_t to each c_e. This is done using Eq. (5.7) by first assigning each word w_t to its closest word cluster c_j. Based on this assignment, we select that w_t with the maximum KL value returned by Eq. (5.7), i.e. that w_t found at the furthest distance from its currently assigned word cluster c_j. Then we reassign this w_t to c_e and remove it from c_j.

Comparing the computational cost of the two algorithms shows one of the advantages of DITC: AIB results in high computational cost of $O(m^3c)$ operations as it runs an agglomerative algorithm until k clusters are obtained. Here m is the total number of words and c is the number of classes in the data set. The fast implementation of the AIB costs $O(m^2c)$. On the other hand, the DITC algorithm requires Eq. (5.7) to be computed for every pair, $P(C|w_t)$ and $p(C|W_j)$ at a cost of $O(mkc\tau)$, where generally $k << m$. The number of required iterations τ to obtain convergence is typically around 15. We found DITC in practice to be computationally superior to AIB, obtaining a speedup between one or two orders of magnitude. On a typical run for obtaining 100 clusters from 20000 words on a data set with 15 classes, AIB (using [54]) took 14460 seconds while DITC converged in 234 seconds using a standard PC.

5.3.2 Experimental Results

In this section, we compare the two algorithms discussed above on the task of constructing compact spatial pyramids. To the best of our knowledge we are the first to apply DITC to visual word vocabulary construction. Lazebnik and Raginsky [94] propose a method for discriminative vocabulary construction which uses ideas of the theory of DITC [38]. However, the word clusters where restricted to lie in Voronoi cells, whereas in the original algorithm words are clustered without restrictions on their location in feature space, and thus allowing for multi model distributions. We show that the pyramid compression based on DITC has a lower loss of discriminative power, and is computationally more efficient compared to compression based on AIB [54].

Method	Level	Size	Sports Events	15 class Scenes
$Pyramid$	2	21000	83.8	84.1
$Pyramid_{AIB}$	2	5000	81.5	81.7
$Pyramid_{AIB}$	2	1000	79.8	80.4
$Pyramid_{AIB}$	2	500	78.8	78.3

Table 5.1: Classification Score (percentage) on both the Sports Events and 15 class Scenes Data sets. The results demonstrates that by applying the AIB compression [54] a considerable loss in performance occurred for compact vocabularies.

Method	Level	Size	Sports Events	15 class Scenes
$Pyramid$	2	21000	83.8	84.1
$Pyramid_{DITC}$	2	5000	84.2	85.4
$Pyramid_{DITC}$	2	1000	85.6	84.4
$Pyramid_{DITC}$	2	500	84.6	84.2

Table 5.2: Classification Score (percentage) on both the Sports Events and 15 class Scenes data sets. The results demonstrates that DITC successfully compresses the vocabularies while preserving their discriminative power.

Table 5.1 shows numerical results obtained by applying AIB on both the Sports Events and 15 Scenes data sets for different sizes. We started by using vocabulary of size 1000 for constructing a three level pyramid of 21000 dimensionality, after which we compress this vocabulary to a dimensionality of 5000, 1000 and 500. We can notice that by applying AIB compression on the pyramids the performance drops significantly, especially when we are going towards lower dimensionality. We attribute this to the fact that the information bottleneck technique is agglomerative in nature and result in a sub-optimal word clusters because it greedily merges just two word clusters at every step and it does not directly optimize the global objective function of Eq. (5.4).

Table 5.2 shows the results obtained using DITC. The main observation is that the DITC approach succeeds in conserving the discriminative power while reducing dimensionality of the image representation. Furthermore, for both sets reducing the dimensionality leads to an improvement of the classification score, and even at the smallest dimensionality of 500 similar results are obtained as with the total 21000 word vocabulary.

Classification accuracies of both compression approaches are shown Figure 5.2 which supports the two main conclusions: first, using DITC compression mechanism leads to a compact pyramid representation that reduces the dimensionality of the original pyramid yet preserves or even improves its performance. Second, compact pyramid representation based on DITC achieve better results than those based on AIB approaches at all the vocabulary sizes. Moreover the performance gain is more significant for smaller vocabularies.

We also perform experiments comparing the performance of DITC compression with Principle Component Analysis (PCA) and Partial Least Square (PLS) techniques. Figure 5.3 shows the comparison on two data sets. We only show the performance for very compact pyramid representations, since PLS is known to obtain better results for compact representa-

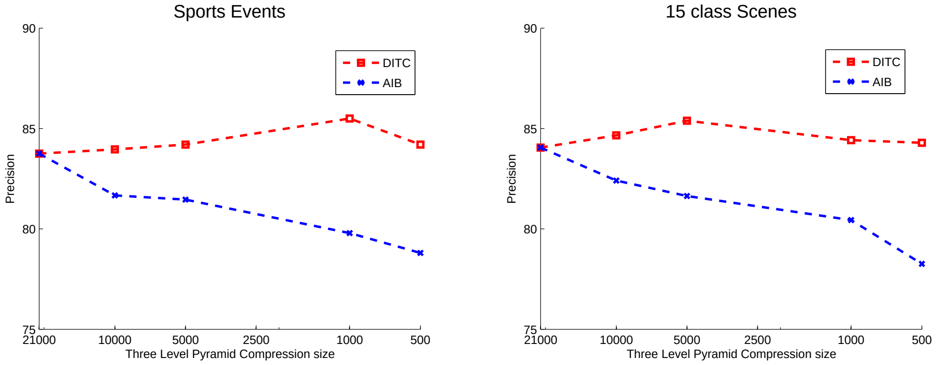


Figure 5.2: Sports Events data set (left) and 15 class Scenes data set (right) classification accuracy for compressing the whole pyramid representation leading to a more compact pyramid representation using the two compression approaches considered namely: DITC vs. AIB.

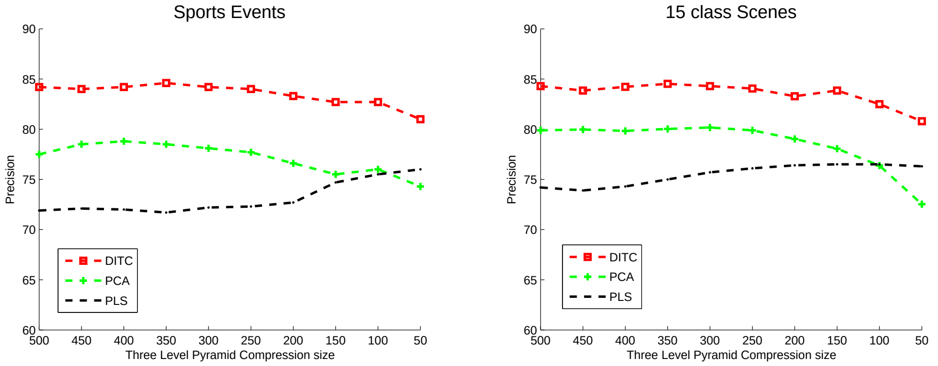


Figure 5.3: Sports Events data set (left) and 15 class Scenes data set (right) classification accuracy for compressing the whole pyramid to a compact representation using approaches namely: DITC, PLS and PCA. Note that DITC based compression also provides superior performance for very compact pyramid representations.

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
Pyramid	72.1	54.9	41.9	62.6	23.9	46.3	71.4	51.4	48.8	37.4	46.8	38.9	72.1	58.1	80.3	25.4	32.4	41	70.5	43.6	50.9
AIB	53.2	28.3	24.6	43.2	11.4	27.5	54.2	29.9	35.6	11.1	13.9	21.1	41.3	32.3	73.3	10.4	13.9	27.9	40.2	27.8	31.1
DITC	61.4	50.6	36.5	49.1	20.3	43.9	68.2	44.1	47.1	29.7	38.8	33.4	69.5	53.6	78.9	23.6	22.9	37.6	64.3	42.3	45.8

Table 5.3: Average-Precision Results for all classes of the PASCAL VOC 2007 database. Comparison on the average accuracy of the original four level pyramid representation of size 25500 compressed to size 200. The second row shows the compression results using the AIB [54] and the third row shows the results using DITC [38].

tion and quickly deteriorates for larger representation. Moreover, the number of dimensions of PCA is bounded by the number of observations. DITC based pyramid compression consistently outperforms the other two compression technqiues. It is worthy to mention that DITC also provides better performance compared to both PCA and PLS with a very small compact pyramid representations (50 bins).

The performance difference between DITC and AIB becomes especially apparent for high compression. An initial pyramid representation of the PASCAL dataset of 25500 words is compressed to 200 clusters. Table 5.3 shows a 14% higher Mean Average-Precision for having compact pyramid representations based on DITC compared to those obtained using AIB on object recognition.

5.3.3 Compact Pyramid Designs

As demonstrated in the last section, we can significantly reduce the dimensionality while preserving or even improving the performance of the original pyramid representation that we started with. We next evaluate and compare two different design strategies for building our final compact pyramid representations. The main aim is to find an optimal design for obtaining compact yet efficient pyramids based on the DITC compression algorithm. The two proposed designs are the following:

1. Compute a vocabulary, compress it using DITC and subsequently build a compact pyramid representation based on the compressed compact vocabulary (schema denoted as *CompPyr* hereafter).

2. Construct pyramid representation for an image and subsequently compress the whole pyramid directly using DITC (strategy denoted as *PyrComp* hereafter).

Table 5.4 shows the results obtained using both of the considered proposed designs on 15 class Scenes and the Sports Events datasets. To compare the classification scores obtained

Method	Level	Size	Sports Events	15 class Scenes
$Pyramid$	2	21000	83.8	84.1
$Pyramid_{AIB}$	2	1000	79.8	80.4
CompPyr	2	1000	81.9	82.1
PyrComp	2	1000	**85.6**	**84.4**

Table 5.4: Classification score on the Sports Events and 15 class Scenes datasets using the DITC approach comparing the two proposed designs: *CompPyr* (compute a vocabulary, compress it, and then build a compact pyramid representation using this compressed compact vocabulary) and *PyrComp* (i.e. construct a pyramid representation for an image, then compress the whole pyramid afterwards).

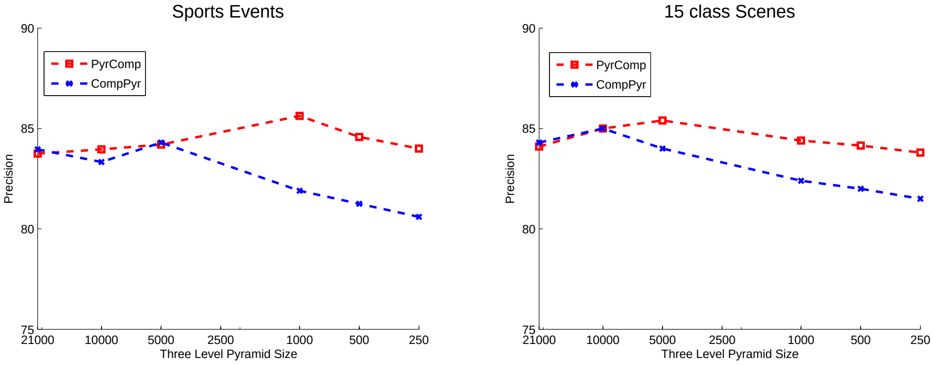


Figure 5.4: Classification comparison between *PyrComp* and *CompPyr* strategies for (left) 15 class Scenes and (right) Sports Events datasets.

from the two designs, we consider the same dimensionality of size 1000. For the 15 class Scenes data set, using *CompPyr* we got a score of 82.1%, while *PyrComp* gives us a performance of 84.4%. For the Sports Events data set, we observe a similar gain in the obtained results.

These quantitative results suggest how optimal compact pyramid representations can be built: although both designs preserve the accuracy of the original pyramid representation, the best results are obtained following the *PyrComp* strategy, since it does not only preserve the original pyramid performance, but slightly improves performance. Additionally Figure 5.4 illustrates another interesting conclusion: the gain in performance using *PyrComp* is obtained throughout all sizes, and this gain is more significant at lower sizes.

The *CompPyr* compresses the vocabulary while ignoring the spatial pyramid image representation to which it will later be applied. This strategy is used by most existing methods for compact vocabulary construction [94, 187, 105]. Our experiment show that compressing the vocabulary within the spatial pyramid, significantly improves results. Compression with *PyrComp* has the same freedom as *CompPyr* to merge words within a sub-window. Additionally, it can also merge words of different sub-windows, something which is impossible within the *CompPyr* strategy.

5.4 Combining Multiple Features in Spatial Pyramids

In the previous section, we have provided an efficient method for the construction of compact pyramid representations. The gained compactness allows us to combine more features at the same memory usage of the image representation. Here we analyze how to optimally combine multiple features in a pyramid representation.

We will look at the particular case of combining color and shape, which was shown to provide excellent results for object and scene recognition [44]. In particular we investigate two approaches to combine multiple features, namely the early and late fusion schemes. In the next section we provide results from combining visual cues other than color and shape.

5.4.1 Early and Late Fusion Spatial Pyramid Matching

In early fusion the local features of color and shape are concatenated into a single feature. Subsequently, the combined color and shape features are quantized into a joint shape-color vocabulary. In general, early fusion results in vocabularies with high discriminative power, since the visual-words describe both color and shape jointly, allowing for the description of blue corners, red blobs, etc. A significant shortcoming of early fusion approach is that it deteriorates for categories which vary significantly over one of the visual cues. For example, man made categories such as cars and chairs which vary considerably in color. In such cases, the visual-words will be contaminated by the ”irrelevant” color information. The relevant shape words will be spread over multiple visual-words, thereby complicating the task of the learning algorithm significantly. On the other hand, early fusion is suitable for categories which are constant over both color and shape cues like plants, lions, road-side signs etc.

The second approach, called late fusion, fuses the two cues, color and shape, by processing the two features independent of each other. Separate visual vocabularies are constructed for color and shape independently, and the image is represented as a distribution over shape-words and color-words. A significant drawback of late fusion is that we can no longer be certain that both visual cues come from the same location in an image. Late fusion is expected to provide better results over early fusion on categories where one cue is constant and the other varies considerably. Example of such categories are man made objects such as car, buses and chairs etc.

Typically within the bag-of-words framework a number of local features f^c_{mn}, m=1...M^n are extracted from training images I_n. Where $n = 1, 2, ..., N$, and $c \in \{1, 2\}$ is an index indicating the different visual features. In case of early fusion, two visual features are concatenated according to :

$$f^{1\&2}_{mn} = \left(\beta \ f^1_{mn}, (1-\beta) f^2_{mn}\right) \tag{5.8}$$

Vector quantization of f^1, f^2, $f^{1\&2}$ yields the corresponding vocabularies V_1, V_2, $V_{1\&2}$. We define $h^V(I)$ to be the histogram representation of image I in vocabulary V. Early fusion representation of the image is given by $h^{V_{1\&2}}(I)$ and the late fusion is obtained by concatenating the separate histograms:

$$h^{(V_1,V_2)}(I) = \left[\beta \ h^{V_1}(I), (1-\beta) \ h^{V_2}(I)\right] \tag{5.9}$$

Note that we have introduced a weight parameter β for both early and late fusion which allows us to leverage the relative weight of the various cues. In our setting this parameter is learned through cross-validation on the training data. Both fusion schemes can easily be extended to accommodate several visual cues.

Before applying the two schemes on spatial pyramids, we will shortly discuss the relation of existing approaches for the combination of multiple features to early and late fusion. Bosch et al. [19] computes the SIFT descriptor on the H,S,V channels and then concatenates the final descriptor into a single representation. Van de Weijer and Schmid [179] compare photometrically invariant representations in combination with SIFT for object recognition. Recently, Van de Sande et al. [166] performed a study on the photometric properties of many color descriptors, and did an extensive performance evaluation. In their evaluation OpponentSIFT was shown to be the best choice to combine color and shape features. Since in all these works color and shape are combined before vocabulary construction, they are considered early fusion methods.

Regarding late fusion, several methods explore the combination of multiple features at the classification stage. These approaches, of which multiple kernel learning MKL is the most well-known, [5, 18, 56, 137, 171] combine kernel combinations of different visual features. A weighted linear combination of kernels is employed, where each feature is represented by multiple kernels. Beside the multiple kernel learning approach, the two conventional approaches that combine different kernels at the classification stage in a specified deterministic way are *averaging* and *multiplying* the different kernel responses. Surprisingly, the product of different kernel responses is shown to provide similar or even better results than MKL in a recent study performed by Gehler and Nowozin [57]. These approaches are considered as late fusion since they perform vocabulary construction separately for the different features. Recently, an alternative method for combining color and shape, called color attention, was proposed by Khan et al. [84]. However, it is unclear how this method can be extended to incorporate spatial pyramids, since the normalization performed in the sub-regions of the pyramid counters the color attention weighting.

For the standard bag-of-features image representation there is no consensus whether early or late fusion is better. Here we investigate the two approaches in the context of spatial pyramids. The common methodology employed in current object recognition frameworks is to build spatial pyramids of early fusion based schemes (such as Opp-SIFT, C-SIFT, HSV-SIFT etc.) [19, 166, 179]. We refer to these spatial pyramids that are based on early fusion scheme as *early fusion spatial pyramids* and the spatial pyramids that are based on late fusion as a *late fusion spatial pyramids*. Figure 5.5 highlights the two spatial pyramid matching approaches.

5.4.2 Experimental Results of Early and Late Fusion based Spatial Pyramids

To evaluate both early and late fusion spatial pyramids, we perform an experiment for both object and scene recognition. For scene classification, the experiments are performed on Sports Events data set. We use the Butterflies data set for the object recognition task. To con-

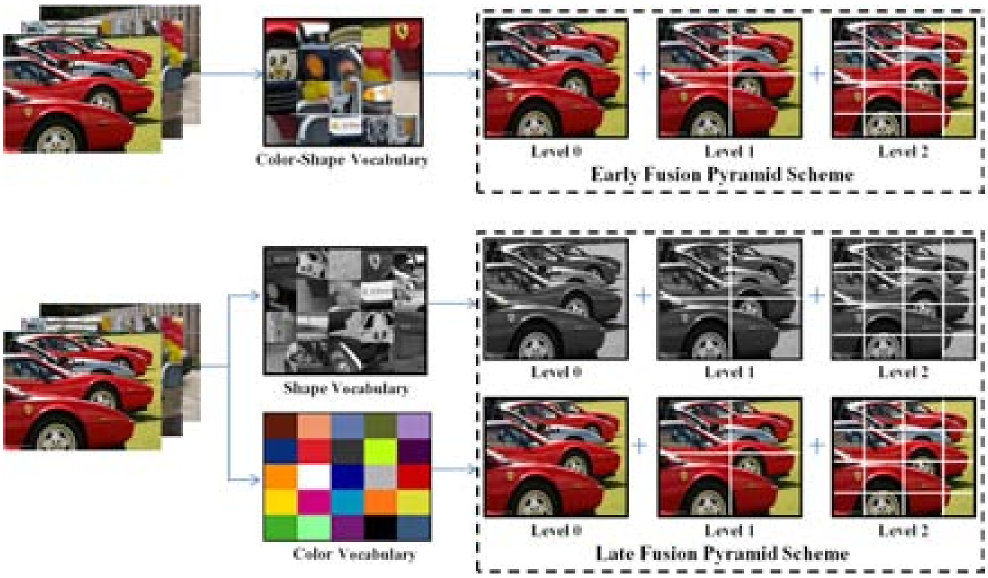


Figure 5.5: Early and Late fusion pyramid schemes. In the early fusion pyramid scheme a combined color-shape vocabulary is constructed as a result of which a single pyramid representation is obtained. To construct a late fusion pyramid, a separate vocabulary is constructed for color and shape and spatial pyramids are obtained for each cue. We show that late fusion is the recommended approach for combining multiple features.

struct a shape vocabulary we use the SIFT descriptor and the Color Names descriptor [169] for creating a color vocabulary. We combine the two cues based on early fusion and late fusion schemes, both at the standard bag-of-words level and at the spatial pyramids level. We also compare with OpponentSIFT which was shown to be the best color-shape descriptor in a recent evaluation [166]. Table 5.5 shows the results obtained on Sports Events data set. For this data set, shape is an important cue and color plays a subordinate role. At the standard bag-of-words level, OpponentSIFT provides the best results but as we move into higher levels of spatial pyramids the performance of both early fusion and OpponentSIFT starts to degrade (the performance of OpponentSIFT at finest pyramid level is below its performance at the standard bag-of-words level). We also combined color and shape at the kernel level with the product rule as advocated by Gehler [57]. However, results were found to be inferior compared to the late fusion spatial pyramid scheme.

Table 5.6 shows the results obtained on Butterflies data set. Shape plays an important role as depicted from the results of individual visual cues. Late fusion provides better results at the standard bag-of-words level than both early fusion and OpponentSIFT. The performance gain of late fusion is further increasing when more pyramid levels are considered.

In conclusion, in a standard bag-of-words representation both early and late fusion obtain comparative results. However, our experiments show that within a spatial pyramid representation late fusion significantly outperforms early fusion.

Method	Level	Size	Score
$Shape$	0	800	80.6
$Color$	0	300	53.9
$Opp-SIFT$	0	1100	**82.9**
$EarlyFusion$	0	1100	80.6
$LateFusion$	0	1100	81.8
$Opp-SIFT$	1	5500	82.3
$EarlyFusion$	1	5500	80.8
$LateFusion$	1	5500	**82.7**
$Opp-SIFT$	2	23100	80.8
$Earlyfusion$	2	23100	82.7
$Latefusion$	2	23100	**84.4**

Table 5.5: Classification Score (percentage) on Sports Events Data set.

Method	Level	Size	Score
$Shape$	0	1000	79.4
$Color$	0	300	53.3
$Opp-SIFT$	0	1500	78.7
$EarlyFusion$	0	1500	79.6
$LateFusion$	0	1300	**81.9**
$Opp-SIFT$	1	7500	79.6
$EarlyFusion$	1	7500	81.7
$LateFusion$	1	6500	**84.4**
$Opp-SIFT$	2	31500	81.0
$Earlyfusion$	2	31500	83.3
$Latefusion$	2	27300	**87.9**

Table 5.6: Classification Score (percentage) on Butterflies Data set.

5.5 Comparison to State-of-the-Art

In the previous section we have investigated how to optimally compute compact and multi-feature spatial pyramids. We have shown that optimal results are obtained by using DITC algorithm for compression, and using the *PyrComp* strategy for the computation of compact pyramids. Furthermore, as demonstrated in the previous section, late fusion pyramids is shown to be more efficient than early fusion pyramids. In this section, we combine these conclusions to construct compact multi-feature spatial pyramids. First we compute compact spatial pyramids for each feature separately and then combine them in a late fusion manner.

We denote our pyramid representation for SIFT with PS, and the compact pyramids of SIFT, SelfSimilarity and Color with PS_C,PSS_C and PC_C respectively. We report the final results on all the four challenging data sets obtaining very good classification scores even when reducing the pyramid histograms significantly. In addition, we compare our results with several recent results reported on these data sets in literature. Table 5.7 shows our final results and a comparison with the best results reported on the four data sets.

For the **Sports Events data set** experiments are repeated five times by splitting the data set into train and test set and the mean average accuracy is reported. As depicted from the results, each feature's compact representation preserves or even improves the performance over its original pyramid histogram. The original three level pyramid representation of SIFT (PSIFT) with dimensionality 21000 gives accuracy of 83.8 while, compressing it to 1000 we improve the score to 85.6. By combining the three compact pyramid representations we obtained a classification score of 87.1 which exceeds the state-of-the-art results obtained on this data set [183, 184, 20, 185, 178]. The final accuracy is obtained with our compact histogram of dimensionality 2000 reduced from the original pyramid histograms of dimensionality 42000.

For the **15 category Scenes data set**, we followed the standard protocol of splitting the data set in to training and testing 5 times and reported the mean classification score. The results of each feature compact pyramid representation preserves or even improves the performance of its original pyramid representation. The original three level pyramid structure of SIFT (PS) with dimensionality 21000 gives accuracy of 84.1 while, compressing it to 1000 we improve the score further to 84.4. Since there is no color in this data set, we only combine the compact pyramids obtained from SIFT and SelfSimilarity. Our final compact representation has a histogram of size 2000 reduced from original pyramid histograms having dimensionality of 42000. We obtained a classification accuracy of 86.7 which is to the best of our knowledge the best performance on this data set [183, 184, 20, 185, 178].

The **Butterflies data set** shows our approach on a object recognition data set. Our compact pyramid representation of SIFT provides comparable results w.r.t. the original pyramids of SIFT. Our final combination yields a score of 91.4 which outperforms the best reported result in [93].

The results on the **Pascal VOC 2007** show we reduce the pyramid histogram of SIFT to one third with a small loss. The final mean average precision of 59.5 is obtained with a histogram size of $25K$. Our final results are close to state-of-the-art, but we have significantly reduced the histogram dimension ($25K$) compared to the approach of Van de Sande [166], where SIFT pyramids are combined with 4 ColorSIFT pyramids, leading to higher histogram dimensions of $160K$. Lastly, it should be noted that better results (63.5) were reported in [68], where authors include additional information of object bounding boxes from object detection to improve image classification.

For the **Pascal VOC 2009**, similar behavior is noticed. Hence, with an original SIFT pyramid of size $84K$ a mean average score of 55.7 is obtained. However, we maintained a score of 55.2 using our $15K$ compact SIFT representation. Finally, the results for multiple features fusion improve the overall mean average score up to 57.6 over the compact SIFT features.

Data Sets	Best Score		PS		PS_C		$PS_C + PC_C + PSS_C$	
	Size	**Score**	Size	**Score**	Size	**Score**	Size	**Score**
Sports	6K	84.2 [183]	21K	83.8	1K	85.6	2K	**87.1**
15 Scenes	21K	84.3 [20]	21K	84.1	1K	84.4	2K	**86.7**
Butterflies	2K	90.6 [93]	21K	89.5	1K	89.0	2K	**91.4**
Pascal 2007	160K	**60.5** [166]	84K	57.4	15K	57.2	25K	59.5
Pascal 2009	4194K	**64.6** [188]	84K	55.7	15K	55.2	25K	57.6

Table 5.7: Classification Score (percentage) on Sports Events, 15 class Scenes, Butterflies, Pascal VOC 2007 and 2009 Data sets.

5.6 Conclusions

A major drawback of spatial pyramids is the high dimensionality of their image representation. In this work we have proposed a method for the computation of compact discriminative pyramids. The method is based on the divisive information theoretic feature clustering algorithm, which clusters words based on their discriminative power. We show that this method outperforms clustering based on the agglomerative information bottleneck both in accuracy and in computational complexity. Results show that depending on the data set dimensionality reductions up to an order of magnitude are feasible without a drop in performance. The gained compactness leaves more room for the combination of features. We investigate the optimal strategy to combine multiple features in a spatial pyramid setting. Especially for higher level pyramids late fusion was found to significantly outperform early fusion pyramids. We evaluated the proposed framework on both scene and object recognition, and obtained state-of-the-art results on several benchmark data sets.

For future work we are particularly interested in applying the compact pyramids to the task of bag-of-words based object detection [89, 68]. The application of bag-of-words based detection has been particularly advanced due to the efficient sub-window search (ESS) algorithm proposed by Lampert et al. [89]. The usage of compact discriminative pyramids to this application could help obtain faster detection methods without loss in accuracy.

Another line of future research includes investigating the application of DITC to sparse image representation [187, 105], which has been shown excellent results in a recent work [82]. Although discriminative vocabularies within the context of sparse image representation have been investigated, these methods still ignore the spatial pyramid for the construction of discriminative vocabularies, whereas our work shows that compressing the vocabulary within the spatial pyramid significantly improves results. Therefore, we expect that combining the strengths of both methods will lead to further improvements.

Chapter 6

Color Constancy using 3D Scene Geometry

The aim of color constancy is to remove the effect of the color of the light source. As color constancy is inherently an ill-posed problem, most of the existing color constancy algorithms are based on specific imaging assumptions (e.g. grey-world and white patch assumption).

In this paper, we propose a method to combine color constancy algorithms by investigating the relation between scene depth, local image statistics and color constancy. The aim is to compute the scene layout first to obtain the depth layers. Then, image statistics are exploited (per layer/depth) to select the proper color constancy method. Our approach enables the estimation of multiple illuminants by distinguishing nearby light source from distant illuminants.

Experiments on large scale image data sets show that the proposed algorithm outperforms state-of-the-art single color constancy algorithms with an improvement of almost 50% of median angular error. When using a perfect classifier (i.e, all of the test images are correctly classified into stages), the performance of the proposed method achieves an improvement of 52% of median angular error compared to the best-performing single color constancy algorithm.

6.1 Introduction

The color of objects is largely dependent on the color of the light source. Therefore, the same object recorded by the same camera but under different illumination conditions may vary in its measured color appearance. This color variation may negatively affect the result of subsequent image and video processing methods for different applications such as object recognition, tracking and surveillance. The aim of color constancy is to remove the effect of the color of the light source. A considerable number of color constancy algorithms have been proposed, see [40, 78, 63] for reviews. Traditionally, pixel values are exploited to estimate the illuminant. Examples of such methods include approaches based on low-level features

[28], gamut-based algorithms [49], and other methods using a learning phase [63]. Recently, methods that use derivatives (*i.e* edges) and even higher-order statistics have been proposed [167].

In general, color constancy algorithms are based on specific assumptions about the illumination or properties of object reflectance. As a consequence, none of them can be considered as universal. Therefore, different methods have been proposed to select or combine different color constancy methods. Higher level visual information is taken into account only recently [167, 14, 61]. In [167], the image is modeled as a mixture of semantic classes, such as sky, grass, road and buildings. Illuminant estimation is steered by different classes by evaluating the likelihood of the semantic content. Similarly, indoor-outdoor image information is used in [14]. Alternatively, image statistics are used in [61] to select the most useful color constancy method. It is shown that images with similar image statistics will select the same color constancy algorithms.

In contrast to previous work, our contribution is to exploit the relation between *depth* and *color constancy*. Image statistics are influenced by depth patterns [**?**], e.g. the signal-to-noise ratio generally decreases as the depth increases [75] while the scale changes when viewing scenes from different depths [161]. Attributes like signal-to-noise and scale are inherently correlated with color constancy methods as they strongly influence the accuracy of the illuminant estimates [61].

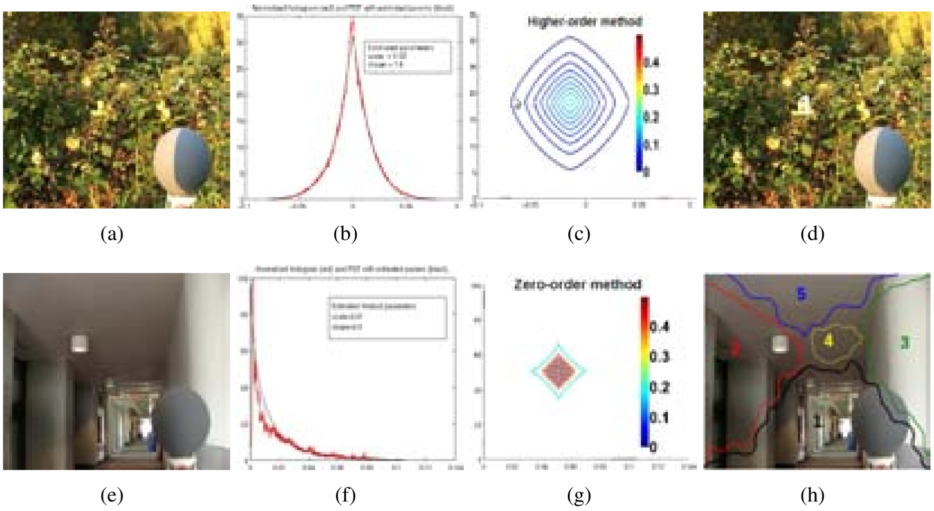
Therefore, in this paper, the relationship between depth, local statistics and color constancy algorithms is investigated. The aim is to compute the $3D$ scene geometry from a single image. In this way, the depth layers are obtained. Then, image statistics are exploited (per layer/depth) to select the proper color constancy method. Color constancy will be applied per depth layer allowing multiple illuminants per scene. The only assumption is that a single light source is illuminating the scene part at a certain distance (layer).

This paper is organized as follows, First, in sections 6.2 and 6.3, we give the motivation of our approach and briefly outline the color constancy framework and $3D$ scene geometry estimation. Then, the proposed method is described in section 6.4. After that, the experimental setup and results are presented in section 6.5. Finally, section 6.6 concludes this paper.

6.2 Motivation

The lighting conditions in a scene may differ at varying depth layers. For example, nearby objects may be illuminated by a different light source than distant ones. If we can derive the scene geometry, this will result in important depth cues for color constancy such as the ordering of scene elements determined by their relative depth order. In this section, we motivate our approach by inferring a relation between the depth of a scene and color constancy.

Image statistics and scene depth: The relation between depth patterns and natural image statistics is studied in [161, 175]. They show that in case of a dominant structure (object or background) gradient histograms correspond to a decaying power-law distribution. When depth increases, the size of objects will decrease showing less texture. Hence, each object



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

(e) (f) (g) (h)

Figure 6.1: First column shows examples of textured samples at different scales (depths). In the second column, we show the corresponding Weibull distribution parameters. Higher values of the shape parameter γ is equivalent to normal distribution, while lower values (i.e., γ=1) is equivalent to exponential distribution as shown in [60]. Third column, shows the edge distribution (approximated by a Weibull fit) that can be considered to be characteristic for the corresponding color constancy algorithm. The last column shows the depth layers indicated by numbers: (d) is an example for the *ground* category, where the depth is small and (h) is an example of the *box* category, with a larger depth (i.e., five segments). The images come from the data set published in [31].

distance is associated with a different power-law.

In this paper, to model natural image statistics, the integrated Weibull distribution is taken as the representation of a decaying power-law [60]:

$$\omega(x) = C \exp(-\frac{1}{\gamma}|\frac{x}{\beta}|^{\gamma}) \tag{6.1}$$

where x is the edge responses in a single color channel of a Gaussian derivative filter, C is a normalization constant, β is the scale parameter (i.e. the width) of the distribution and γ is the shape parameter (i.e. the peakedness) of the distribution. The parameters of this distribution are indicative for the edge statistics of a scene [60]. In general, the Weibull parameter β encodes image (edge) contrast, and γ corresponds to the grain size and is related to the amount of texture. A higher value for β indicates more contrast, while a higher value for γ indicates a smaller grain size (more fine texture). For example, $\gamma = 2$ the Weibull distribution is equivalent to the normal distribution and for $\gamma = 1$ it is a double exponential. The exponent δ indicates the fractal dimension.

When the depth of a scene becomes smaller, object surfaces will become larger and coarser showing more contrasting details. In this case, natural image statistics computed from the image follow a Weibull distribution with increasing β and γ [175]. In other words, more contrasting details and texture patterns are reflected by their corresponding β and γ values. When the depth of a scene becomes larger, object surfaces will become smaller and will contain fewer details and therefore become smoother. Hence, scene elements appear increasingly fuzzier with depth. With varying depth, image statistics follow a Weibull distribution with associated β and γ values. Hence, a relation exists between natural image statistics and depth patterns of a scene [161, 175]. Further, these observations are coherent with [60] showing that for scenes with smaller depths, higher values β and γ are obtained, reflecting more contrasting details and texture patterns.

Figure 6.1 shows examples of textured samples (first column) at different scales *i.e. depths*, together with their Weibull distributions (second column). The estimated parameters of the Weibull distribution obtained for image 6.1(a) are: *scale (β)* $= 0.02$, *shape (γ)* $= 1.4$ and the shape of the Weibull distribution is similar to a normal distribution, see fig. 6.1(b). On the other hand, for image 6.1(e), we obtained *scale* $= 0.01$, *shape* $= 0.8$ and the shape of the Weibull distribution corresponds to a double exponential curve, as shown in fig. 6.1(f).

Image statistics and color constancy: It has been shown in [62], that natural image statistics, represented by Weibull distributions, are useful to select the proper color constancy method. It is derived that if the image contains a limited number of edges, pixel-based color constancy is preferred. In case of sufficient edges (e.g., more than eight different edges), edge-based color constancy is preferred. The order of the best performance in terms of the number of edges (from low to high) is as follows: zeroth-order statistics first, followed by first-order and second-order statistics. If the image contains contrasted edges (i.e. β and γ are small), edge-based color constancy is used, otherwise pixel-based color constancy is preferred. Hence, a relation exists between image statistics (Weibull parameterization) and color constancy where contrast β and grain size (γ) are related to the number of edges, amount of texture, and signal-to-noise ratio to which the used color constancy methods are sensitive.

In figure 6.1, example images are shown with corresponding edge distribution which are approximated by a Weibull-fit (third column). The intensity channel is chosen for the ease of illustration. The images are examples on which the different color constancy algorithms perform best (i.e. pixel-values, higher-order methods). The relation between the images in figure 6.1 and their corresponding color constancy algorithm becomes clear from the edge distributions (third column) that are shown together with the images in figure 6.1. Pixel-based algorithms (i.e. 0^{th}-order) perform better than higher-order methods (i.e. 1^{st} and 2^{nd}-order) on images with only little texture (*i.e.* box scene images). This is reflected by an edge distribution that is densely sampled around the origin, *i.e.* many edges with little or zero energy. Higher-order methods require more edge information for an accurate illuminant estimate, which is reflected by an edge distribution that is less sharply peaked.

Depth and color constancy: In contrast to previous work, the contribution of this paper is to exploit the relation between depth and color constancy. Contrasted details (edges) are common for close-by objects (edge-based color constancy is preferred) than distant objects (pixel-based color constancy is preferred). This novelty is used to improve color constancy: by inferring the scene geometry and consequently the depth of each layer, image statistics are exploited (per layer/depth) to select the proper color constancy method. Further, scene layers at different distances may be illuminated by different light sources. Color constancy will be applied per layer allowing multiple illuminants per scene. The only assumption is that a single light source is illuminating the scene part at a certain distance (layer).

6.3 Preliminaries

In this section, we briefly discuss the computational methods to estimate the illuminant and to determine scene geometries (stages).

6.3.1 Color Constancy

Assuming Lambertian reflection, the image color $\mathbf{f} = (R, G, B)^T$ depends on the color of the light source $e(\lambda)$, the surface reflection $s(\mathbf{x}, \lambda)$ and the camera sensitivity function $\mathbf{c}(\lambda)$:

$$\mathbf{f}(\mathbf{x}) = \int_{\omega} e(\lambda)\mathbf{c}(\lambda)s(\mathbf{x}, \lambda)d\lambda \ , \tag{6.2}$$

where ω is the visible spectrum, λ is the wavelength of the light and $\mathbf{x}$ is the spatial coordinate. Under the assumption that the recorded color of the light source $\mathbf{e}$ depends on the color of the light source $e(\lambda)$ and the camera sensitivity function $\mathbf{c}(\lambda)$, the color of the light source is estimated by

$$\mathbf{e} = \int_{\omega} e(\lambda)\, \mathbf{c}(\lambda)\, d\lambda \ . \tag{6.3}$$

Since both $e(\lambda)$ and $\mathbf{c}(\lambda)$ are unknown, color constancy is an under-constrained problem. Therefore, in order to solve the color constancy problem, a number of assumptions are made such as the Grey-World assumption (i.e. the average pixel value is grey) and the White-Patch assumption (i.e. the maximum pixel value is white) [40].

To incorporate both pixel values and higher-order derivative information, in this paper, the following color constancy framework is used [167],

$$\left(\int \left| \frac{\partial^n \mathbf{f}^{\sigma}(\mathbf{x})}{\partial \mathbf{x}^n} \right|^p d\mathbf{x} \right)^{\frac{1}{p}} = k\, \mathbf{e}^{n,p,\sigma}, \tag{6.4}$$

where n is the order of the derivative, p is the Minkowski-norm and $\mathbf{f}^{\sigma}(\mathbf{x}) = \mathbf{f} \otimes G^{\sigma}$ is the convolution of the image with a Gaussian filter with scale parameter σ. Using Equation 6.4, different color constancy algorithms are generated by varying one or more of parameter values. For example,

1. when n=0, pixel-based color constancy algorithms are obtained, such as the Grey-World algorithm ($\mathbf{e}^{0,1,0}$), the White-Patch algorithm ($\mathbf{e}^{0,-1,0}$) and the general Grey-World ($\mathbf{e}^{0,13,2}$);
2. when n=1, color constancy algorithms are obtained using first-order derivative information, i.e. image edges information. The Minkowski-norm p and smoothing parameter σ depend on each specific data set. The instantiation $\mathbf{e}^{1,1,6}$ is applied in this paper;

3. when n=2, the framework provides color constancy methods based on second-order statistics. Similarly, the other two parameters p and σ vary with the data set. We use $\mathbf{e}^{2,1,5}$ in our experiments.

Although other color constancy methods can be used, we focus on the above instantiations which include pixel and derivative-based methods. Moreover, to exploit the relation between natural image statistics and color constancy, this set of color constancy methods is required [62].

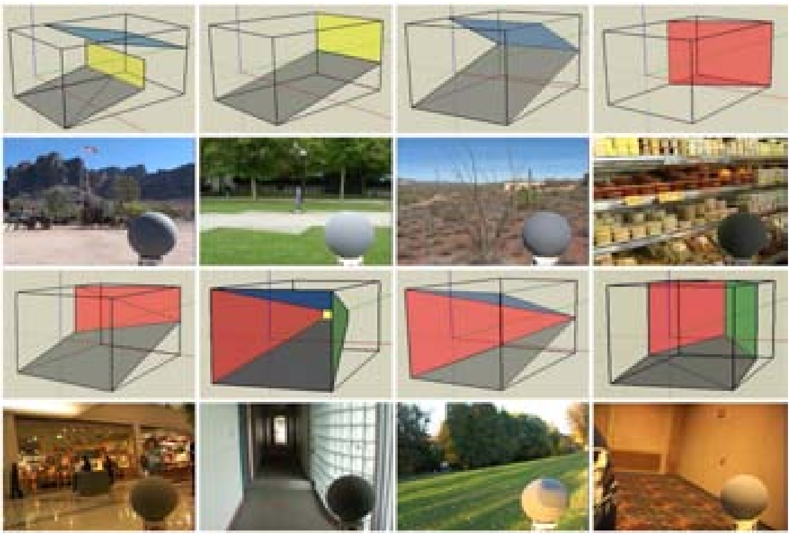
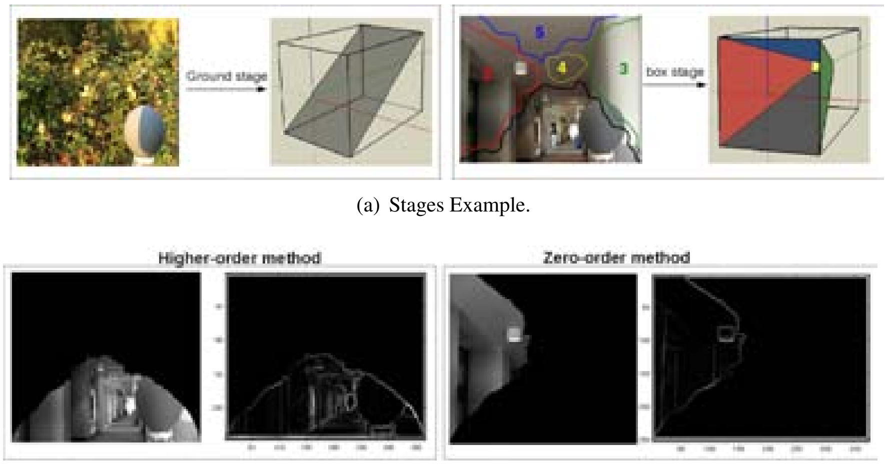


Figure 6.2: Stage models and their corresponding instantiations: top two rows, from left to right: sky + background + ground, background + ground, sky + ground, ground + diagalBackgroundLR; bottom two rows: diagalBackgroundLR, box, 1side-wallLR, corner.

6.3.2 Depth from Stage Models

A number of methods have been proposed to estimate the rough scene geometry from single images [36, 154, 77]. However, these methods are restricted to a number of classes limiting their applicability. To this end, *stages* are taken which correspond to generic categories [175]. Stages are defined as a set of prototypes of common scene configurations. They can be seen as discrete classes of scene geometries. Typical classes of discrete $3D$ scene geometries (i.e. stage models) include single-side backgrounds (e.g. walls and buildings) or three sides (e.g. corridor and narrow streets). A number of stage models, together with corresponding image examples are shown in Figure 6.2. These models are dependent on the inherent structure of images. It has been shown that images can be classified into one of the different stages [175]. Depth is estimated via stage classification. Each stage has a certain depth layout in terms of layers at a certain distance to the camera. In this way, color constancy can be applied per depth layer. In this paper, 13 different stages are used excluding *noDepth* or *tab+pers+bkg*, as these stages are specific characteristics of the data set used in [175].



(a) Stages Example.

(b) Color Constancy per Depth.

Figure 6.3: Example depth images together with their corresponding stages are shown in the first row. The "Ground" stage has one depth and therefore its corresponding image example is not segmented (top-right). While, the "Box" stage has five depths, and therefore its corresponding image example is segmented into five segments indicated by numbers (top-left). The second row shows example depth layers for close-by and far-away objects wrt the most appropriate color constancy algorithm per depth.

In the first row of figure 6.3, we show that the sample images shown in figure 6.1 are example images from the *ground* (one depth layer) and *box* (five depth layers) stages, respectively. Each depth layer is equivalent to an image segment (scene part). In the second row of figure 6.3, we also show example image segments for the *box* stage together with their Gaussian derivatives. For each image segment, the most appropriate color constancy algorithm is selected based on its (local-per segment) natural image statistics. For an image segment containing near-by objects and therefore showing more details (edges), higher-order color constancy algorithm is used for estimating its illuminant. This is demonstrated in the bottom row (left) of figure 6.3. While, for image segments containing distant objects, pixel-based color constancy methods are selected for estimating their illuminant, as illustrated in the bottom row of figure 6.3 (right).

As shown in Figure 6.2, the depth structures of the stage models are demonstrated in different colors. Each stage has a unique depth pattern. The stage models are used to determine how an image is divided in depth layers. For instance, images of stage *sky* + *background* + *ground* will be divided into three layers: sky (in blue), background (in yellow) and ground (in brown). While, images of stage *box* will be divided into five layers: top (in blue), bottom (in brown), right (in green), left (in red), and middle (in yellow). On the other hand, images of stage *ground* will not be divided as it only contains one depth layer. In the next section, scene depth (*i.e.* scene geometry) is used together with natural image statics in order to achieve a proper selection of color constancy algorithms per depth layer.

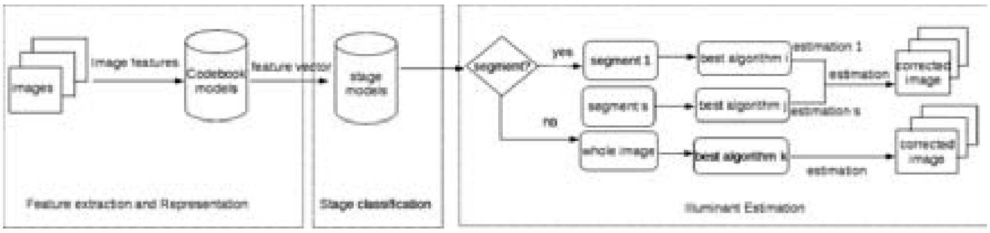


Figure 6.4: Outline of color constancy using 3D scene geometry. Note that the codebook models and the stage models are obtained off-line. Stage segmentation and color constancy algorithm selection are trained on the data set beforehand.

6.4 Color Constancy using 3D Scene Geometry

The method using scene geometry for color constancy consists of the following steps. First, images are classified into stages. Then, according to their stage models, images are divided into depth layers. Then, for each layer, the proper color constancy algorithm is selected. Selection is based on natural image statistics. The final result of the whole image is a weighted illuminant correction per layer. In this way, each layer is allowed to be illuminated by a different (single) light source. In the case of a single layer (i.e. the whole image), illumination estimation will be computed using the best color constancy algorithm for that stage. The whole process is shown in Figure 6.4. To be precise, the different components of our method are as follows.

6.4.1 Stage Classification

In [175], stage classification is based on simple image features and classifiers. However, in this paper, the Bag-of-Words method is used to classify stages. The classification system is selected proposed by Van de Sande et al. [165] using the color constant RGB SIFT descriptor. We used a vocabulary of size 4000 visual words with a two level spatial pyramid image representation [97]. Furthermore, *1-vs-all* SVM classifiers with χ^2 kernel are used, see [29, 165] for more details.

A standalone data set is selected for the sake of training the stage classifier exclusively. We refer to it as *“stages data set’*. The data set consists of 3589 images classified as 15 different categories representing the standard generic scene geometries (stages): 151 *sky + background + ground*, 333 *background + ground*, 81 *sky + ground*, 212 *ground*, 139 *ground + DiagBkgLR*, 132 *ground + DiagBkgRL*, 75 *diagBkgLR*, 71 *diagBkgRL*, 84 *box*, 57 *1sidewallLR*, 69 *1sidewallRL*, 266 *corner*, 960 *persBkg*, 833 *noDepth*, and 126 *tabPersBkg*. Images are take are under a large variety of lighting conditions (including indoors, outdoors, deserts, cityscapes, and other settings). Some example images are shown in figure 6.5.

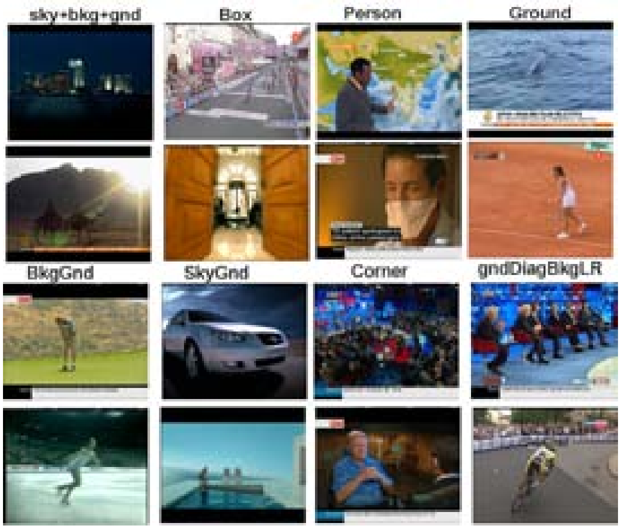


Figure 6.5: Example images of stages data set.

6.4.2 Image Segmentation to Obtain Depth Layers

After the stage classification, the different depth layers (image segments) correspond to a scene part at a certain depth (layer). Each layer represents geometrical entities like walls, ground, and sky. We will use the image division provided by the stage model to learn which color constancy algorithm performs best for each layer. Both hard and soft segmentation is considered, the latter taking the uncertainty into account due to the rough outline of the stage geometry. Both of them are based on the occurrence probability in the training set. Ground truth is obtained by manual annotation, thereby dividing the training set according to the stage patterns, and fitting the parameters of each stage model (horizon, vanishing points) such as to visually best fit the underlying data. For this purpose, we used the *stages* data set described earlier in section 6.4.1 for obtaining the hard and soft segmentation masks used to represent each scene geometry category.

More precisely, suppose that an image belongs to stage S, which is composed of N layers, correspondingly there will be N mask maps. The mask map for the i^{th} partition T_i is obtained by taking the average of the mask maps for each image:

$$T_i(x) = \frac{\sum_{j=1}^{n} M_{j,i}(x)}{n}, \tag{6.5}$$

where *n* is the total number of images in the training data set, and $M_{j,i}(x)$ is the mask map of the j^{th} image for i^{th} partition. Note that $M_{j,i}(x)$ is an indicator function: $M_{j,i}(x) = 1$, if x belongs to the i^{th} partition and 0 otherwise.

Hard Segmentation

[62] proposes a global selection mechanism based on Weibull distributions to choose the proper color constancy algorithms. As there exists a relation between natural image statistics and color constancy, the Weibull parameterization (grain size and contrast) is taken to select the proper color constancy method per image. In contrast, we propose a *local* method in which the proper color constancy method is selected per layer (image segment). Moreover, we exploit the relation between depth and color constancy. Contrasted details (edges) are common for close-by objects (edge-based color constancy is preferred) than distant objects (pixel-based color constancy is preferred).

Therefore, mask maps are used to automatically divide the images. Assuming that the images of a certain stage can be partitioned into N layers, there exist N mask maps corresponding to the partitions in the training data set. Then, the binary mask map is defined as follows:

$$\mathrm{T}'_i(\mathbf{x}) = \begin{cases} 1, & \mathrm{T}_i(\mathbf{x}) = \max\limits_{\mathbf{j=1}}^{\mathbf{N}} \mathrm{T}_{\mathbf{j}}(\mathbf{x}), \\ 0, & otherwise. \end{cases} \tag{6.6}$$

As a consequence, the values in the hard mask map are either 0 or 1, as shown in Figure 6.6(b).

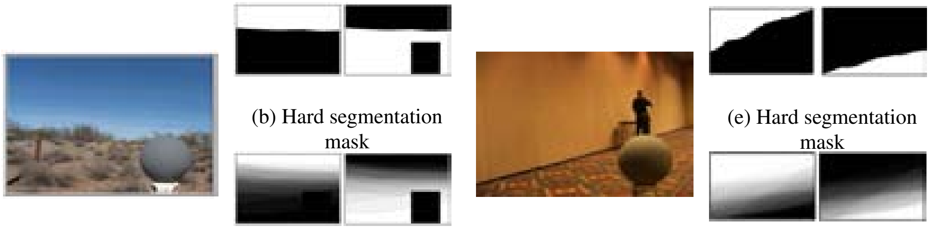
After the maximum mask map is obtained, the color of the light source is estimated using pixels from one layer while other pixels are ignored. In this way, multiple light sources are allowed per scene (one per layer).

Soft Segmentation

As stage classification is a rough estimation of the scene geometry, some locations (pixels) are more reliable than others to belong to a certain layer (image segment). To this end, we assign different confidence values to locations. We set the confidence values of pixels to $\mathrm{T}_\mathrm{i}(\mathbf{x})$, which indicates the occurrence frequency of pixel positions appearing in the training data set. Hence, given a location (pixel position), the larger the confidence value, the more probable it belongs to that layer (image segment). Note that the values of the soft segmentation mask map are between 0 and 1, as shown in figure 6.6(c) of stage "*sky+ground*", and figure 6.6(f) of stage "*ground + DiagBkgLR*".

6.4.3 Illumination Estimation

Without segmentation. After stage classification, the color constancy algorithm is selected for each stage by considering the angular errors of the five different color constancy algorithms. Algorithms are applied on the training images of a specific stage. The algorithm with the lowest angular error is assigned to the stage under consideration. In this way, for each stage, the most proper color constancy algorithm is assigned. Note that this step is processed offline. Then, the on-line processing is to predict which stage an (unknown) image belongs to by using the trained classifier. Finally, the color constancy algorithm that has been assigned to



(a) Original image

(b) Hard segmentation mask

(c) Soft segmentation mask

(d) Original image

(e) Hard segmentation mask

(f) Soft segmentation mask

Figure 6.6: An example of hard and soft segmentation mask maps. The original image (a) belongs to stage *"sky+ground"* and (d) belongs to stage *"ground + DiagBkgLR"*. The mask maps are of the same size as the original image. The difference between hard segmentation map, shown in figures (b), (e), and soft segmentation map, shown in figures (c), (f), is that values in figure (b), (e) are either 0 or 1 while values in figure (c), (f) between 0 and 1.

that stage will be used to estimate the light source to correct the image. This method implies *global* illumination.

With segmentation. After the mask map has been obtained by hard or soft segmentation, images in the training data set will be divided into several segments. The most suitable color constancy algorithm will be selected from the existing color constancy algorithms for each layer. This is achieved by analyzing the angular errors for all color constancy algorithms on images of this stage in the training data set (see section 6.5.2 for details). The color constancy algorithm with the lowest angular error is assigned to this layer of the stage. In other words, the stage model is labeled with the layers and their corresponding color constancy algorithms which provide the highest color constancy accuracy (*i.e. lowest angular error*). The weight of each layer is inversely to its angular error. This method takes advantage of *local* illumination in which the color of the light source is estimated using pixels from one layer while other pixels are ignored. In this way, multiple light sources are allowed per scene (one per layer).

For each unseen image in the test data set, the on-line process is as follows: first, the image is classified into stage S; then it is divided according to the mask maps of stage S. This has been obtained by the training data set beforehand. Further, the color of the light source is estimated for each layer using its selected color constancy algorithm. Finally, we use a weighted average *global illumination* for evaluating our approach, this is due to the unavailability of the color constancy data sets (according to our knowledge), which provide *local illumination* ground-truth information.

Illumination Estimation using Natural Image Statistics

In this section, the novel strategy is discussed based on natural image statistics to select the color constancy method which performs best for a specific depth layer. The selection of the most appropriate color constancy algorithm for a specific image segment is done based on post-supervised layer classification [87]. This algorithm aims at combining the estimates

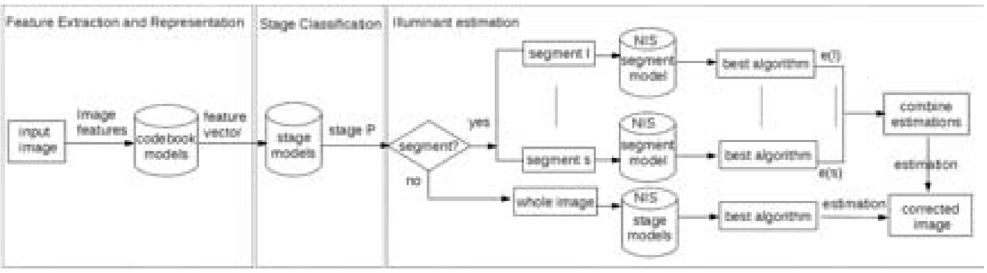


Figure 6.7: Outline of color constancy using $3D$ scene geometry. Note that the codebook models and the stage models are obtained off-line. Stage segmentation and best color constancy algorithm selection are trained beforehand using Natural Image Statistics features (NIS).

obtained for each layer (image segment) into a single more accurate one. The whole process is demonstrated in Figure 6.7.

More precisely, let M be the set of color constancy algorithms that are considered, where M_i denotes algorithm i. Further, the accuracy of the estimate of algorithm i on layer j is denoted by $\epsilon_i(j)$. This is computed by analyzing the angular errors (*i.e.*, a performance measure that determines the angular distance between the estimated illuminant and the true illuminant) for all color constancy algorithms for layer j. The color constancy algorithm with the lowest angular error is assigned to this segment. The learning algorithm consists of the following steps.

After the mask map has been obtained by hard or soft segmentation, the image statistics ω for $segment_j$ for all stage images are computed, *i.e.* ω_{ij} is the i^{th} feature of the j^{th} segment. For simplicity, the subscript i is omitted, so ω_j denotes the feature vector representing the image statistics of the j^{th} segment.

Then, all $segment_j$ images that are in the training set are labeled. The label y_j of image $segment_j$ is derived using the performance of the algorithms on the image segment j.

$$y_j = min_i(\epsilon_i(j)) \tag{6.7}$$

Further, a classifier is created for $layer_j$ using the training data. The labels, determined in the previous step, are used as classes.

For an unknown test image: first, apply the stage classifier. After knowing the corresponding stage and segmentation mask, apply $layer_j$ learned classifier to $imagesegment_j$. The output of $layer_j$ classifier is a color constancy algorithm. This algorithm is assigned to $imagesegment_j$.

Finally, the estimated illuminant for the considered test image is obtained by combining the estimated illuminants for each specific segment j belonging to its assigned segmentation mask.

For the purpose of evaluating our approach, we propose two different experiments for training the required classifiers. The first experiment is based on cross-validation: the *same data set* (we refer to as *real-world data set*) will be used for both training and evaluating our

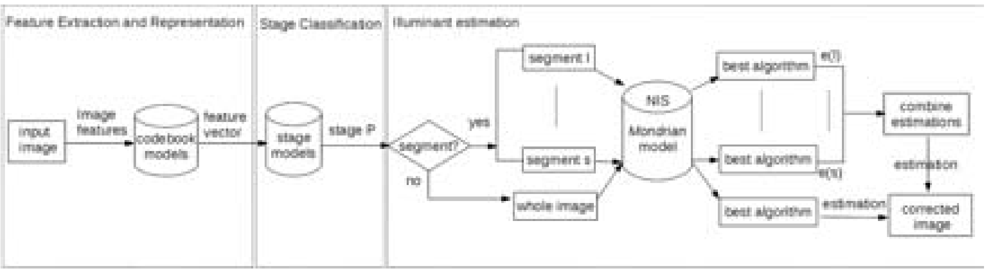


Figure 6.8: Outline of color constancy using $3D$ scene geometry. Note that the codebook models and the stage models are obtained off-line. Stage segmentation and best color constancy algorithm selection are trained beforehand using Natural Image Statistics features (NIS).

approach. Concretely, the *real-world data set* will be divided into 15 parts, then the method will be trained on 14 parts of the data and tested on the remaining part. This procedure is repeated 15 times, so that every image is in the test set exactly once. The whole process is demonstrated in Figure 6.7.

The second experiment consists of using an *independent data set* (we refer to it as *Mondrian data set*) for the training phase, while the *real-world data set* will be used for the testing phase. We use the *Mondrian data set* presented in [8], which is created based on spectral reflectance data for several reasons. First, to simulate the scenario where few knowledge is known on the data set used for testing our approach. This makes the whole approach independent of the training set and therefore, allows for evaluating our approach in a more generic manner. Second, the main advantage of hyperspectral data is that many different illuminant can be used to realistically render the same scene under various light sources. The data set images have different properties in the number of edges, the amount of texture and contrast. Note that, the resulting images contain up to tens of different surfaces and hence, many different transitions simulating the statistics of the real-world images. The whole process is demonstrated in Figure 6.8. Ideally both types of data should be used for a thorough evaluation of color constancy methods [6, 7]. We summarize the whole procedure of our proposed approach in Algorithm 3.

Combination of NIS-based Illuminate Estimation

In this section, different methods are proposed to fuse the most appropriate color constancy algorithms assigned to each segment j of an image based on natural image statistics. When using the output of multiple algorithms to generate a new estimate of the illuminant, the simplest method is to take the average of the estimates over all the algorithms. A straight forward extension is to take the weighted average of the estimated illuminants. If n algorithms corresponding to the n segments of an image are to be combined, then the weighted average is defined as:

$$\bar{\text{e}}= \sum_i^n w_i e_i,$$

Algorithm 3 Color Constancy using 3D Scene Geometry.

```
Input:
$M$ is the set of color constancy algorithms considered, where $M_l$ denotes algorithm $l$.
$\varepsilon$ is the accuracy of the estimation of algorithm $l$.
$C$ is the set of stages.
$k$ is the number of stages.
$S$ is the set of image segments after the mask map has been obtained.
$i$ is the number of stage images.
$j$ is the number of stage segments.
Output:
E is the final illumination estimation obtained.
Training Phase:
if using same data set then
    For every image segment $S_{ij}$ in the current stage $C_k$, compute the natural image statics $\omega_{ij}$.
    For each segment $S_j$, create a classifier using its training data $S_{ij}$. The label $y_{ij}$ is derived based on the performance $\epsilon_l$ of the algorithms $M_l$ on each image segment $ij$.
        $y_{ij} = min_l(\epsilon_l(ij))$    (6.8)
else if using independent Mondrian data set then
    Create a classifier using the Mondrian data set. The label $y_i$ is derived based on the performance $\epsilon_l$ of the algorithms $M_l$ on image $i$:
        $y_i = min_l(\epsilon_l(i))$    (6.9)
end if
Testing Phase:
Apply the stage classifier. After knowing the corresponding stage and consequently, segmentation mask, apply the appropriate learned segment classifier for each segment$_j$. The output of the classifier is a color constancy algorithm used to estimate the illuminant of that segment.
if using same data set then
    For each image segment $S_{ij}$, apply the specific classifier learned over the training data of that specific segment.
else if using independent Mondrian data set then
    For each image segment $S_{ij}$, apply the same classifier learned over on the independent Mondrian data set.
end if
The final illuminant estimation $E$ of a test image is obtained by combining the weighted estimated illuminants $e_j$ on each of its segments.
```

The weight of the estimated illuminant for each segment is derived from the stage category it belongs to, based on the ground-truth illuminant of its training images. For this purpose, we model the ground-truth illuminant for each stage based on several weighting methods, namely, *Bayesian*, *Kernel density*, *Histogram smooth*, *Mixture of Gaussians (MOG)* and *Inverse error* in order to obtain the illuminant model of each stage.

To this end, we first model the ground-truth illuminant for each stage based on well-known classifiers, such as: *Bayesian*, and *MOG*. For a test image, the learned model parameters are used to derive the weight of the estimated illuminant for segment j. In *Kernel density* and *Histogram smooth*, a histogram that measures the frequency of occurrence of the ground-truth illuminations of the training data is obtained. However, in *Histogram smooth*, the obtained histogram using the *Kernel density* method is smoothed to reduce the noise effect due to the existence of the empty bins. For a test image, weighting the estimated illuminant for segment j is determined using the weight (frequency) of the closest ground-truth illuminant. Finally, we examine the *Angular Difference Confidence* ADC criteria which weights the estimated illuminant of each segment with respect to the inverse of its angular error.

6.5 Experiments

In this section, the proposed method is evaluated and compared with the state-of-the-art color constancy methods on a large-scale data sets including hyper-spectral, linear and non-linear real-world data sets. The main advantage of hyperspectral data is that many different illuminants can be used to realistically render the same scene under various light sources. However, the simulation of illuminants generally does not include real-world effects like interreflections and non-uniformity. Consequently, the evaluation of *real-world* RGB-images results in more realistic performance evaluations. Ideally, both types should be used for a thorough evaluation of color constancy methods. In section 6.5.1 the data sets that are used for training and testing are discussed in detail. In section 6.5.2, the performance measures are given. In section 6.5.3, the stage classification framework is presented. In sections 6.5.4 and 6.5.5, the performance of several state-of-the-art algorithms on a large-scale data set is given. In section 6.5.6, the performance of the baseline and the proposed method is provided. Finally, in section 6.5.7, experiments using scene geometry and natural image statistics together with the proposed fusion algorithms are given.

6.5.1 Data sets

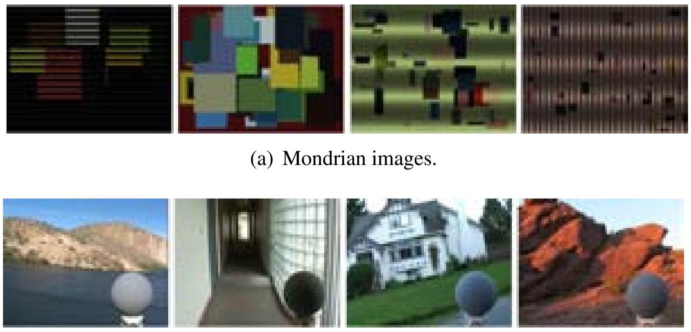
Three independent data sets are used in the experiments. In order to obtain a classifier with good generalization capabilities, a large data set (denoted as *stages data set*) with more than $3,500$ images is used to train the stage classifiers [175].

As a second data set, the color constancy data set of Ciurea and Funt [31] referred to as *real-world data set* is used for testing. Note that this data set contains the ground truth for color constancy while, no ground truth is available for *stages data set*. Therefore, the color constancy algorithms are evaluated on *real-world data set*. The *stages data set* is used to provide an independent data set for stage classification ensuring generalization of the proposed method.

The *real-world data set*, consists of more than $11,000$ images, extracted from 2 hours of video for a wide variety of settings such as indoor, outdoor, desert, cityscape, etc. There are in total 15 different video clips taken at different places and with varying lighting conditions. As there exists a correlation among images of the same video clip, we test the color constancy algorithms on a subset of uncorrelated images composed of 711 images. These images are manually selected and annotated. A few example images that are in the data set are shown in figure 6.9(b). In each image, there is a grey ball at the right bottom, which is used to capture the ground truth of the light source. Note that the grey ball is masked when the illuminant is estimated.

As a third set, the training data set that is created based on spectral reflectance data presented in [8]. This data set originally comprises only surface and illuminant spectra, which are first combined into (R, G, B)-values. Then, using these generated pixel colors, several Mondrian-like images are created, which all have different properties in the number of edges, the amount of texture and contrast. Since only material surfaces are present in the original data set, shadow gradients are added to several images to enlarge their photo-metrical variety. Note that the resulting images contain up to tens of different surfaces, and hence many different transitions, simulating the statistics of real-world images as close as possible. A few example images that are in the data set are shown in figure 6.9(a). This data set will be called the *Mondrian data set* in the remainder of the paper.

One of the basic assumptions in machine learning, is that the distribution of the test data should be similar to the distribution of the training data. Hence, the variety of images used for training the algorithm, should be similar to the variety of the images that are used to test it. To this end, we propose two different experiments. The first experiment is based on cross-validation: the data will be divided into 15 parts. Next, the method is trained on 14 parts of the data, and tested on the remaining part. This procedure is repeated 15 times, so every image is in the test set exactly once, as shown in Figure 6.7.



(a) Mondrian images.

(b) Real-world images.

Figure 6.9: Examples of images that are in the two data sets that are used in this paper. The first data set consists of images that are generated using surface reflectance spectra combined with illuminant spectra [8]. The second data set consists of real-world images [31].

The second experiment consists of a training phase using the Mondrian-like images shown in figure 6.9(a) and a test phase using the *real-world data set* figure 6.9(b). The whole scenario is shown in Figure 6.8 for more illustration. In this way, the data sets for training and testing are completely different. This scenario reflects the case when the data set (used for testing the method) is unknown which is the most general case.

6.5.2 Performance Measures

Two performance measures are used in this paper: stage classification is evaluated using the average precision, while the angular error is used to validate the performance of the color constancy algorithms.

Average precision. The average precision is equivalent to the area under a precision-recall curve. It combines precision and recall in a single number. Mean average precision MAP is used to evaluate the performance of the features over all the stages, which is obtained by averaging the average precisions over all stages.

Angular error. In order to evaluate the performance of the color constancy algorithms, the angular error ε is used,

$$\varepsilon = \cos^{-1}(\hat{e_l}.\hat{e_e}), \tag{6.10}$$

where $\hat{e_l}$ is the normalized ground truth of the illuminant, while $\hat{e_e}$ is the normalized estimation. Both mean and median angular errors [78] are taken as performance indicator.

6.5.3 Stage Classification

For the purpose of building the stage classifier, we use the Bag-of-Words *(BoW)* classification framework. The RGB-SIFT feature is used as it outperforms other variants of the SIFT-feature [104]. The performance of the RGB-SIFT versus intensity SIFT for the entire *real-world data set* is demonstrated in Table 6.1. Performance of rgSIFT, OpponentSIFT, and SIFT are quite similar. By contrast, the RGB-SIFT feature is more accurate. This is explained by the color constant properties of the RGB-SIFT, *i.e.* the invariance to both light color changes and shifts [165]. This is an advantage for our color constancy data set, as images with similar stage layout may be captured under different lighting conditions. Consequently, we focus on the RGB-SIFT feature in the sequel.

For the vocabulary construction, we use a standard k-means to build a vocabulary of size 4000. After the assignment stage, we use the spatial pyramid image representation [97]. A compact spatial pyramid image representation is obtained by compressing the original spatial pyramid histograms based on the Divisive Information Theoretic feature Clustering algorithm (*DITC*) [38] as proposed by [42]. This results in a compact spatial image representation, which maintains the original pyramid performance and speeds up the classification process afterwards. For the classification stage, we use generic $1-vs-all-$-based classifiers. There are a total of 13 classifiers corresponding to 13 stages concerned. The output of the classifier is a single stage label.

The performance of stage classification on each stage is shown in Table 6.2. From this

Feature	Mean Average Precision
C-SIFT	0.283
rgSIFT	0.296
OpponentSIFT	0.303
SIFT	0.304
RGB-SIFT	0.320

Table 6.1: The performances of stage classification for color vs. intensity SIFT features over all stages.

Name	% in data set	AP
skyBkgGnd	9.1%	0.65
bkgGnd	9.9%	0.34
skyGnd	2.7%	0.34
gnd	12.1%	0.67
gndDiagBkgLR	6.6%	0.16
gndDiagBkgRL	4.6%	0.16
diagBkgLR	4.6%	0.12
diagBkgRL	3.8%	0.15
box	8.0%	0.37
1side-wallLR	12.9%	0.46
1side-wallRL	15.6%	0.41
corner	6.5%	0.15
persBkg	3.5%	0.19
MAP	0.320	

Table 6.2: Stage classification results for each stage using the RGB-SIFT feature, as well as relative occurrence within the *real-world data set*. Note that the last row gives the mean average precision over all stages.

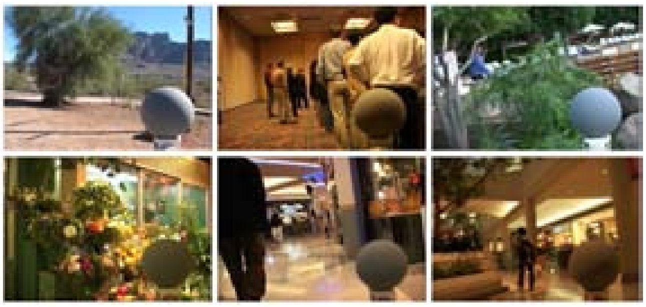


Figure 6.10: Examples of misclassified images.

Method	**Mean**		**Median**	
Grey-World	7.4°		7.0°	
White-Patch	7.3°		6.1°	
general Grey-World	6.4°		5.8°	
1^{st}-order Grey-Edge	**6.0°**		**5.2°**	
2^{nd}-order Grey-Edge	6.0°		5.4°	
Combination using indoor-outdoor classification	7.0°	(+17%)	6.5°	(+25%)
Combination using natural image statistics	5.7°	(−5%)	4.7°	(−10%)
Proposed (auto): without segmentation	5.7°	(− 5%)	4.8°	(− 8%)
Proposed (auto): hard segmentation	**5.4°**	**(−10%)**	**4.5°**	**(−14%)**
Proposed (auto): soft segmentation	5.4°	(−10%)	4.6°	(−12%)
Proposed (manual): without segmentation	5.5°	(− 8%)	4.6°	(−12%)
Proposed (manual): hard segmentation	4.7°	(−22%)	3.7°	(−29%)
Proposed (manual): soft segmentation	**4.7°**	**(−22%)**	**3.6°**	**(−31%)**

Table 6.3: Performance of color constancy algorithms over the *real-world data set*. Proposed (auto) means that the proposed methods are applied to automatically classified images, while proposed (manual) indicates that our methods are evaluated on manually classified images.

table, it can be derived that for some stages, such as *sky+bkg+gnd*, and *gnd*, the results are satisfying. For other stages, like *diagBkgLR* and *diagBkgRL*, the results still leave room for improvement. This is due to occlusions appearing in these categories, making it hard to classify them correctly. A few examples of misclassified images are given in Figure 6.10.

6.5.4 Single Color Constancy Methods

The algorithms that are evaluated here are the five instantiations discussed in Section 6.3. The results for single algorithms are shown in Table 6.3. These methods are applied to each image in D_2. Table 6.3 shows that the edge-based methods (i.e. 1^{st}-order Grey-Edge and 2^{nd}-order Grey-Edge) outperform the pixel-based methods (i.e Grey-World, White-Patch, and general Grey-World). Differences between 1^{st}-order Grey-Edge and 2^{nd}-order Grey-Edge are small, but the median error of the 1^{st}-order Grey-Edge is slightly lower, so the 1^{st}-order Grey-Edge is considered our *baseline* in the remainder of this section.

6.5.5 Combining Algorithms

In addition to using only single algorithms, two combination algorithms are evaluated. The first method is proposed by [14] and distinguishes between indoor and outdoor images. An indoor-outdoor classifier is proposed that is used to apply different color constancy algorithms to indoor and outdoor images. They propose to use the shades-of-grey method for indoor images and the 2^{nd}-order Grey-Edge method for outdoor images. For convenience, we used manual annotation of indoor and outdoor images instead of the indoor-outdoor classifier proposed in [14]. As can be seen in Table 6.3, the accuracy of the illuminant estimates does not improve with respect to the single color constancy algorithms.

Another combination method is proposed by [61], and use global image statistics for the selection of the most appropriate color constancy algorithm. Results indicate that the performance indeed improves with respect to the best-performing single algorithm, see table 6.3. However, the scene geometry is not taken into account as this method uses global selection of the most appropriate color constancy algorithm.

6.5.6 Color Constancy using Stage Classification

Evaluation of the proposed methods is performed using the leave-one-out cross validation method. To obtain the mask map, all the images in the training data set are manually annotated and segmented.

Without Segmentation Illuminant estimation is computed using the entire image. The median angular error of the proposed method without segmentation is $4.8°$ as shown in Table 6.3. Compared with the best-performing algorithm, i.e., the 1^{st}-order Grey-Edge, an increase of almost 8% on the median angular error is reached. Results on individual stages reveal that most of the color constancy algorithms have a preference for specific stages. For instance, 0^{th}-order methods like the White-Patch and the general Grey-World prefer stages where the depth can become quite high, like the stage *sky+background+ground*. Such stages with a large depth can contain haze, which causes a relatively low signal-to-noise ratio, and it is known from [61] that methods that are based on higher-order statistics like the 2^{nd}-order Grey-Edge do not perform well on such images. On the other hand, the 2^{nd}-order Grey-Edge algorithm performs better on images with a high amount of information, e.g. many edges. This is reflected in a preference for stages like *diagonal Background LR* and *diagonal Background RL* that generally contain images with much contrast and many edges.

Segmentation	Estimation	Groundtruth
sky	(0.60, 0.59, 0.54)	(0.55, 0.57, 0.60)
background	(0.48, 0.56, 0.67)	
ground	(0.58, 0.58, 0.58)	
1sidewallLR	(0.61, 0.56, 0.57)	(0.59, 0.59, 0.54)
ceil	(0.62, 0.45, 0.64)	
floor	(0.68, 0.58, 0.45)	

Figure 6.11: Examples of the proposed method using hard segmentation. The red lines in images are the boundaries of rough segmentation. Illuminant estimation of each region is obtained by 1^{st}-order Grey-Edge, which is the best among the considered single algorithms on the *real-world data set*.

Hard Segmentation. The performance of the proposed method using hard segmentation on each stage is shown in Figure 6.12. The performance of the proposed method using hard segmentation on the entire data set *real-world data set* is given in Table 6.3: the median

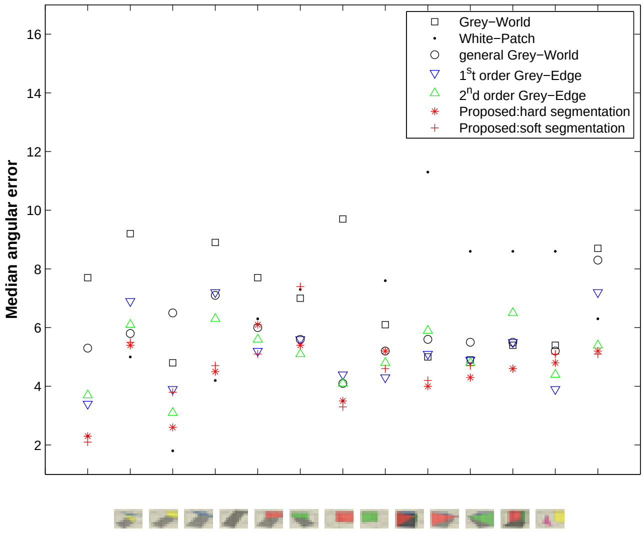
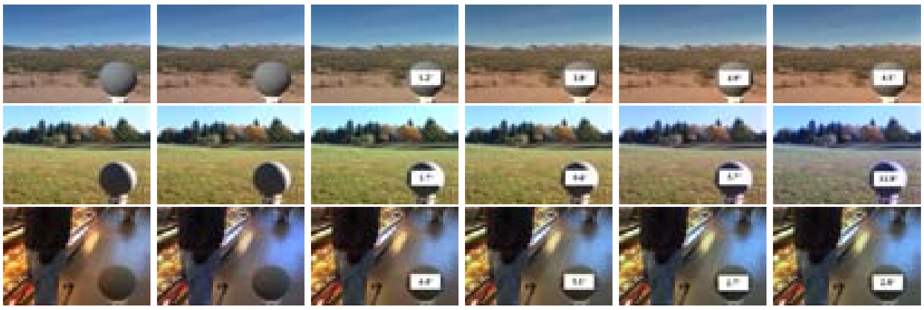


Figure 6.12: Median angular errors of color constancy algorithms for each stage in the *real-world data set*. The stage models are shown on the horizontal axis: "sky + background + ground", "background + ground", "sky + ground", "ground", "ground + diaganalBackgroundLR", "ground + diaganalBackgroundRL", "diaganalBackgroundLR", "diaganalBackgroundRL", "box", "1sidewallLR", "1sidewallRL", "corner", " person+Background".

angular error equals $4.5°$. Compared to the baseline, the median angular error is reduced by almost 14%. Specific examples are shown in Figure 6.11. Note that, due to the stage classification, we do not only improve the overall illuminant estimation accuracy, but we can assess the illuminant color of the various geometrical constellations of a scene. This is outlined in Figure 6.12, where each stage is represented by its best estimation algorithm. Expanding from this, we propose to estimate the light source color at the various depth layers as indicated by the $3D$ stage model. This allows the estimation of a distant light source, and to distinguish a nearby illuminant (indoor, shadow) from a far away illuminant (outdoor, sunlight).

Soft Segmentation. The performance of the proposed method using soft segmentation on each stage is demonstrated in Figure 6.12 while the result over the whole data set is shown in Table 6.3: the median angular error is $4.6°$, which is quite similar to the proposed method using hard segmentation. The proposed method using soft segmentation makes an improvement of 12% in median angular error over the baseline.

Discussion. In addition to automatic classification, manual classification is used to determine how the stage classification performance influences the final results. Results are shown in Table 6.3. Using this ideal classifier (i.e. the mean average precision is 1), the median angular errors of the method without segmentation is reduced to $4.6°$. Using hard segmentation, the best-performance that can be obtained is $3.7°$ for the median angular error,



(a) Original image (b) Ground truth (c) Correction using hard segmentation (d) Correction using soft segmentation (e) Correction using 1st-order Grey-edge (f) Correction using grey-world

Figure 6.13: Results of color constancy. The angular error is given on the grey ball, which is masked during illuminant estimation.

Segmentation	Estimation	Groundtruth
sky (M)	(0.57, 0.58, 0.58) ±0.95	
background (M)	(0.59, 0.58, 0.55) ±0.96	(0.56, 0.58, 0.58) ±0.93
ground (M)	((0.60, 0.58, 0.54) ±0.96	

Table 6.4: The mean of the Ground-truth Illumination (M) for stage *sky+background+ground* along with the euclidean distance between M and the standard deviation (std), Compared to those obtained using the illuminant of the White patches within each specific segment.

while the median angular error can be further reduced to 3.6° by using soft segmentation. In conclusion, improving stage classification will further improve the color constancy results significantly.

Figure 6.13 presents three images, two of which are correctly classified while the other is misclassified due to the occlusion. The proposed method using soft segmentation is more effective in the presence of shadow or shading edges.

6.5.7 Stage Classification using Scene Geometry and Natural Image Statistics (NIS)

Global Illumination Assumption. The performance of the estimated illuminant of the proposed method using NIS-based hard/soft segmentation is to be compared with the ground truth illuminant of the corresponding segment. As such ground truth is not yet available in the current color constancy data sets, we did not pursue evaluation of these extensions. However, in our experiments we use the available ground truth illuminant of the whole image (*global illuminant*) as a baseline to compare with the estimated illuminant of each segment.

Expanding from this, we propose an experiment based on the global image illumination

Table 6.5: Performance of color constancy algorithms applied on hard and soft segmented images, trained over the *real-world data set* using cross-validation *(cv)* or using the independent Mondrian data set. The proposed methods are applied to automatically classified images. *NIS*, *GE*, *ADC* and *KD* refer to *Natural Image Statistics*, *Grey-Edge* algorithms, *Angular Difference Confidence* and *Kernel Density* weighting schemes (see text).

Method	**Mean**	**Median**
Baseline: 1^{st}-order GE	6.0°	5.2°
Baseline: Global NIS	5.7° (−5%)	4.7° (−10%)
NIS hard-cv-Average	5.4° (−10%)	4.6° (−12%)
NIS hard-cv-Bayesian	4.9° (−18%)	3.7° (−29%)
NIS hard-cv-KD	4.7° (−22%)	3.7° (−29%)
NIS hard-cv-HistSmooth	4.7° (−22%)	3.6° (−31%)
NIS hard-cv-MOG	4.7° (−22%)	3.4° (−35%)
NIS hard-cv-ADC	**3.9**° (−**35**%)	**2.8**° (−**46**%)
NIS hard-Mondrian	4.7° (−22%)	3.3° (−37%)
NIS soft-cv-Average	5.3° (−12%)	4.6° (−12%)
NIS softcv-Bayesian	4.6° (−23%)	3.5° (−33%)
NIS softcv-KD	4.5° (−25%)	3.3° (−37%)
NIS softcv-HistSmooth	4.5° (−25%)	3.3° (−37%)
NIS softcv-MOG	4.5° (−25%)	3.2° (−38%)
NIS soft-cv-ADC	**3.9**° (−35%)	**2.6**° (−50%)
NIS soft-cv-Mondrian	5.4° (−10%)	4.4° (−15%)

and the well-known White-Patch assumption, which states that: a surface with perfect reflectance properties will reflect the full range of light that it captures. Consequently, the color of this perfect reflectance is exactly the color of the light source. Accordingly, we manually select White-Patches from each segment of a specific stage. Then, for each specific segment we compare the mean illuminant of those manually selected patches with the mean ground truth illuminat of all the considered stage images. Specific example for stage *sky* + *background* + *ground* is shown in Table 6.4.

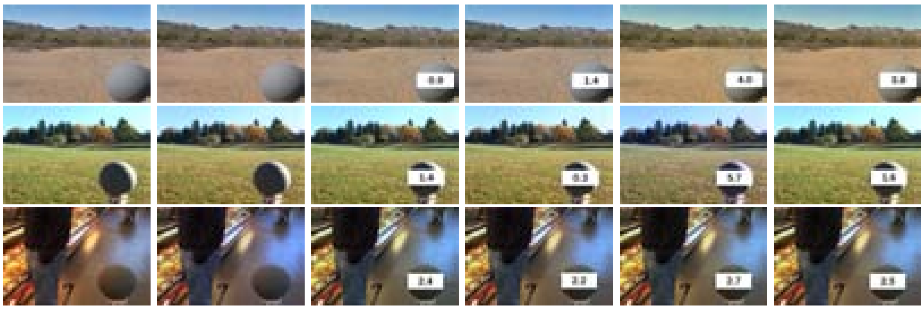
From the quantitative results shown in table 6.4, we conclude that when the object is near the light source, its estimated illuminant is closer to the illuminant of the light source. However, when the object is far away from the light source then its estimated illuminant is deviated from the light source illuminant. For instance, for the *sky* + *background* + *ground* scene geometry category, which is an outdoor scene category, and, consequently, its light source illuminant is the sunlight illuminant: for the *sky* layer, which is the nearest to the light source, its estimated illuminant is the closest to the ground-truth illuminant. However, for the *ground* layer which is the farthest from the light source, it has the most deviated illuminant estimation from the light source (ground-truth).

NIS Hard Segmentation-*real-world*. The performance of proposed method, (denoted by NIS-hard-*cv*) in table 6.5 corresponds to the situation where the circumstances under which the system is used are known apriori. Training is done using part of the real-world data, and tested is done using an independent part. Hence, no direct relation exist between the training data and the testing data (see Figure 6.7). A number of methods for combining the estimated illuminants using the proposed method are investigated. A simple average of the output illuminants *(denoted as "Average") in table 6.5*, improves the results compared to the baseline algorithms (see table 6.5). Compared to the *global Natural Image Statistics (NIS)* the median angular error is reduced by almost 5%. While, a reduction of 12% in the median angular error is obtained compared to 1^{st}*−order Grey Edge* baseline.

Using a weighted average instead of a simple average, performs significantly better than the baseline. We evaluated different weighting strategies (see section 6.4.3) for combining the illuminant estimates obtained for each image segment. Table 6.5, shows the obtained results using the considered weighting schemes, namely, Bayesian, Mixture-of-Gaussians *"MOG"*, Kernel-Density *"KD"*, Histogram-Smooth *"HistSmooth"*, and Angular Difference confidence *"ADC"*, respectively. The median error decreased by around 29% based on the *Bayesian* weighting scheme with respect to the baseline as shown is Table 6.5. Compared to the *Bayesian* method, the *Kernel-density* decreased the mean error from 4.9° to 4.7°, while, maintaining the same median error 3.7°. In addition, the use of the *Histogram-smooth* method results in a reduction of the median error to 3.6° due to the reduction of the noise level obtained while smoothing the histogram empty bins. Further, the proposed method based on *MOG* scheme decreased the median error to 3.4° using two Gaussians. Finally, the *ADC* weighting scheme leads to major drop on both the median and the mean angular errors up to 2.8° and 3.9°, respectively. These obtained results excel the state-of-art results significantly; a reduction of 35% in the mean angular error and 46% in the median error is obtained w.r.t. the baseline (see table 6.5). When using a perfect classifier (i.e., all of the test images are correctly classified into stages), the performance of the proposed method achieve an improvement of 48% of median error compared to the best performing single color constancy algorithm (see table 6.6).

NIS Hard Segmentation-Mondrian. Results of the second experiment, learned on the Mondrian data set and tested on the real-world data set based on the hard segmentation masks (denoted by NIS hard-Mondrian in table 6.5) performs significantly better than the baseline. Compared to the *Global NIS* baseline, the median angular error is reduced by around 30%. While, a 37% reduction on the median error is obtained (i.e., from 5.2° up to 3.3°) compared to the 1^{st}*−order Grey Edge* baseline.

NIS Soft Segmentation-real-world. The performance of the proposed method using soft segmentation over the *real-world data set* is shown in Table 6.5. The proposed method based on the simple average of estimated illuminants (denoted as NIS-soft-cv-average), results in a performance improvement of almost 12% compared to the baseline. Results obtained using the weighted average of the estimated illuminants improve the performance over the simple averaging combination scheme. The *Bayesian* method decreased the median error to 3.5° compared to 4.6° using the simple averaging scheme. Additionally, the median errors ob-



(a) Original image (b) Ground truth (c) Correction using NIS+hard segmentation (d) Correction using NIS+soft segmentation (e) Correction using 1st-order Grey-edge (f) Correction using global-NIS

Figure 6.14: Results of color constancy. The angular error is given on the grey ball, which is masked during illuminant estimation.

tained using the *MOG*, *kernel density*, and *Histogram smooth* weighting methods are reduced further over the *Bayesian* up to 3.3°, 3.3°, and 3.2°, respectively. Finally, the proposed method *ADC* makes an a major reduction in the median angluar error up to 2.6°. This results in an overall performance improvement of almost 50% with respect to the baseline (i.e., 5.2°). When using a perfect classifier (i.e., all of the test images are correctly classified into stages), the performance of the proposed method achieve an improvement of 52% of median error compared to the best performing single color constancy algorithm (see table 6.6).

NIS Soft Segmentation-Mondrian. The proposed algorithm learned on the Mondrian data set and tested on the *real-world data set* based on the soft segmentation masks (denoted by NIS soft-Mondrian in table 6.5) performs significantly better than the baseline. An improvement on the median angular error from 5.2° up to 4.4° is obtained (i.e., 15% improvement).

In addition to automatic classification, manual classification is used to determine how the stage classification performance influences the final results. Results are shown in Table 6.6. Using this ideal classifier, the best-performance that can be obtained is 3.3° (i.e., 37% improvement) for the median angular error based on soft segmentation, while the median angular error can be further reduced to 3.2° (i.e., 38% improvement) by using hard segmentation.

Discussion. In *Mondrian* experiment: the test-data is unknown, and corresponds to the most generic approach of the proposed method. No Learning step is required for a new data set, since the results of the classifier is independent of the test data. On the other hand, the proposed algorithm learned over the *real-world data set*, requires data with enough variety, to train the method. In this experiment, test-data is known making the proposed method less generic. Hence, a learning step is required for each new data set.

Results show significant improvement of the *real-world data set* case over the *Mondrian* case (with no a priori knowledge): the median angular error drops to 2.8° and 2.6° based

Table 6.6: Performance of color constancy algorithms applied on hard and soft segmented images, trained over the *real-world data set* using cross-validation *(cv)* or using the independent Mondrian data set. The proposed methods are evaluated on manually classified images. *NIS*, *GE*, *ADC* and *KD* refer to *Natural Image Statistics*, *Grey-Edge* algorithms, *Angular Difference Confidence* and *Kernel Density* weighting schemes (see text).

Method	**Mean**	**Median**
Baseline: 1^{st}-order GE	6.0°	5.2°
Baseline: Global NIS	5.7° (-5%)	4.7° (-10%)
NIS hard-cv-Average	4.9° (-18%)	4.0° (-23%)
NIS hard-cv-Bayesian	4.9° (-18%)	3.8° (-27%)
NIS hard-cv-KD	4.7° (-22%)	3.6° (-31%)
NIS hard-cv-HistSmooth	4.7° (-22%)	3.5° (-33%)
NIS hard-cv-MOG	4.7° (-22%)	3.4° (-35%)
NIS hard-cv-ADC	$\mathbf{3.8}^\circ$ $(\mathbf{-37}\%)$	$\mathbf{2.7}^\circ$ $(\mathbf{-48}\%)$
NIS hard-Mondrian	4.4° (-27%)	3.2° (-38%)
NIS soft-cv-Average	4.8° (-20%)	3.8° (-27%)
NIS softcv-Bayesian	4.5° (-25%)	3.4° (-35%)
NIS softcv-KD	4.4° (-27%)	3.2° (-38%)
NIS softcv-HistSmooth	4.4° (-27%)	3.2° (-38%)
NIS softcv-MOG	4.4° (-27%)	3.1° (-40%)
NIS soft-cv-ADC	$\mathbf{3.8}^\circ$ (-37%)	$\mathbf{2.5}^\circ$ (-52%)
NIS soft-Mondrian	4.5° (-25%)	3.3° (-37%)

Table 6.7: Performance of color constancy algorithms applied on *hard* and *soft* segmented images, trained over Linear *real-world data set* using cross-validation *(cv)* or using the independent Mondrian data set. The proposed methods are applied to automatically classified images. *GE*, *ADC* and *KD* refer to *Grey-Edge* algorithm, *Angular Difference Confidence* and *Kernel Density* weighting schemes, respectively (see text).

Method	**Mean**	**Median**
Baseline: 1^{st}order GE	$14.5°$	$13.0°$
Baseline: Global NIS	$12.6°$ (-13%)	$12.2°$ (-6%)
NIS hard-cv Average	$11.4°$ (-21%)	$9.5°$ (-27%)
NIS hard-cv-Bayesian	$11.1°$ (-23%)	$9.4°$ (-28%)
NIS hard-cv-KD	$11.1°$ (-23%)	$9.4°$ (-28%)
NIS hard-cv-HistSmooth	$10.9°$ (-25%)	$9.4°$ (-28%)
NIS hard-cv-MOG	$10.9°$ (-25%)	$9.0°$ (-31%)
NIS hard-cv-ADC	**10.6**° (-27%)	**9.2**° (-29%)
NIS hard-Mondrian	$10.8°$ (-26%)	$8.7°$ (-33%)
NIS soft-cv-Average	$11.4°$ (-21%)	$10.0°$ (-23%)
NIS soft-cv-Bayesian	$11.0°$ (-24%)	$9.5°$ (-27%)
NIS soft-cv-KD	$11.0°$ (-24%)	$9.5°$ (-27%)
NIS soft-cv-HistSmooth	$11.0°$ (-24%)	$9.5°$ (-27%)
NIS soft-cv-MOG	$11.0°$ (-24%)	$9.4°$ (-28%)
NIS soft-cv-ADC	**10.6**° (-27%)	**9.4**° (-28%)
NIS soft-Mondrian	$11.0°$ (-24%)	$8.6°$ (-34%)

on hard and soft segmentations, respectively. However, from the experiments, it can be concluded that the proposed method can be trained using completely independent training set, and still perform significantly better than (i.e., $3.3°$) the baseline algorithm (i.e., $5.2°$).

Figure 6.14 presents three images, which are correctly classified based on our proposed approach. The proposed method using soft segmentation is more effective in the presence of shadow or shading edges.

Linear Real-World Evaluation. Finally, we evaluate our method on the linear *real-world* data set. The results in table 6.7 show the obtained results based on our proposed method with both hard and soft segmentation. As expected, the various weighted combination schemes applied on the estimated illuminant of each segment (for obtaining the final image estimated illuminant) improve the results over the simple averaging combination scheme. In addition, the *ADC* method results in achieving the minimum angular error between the estimated illuminant and the ground-truth illuminant (i.e., the best performance). An improvement of almost 29% is obtained compared to the baseline based on our proposed approach.

6.6 Conclusions

We have investigated the relation between scene depth, local image statistics and color constancy. The scene geometry has been computed first to obtain image depth layers. Then, image statistics are exploited (per depth) to select the proper color constancy method. Our approach enables the estimation of multiple illuminants by distinguishing nearby light source from distant illuminants.

Experiments on large scale image data sets show that the proposed algorithm outperforms state-of-the-art single color constancy algorithms with an improvement of almost 50% of median angular error. When using a perfect classifier (i.e, all of the test images are correctly (manually) classified into stages), the performance of the proposed method improves the median angular error as much as 52%. Using Linear large scale data set, leads to an improvement of 29% compared to the state-of-the-art single color constancy algorithms. This gain in performance can largely be explained by the fact that most color constancy algorithms are specifically suited for images with certain image statistics, like a high (or low) signal-to-noise ratio. Further, it is shown that extracting local geometry features is more efficient than applying a global selection or combination algorithm.

Chapter 7

Conclusions

In the final chapter of this dissertation, we briefly recapitulate the main contributions of our research and discuss possible directions for future work. Finally, publications which are directly related to this thesis are listed.

7.1 Summary and contributions of the thesis

Spatial pyramids *(SPs)* have been successfully applied to incorporate the important spatial information into the bag-of-words-based image classification framework. However, the resulting *SP* representation suffers from two major drawbacks, which are (i) the high dimensionality, and (ii) the lack of flexible image representation (i.e., rigidity). In this thesis, we addressed these problems and proposed a variety of models and techniques for obtaining compact and adaptive spatial knowledge that is capable of improving the task of image classification. In particular, we focused on classifying an image by the scene it belongs (e.g. coast, forest, living room, etc.) and classifying an image by the object it contains (e.g. sail boat, dolphin,cactus). The major contributions together with their corresponding conference or journal publications are summarized below:

Chapter 3, Compact and Adaptive Spatial Pyramids for Scene Recognition. To address the above mentioned SP problems, we proposed a technique for obtaining a Compact and Adaptive Spatial Pyramid *(CASP)* representation. Our *CASP* approach, consists of a two-stages pyramid compression strategy which is based on the Agglomerative Information Bottleneck (AIB) theory for (i) compressing the least informative SP features within the spatial pyramids levels, and, (ii) automatically learning the most appropriate spatial pyramid shape for each particular category. Moreover, we proposed a new texture descriptor that supports spatial pyramid matching (P-TPLBP) for scene classification tasks. Our approach is based on the insight that scenes can be seen as a composition of micro-texture patterns. It generalizes the *PHOG* representation of Bosch et al. [18] from shape (edge distributions) alone and the *PHOW* representation of Lazebnik et al. [97] from appearance (visual words) alone to local texture (micro-texture patterns distributions). We demonstrated that this descriptor is an efficient and complementary cue to state-of-the-art spatial shape and appearance features. We also showed that our proposed method exceeds the state-of-the-art results on several challenging scene classification datasets. The hybrid pyramid compression approach together with the spatial texture descriptor is published to the *Image and Vision Computing Journal*.

Chapter 4, Spatial Pyramids using 3D Scene Geometry for Improved Object Recognition The spatial pyramids are often rigid in nature and are based on predefined grid configurations (e.g. 2×2, 4×4 and 1×3 image divisions). As a consequence, spatial pyramids often fail to coincide with the underlying spatial structure of images. To counter the spatial pyramid rigidity problem, we proposed adaptable spatial pyramids which are steered by the 3D scene geometry present in images. The geometry of a scene is measured based on the image statistics taken from a single image. After the estimation of the 3D scene geometry of an image, the corresponding spatial pyramid is selected as the geometrical representation. From large scale experiments on the Pascal VOC 2007 and Caltech-101, it can be derived that SPs which are obtained by selective search outperforms the standard SPs with 12.4% and 14.0% for Pascal VOC 2007 and Caltech101 respectively. The use of 3D scene geometry, to select the proper SP configuration, provides an even higher improvement of 16.0% and 19.7% respectively. Finally, selective spatial pyramids converge to one of the thirteen generic stages yielding a generic structure for spatial pyramids. The proposed method is then tested on several challenging benchmark object classification datasets. The results clearly demonstrated the effectiveness of learning those adaptive spatial partitionings, which are steered by the standard generic 3D scene geometries. This work is published to the *European Conference of Computer Vision*

Chapter 5, Discriminative Compact Pyramids for Object and Scene Recognition. Due to the recent advances in the information-theoretic clustering techniques [38, 54, 152], the problem of constructing compact and discriminative spatial pyramids image representations is revisited. To this end, we presented a novel framework for obtaining compact pyramid representation. Firstly, we investigate the usage of the *divisive information theoretic feature clustering (DITC)* algorithm in creating a compact pyramid representation. In many scenarios this method is shown to reduce the size of a high dimensional pyramid representation up to an order of magnitude without a significant loss in accuracy. Furthermore, comparisons with standard clustering and compression techniques, such as the *agglomerative information bottleneck (AIB)*, show that our method obtained superior results at significantly lower computational costs. Moreover, we also investigated the optimal combination of multiple features in the context of our compact pyramid representation. Finally, experiments showed that our method can obtain state-of-the-art results on several challenging scene and object classification datasets. This work is published to the *Pattern Recognition Journal*.

Chapter 6, Color Constancy using 3D Scene Geometry and Natural Image Statistics The aim of color constancy is to remove the effect of the color of the light source. As color constancy is inherently an ill-posed problem, most of the existing color constancy algorithms are based on specific imaging assumptions such as the grey-world and white patch assumptions. Our approach proposed the usage of 3D geometry models to determine which color constancy method to use for the different geometrical regions found in images. We have investigated the relation between scene depth, local image statistics and color constancy. The scene geometry has been computed first to obtain image depth layers. Then, image statistics are exploited (per depth) to select the proper color constancy method. Our approach enables the estimation of multiple illuminants by distinguishing nearby light source from distant illuminants. Experiments on large scale image data sets show that the proposed algorithm outperforms state-of-the-art single color constancy algorithms with an improvement of almost 50% of median angular error. When using a perfect classifier (i.e, all of the test images are correctly (manually) classified into stages), the performance of the proposed method improves the median angular error as much as 52%. Using Linear large scale data set, leads to an improvement of 29% compared to the state-of-the-art single color constancy algorithms. This gain in performance can largely be explained by the fact that most color constancy algorithms are specifically suited for images with certain image statistics, like a high (or low) signal-to-noise ratio. Further, it is shown that extracting local geometry features is more efficient than applying a global selection or combination algorithm. This work is published to the *International Journal of Computer Vision*.

7.2 Future Work

In the future, we will work towards improving the adaptability, the discriminative power, and the expressiveness of the proposed spatial image models to make them applicable to a wider range of challenging application domains and imagery conditions, including objects that are highly non-rigid and articulated. There are several ways further research could go, but there are also a variety of obvious extensions to the existing frameworks we have presented here in this thesis. Next, we briefly describe them:

- Texture in videos. We have explored texture features in the context of scene recognition in still images (see chapter 3). A further work would be to explore how the texture information can help in video scenes. We would like to develop algorithms for investigating our proposed texture descriptor for videos. By taking advantage of applying this novel and efficient spatial representation for representing video frames, in this fashion it can then be used in videos applications. Recently, promising recent work [148, 85] has shown excellent results based on motion features for scene classification within scene videos. With this insight, we plan to exploit the developed techniques for scene classification in still images complementary to motion features and extend them to video scenes. Therefore, we expect that combining the strengths of both methods will lead to further improvements.

- *CBIR*. Image classification is a special case of image retrieval where the query corresponds to the image category being searched for [177]. We have demonstrated that we can build a good image representation so our method could easily be used for image retrieval tasks providing an automatic ranking of images.

- Hierarchical ROI-based adaptive spatial model. The development of better adaptive spatial models, as well as their quantitative evaluation for recognition and localization tasks are important future directions. We plan to develop more sophisticated methods for learning more powerful class-specific spatial shape models. In particular, we will extend our system to automatically learn a hierarchal class-specific adaptive shape model, where the highest levels will incorporate the important localization and/or segmentation knowledge for efficiently capturing the *Region Of Interest (ROI)*, for restricting the objects spatial location to work with. While, for lower levels we will learn the proper spatial partitioning upon the learned *ROIs*. This hierarchical approach will allow for obtaining more efficient spatial shape models for the objects which they do represent and, consequently, helps in distinguishing among the most ambiguous categories. We expect that combining the strengths of both detection (i.e., ROIs) and a proper adaptive spatial representation, will improve the spatial models required and will lead to significant performance improvements.

- We can also improve the quality of the adaptive spatial shape models by choosing a class-specific subset *(s)* (the number of images that have corresponding object instances) of images for each category instead of using the same number s for all the categories, as it has been from our initial experiments that classes with higher intra-class variability can work better with lower values of s. This will enable us to handle complex classes of articulated objects such as human beings [30, 64], animal species [144], and more general non-rigid shapes.

- Multiple ROI detection. Another thing to bear in mind is the number of object instances that appear in the images. Not only a single object instance can be spatially modeled, a further work would be to automatically detect the regions of all the existing object instances in the images and then learn the most adequate adaptive spatial models that can efficiently represent them.

- Adaptive pyramid weights. At the moment no weighting scheme is introduced within the adaptive pyramid, where each of the foreground and background image representations are equally weighted. We could improve this spatial representation by giving appropriate weighting at each region, which currently we are not taking this into account. We could do pyramid weightings by learning appropriate weight for foreground and background features (as we did for CLW optimization) and, consequently, efficiently using all the information that we have. For example we can choose random weights ranging from 0 to 1 for weighting the adaptive pyramid regions. If the weight is 0 or 1, then we have either foreground or background representation. However if the weight is a value between 0 and 1, let's say 0.3 for foreground and 0.7 for background, both of these pyramid regions will be taken into account while more emphasize is given to the background (contextual knowledge). This will lead to better adaptive image representation for categories which are better characterized by their foreground features (e.g. car, bikes), and also for the other categories which are better characterized by their contextual information (e.g. cows, sheep).

- DITC-based adaptive pyramids. We showed in chapters 3 and 4 an efficient learning approach for learning adaptive spatial pyramids based on the Agglomerative Information Bottleneck *(AIB)* theory. However, due to the recent advances in the information theory, a further work would be to explore the effect of using recent techniques such as *DITC* algorithm, which has shown its effectiveness to obtain compact and discriminative visual words besides its major improvement in-terms of the speed (see chapter 5) for learning adaptive spatial pyramids. In particular, we aim at learning the most discriminative image regions based on the *DITC* algorithm for obtaining fast, compact, yet adaptive spatial image representations and further comparing them to those learned based on the *AIB* algorithm.

- Sparse image representation. Another line of future research includes investigating the application of DITC to sparse image representation [187, 105], which has shown excellent results in a recent work [82]. Although discriminative vocabularies within the context of sparse image representation have been investigated, these methods still ignore the spatial pyramid for the construction of discriminative vocabularies, whereas our work shows that compressing the vocabulary within the spatial pyramid significantly improves results. Therefore, we expect that combining the strengths of both methods will lead to further improvements.

- Large scale image classification. The compact pyramid representation introduced in chapter 5 allows for efficient combination of multiple visual cues. For complex problems, such as large scale image classification and *BoW-based* object detection [89, 68], reducing the computational complexity and memory usage is of paramount importance. In such applications, compact pyramid representations of multiple visual cues will allow to achieve higher classification performance without increasing the complexity. Therefore, we aim at applying the proposed approach for large scale image classification datasets such as *ImageNet*. We are also interested in applying the compact pyramids to the task of object detection. The application of *BoW-based* detection has been advanced due to the efficient sub-window search *(ESS)* algorithm proposed by Lampert et al. [89]. The usage of compact discriminative spatial pyramids for detection applications could help in obtaining faster detection methods without a significant loss in accuracy.

- Unknown categories. The current system is unable to recognize categories which have not been considered in the learning stage. For example if we train the system to recognize cars and motorbikes, and the test image contains a bicycle it would be wrongly classified as car or motorbike. However, the ideal solution would be a system able to say that it does not know the category in the test image.

- Learning compact, adaptive yet discriminative spatial image representation using the methods proposed in this thesis has shown excellent for image classification. However, the method is general so it can be applied in various domains such as action recognition both in videos and in still images, object detection, etc. The problem of finding compact, adaptable yet efficient spatial image representation in these applications is still an open research direction. We do believe that these novel spatial representation can be very beneficial for these tasks as they do contain rich spatial knowledge, which can improve the overall performance.

7.3 Related Publications

Journal Articles:

- Noha Elfiky, Fahad Shahbaz Khan, Joost van de Weijer, and Jordi Gonzàlez. Discriminative Compact Pyramids for Object and Scene Recognition. Pattern Recognition (PR). Volume 45, Issue 4, pages 1627-1636, April 2012.
- Noha Elfiky, Jordi Gonzàlez, and Xavi Roca. Compact Adaptive Pyramids for Scene Recognition. Image and Vision Computing (IVC), (in press 2012).
- Noha Elfiky, Arjan Gijsenij, Theo Gevers, and Jordi Gonzàlez. Color Constancy using 3D Scene Geometry. International Journal of Computer Vision (IJCV), (submitted 2012).

International Conferences:

- Noha Elfiky, Theo Gevers, and Jordi Gonzàlez. Spatial Pyramids derived from 3D Scene Geometry for Improved Object Recognition. European Conference of Computer Vision (ECCV), (submitted 2012).

Book Chapters:

- Noha Elfiky, and Jordi Gonzàlez. Image Description using Local Binary Patterns: Application to Scene Classification. In 2^{nd} CVC Workshop: Progress of Research and Development (CVCRD 2009). Cerdanyola del Vallès, Barcelona, Spain, October 2009.
- Noha Elfiky, and Jordi Gonzàlez. Learning Pyramid Level Specific Vocabulary for Image Classification. In 3^{rd} CVC Workshop: Progress of Research and Development (CVCRD 2010). Cerdanyola del Vallès, Barcelona, Spain, October 2010.
- Noha Elfiky, and Jordi Gonzàlez. Automatic Learning of Pyramid Shapes. In 4^{th} CVC Workshop: Progress of Research and Development (CVCRD 2011). Cerdanyola del Vallès, Barcelona, Spain, October 2011.

Technical Reports:

- Noha Elfiky, and Jordi Gonzàlez. Enhancing Local Binary Patterns using Spatial Pyramid Kernel: Application to Scene Classification. CVC Technical Report, Computer Vision Center (Universitate Autonoma de Barcelona), July (2008).

Appendix A

Termnology and Abbreviations

A.1 Terminology

Many terms related to image classification are used in a somewhat loose manner in the literature so to avoid confusion we give definitions of the terms used in the thesis:

- Image category: Refers to the label for the whole image. Thus we can have coast image (image representing a coast scene) or a dolphin image (image which contains a dolphin).
- Recognition: The classical problem in computer vision, image processing, and machine vision is that of determining whether or not the image data contains some specific object, feature, or activity. This task can normally be solved robustly and without effort by a human, but is still not satisfactorily solved in computer vision for the general case: arbitrary objects in arbitrary situations. Different varieties of the recognition problem are described in the literature, such as: content-based image retrieval, pose estimation, classification, detection, etc.
- Scene/Object classification: Is the task of correctly classifying/recognizing an image category based on the scene or the object it contains. In this thesis, we use the terms classification and recognition interchangeably to refer to the same task.
- Scene/Object annotation. Also scene object labeling: Consists of manually annotating/labeling the images as a kind of the scene or the object they contain.
- Category: Refers to a visually consistent set of scenes or objects.
- Visual word: Is the analogy of the term word of the text analysis. It denotes specific informative parts of an image.
- Visual Vocabulary: It is composed for a set visual words.
- Bag-of-words: Sometimes called bag-of-features. The image is represented as a ”bag of representative features. Each image is further encoded by a binary vector whether it contains certain visual words or not. In a more general way it refers to the visual word histogram of an image.

- Supervised learning: A learning method is called supervised if the method needs the labels of the training images (what kind of scene) and the segmentation and localization of objects in images.

A.2 Abbreviations

Here below we summarize the abbreviations used in the thesis.

AIB: Agglomerative Information Bottleneck

CASP: Compact Adaptive Spatial Pyramid

CBIR: Content Based Image Retrieval

CFW: Class-specific Feature Weight optimization

CLW: Class-specific Level Weight optimization

DITC: Divisive Information Theoretic feature Clustering

GFW: Global Feature Weight optimization

GLW: Global Level Weight optimization

KNN: K-Nearest Neighbour

NIS: Natural Image Statistics

OT: Oliva and Torralba [125] dataset

PHOG: Pyramid Histogram of Orientation Gradients

PHOW: Pyramid Histogram Of visual Words

PC-TPLBP: Pyramid of Colored Three-Patch Local Binary Patterns

pLSA: probabilistic Latent Semantic Analysis

P-rbf: Pyramid-radial basis function kernel

ROI: Region Of Interest

SP: Spatial Pyramid

SVM: Support Vector Machine

VS: Vogel and Schiele [176] dataset

Bibliography

[1] A.Bosch, A.Zisserman, and X.Munoz. Scene classification using a hybrid generative/discriminative approach. *PAMI*, 30(4):712–727, 2008.

[2] A. Agarwal and B. Triggs. Hyperfeatures - multilevel local coding for visual recognition. In *Technical report, INRIA.*, 2005.

[3] T. Ahonen, A. Hadid, and M. Pietikinen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:20372041, 2006.

[4] J.M. Alvarez, T. Gevers, and A.M. Lopez. 3d scene priors for road detection. In *CVPR*, pages 57 –64, 2010.

[5] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *ICML*, 2004.

[6] K. Barnard, V. Cardei, and B. Funt. A comparison of computational color constancy algorithms, part i: Methodology and experiments with synthesized data. *IEEE Transactions on image processing*, 11(9):972–984, 2002.

[7] K. Barnard, L. Martin, A. Coath, and B. Funt. A comparison of computational color constancy algorithms, part ii: Experiments with image data. *IEEE Transactions on image processing*, 11(9):985–996, 2002.

[8] K. Barnard, L. Martin, B. V. Funt, and A. Coath. A data set for color research. *Color Research & Application.*, 27(3):147–151, 2002.

[9] H.G. Barrow, J.M. Tenenbaum, R.c. Bolles, and H.C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *International Joint Conference on Artifficial Intelligence.*, 1977.

[10] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509522, 1988.

[11] A. C. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[12] Brent Berlin and Paul Kay. Basic color terms. their universality and evolution. In *University of California Press, Berkeley, CA*, 1969.

[13] D. Berwick and S.W. Lee. Spectral gradients for color-based object recognition and indexing. *Computer Vision and Image Understanding.*, 94:2843, 2004.

[14] S. Bianco, G.Ciocca, C. Cusano, and R. Schettini. Improving color constancy using indoor-outdoor image classification. *TIP*, 17(12):2381–92, 2008.

[15] D.M. Blei, A. Ng, and M.I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research.*, 3:9931022, 2003.

[16] O. Boiman, I. Rehovot, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *CVPR*, 2008.

[17] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 10(6):849865, 1988.

[18] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *CIVR*, 2007.

[19] A. Bosch, A. Zisserman, and X. Mun oz. Scene classification via plsa. In *European Conference on Computer Vision.*, 2006.

[20] Y-Lan Boureau, Francis Bach, Yann LeCun, and Jean Ponce. Learning mid-level features for recognition. In *CVPR*, 2010.

[21] G. Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute.*, 310(1):126, 1980.

[22] S.D. Buluswar and B.A. Draper. Color machine vision for autonomous vehicles. *International Journal of Engineering Applications of Artificial Intelligence.*, 11(2):245256, 1998.

[23] S.D. Buluswar and B.A. Draper. Color models for outdoor machine vision. *Computer Vision and Image Understanding.*, 85:7199, 2002.

[24] G.J. Burghouts and J. Geusebroek. Performance evaluation of local colour invariants. *CVIU*, 13(113):48–62, 2009.

[25] N.W. Campbell, B.T Thomas, and T. Troscianko. A two-stage process for accurate image segmentation. In *IEEE International Conference on Image Processing.*, 1997.

[26] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 8:679698, 1986.

[27] M Celenk. A color clustering technique for image segmentation. In *Computer Vision,Graphics and Image Processing.*, 1990.

[28] A. Chakrabarti, K. Hirakawa, and T. Zickler. Color constancy beyond bags of pixels. In *CVPR*, 2008.

[29] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines., 2001.

[30] E. Chang, K. Goh, G. Sychay, and G. Wu. Cbsa: Content-based soft annotation for multimodal image retrieval using bayes point machines. *IEEE Transactions on Circuits and Systems for Video Technology Special Issue on Conceptual and Dynamical Aspects of Multimedia Content Description.*, 13(1):2638, 2003.

[31] F. Ciurea and B. Funt. A large image database for color constancy research. In *CIC*, pages 160–164, 2003.

[32] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *European Conference of Computer Vision Workshop on Statistical Learning in Computer Vision.*, 2004.

[33] O.G. Cula and K.J. Dana. Compact representation of bidirectional texture functions. In *Conference on Computer Vision and Pattern Recognition.*, 2001.

[34] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*, 2005.

[35] K.J. Dana, B. van Ginneken, S.K. Nayar, and J.J. Koenderink. Reflectance and texture of real world surfaces. *TOG.*, 18(1):134, 1999.

[36] E. Delage, H. Lee, and A. Y. Ng. A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. In *CVPR*, pages 2418–2428, 2006.

[37] R. Deriche. Using cannys criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision.*, 1(2):167187, 1987.

[38] I. S. Dhillon, S. Mallela, R. Kumar, I. Guyon, and A. Elisseeff. A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research.*, 3:12651287, 2003.

[39] G. Dorko and C. Schmid. Selection of scale-invariant parts for object class recognition. In *ICCV*, 2003.

[40] M. Ebner. Color constancy. In *Wiley*, 2007.

[41] M. Ebner. Color constancy based on local space average color. *Machine Vision and Applications.*, 20(5):283301, 2009.

[42] N. Elfiky, F. Shahbaz Khan, J. van de Weijer, and J. Gonzlez. Discriminative compact pyramids for object and scene recognition. *Pattern Recognition.*, 45(4):1627–1636, 2012.

[43] M. Everingham, L. Van Gool, C. K. I.Williams, J.Winn, and A. Zisserman. The PASCAL Visual Object Classes challenge 2007 results.

[44] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes challenge 2008 (voc2008) results.

[45] R. Fergus F. Li and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *In Workshop on Generative-Model Based Vision, CVPR*, page 178, 2004.

[46] L. Fei-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *International Conference on Computer Vision*, 2003.

[47] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*, 2005.

[48] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

[49] G. Finlayson, S. Hordley, and C. Lu. On the removal of shadows from images. *TPAMI*, 28(1):59–68, 2006.

[50] G.D. Finlayson, S.D. Hordley, and P.M. Hubel. Color by correlation: a simple, unifying framework for color constancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 23(11):12091221, 2001.

[51] G.D. Finlayson, S.D. Hordley, and I. Tastl. Gamut constrained illuminant estimation. *International Journal of Computer Vision.*, 67(1):93109, 2006.

[52] G.D. Finlayson and E. Trezzi. Shades of gray and colour constancy. In *IST/SDI's Color Imaging Conference. IST-The Society for Imaging Science and Technology.*, 2004.

[53] D.A. Forsyth. A novel algorithm for color constancy. *International Journal of Computer Vision.*, 5(1):536, 1990.

[54] B. Fulkerson, A. Vedaldi, and S. Soatto. Localizing objects with smart dictionaries. In *European Conference on Computer Vision.*, 2008.

[55] D. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In *International Conference on Computer Vision*, 1999.

[56] P. V. Gehler and S. Nowozin. Let the kernel figure it out:principled learning of preprocessing for kernel classifiers. In *CVPR*, 2009.

[57] P. V. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *International Conference on Computer Vision*, 2009.

[58] T. Geodeme, T. Tuytelaars, G. Vanacker, M. Nuttin, and L. Van Gool. Omnidirectional sparse visual path following with occlusion-robust feature tracking. In *In OMNIVIS Workshop, International Conference on Computer Vision.*, 2005.

[59] J. Geusebroek, R. van den Boomgaard, and A. Smeulders. Color invariance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):12281350, 2001.

[60] J.M. Geusebroek and A.W.M. Smeulders. A six-stimulus theory for stochastic texture. *IJCV.*, 62(1-2):7–16, 2005.

[61] A. Gijsenij and T. Gevers. Color constancy using natural image statistics. In *CVPR*, 2007.

[62] A. Gijsenij and T. Gevers. Color constancy using natural image statistics and scene semantics. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 33(4):687–698, 2011.

[63] A. Gijsenij, Th. Gevers, and J. van de Weijer. Computational color constancy: Overview and experiments. *IEEE Trans. on Image Processing (TIP).*, 20(9):2475–2489, 2011.

[64] R.C Gonzalez and R.E Woods. Digital image processing. In *Addison-Wesley.*, 1993.

[65] R.C Gonzalez and R.E Woods. Digital image processing. In *Addison-Wesley, 3rd edition, Boston.*, 2001.

[66] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *International Conference on Computer Vision*, 2005.

[67] H. and Bay. Surf: Speeded up robust features. In *European Conference on Computer Vision.*, 2006.

[68] Hedi Harzallah, Frederic Jurie, and Cordelia Schmid. Combining efficient object localization and image classification. In *ICCV*, 2009.

[69] E. Hayman, B. Caputo, M. Fritz, and J.-O. Eklundh. On the significance of real-world conditions for material classification. In *European Conference on Computer Vision.*, 2004.

[70] A. Hegerath, T. Deselaers, and H. Ney. Patch-based object recognition using discriminatively trained gaussian mixtures. In *British Machine Vision Conference*, volume 2, page 519528, 2006.

[71] M. Heikkil and M. Pietikinen. A texture-based method for modeling the background and detecting moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 28(4):657662, 2006.

[72] M. Heikkil, M. Pietikinen, and C. Schmid. Description of interest regions with local binary patterns. *Pattern Recognition.*, 42(3):425436, 2009.

[73] M. Heikkil, M. Pietikinen, and Cordelia Schmid. Description of interest regions with local binary patterns. *Pattern Recogn.*, 42(3):425–436, 2009.

[74] Marko Heikkil and Matti Pietikinen. A texture-based method for modeling the background and detecting moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):657–662, 2006.

[75] R. Henry, S. Mahadev, S. Urquijo, and D. Chitwood. Color perception through atmospheric haze. *J. Opt. Soc. Am. A*, 17(5):831–835, 2000.

[76] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning.*, 41(2):177196, 2001.

[77] D. Hoiem, A. A. Efros, , and M. Hebert. Geometric context from a single image. In *ICCV*, pages 654–661, 2005.

[78] G. Hordley. Scene illuminant estimation:past, present, and future. *Color Res. and App.*, 31(4):303–314, 2006.

[79] N. Ikizler-Cinbis and S. Sclaroff. Object, scene and actions: Combining multiple features for human action recognition. In *ECCV (1)*, pages 494–507, 2010.

[80] S. Ito and S. Kubota. Object classification using heterogeneous co-occurrence features. In *ECCV*, 2010.

[81] J. Shepherd J. Shen and A. H. H. Ngu. Semantic-sensitive classification for large image libraries. In *International Multimedia Modelling Conference*, 2005.

[82] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for sparse hierarchical dictionary learning. In *ICML*, 2010.

[83] Y. Ke and Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.

[84] Fahad Shahbaz Khan, Joost van de Weijer, and Maria Vanrell. Top-down color attention for object recognition. In *ICCV*, 2009.

[85] Kihwan Kim, M. Grundmann, A. Shamir, I. Matthews, J. Hodgins, and I. Essa. Motion fields to predict play evolution in dynamic sport scenes. In *CVPR*, pages 840 –847, 2010.

[86] S. Kumar, A. C. Loui, and M. Hebert. An observation-constrained generative approach for probabilistic classification of image regions. *Image and Vision Computing.*, 21:8797, 2003.

[87] L. Kuncheva and J. Bezdek. Presupervised and postsupervised prototype classifier design. *IEEE Trans. on Neural Networks.*, 10(5):1142–1152, 1999.

[88] Y. Lamdan, J. T. Schwartz, and H. J. Wolfson. Object recognition by affine invariant matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*, 1988.

[89] C.H. Lampert, M.B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *CVPR*, 2008.

[90] Christoph H. Lampert, Matthew B. Blaschko, and Thomas Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *PAMI*, 2009.

[91] E.H. Land. The retinex theory of color vision. *Scientific American.*, 237(6):108128, 1977.

[92] I. Laptev. Improvements of object detection using boosted histograms. In *British Machine Vision Conference*, volume 3, page 949958, 2006.

[93] D. Larlus and F. Jurie. Latent mixture vocabularies for object categorization and segmentation. *IMAVIS*, 27(5):523–534, 209.

[94] S. Lazebnik and M. Raginsky. Supervised learning of quantizer codebooks by information loss minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 31(7):12941309, 2009.

[95] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using affine-invariant regions. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

[96] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *PAMI*, 27(8):1265–1278, 2005.

[97] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, page 21692178, 2006.

[98] B. Leibe, K. Micolajckzyk, and B. Schiele. Efficient clustering and matching for object class recognition. In *BMVC.*, 2006.

[99] B. Leibe and B. Schiele. Scale-invariant object categorization using a scale-adaptive means-shift search. In *DAGM.*, 2004.

[100] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision.*, 43(1):2944, 2001.

[101] Li-Jia Li and Li Fei-Fei. What,where and who? classifying events by scene and object recognition. In *ICCV*, 2007.

[102] Li-Jia Li, Richard Socher, and Li Fei-Fei. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *CVPR*, 2009.

[103] J. Liu and M. Shah. Scene modeling using co-clustering. In *International Confernece on Computer Vision.*, 2007.

[104] D. Lowe. Distinctive image features from scale invariant keypoints. *International Journal of Computer Vision.*, 60(2):91–110, 2004.

[105] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *ICML*, 2009.

[106] Subhransu Maji, Alexander C. Berg, and Jitendra Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR*, 2008.

[107] M. Marszalek, C. Schmid, H. Harzallah, and J. van de Weijer. Learning object representation for visual object class recognition. In *Visual recognition Challenge Workshop, in conjuncture with ICCV*, 2007.

[108] Marcin Marszalek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *CVPR*, 2009.

[109] J. Mart, J. Freixenet, J. Batlle, and A. Casals. A new approach to outdoor scene description based on learning and top-down segmentation. *Image and Vision Computing.*, 19(1):10411055, 2001.

[110] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 26(5):530549, 2004.

[111] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision.*, 60(1):6386, 2004.

[112] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 27(10):16151630, 2005.

[113] K. Mikolajczyk, t. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2005.

[114] Alastair P. Moore, Simon J. D. Prince, Jonathan Warrell, Umar Mohammed, and Graham Jones. Superpixel lattices. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[115] H. Mori, K. Kobayashi, N. Ohtuki, and S. Kotani. Color imipression factor: an image understanding method for outdoor mobile robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems.*, 1993.

[116] T. Menp and M. Pietikainen. Classification with color and texture: jointly or separately? *Pattern Recognition*, 37(8):1629–1640, 2004.

[117] H. Nakayama, T. Harada, and Y. Kuniyoshi. Global gaussian approach for scene categorization using information geometry. In *CVPR*, 2010.

[118] Chetan Nandakumar and Jitendra Malik. Understanding rapid category detection via multiple degraded images. *Journal of Vision.*, 9(19):18, 2009.

[119] V. Nedovic, A. Smeulders, A. Redert, and J.-M. Geusebroek. Depth information by stage classification. In *International Conference on Computer Vision*, 2007.

[120] Eric Nowak, Frederic Jurie, and Bill Triggs. Sampling strategies for bag-of-features image classification. In *European Conference on Computer Vision*, 2006.

[121] Sangmin Oh, Anthony Hoogs, Matthew W. Turek, and Roderic Collins. Content-based retrieval of functional objects in video using scene context. In *ECCV (1)*, pages 549–562, 2010.

[122] Y. Ohta, T. Kanade, and T. Sakai. Color information for region segmentation. In *Computer Graphics and Image Processing.*, 1980.

[123] T. Ojala, M. Pietikinen, and T. Menp. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 24(7):971987, 2002.

[124] T. Ojala, M. Pietikinen, and T. Menp. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *PAMI*, 24(7):971–987, 2002.

[125] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[126] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42:145–175, 2001.

[127] C.F. Olson and D.P. Huttenlocher. Automatic target recognition by oriented edge pixels. *IEEE Transactions on Image Processing.*, 6(1):103113, 1997.

[128] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *European Conference on Computer Vision.*, 2006.

[129] S. Paek and S-F. Chang. A knowledge engineering approach for image classification based on probabilistic reasoning systems. In *IEEE International Conference on Multimedia and Expo.*, 2000.

[130] Marius V. Peeleen, Li Fei-Fei, and Sabine Kastner. Neural mechanisms of rapid natural scene categorization in human visual cortex. *Nature.*, page 9497, 2009.

[131] A. Perina, M. Cristani, U. Castellani, V. Murino, and N. Jojic. A hybrid generative/discriminative classification framework based on free-energy terms. In *ICCV*, 2009.

[132] Florent Perronnin, Jorge Snchez, and Yan Liu. Large-scale image categorization with explicit data embedding. In *CVPR*, 2010.

[133] V. Perronnin, C. Dance, G. Csurka, and M. Bressan. Adapted vocabularies for generic visual categorization. In *European Conference on Computer Vision*, 2006.

[134] D. Purves, R.G. Lotto, and S. Nundy. Why we see what we do. *American Scientist.*, 90(3):236243, 2002.

[135] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *Conference on Computer Vision and Pattern Recognition.*, 2009.

[136] P. Quelhas, F. Monay, J.M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. Van Gool. Modeling scenes with local descriptors and latent aspects. In *International Conference on Computer Vision*, 2005.

[137] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. More efficiency in multiple kernel learning. In *ICML*, 2007.

[138] C. Rothwell, A. Zisserman, D. Forsyth, and J. Mundy. Planar object recognition using projective shape representation. *International Journal of Computer Vision.*, 16(2):57–99, 1995.

[139] Lu Rui, A. Gijsenij, T. Gevers, V. Nedovic, Xu De, and J.M. Geusebroek. Color constancy using 3d scene geometry. In *International Conference on Computer Vision.*, 2009.

[140] G. Salton. Automatic information organization and retrieval. 1968.

[141] J. Sanchez, X. Binefa, and J. Vitria. Shot partitioning based recognition of tv commercials. *Multimedia Tools Applications.*, 18(3):233247, 2002.

[142] C. Schuldt, I. Laptev, , and B. Caputo. Recognizing human actions: A local svm approach. In *International Conference of Pattern Recognition.*, 2004.

[143] William Robson Schwartz, Aniruddha Kembhavi, David Harwood, and Larry S. Davis. Human detection using partial least squares analysis. In *ICCV*, 2009.

[144] N. Serrano, A.E. Savakis, and J. Luo. Improved scene classification using efficient low-level features and semantic cues. *Pattern Recognition.*, 31(37):17731784, 2004.

[145] Eli Shechtman and Michal Irani. Matching local self-similarities across images and videos. In *CVPR*, 2007.

[146] J. Shen and S. Castan. An optimal linear operator for step edge detection. *Computer Vision, Graphics and Image Processing.*, 54(2):112133, 1992.

[147] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. *International Conference on Computer Vision.*, 1:503510, 2005.

[148] N. Shroff, P. Turaga, and R. Chellappa. Moving vistas: Exploiting motion for describing scenes. In *CVPR*, pages 1911 –1918, 2010.

[149] J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, and W. T Freeman. Discovering objects and their locations in images. In *International Conference on Computer Vision.*, 2005.

[150] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision.*, 2003.

[151] S.Lazebnik and M.Raginsky. Learning nearest-neighbor quantizers from labeled data by information loss minimization. In *Confernece on Artificial Intellligence and Statistics.*, 2007.

[152] N. Slonim and N. Tishby. Agglomerative information bottleneck. In *Neural Information Processing Systems.*, 1999.

[153] A.R Smith. Color bamut transform pairs. *Computer and Graphics.*, 12(3):1219, 1978.

[154] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Depth from familiar objects: A hierarchical model for 3d scenes. In *CVPR*, pages 2410–2417, 2006.

[155] M. Szummer and R. W. Picard. Indoor-outdoor image classification. In *ICCV Workshop on Content-based Access of Image and Video Databases*, pages 42–51, 1998.

[156] Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. Hierarchical dirichlet process. In *Neural Information Processing Systems.*, 2005.

[157] J.N. Tenenbaum. An interactive facility for scene analysis research. In *Technical report, Stanford Research Institute.*, 2006.

[158] J. Thureson and S. Carlsson. Appearance based qualitative image description for object class recognition. In *European Conference on Computer Vision*, 2004.

[159] S. Todorovic and N. Ahuja. Learning subcategory relevances for category recognition. In *CVPR*, 2008.

[160] A. Torralba. Contextual priming for object detection. *Inetnational Journal of Computer Vision.*, 53(2):169–191, 2003.

[161] A. Torralba and A. Oliva. Depth estimation from image structure. *TPAMI*, 24(9):1226–1238, 2002.

[162] A. Vailaya, A. Figueiredo, A. Jain, and H. Zhang. Image classification for content-based indexing. *IEEE Transactions on Image Processing.*, 10(1):117130, 2002.

[163] A. Vailaya, M. Figueiredo, A. Jain, and H.J. Zhang. Content-based hierarchical classification of vacation images. In *IEEE International Conference on Multimedia Computing and Systems*, 1999.

[164] A. Vailaya, A. Jain, and H. Zhang. On image classification: City vs. landscapes. *Pattern Recognition.*, 31(12):19211935, 1998.

[165] K. van de Sande, T. Gevers, and C. Snoek. Evaluation of color descriptors for object and scene recognition. In *CVPR*, 2008.

[166] Koen E. A. van de Sande, Theo Gevers, and Cees G. M. Snoek. Evaluating color descriptors for object and scene recognition. *PAMI*, 32(9):1582–1596, 2010.

[167] J. van de Weijer, T. Gevers, and A. Gijsenij. Edge-based color constancy. *TIP*, 16(9):2207–2214, 2007.

[168] J. van de Weijer and Th. Gevers. Robust optical flow from photometric invariants. In *ICIP*, pages 1835–1838, 2004.

[169] J. van de Weijer, C. Schmid, Jakob J. Verbeek, and D. Larlus. Learning color names for real-world applications. *TIP*, 18(7):1512–1524, 2009.

[170] Joost van de Weijer, Theo Gevers, and Andrew D. Bagdanov. Boosting color saliency in image feature detection. *PAMI.*, 28(1):150–156, 2006.

[171] M. Varma and D. Ray. Learning the discriminative powerinvariance trade-off. In *ICCV*, 2007.

[172] M. Varma and A. Zisserman. Classifying images of materials: Achieving viewpoint and illumination independence. In *European Conference on Computer Vision.*, 2002.

[173] M. Varma and A. Zisserman. Texture classification: Are filter banks necessary? In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

[174] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009.

[175] Andr Redert Vladimir Nedovi, Arnold W. M. Smeulders and Jan-Mark Geusebroek. Stages as models of scene geometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(9):1673–1687, 2010.

[176] J. Vogel and B. Schiele. Natural scene retrieval based on a semantic modeling step. In *International Conference on Image and Video Retrieval*, volume 3115, pages 207–215, 2004.

[177] J. Vogel and B. Schiele. Semantic modeling of natural scenes for content-based image retrieval. *International Journal of Computer Vision.*, 72(2):133157, 2007.

[178] Zhengxiang Wang, Yiqun Hu, and Liang-Tien Chia. Image-to-class distance metric learning for image classification. In *ECCV*, 2010.

[179] J. Weijer and C. Schmid. Coloring local feature extraction. In *European Conference on Computer Vision.*, 2006.

[180] Joost Van Weijer and C. Schmid. Applying color names to image description. In *ICIP*, 2007.

[181] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *International Conference on Computer Vision.*, 2005.

[182] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. in real-life images. In *European Conference on Computer Vision.*, 2008.

[183] J. Wu and J.M. Rehg. Beyond the euclidean distance: Creating effective visual codebooks using the histogram intersection kernel. In *ICCV*, 2009.

[184] Jianxin Wu. A fast dual method for hik svm learning. In *ECCV*, 2010.

[185] Nianhua Xie, Haibin Ling, Weiming Hu, and Xiaoqin Zhang. Use bin-ratio information for category and scene classification. In *CVPR*, 2010.

[186] D. Yagi, K. Abe, and H. Nakatami. Segmentation of color aerial photographs using hsv color models. In *IAPR Workshop on Machine Vision Applications.*, 1992.

[187] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[188] Jianchao Yang, Kai Yu, and Thomas Huang. Efficient highly over-complete sparse coding using a mixture model. In *ECCV*, 2010.

[189] L. Yeffet and L. Wolf. Local trinary patterns for human action recognition. In *International Conference on Computer Vision*, 2009.

[190] H. Zhang, A. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

[191] J. Zhang, M. Marszaek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, 2007.