Scene Image Classification and Segmentation with Quantized Local Descriptors and Latent Aspect Modeling

THÈSE N^o 3743 (2007)

PRÉSENTÉ A LA FACULTE SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
Institut de traitement des signaux

SECTION DE GÉNIE ÉLECTRIQUE ET ÉLECTRONIQUE
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Pedro Manuel da Silva Quelhas

Master by research in Image Processing and X-Ray Physics,
King's College, London, UK

et de nationalité portugais

acceptée sur proposition du jury:

Thesis committee members:

Prof. Hervé Bourlard, directeur de thèse

Dr. Jean-Marc Odobez, co-directeur de thèse

Prof. Bernt Schiele, rapporteur, Darmstadt University of Technology, Darmstadt, Germany

Dr. Nicu Sebe, rapporteur, University of Amsterdam, Amsterdam, The Netherlands

Prof. Pierre Vandergheynst,rapporteur, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland

xx, EPFL, Switzerland

Ecole Polythechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
Décembre 2006

Abstract

The ever increasing number of digital images in both public and private collections urges on the need for generic image content analysis systems. These systems need to be capable to capture the content of images from both scenes and objects, in a compact way that allows for fast search and comparison. Modeling images based on local invariant features computed at interest point locations has proven in recent years to achieve such capabilities and to provide a robust and versatile way to perform wide-baseline matching and search for both scene and object images.

In this thesis we explore the use of local descriptors for image representation in the tasks of scene and object classification, ranking, and segmentation. More specifically, we investigate the combined use of text modeling methods and local invariant features.

Firstly, our work attempts to elucidate whether a text like bag-of-visterms representation (histogram of quantized local visual features) is suitable for scene and object classification, and whether some analogies between discrete scene representations and text documents exist. We further explore the bag-of-visterms approach in a fusion framework, combining texture and color information for natural scene classification.

Secondly, we investigate whether unsupervised, latent space models can be used as feature extractors for the classification task and to discover patterns of visual co-occurrence. In this direction, we show that Probabilistic Latent Semantic Analysis (PLSA) generates a compact scene representation, discriminative for accurate classification, and more robust than the bag-of-visterms representation when less labeled training data is available. Furthermore, we show through aspect-based image ranking experiments, the ability of PLSA to automatically extract visually meaningful scene patterns, making such representation useful for browsing image collections.

Finally, we further explore the use of the latent aspect modeling in an image segmentation task. By extending the representation resulting from the latent aspect modeling, we are able to introduce contextual information for image segmentation that goes beyond the traditional regional contextual modeling found for instance in Markov Random Field approaches.

Keywords: Scene classification, image modeling, latent aspect modeling, contextual segmentation modeling, quantized local descriptors.

Resume

La disponibilité croissante d'images digitales dans les domaines publiques et privés rend de plus en plus nécessaire la création des systèmes automatiques d'analyse de contenu d'images. Ces systèmes doivent avoir la capacité d'analyser le contenu des images à partir des scène globale et des objects, de façon compacte permettant une analyse rapide. La modélisation d'images basée sur certaines characteristiques invariantes locales, calculées en quelques points informatifs, a récemment prouvée être une méthodologie robuste et adaptable de recherche et de comparaison de scènes et d'objets dans des images.

Dans cette thèse, nous explorons l'utilisation de descripteurs locaux pour la classification, la hiérarchisation selon certains critères de similarités et la segmentation de scènes et d'objets dans des images. Plus précisement, nous faisons des investigations a propos de l'utilisation combinée de méthodes de modélisation de textes et des caractéristiques invariantes locales.

Premièrement, ce travail a pour objectif d'élucider dans quelles mesures, une représentation en sac-de-terme-visuels empruntée au domaine de la modélisation de texte, est appropriée pour la classification de scènes et d'objets. Dans nos investigations, nous vérifierons l'existence de similarités entre une représentation discrète de scènes et d'objets dans une image et un texte. De plus, nous explorons l'approche représentation en sac-de-terme-visuels a travers une méthodologies jointe, combinant informations de texture et information de couleur pour la classification de scènes naturelles.

Deuxièmement, nous étudions l'utilisation de modeles d'espaces latent non-supervisés pour l'extraction de caractéristiques pour une tache de classification et pour la découverte d'occurrence jointe de structures visuelles. Dans cette étude, nous montrons que l'analyse probabiliste de sémantique latente génère une représentation de compacte scène efficace pour des taches de classification de pointe, et plus robuste que que la représentation en sac-de-terme-visuels lorsque l'ensemble des données d'entraînement annotés est petit. De plus, nous montrons, a travers la hiérarchisation d'image, basée sur l'aspect, de la capacité de l'analyse probabiliste de sémantique latente d'extraire automatiquement des structures scéniques visuelles sensées, rendant cette représentation utile pour la recherche d'images dans des bases de données.

Pour finir, nous explorons l'utilisation de la modélisation d'aspects latents pour une tache de segmentation d'images. En généralisant la représentation résultant de la modélisation d'aspects

latents, nous sommes en mesure de produire de l'information contextuelle plus riche que celle produite par des approches classiques telles que celles basées sur les champs de Markov.

Mots-clés : Classification de scène, modélisation de la image, modeles d'espaces latent non-supervisés, segmentation contextuelle, quantification des descripteurs locaux.

Acknowledgements

'I did it!'
- Hiro Nakamura

'The most exciting phrase to hear in science, the one that heralds new discoveries, is not 'Eureka!' (I found it!) but 'That's funny ...'
- Isaac Asimov

I want to thank my family, I could never have done this work without their love and support. Even from far away they were always my greatest supporters.

I would like to thanks my supervisor Jean-Marc Odobez for supervising my work, for all the discussions and all the exchange of ideas that lead to this thesis. I would also like to thank Daniel Gatica-Perez for contributing to many of the ideas presented in this thesis and for always having time to talk. Thanks also to Florent for the collaboration in much of this work, which made it so much more interesting.

To Mael, Kevin, Florent and Mark who have shared an office with me, thanks for the companionship and the brainstorming discussions that resulted in many ideas, some of which are part of this thesis. To Mike Flynn, Mike Perow, Anna, Michael, John, Agnes, Siley, Ronan and Sammy, thanks for always being willing to take a break and discuss anything, work related or otherwise.

A special thanks too all the friends I made during my time in Martigny, you guys are great, I will never forget you.

Contents

Chapter 1

Introduction

THE need for image content analysis systems continues to increase due to the ever growing number of digital images in both public and private collections. The need to organize digital images based on their content is essential since any digital image collection is only usable if a user can extract from it the desired content. However, in the general case, we know little about which images are contained in a given digital image collections. Images may vary in size (resolution), quality (compression or post acquisition processing) and more importantly semantic visual content. This makes image content analysis one of the great challenges of computer vision with great importance for image data management and the advance of the field of artificial intelligence. In an image content analysis system one important goal is to bridge the semantic gap between the image's visual content and the labels we want to attribute to each image. While matching the performance of humans in interpreting the semantic content of images is still beyond the capabilities of current computer vision systems, the aim of this thesis is to propose new techniques to achieve those goals. However, due to the large diversity of image content it is important to follow principled approaches to address the content modeling of these images as well as their retrieval

If we consider as an example the personal images and video collections of an average user, there are different broad types of images that can be of interest. Figure 1.1 illustrates some of these images types that apply to most of the images found in a personal image and video collection. Each of these types poses different challenges for image content analysis methodology.

When analyzing an image, we can interpret its semantic content in a general way, or we can recognize the specific content of that image. For example, we may say that an image contains a mountain, or we may say that the image is from 'Mont Blanc' in the French Alps. The same can be said for object. For example, we may say that an image contains a car, or we may say that an image if from a '1967 ford mustang'. For the task of image content analysis, both generic and specific levels of recognition are useful, as we never know what the user is searching for, and both of them have been investigated in the past. Similarly, interpreting an image by recognizing

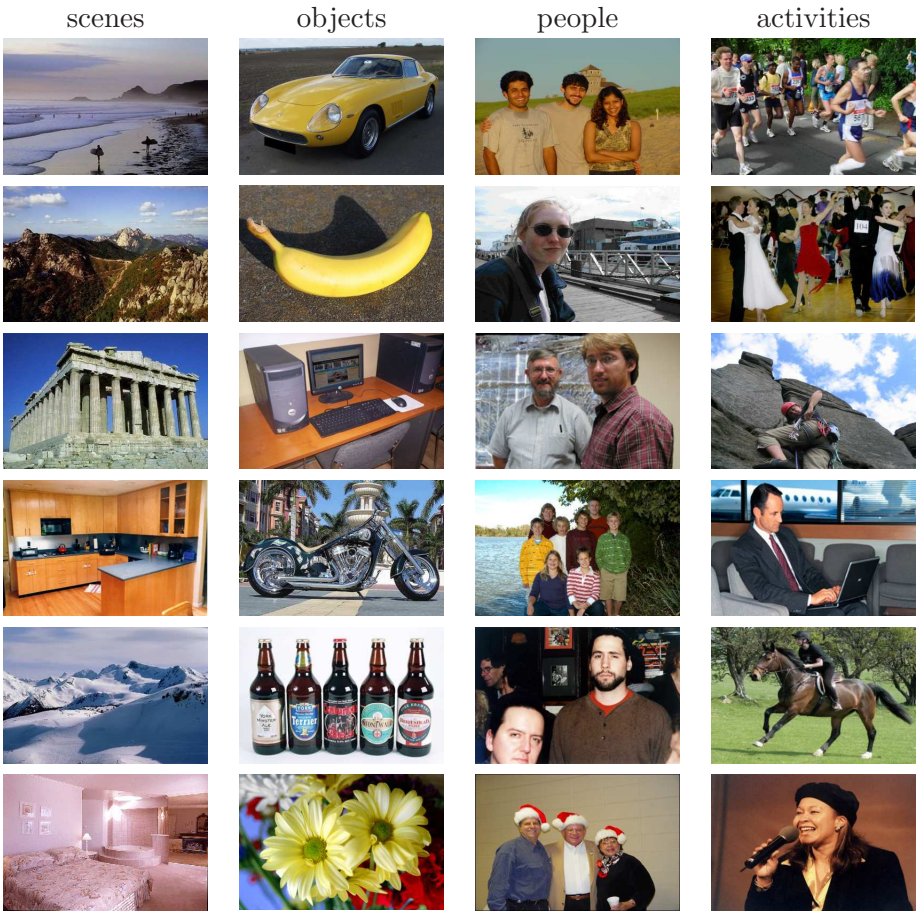


Figure 1.1. Examples of different types of images which can be commonly found in a digital collection of an average user. From left column to right column: scene, objects, people, and activities. Each type of image poses different challenges (not necessarily disjoint) for computer vision methods: scene classification/recognition, object classification/recognition, face and pedestrian detection/recognition, and activity recognition.

both objects and scene types illustrated in an image can provide us with valuable information to understand the semantic content of each image. To address all the above interpretation issues, the use of image modeling based on local features has provided for a significant progress in terms of the robustness, versatility, efficiency and quality of results. In this thesis we will thus focus on the problems related to image representation for scene and object classification, segmentation and retrieval, based on local features.

In this chapter we first present the principal tasks and issues related to the work done during the thesis. Then we present our contributions and give an overview of the remaining manuscript.

1.1 Tasks

The design and validation of representations based on quantized local interest point descriptors (visterms), for the task of scene and object classification, is the central theme of this thesis. We explore both a histogram of visterms approach, entitled bag-of-visterms (BOV), and a latent aspect modeling representation based on that same representation. However, as we will see, using the same representations we can perform other more complex tasks, like segmentation and ranking or retrieval. In this section we discuss the object and scene classification tasks, among other tasks that are of great importance for the managing of digital image libraries. In what follows, we will discuss the uses, image types, and issues of each particular task.

1.1.1 Content-based image retrieval (CBIR)

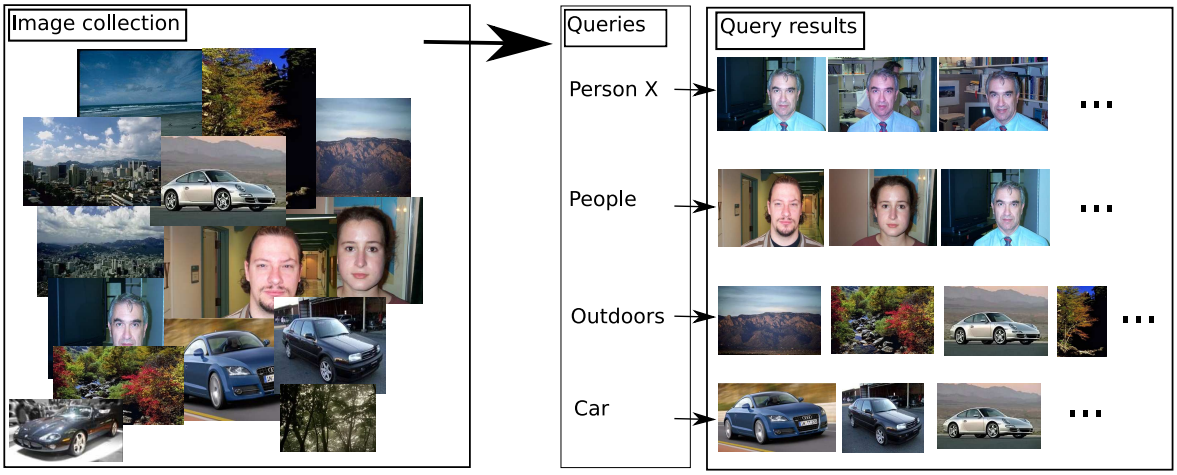


Figure 1.2. Example of typical image queries in an image collection.

Content-based image retrieval is the task of retrieving relevant images from a large collection on the basis of a user query. In large commercial digital image collections, retrieval is a very important task to solve. An inefficient retrieval system may retrieve irrelevant images and cause the loss of a client. One example of a commercial digital image collections is the CORBIS stock photo library (http://pro.corbis.com/), see Figure 1.3 for a screen capture of the web query interface. Companies that maintain commercial stock photography collections attempt to sell image rights for use in publications or publicity. Depending on the particular image and the envisioned use, the price for such rights can reach considerable amounts. CBIR is a very difficult task since, in most cases, the images in large commercial digital image collection have a large variability of content. For this reason most commercial digital image collection rely on retrieval systems based on image annotation, done by trained operators and user textual search.

In the case of personal digital image collections, there has been a rising interest in content based

query systems. This is due to the amount of images that an average digital camera owner stores in his computer, which has increased dramatically in the last few years. Contrary to commercial collections, in personal collections the user usually neglects a proper organization or annotation of his images. Since there is no proper structure or annotation to rely on, queries based on content would be the most appropriate approach to search for images in the collection. The capability to search for a particular person or location, among other queries, would be welcome by most digital camera users.

In general, content-based image retrieval can be performed based on several types of user inputs: query by example, query by sketch, and semantic retrieval. Retrieval using the query by example (Sivic and Zisserman (2003); Smeulders *et al.* (2000); Carson *et al.* (1999)) has seen some success. This approach is based on the extraction of low level features from all images in the collection and the query image. These features provide a similarity ranking between the query example and the images in the collection, which enables the retrieval of the most relevant images. However, the need for an image which is similar to the image we want to retrieve is a difficult requirement to meet, and this limits the use of this approach. One solution to eliminate the need for an example image is the query by sketch (Bimbo and Pala (1997)) approach. In this case the query is performed by a user provided sketch, which can be changed to refine the query. However, the need for a sketch is still not always easy to satisfy. The ideal CBIR system from a user perspective would involve semantic retrieval, where the user makes a request like "find pictures of Paris in the winter". However, this is a more difficult system to implement using current computer vision technology.

1.1.2 Image browsing

An image browsing system enables the user to navigate from image to image, or from group of images to group of images, according to a particular element of interest (time, location, image content,...). Browsing through a collection of images can be a very difficult task depending on the organization and type of the data contained in the collection. In the past, generic image browsing systems have been considered as computer programs that display stored images, handling various graphics file formats. With this definition, the browsing was often limited to a certain arbitrary order of the images, generally dictated by the directory structure. However, in recent years, more interesting approaches for browsing image collections have been proposed. Most of them exploit the pictorial content statistics of one image, or a group of images, or the whole collection, to navigate through the collection (Barecke *et al.* (2006); Carson *et al.* (1999)). These systems are somewhat related to the tasks of retrieval, since given the currently browsed image, or group of images, we want to navigate to related images. This type of browsing can also be seen as an iterative search for an image within the collection, in which the user chooses the next image based on the current viewed image, towards a final objective of retrieving a specific image. This relates to retrieval with user feedback, in which the actions of the user help the system in the retrieval process. In the case of large professional content data libraries, browsing occurs after an initial retrieval step based on a query. Browsing images relates heavily to the

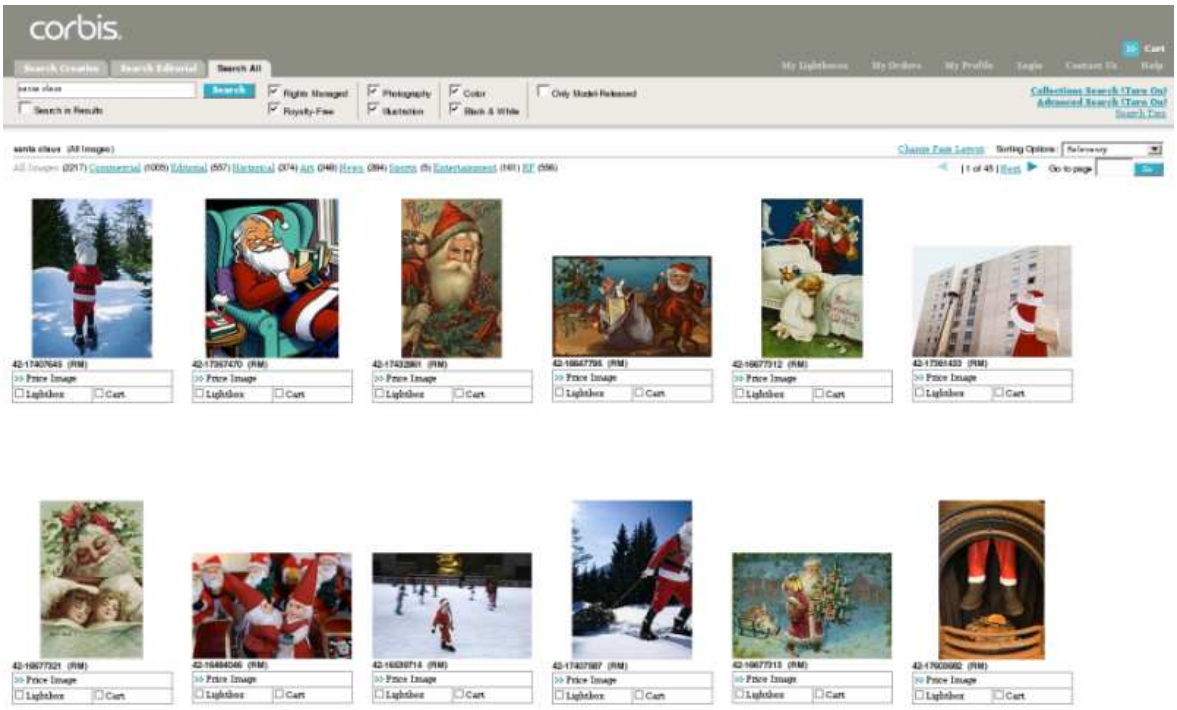


Figure 1.3. CORBIS stock photography web interface with results for the query 'Santa Claus'.

computation of distances between any two images, as well as the finding 'principal' axis in the collection.

One particular instantiation of this browsing methodology is entitled hyper-linking, and has become very popular in recent years. In hyper-linking, the user can click on (or select) part of the current image to navigate to other images which have a similar content to the selected part. For example given the image with a certain person, a user may select that person's face and indicate that he/she wants to browse images of that particular person. This is currently available in some systems (i.e. http://www.riya.com/) based on manual annotation. However, recent studies have shown that to some extent, this could be done automatically. For instance, using multi-dimensional color histograms (Gatica-Perez *et al.* (2002)) or local descriptors (Sivic and Zisserman (2003)), hyper-linking within a video have been demonstrated.

The effectiveness of a browsing system is very difficult to assess since the task is not well defined in quantitative terms. In some cases, performance can be measured by the average time spent by a user on specific searches, across different browsing systems. This provides a quantitative comparison between systems. Performance, as measured by the retrieval speed is valuable for commercial applications. However, in the case of personal user browsing their collections the focus is on user satisfaction, which is much more difficult to assess.

1.1.3 Object recognition

Object recognition is a fundamental task of computer vision, where given an image we try to automatically recognize the identity of the object visual represented in the image. This task relates to the basis of artificial intelligence where we try to understand the human cognitive system. Originally, in computer vision, object recognition studies were conducted using clutter free setups. Databases like Coil-100 (Nene *et al.* (1996)) were constructed by displaying a single object per image, taken from different points of view. In this context, the task was to model each individual object, based on one or more images, and recognize the other occurrences of that same object in other images. This is clearly different from the task of recognizing objects in non specialized personal or commercial digital image collections, where both the localization and recognition of objects have to be performed in parallel. In addition, while most of these early research in computer vision focused on recognizing specific object instances, which could to a great extent rely on image or feature matching, recent studies address the more fundamental task of detecting and recognizing object categories.

Object recognition in cluttered real-world scenes requires local image features that are unaffected by nearby clutter or partial occlusion. In a real-world scene objects appear in a cluttered environment, and the task of object recognition becomes a conjugation of object detection and recognition. To handle cluttered environments, we must at the same time identify the location and recognize the object, since these two tasks are depending on each other. Besides handling clutter, the object representation must also handle changes in the object's appearance. The features for such task must be at least partially invariant to illumination, 3D projective transforms, and other common object variations. On the other hand, the features must also be sufficiently distinctive to identify any specific object among all possible objects. The difficulty of the object recognition problem is due in large part to the lack of success in finding such image features, along with appropriate representations.

1.1.4 Scene classification

Scene classification is an important task in computer vision. Like object recognition it tries to recognize the visual content of the image. However scene classification is different from object recognition. On one hand, images of a given object are usually characterized by the presence of a limited set of specific visual parts, tightly organized into different view-dependent geometrical configurations. On the other hand, a scene is generally composed of several entities (e.g. car, house, building, face, wall, door, tree, forest, rocks), organized in often unpredictable layouts. Hence, the visual content (entities, layout) of a specific scene class exhibits a large variability, characterized by the presence of a large number of different visual descriptors. In scene classification, we attempt to give a label to the context of the visual content of each specific image. Examples of labels can be specific like 'Mont Blanc' or more general like 'mountain' or 'landscape'.

Scene classification is a difficult problem, interesting in its own right, but also as a means to provide contextual information to guide other processes such as object recognition (Torralba *et al.* (2003)). From the application viewpoint, scene classification is relevant in systems for organization of personal and professional image and video collections. In many occasions a user searches for a specific scene, like a geographic location or a scene type, like beach or bar. It can also occur that the user may search for an object in a specific context (scene), like searching for a family member at the beach. In some cases, scenes may be difficult to define, such as in specialized collections containing portraits or close-up of objects. In these cases, scene classification is not useful. Nevertheless, in most personal image collections knowing the scene context is often valuable.

1.1.5 Image segmentation

Segmentation is the task by which we divide an image into regions that are coherent with respect to a specific visual criteria. In a classical point of view, segmentation is the result of a threshold operation that separates the image into regions with the same pixel or texture properties. One current application of a classical segmentation approach is industrial image segmentation where a system extracts the shapes of parts, for verification or separation. However, in a modern approach point of view segmentation extends to the association of semantic labels to image regions. Shotton *et al.* (2006) proposed a textons based approach with boosting, which enable the learning of discriminative models for a large number of object classes. This approach incorporated appearance, shape and context information. To obtain the final image segmentation a conditional random field was used. Image segmentation relates to object detection, where the objective is to obtain an area of the image which can be classified as a certain object. More generally, image segmentation can detect regions in the image which have visually meaning, and by doing so it can help us understand the complete image content better. This can be extremely useful in the context of medical and satellite images.

1.1.6 Automatic image annotation

Automatic image annotation systems attempt to provide a textual annotation which describes the main visual concept represented in the image. These systems take advantage of previously human annotated data to learn the connection between textual annotation and the image content, trying in this way to bridge the semantic gap between concepts and visual content (Duygulu *et al.* (2002); Li and Wang (2006); Monay and Gatica-Perez (2003, 2004); Barnard *et al.* (2003)). Annotation can be seen as a generalization of object and scene recognition. Indeed, associating scene and object labels to images corresponds to performing image annotation. However, image annotation goes beyond those tasks. The terms with which image are annotated can represent any concept present in the image, not being limited to objects or scene types. Also, in annotation we want both scene and object recognition at the same time, and in any image we may have any number of object or scene types. In automatic annotation approaches, even with a

building , historical, ocean
water, man-made, castle, landmark

rock, skyline, landscape

snow, winter, panda

ocean, boat, rock

Figure 1.4. Examples of automatically annotated images. These images were obtained using the ALIPR (http://alipr.com/) real-time annotation system, developed at Penn State by Professors Jia Li and James Z. Wang (Li and Wang (2006).

constrained vocabulary to annotate each image, the range of semantic content that is captured is too large to allow for specific features for each term. As such, automatic annotation systems tend to use generic feature. One problem that arises from the annotation of images is that the resulting annotation contains the inherent semantic ambiguities of text. A common example is the annotation term bank, which can be either the institution bank or the river bank.

Automatic annotation systems are used to give the possibility of querying any large digital image library using concepts described by textual terms. The main advantage of using such systems is that users find it easy to express a query in textual terms and that these systems can handle most type of images. There are some systems that target video data, where the annotation terms tend to be related also to actions. In the case of personal users, automatic annotation systems are not commonly used.

1.2 Issues

A central issue with the design of a system capable of solving one of the tasks described in the previous section is image representation. In some tasks such as scene or object recognition, it

is possible to find an appropriate representation that takes into account the specificities of the classes which we are attempting to recognize. However, browsing and retrieval systems applied to general content digital library (personal or professional) cannot be built under the assumption that we are targeting a limited set of image content. Thus, in general, it is important to find principled image representations that can be applied to a large variety of data.

Modeling images based on quantized local interest point descriptors has proven in recent years to provide a robust and versatile way to model images. Good classification performance has been obtained on objects (Willamowski *et al.* (2004); Leibe and Schiele (2004)) and scenes (Quelhas *et al.* (2005); Fei-Fei and Perona (2005)). Similarly, good image retrieval (Sivic and Zisserman (2003); Sivic *et al.* (2005)) and image segmentation (Quelhas *et al.* (2005); Dorko and Schmid (2003)) capabilities have been demonstrated. The great advantage of modeling images based on local invariant features for the tasks of retrieval and classification is that the same methodology can be used for different image categories and that performance is normally competitive to existing task specific state-of-the-art approaches.

In addition, by quantizing the local descriptors, we can obtain image representations which have a low dimensionality and great simplicity. Although the quantization step may reduce the capacity to describe image content, the resulting representations are compact and allow to apply many of the fast techniques that were applied to the modeling and retrieval of text documents. This is an essential issue when dealing with large image and video data collections.

1.3 Contributions

In this section we introduce the main contributions of the work presented in this thesis.

In our work, we explored the use of local descriptors for image representation applied to the tasks of scene and object classification, ranking, and segmentation. More specifically, we investigated the combined use of text modeling methods and local invariant features.

Bag-of-visterms approach: we explored the use of quantized local interest point descriptors (called visterms) representation of images, applied to scene classification, segmentation, and object classification. First, we propose to use the histogram of quantized local interest point descriptors, named bag-of-visterms (BOV), to perform scene classification. We validated our approach by comparing the results obtained using BOV with a state-of-the-art approach for scene classification (Vailaya *et al.* (2001)), and by exploring several vocabulary sizes and using different datasets to build our vocabulary. Secondly, taking into account the previously suggested analogy between visterms in images and words in text documents (Sivic and Zisserman (2003)), we tested the use of Tf-idf weighting in our visterm vocabulary, with no significant alterations in performance. Finally, to address possible drawbacks from the visterm quantization we proposed a GMM based alternative to the BOV representation, which showed some improvement in the final classification performance.

Latent aspect modeling: to handle possible problems of polysemy -one visterm having several visual meaning - and synonymy - several visterms corresponding to the same visual meaning- we proposed the use of probabilistic latent aspect (PLSA) modeling to represent the content of the BOV feature vector in a more robust way. We investigated the advantages of the use of PLSA as an image representation and provide results that show that using PLSA aspect modeling allow to automatically extract co-occurrence information that strongly relates to the object or scene classes in our datasets. Importantly we showed that the resulting PLSA based representation, with an aspect distribution learned on unlabeled data, can provide better classification results when the amount of labeled training data is reduced. In addition, we analyzed the use of the PLSA aspect representation in an image ranking task.

Visterms text analogy: We investigate the analogy between visterms in images and words in text documents, presenting a study of the characteristics of both the BOV and the BOW representations in image and text collections respectively. The resulting study suggests that vocabularies of visterms in images and words in text do not share important characteristics such as sparsity and specificity. On the other hand, we encounter the occurrence of polysemy and synonymy in our visterm vocabulary.

Fusion scheme applied to local descriptors: Motivated by the importance of color to describe certain scenes (natural scenes) and objects (household objects), we study feature fusion approaches to combine color information with the local descriptors extracted from our images. The proposed fusion schemes were tested on both an object recognition task, and a natural scene classification task. In both cases, an improvement of the performance is obtained.

Contextual modeling for segmentation: we propose novel computational models to perform the segmentation of images. These models exploits the visual context captured by the co-occurrence analysis of visterms in the whole image Which departs from the more traditional contextual approaches based on spatial proximity. Nevertheless, we show that the combination of the novel PLSA contextual modeling with a traditional Markov Random Field modeling lead to better results than those obtained by applying each modeling individually.

1.4 Thesis organization

In this section we briefly describe the contents of each chapter of the thesis.

In Chapter 2 we will present works that are related to our research, and describe the state-of-the-art approaches for the tasks of scene and object classification as well as scene segmentation.

In Chapter 3 we will describe the methodologies used to obtain local interest point detectors and descriptors invariant to different image transformations as well as the particular implementation of several of the most common detectors/descriptors. We will finish the chapter by introducing (and applying) a framework for comparing several invariant local interest point

detectors and descriptors in a wide-baseline matching task.

In Chapter 4 we will introduce the main image representation studied in this thesis. We will start by describing the bag-of-visterms (BOV), a representation borrowed from the text retrieval literature and then describe several variations of this model. These include a GMM based representation which relies on a soft attribution of the local features to the visterms counts, and different fusion schemes as a mean to include additional information like color into the extracted local descriptors. To study the properties of the BOV representation, we will investigate the analogy between visterms in images and words in text documents, presenting a study of both the BOV and the BOW representations. Finally, we will investigate the use of latent aspect modeling applied to the BOV representation to build a new image representation which takes advantage of the existing visual patterns co-occurrence in the image.

In Chapter 5 we will evaluate the BOV and PLSA based representation, introduced in Chapter 4, on scene image classification tasks. First, we will consider two standard, unambiguous binary classification tasks: indoor vs. outdoor, and city vs. landscape (Vailaya *et al.* (2001)), and a related three-class problem (indoor vs. city vs. landscape). Additional 5-class, 6-class (Vogel and Schiele (2004a), and 13-class (Fergus *et al.* (2005a))) scene classification tasks are also considered. The performance of the proposed representations are evaluated on these tasks, as well as their sensitivity to different issues (e.g. vocabulary size, vocabulary and latent aspect training dataset, local descriptors).

In Chapter 6 we will first study in more detail the visual aspects extracted by PLSA modeling. This will be done through the analysis of ranking experiments and by observing qualitatively the effective spatial decomposition of images into mixtures of aspects. Taking advantage of the observed correlation between the PLSA learned aspects and the dataset classes will motivate the use of the PLSA approach as a good browsing and exploration tool for image collections. Secondly, we will also propose novel computational models to perform contextual segmentation of images, based on the visual context captured by the co-occurrence analysis of visterms in the whole image rather than on the more traditional spatial relationships. Finally, we explore the use of MRFs applied to local invariant interest points for segmentation, with or without using the proposed co-occurrence context modeling.

In Chapter 7 we will focus on object recognition tasks. In a first study we will explore the use of local interest descriptors to model household objects, from one single image, and in the presence of varying view angle and reduced resolution. More specifically, to improve the recognition performance we will propose the use of color local descriptors in a fusion framework. In a second study, we will apply the BOV and PLSA representation in an object recognition task. Like in the case of scene images classification, using PLSA on a bag-of-visterms representation (BOV) produces a compact, discriminative representation of the data, outperforming the standard BOV approach in the case of small amount of training data. Also, we will show that PLSA can capture semantics in the BOV representation allowing for both unsupervised ranking of object images and description of images as a mixture of aspects.

In Chapter 8 we will summarize the contributions and results presented in this thesis and discuss possible new challenges to explore.

Chapter 2

Background

IN this chapter, we provide an overview of the work related to the issues investigated in this thesis. One central issue explored in this thesis is the representation of the visual content in images. From the task point of view, the main problems addressed are the recognition of both scenes and objects. Thus, in the following we will start by describing the use of global image representations for both scene and object recognition and then review local representation approaches applied to the same tasks.

2.1 Global representation approaches

Global image representations are constructed based on all the pixels of an image. As such, any variation in image content will potentially result in an alteration of the image representation. This makes it difficult to obtain invariance to image transformations while using global representation approaches.

2.1.1 Scene modelling

Scene modelling approaches try to capture the overall semantic image content of the image. Traditional approaches to scene classification use global features to extract image content. Color and edge information, both in the full image or in grid subdivisions, collected in the form of histograms, among other global features, have been extensively used to deal with small number of scene classes (Swain and D.Ballard (1991); Vailaya *et al.* (2001, 1998); Gorkani and Picard (1994); Oliva and Torralba (2001); Smeulders *et al.* (2000); Naphade and Huang (2001); Paek and S.-F. (2000); H.Yu and W.Wolf (1995)). Regarding global image representations for scene classification, the work by Vailaya *et al.* (2001) is regarded as representative of the literature in the field. In this work, the approach relies on a combination of distinct low-level cues for dif-

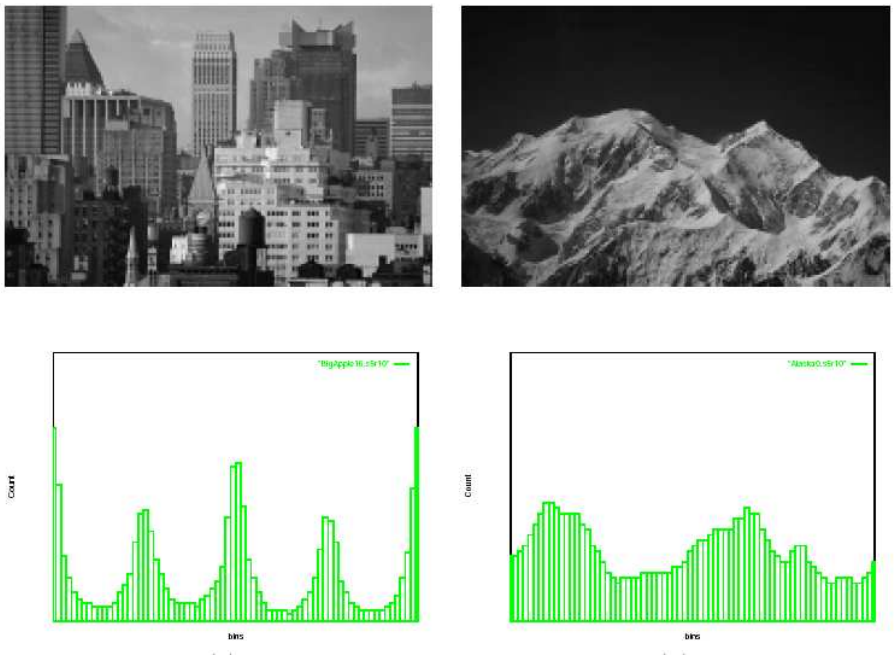


Figure 2.1. Example images and edge orientation histograms from Vailaya *et al.* (1998). Edge histograms are used to distinguish between city and landscape scenes. The histogram of city scenes exhibits peaks related to the dominance of horizontal and vertical edges.

ferent two-class problems (global edge histograms for classifying images into city vs. landscape - see Figure 2.1, and local LUV color moments for indoor vs. outdoor classification) and the performed recognition is obtained using a Bayesian framework. Also to tackle the city/landscape classification of scenes, Gorkani and Picard (1994) proposed the use of multi-resolution steerable filter to extract image dominant orientations on a 4×4 sub-blocks pyramid. Images are classified based on the dominant modality of orientation of the sub-blocks. If the orientation is dominated by horizontal and vertical direction, a city image is assumed. However, as the number of categories increases, the issue of overlapping between scene classes in images arises. To handle this issue, a continuous organization of scene classes (e.g. from man-made to natural scenes) has been proposed by Oliva and Torralba (2001). In that work, an intermediate classification step into a set of global image properties (*naturalness*, *openness*, *roughness*, *expansion*, and *ruggedness*) is proposed. Images are manually labeled with these properties, and a Discriminant Spectral Template (DST) is estimated for each property. The DSTs are based on the Discrete Fourier Transform (DFT) extracted from the whole image, or from a four-by-four grid to encode basic spatial information. A new image is represented by the degree of each of the five properties based on the corresponding estimated DST, and this representation is used for the classification into semantic scene categories (*coast*, *country*, *forest*, *mountain*,...). Alternatively, the issue of scene class overlap can be addressed by doing scene annotation (e.g. labeling a scene as depicting multiple classes). This approach is followed by Boutell *et al.* (2004), which exploits

the output of one-against-all classifiers to derive multiple class labels.

Global features/approaches are clearly an adequate approach for many scene types, since the visual information that characterizes those scenes is spread throughout all the image. However, global feature have more difficulties handling large number of classes.

2.1.2 Object modeling

Object modeling is concerned with finding ways to represent object using information extracted from image data. Object models make use of visual cues (descriptive features) extracted from image data. Global approaches to object modeling can be generally separated into shape based representations and appearance based representations.

Shape can be a powerful source of information to characterize an object. Early shape object recognition approaches used 3D models or a decomposition into parametric surfaces or volumetric primitives (Marr (1982); Biederman (1987)). To obtain invariance these methods used a object centered coordinate system. The outlines and edges of an object can also be used to characterize the shape of that object, which can be described by basic geometric elements such as lines and curves. For example, a simple edge based human head model can be constructed by modeling the head using its surrounding contour (Blake and Isard (1998)). In another approach, Kass *et al.* (1987) introduced the concept of *snakes*, an active shape model, in which deformable splines are used to describe the edges occurring in the image. The shape of the object is defined by a set of control points on the spline. Another approach, Active Shape Models (ASM) allow for the inclusion of prior knowledge on the objects shape into the design of the representation. Active shape models have been proposed that use energy minimizing techniques to estimate the models' shape parameters that best fit the image edges which and the prior knowledge (Cootes *et al.* (1992); Blake and Isard (1998)).

Appearance based approaches model the object based on grey-scale image values of images of that object. The object models can be either based on templates or learned using statistical analysis. In the first case, templates representing the object's appearance over a range of different configurations are chosen by hand (Niyogi and Freeman (1996)). Recognition is performed by searching for the template which maximizes correlation with the image. In these approaches a viewer centered coordinate system is used.

Statistical models of appearance are usually generative, in the sense that they attempt to build representations that can synthesize any instance of the object class under study. Principal Component Analysis (PCA) is a typical example which is illustrated in Figure 2.2.

More recently Active Appearance Models (AAM) were introduced to model both shape and appearance in one unified approach (Cootes *et al.* (1998)). An AAM contains a statistical model of the shape and grey-level appearance of the object of interest, which can generalize to almost any valid example. These models are very precise but also complex to design and implement. These

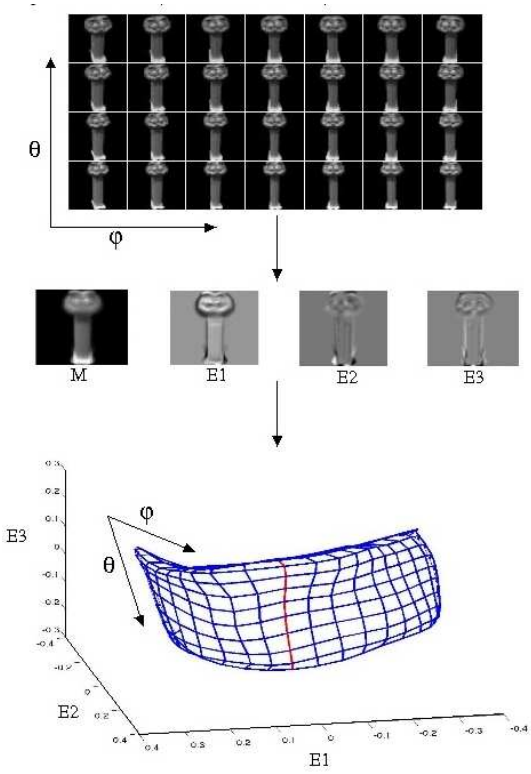


Figure 2.2. Example of object appearance modeling using PCA (object is a Pez dispenser), from Murase and Nayar (1995). (top) images of the object with different vertical (θ) and horizontal (φ) viewing angle. (middle) average image (M) and 3 first eigen images (E1,E2,E3). These images will form the basis for the representation of the object. (bottom) object representation canonical manifold in the 3D space defined by the 3 first eigen images. (images obtained from: http://www.prip.tuwien.ac.at/Research/RobotVision/or.html)

models are specially adapted, and mostly advantageous, for the tasks of tracking, animation and synthesis of objects (mostly faces) (Dornaika and Davoine (2004); Cootes *et al.* (1998)). In the case of a more general object recognition approach, these models are not normally used due to the need for detection of the object location in the image.

Global models are usually too complex to perform object detection in most images, they are mostly used in constrained environments or in tracking and animation tasks. The main disadvantage of these models is the need to perform some detection method before classification, and the fact that these models are sensitive to partial occlusion. Also, since the full shape or/and appearance is modeled global representation models can only handle a limited amount of image variability within the object class. Overall it is generally agreed that the features on which to base object detection/recognition should be local in nature, to cope with noise and occlusion.

2.2 Local representation approaches

Local representation approaches model the image content by subdividing the image into regions or parts on which individual features are computed. The resulting representation is then built as a collection of these local descriptors. As such, an alteration of an image part affects only some of the representation components, which render this approach particularly robust to partial occlusion, and allows for an accurate image content description.

2.2.1 Scene modeling

To handle scene classification in a more efficient way, and allow for a more precise scene description, several authors have proposed approaches which use local feature based representations. In these approaches, the image is split into parts which are then classified into an intermediate supervised region classification step, which is used to obtain the final scene classification (Naphade and Huang (2001); Serrano *et al.* (2002); Fauqueur and Boujemaa (2003); Vogel and Schiele (2004a)). Based on a Bayesian network formulation, Naphade and Huang (2001) defined a number of intermediate regional concepts (e.g. sky, water, rocks) in addition to the scene classes. The relations between the regional and the global concepts are specified in the network structure. Serrano *et al.* (2002) propose a two-stage classification of indoor/outdoor scenes, where features of individual image blocks from a spatial grid layout are first classified into indoor or outdoor. These local classification outputs are further combined to create the global scene representation used for the final image classification. Similarly, Vogel and Schiele (2004a) used a spatial grid layout in a two-stage framework to perform scene retrieval and classification, based on texture and color features extracted on each grid-block. The first stage does classification of image blocks into a set of regional classes, different from the scene target labels which extends the set of classes defined in Naphade and Huang (2001) (this is thus different from Serrano *et al.* (2002) and requires additional block ground-truth labeling). The second stage performs retrieval or classification based on the occurrence of such regional concepts in query images. Figure 2.3 shows an illustration of this system. Alternatively, Lim and Jin (2004) successfully used the soft output of semi-supervised regional concept detectors in an image indexing and retrieval application. In a different formulation, Kumar and Herbert (2003b) used a conditional random field model to detect and localize man-made scene structures, doing in this way scene segmentation and classification. Overall, the use of local approaches for scene classification is a new research area with promising results.

2.2.2 Object modeling

Local approaches in the field of object recognition have since long been proposed to handle the problem of detection and recognition in an unified and efficient way (Viola and Jones (2001); Schneiderman and Kanade (1998); Agarwal and Roth (2002)). The motivation for representing

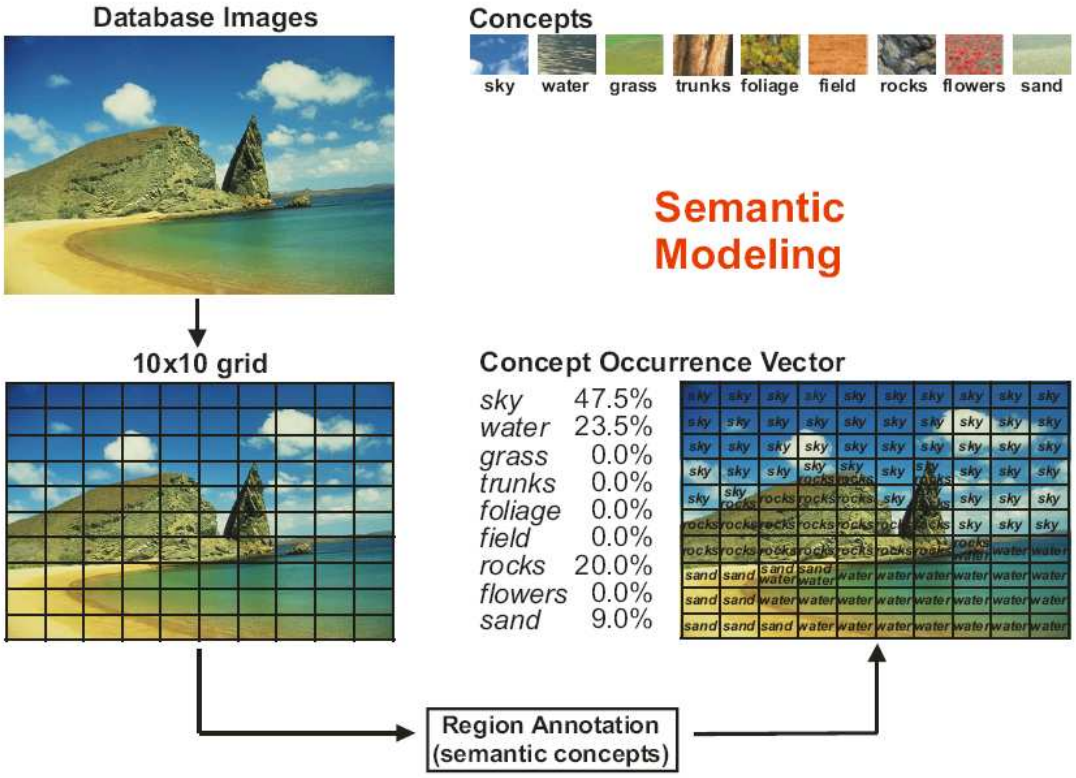


Figure 2.3. Illustration of the system used by Vogel and Schiele (2004a) to perform natural scene classification. The authors use a block subdivision of the image, for which they define intermediate semantic labels. (taken from (Vogel and Schiele (2006))

objects in images by parts comes to a certain extent from biological theory, which explains object detection on the basis of decomposition of objects into constituent parts. According to this theory, the representation used by humans for identifying an object consists of the parts that constitute the object, together with structural relations over these parts that define the global geometry of the object (Agarwal and Roth (2002)). Schneiderman and Kanade (1998) proposed a trainable object detector for detecting faces and cars, which is invariant to scale and position. To cope with variation in object viewing angle, the detector uses multiple classifiers, each spanning a different range of 3D orientation. Each of these classifiers determines whether an object part is present at a specified location within a fixed-size image window. Classification of each window is based on a wavelet decomposition of the local image window. To find the object at any location and size, these classifiers scan the image exhaustively. Figure 2.4 shows the results of applying the Schneiderman and Kanade (1998) face detector in a crowded scene. Viola and Jones (2001) defined a set of rectangle operators on a 24×24 image block, which are used as weak classifiers to perform face recognition, using Adaboost. The rectangle operator value is calculated by computing difference of the sum of pixel values in different areas. The system is based on a large number of such operators (more than 150,000) defined on each image

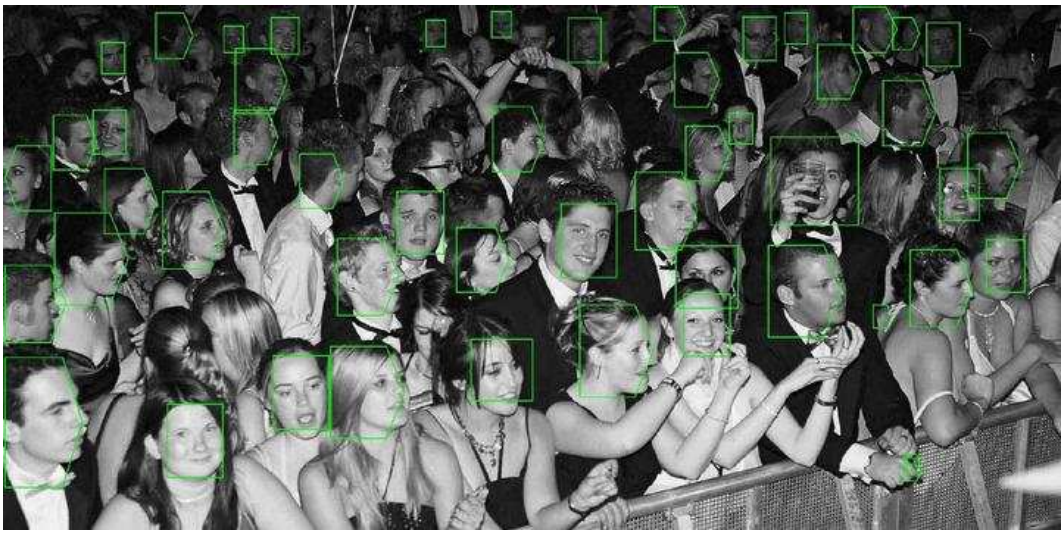


Figure 2.4. Example image with faces obtained using the face detector from Schneiderman and Kanade (1998). A rectangle appear where a frontal face is detected, and in the case of a non frontal face, a polygon with an arrow side indicating the orientation of the detected face.

block. These features can be calculated very efficiently by using integral images. Each rectangle operator is equivalent to a simple, weak classifier if it is coupled with a threshold value. See Figure 2.5 for examples of the proposed operators.

While the above mentioned works implicitly represent objects by parts, they do not model explicitly the geometrical relationships between these parts, which decreases their ability to model object classes with medium and large within class geometric variability. As a remedy, several approaches that model both parts and geometrical constraints have been proposed, and often relied on the use of interest points and local descriptors.

The combination of interest point detectors and local descriptors are increasingly popular for object detection, recognition, and classification (Lowe (2004); Fergus *et al.* (2003); Fei-Fei *et al.* (2003); Dorko and Schmid (2003); Opelt *et al.* (2004); Fei-Fei *et al.* (2004); Willamowski *et al.* (2004)). Most existing works have targeted a relatively small number of object classes, an exception is Fei-Fei *et al.* (2004) where 101 classes are use to classify. Fergus *et al.* (2003) optimized, in a joint unsupervised model, a scale-invariant localized appearance model and a spatial distribution model. In this approach objects are modeled as flexible constellations of parts. Parts are selected using a entropy-based local interest point detector(Kadir *et al.* (2004)). The modeling of the appearance, shape, occlusion and scale of the objects is handled by a probabilistic representation, for which the parameters are obtained using the Expectation Maximization algorithm. Recognition is performed by using the resulting model in a Bayesian manner. Fei-Fei *et al.* (2003) proposed a method to learn object classes from a small number of training examples. The same authors extended their work to an incremental learning procedure, and tested it on a large number of object categories (Fei-Fei *et al.* (2004)). Dorko and Schmid

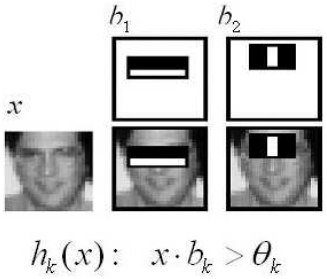


Figure 2.5. Examples of two feature applied to an image and the definition of one possible weak classifier from Viola and Jones (2001).

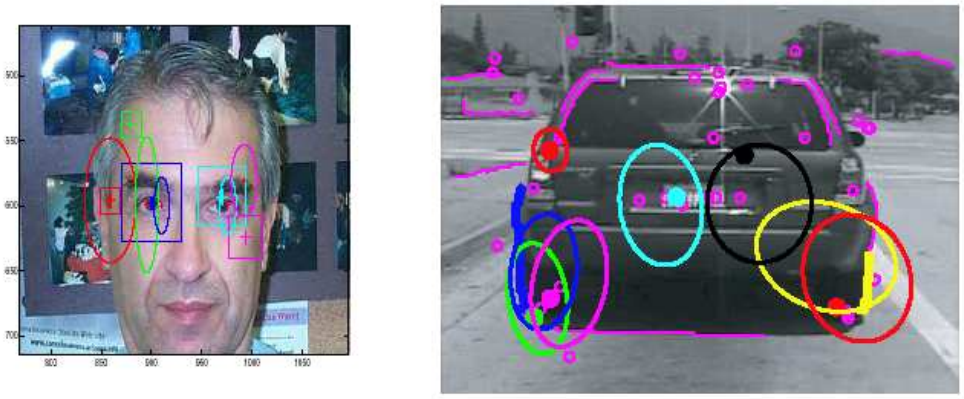


Figure 2.6. Images of a face and a car from the Caltech object database with the automatically learned parts obtained by the method from Fergus *et al.* (2003).

(2003) performed feature selection to identify local descriptors relevant to a particular object class, given weakly labeled training images. Opelt *et al.* (2004) proposed to learn classifiers from a set of visual features, including local invariant ones, via boosting. More recently, Mikolajczyk *et al.* (2006) introduced a generative modeling approach for multiple object detection. This approach uses a hierarchical representation of visual object parts, in a generative framework. This allows the modeling of the parts' relationships together with the overall scale and orientation of the detected object, being able to model multiple object in each image.

2.2.3 Descriptor quantization and text modeling approaches

The analogy between invariant local descriptors and words has also been exploited recently (Sivic and Zisserman (2003); Sivic *et al.* (2005); Willamowski *et al.* (2004)). Sivic and Zisserman (2003) proposed to cluster and quantize local invariant features into visterms, to increase speed

and produce a more stable representation, for object matching in frames of a movie. However, quantizing local descriptors decreases the discriminative power, reducing the capacity to describe the local structure and texture. Nevertheless, such approaches allow to reduce noise sensitivity in matching and to search efficiently through a given video for frames containing the *same* visual content (e.g. an object) using inverted files. Willamowski *et al.* (2004) extended the use of visterms by creating a system for object matching and classification based on a bag-of-words (BOV) representation built from local invariant features and various classifiers. Despite the fact that no geometrical information is kept in the representation good results were achieved.

Recently, in parallel to our work, the joint use of local invariant descriptors and probabilistic latent aspect models has been investigated by Sivic *et al.* (2005) for object clustering in image collections, and by Fei-Fei and Perona (2005) for scene classification. Sivic *et al.* (2005) investigated the use of both Latent Dirichlet Allocation (LDA) (Blei *et al.* (2003)) and Probabilistic Latent Semantic Analysis (PLSA) (Hofmann (2001)) for clustering objects in image collections. Using the BOV representation, they showed that latent aspects closely correlates with object categories from the Caltech object database, though these aspects are learned in an unsupervised manner. The number of aspects was chosen by hand to be equal (or very close) to the number of object categories, so that images are seen as mixtures of one 'background' aspect with one 'object' aspect. This allows for a direct match between object categories and aspects, but at the same time implies a strong coherence of the appearance of objects from the same category: each category is defined by only one multinomial distribution over the quantized local descriptors. Fei-Fei and Perona (2005) proposed two variations of LDA (Blei *et al.* (2003)) to model scene categories. They tested different region detection processes - ranging from random sampling to fixed-grid segmentation - to build an image representation based on quantized local descriptors. Contrarily to Sivic *et al.* (2005), Fei-Fei and Perona (2005) propose to model a scene category as a mixtures of aspects, and each aspect is defined by a multinomial distribution over the quantized local descriptors. This is achieved by the introduction of an observed class node in their models, which explicitly requires each image example to be labeled during the learning process.

Finally, in another research direction, a number of works have also relied on the definition of visterms and/or on variations of latent semantic modeling to model annotated images, i.e. to link images with (semantic) words (Mori *et al.* (1999); Barnard *et al.* (2003); Blei and Jordan (2003); Jeon *et al.* (2003); Monay and Gatica-Perez (2003); Zhang and Zhang (2004)). However, these methods have mostly relied on traditional regional image features without much invariance to image transformations. As an example, Zhang and Zhang (2004) explored the use of a latent space model to discover semantic concepts for content-based image retrieval. The model is learned from a set of quantized regions per image, and the similarity between images is computed from the estimated posterior probability over aspects.

2.3 Chapter conclusion

In this chapter we provided an overview of the work related to image representation, reviewing literature related to both global and local image representation. Overall, it is noticeable that the simplicity of global approaches is still a major motivation for the use of global approaches in the case of scene images, since scenes are defined by the overall content of the image. However, as more classes are introduced, local approaches can outperform global approaches. In the case of object image representation, it as been established across the literature that due to the need for invariance to image transformation, modeling the image globally is not efficient. Also, in the case of object classification in a cluttered environment or scene class overlap, detection becomes an issue and global approaches are not suitable for those tasks.

An increasing number of published work, on the subject of image representation, uses in some way local interest points to obtain the resulting image representation. These approaches have the advantage of being more general, normally easily adaptable to other problems, and capable of extracting a more complete content description of the image. The results obtained until now using local interest point detectors and descriptors are promising, in most cases these new approaches outperform previous baselines which where specifically designed for each task. Even if not necessarily the best in every case, approaches based on local interest point detectors are very promising for most applications.

Chapter 3

Local interest point detectors and descriptors

LOCAL interest point detectors and descriptors are designed to extract specific points from images and produce features that allow for a robust matching between similar points across images. Point matching is an essential task in the wide-baseline matching process(Hartley and Zisserman (2000)). Wide-baseline matching is the task of finding corresponding points between images of the same scene or object, in the case where the images are taken from widely separated viewing angles.

Local interest point detectors are designed to localize points that contain distinctive information in their local surrounding area and whose extraction is stable with respect to geometric transformations and noise. These points are *characteristic* points in the image, where the signal changes bi-dimensionally. In addition, local interest point detectors need to automatically specify an area around the characteristic point that will have a certain amount of invariance to image transformations. We will refer to this area as *local interest area*. Invariance to transformations means that given two images of a certain object taken from different viewing angle, the detector will be able to extract local points and areas in both images that correspond to the same point on the surface of the object (see Figure 3.1).

Local descriptors are compact and distinct features, extracted from local interest areas. These descriptors are designed to be as specific as possible, while providing some invariance to imaging conditions and to compensate for possible errors in the local interest area definition. Local interest points must be as specific as possible because each local interest point is compared with a large amount of other local interest points to assess the similarity between each possible pair of points. This is especially important in the case of wide-baseline matching, where the point-to-point correspondence between images is the final objective.

Local point detectors and descriptors were originally proposed to enable efficient point-to-point

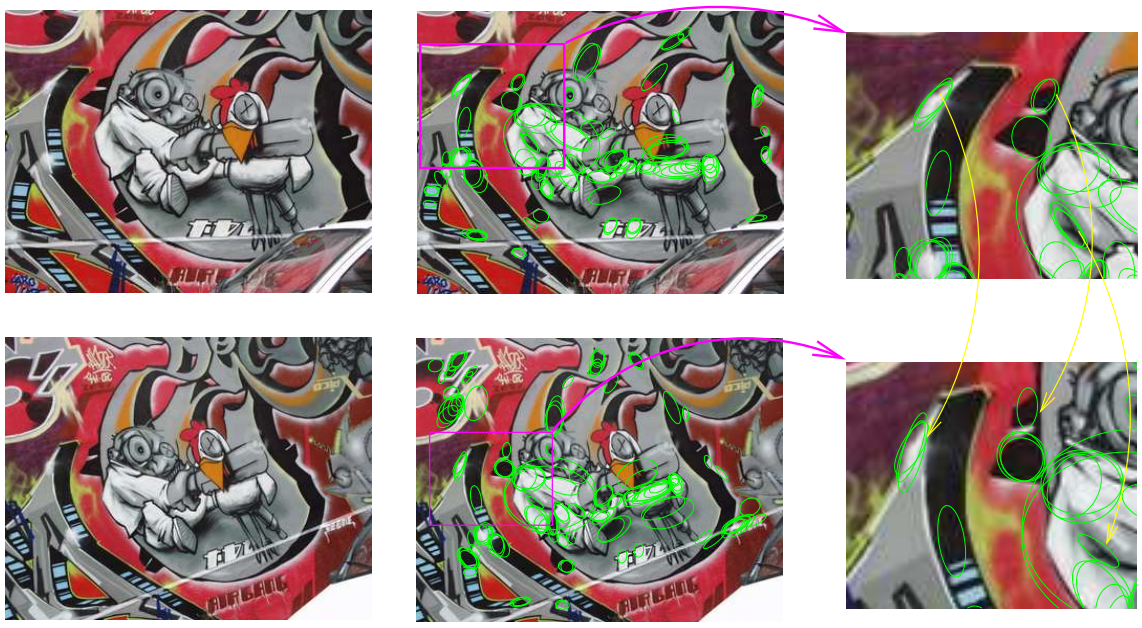


Figure 3.1. An example of two images from one scene, from two viewing angles. For each image we extracted the local invariant interest points. In each image we display the local interest areas and the correspondences between some of the detected areas.

matching in wide-baseline matching problems (Lowe (2004); Mikolajczy and Schmid (2004); Tuytelaars and Gool (2000); Kadir *et al.* (2004); Baumberg (2000); Schaffalitzky and Zisserman (2002)). In more recent work these techniques have been exploited in other areas like object recognition (Willamowski *et al.* (2004); Monay *et al.* (2005)), scene recognition (Quelhas *et al.* (2005); Bosch *et al.* (2006); Fei-Fei and Perona (2005)), image annotation (Fergus *et al.* (2005a)), image segmentation (Monay *et al.* (2006)) and video browsing (Sivic and Zisserman (2003)). These new applications explore the use of local descriptors in quantized form, where quantized local descriptors are usually entitled *visterms*. This quantization allows, by counting the number of visterms' instances in an image, to produce a global image representation, the bag-of-visterms (BOV). Due to the different nature of the application of local descriptors for image categorization, instead of point-to-point matching, it has been argued that some properties that are important when performing point-to-point matching may not be necessary when applying visterms to image categorization, and may even by detrimental to the performance (Quelhas *et al.* (2005); Fei-Fei and Perona (2005)). In this chapter we will explore and analyze local interest point detectors and descriptors from the wide-baseline application point of view only. We will explore the usage of these techniques for scene and object categorization using the BOV approach in Chapter 5 and Chapter 7 respectively.

In the next sections of this chapter we explore the properties of local point detectors and descriptors, and review several particular implementations of detectors and descriptors. Based on a performance comparison framework, we will then present a comparative study on the task of

wide-baseline matching. We will present a comparison of quantized local descriptors for the task of image classification in Chapter 5.

3.1 Local interest Point Detectors

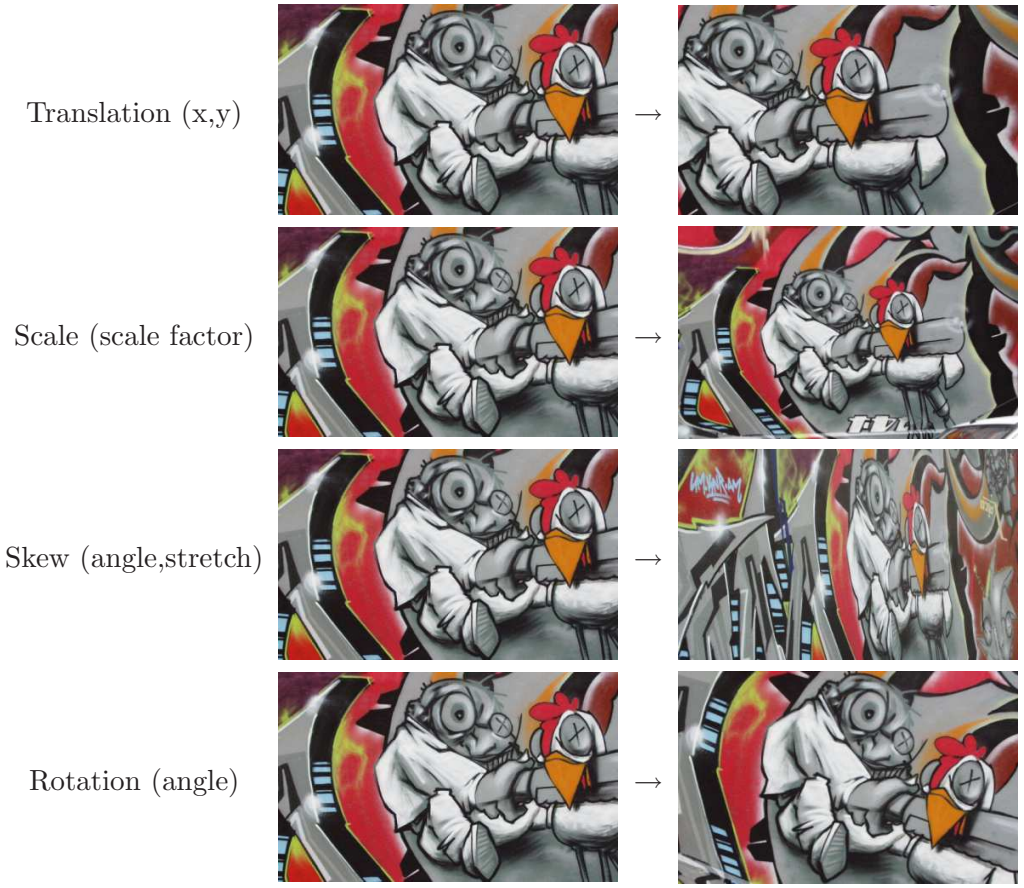


Figure 3.2. Images illustrating the possible camera view changes of a scene.

The goal of the local interest point detector is to automatically extract characteristic points, and more generally *regions*, which are invariant to geometric and photometric transformations of the image. This invariance property is interesting, as it ensures that given an image and its transformed version, the same image points will be extracted from both images and the same local areas will be detected. After the local interest areas are extracted from an image we can compute local descriptors on those areas to obtain an image representation. Several interest point detectors exist in the literature. They vary mostly by the amount of invariance they theoretically ensure, the image property they exploit to achieve invariance, and the type of image structures they are designed to detect (Mikolajczy and Schmid (2004); Tuytelaars and

Gool (1999); Lowe (2004); Mikolajczyk *et al.* (2005); Kadir *et al.* (2004); Baumberg (2000); Schaffalitzky and Zisserman (2002)).

A local interest point detector not only extracts locations in the image, but also defines a *local interest area* ($\mathcal{A}$). The characteristic points detected in the image are, by design, distinct from the surrounding neighborhood and are selected by the detector due to their properties that insure that they remain distinguishable and identifiable after a range of transformations. The same applies for the local interest areas defined by the detectors. Detectors for different characteristic points will have different degrees of invariance to image transformations.

Interest point detectors provide as their output a list of the coordinates of all local interest points detected in an image,$\{\mathbf{x}_i, i = 1 \dots N_{\mathbf{x}}\}$, where $N_{\mathbf{x}}$ is the total number of detected points in the image. For each of the detected points $\mathbf{x}_i$ the detector also defines the characteristics of the local interest area $\mathcal{A}_i$, the characteristic orientation θ_i, the characteristic scale σ_i, and in the case of affine invariant local interest point detector, the affine scale defined by Σ_i. We can define the local interest point detector as a function of an image $d(I)$ which returns a list of local interest point and their associated local interest areas,

$$d(I) \longmapsto \mathcal{L}_{\mathbf{x}} = \{(\mathbf{x}_i, \mathcal{A}_i), i = 1 \dots N_{\mathbf{x}}\}, \text{ where } \mathcal{A}_i = (\theta_i, \Sigma_i). \tag{3.1}$$

For a stable image representation, it is important to make sure the local interest areas are as invariant as possible to image transformations. The more invariant the local interest area, the more confident we are that the local extracted information will remain the same from image to image. As we will see, it is with the purpose of achieving this stable area property that most invariance approaches are developed. In the rest of this thesis, whenever we refer to the invariance of the local interest point detectors, we are in fact referring to both the stability of the extracted points' location and the stability of the local interest areas' properties.

In the next sections, we will explore the different approaches used in the design of local interest point detectors to gain invariance to image transformations. To simplify the discussion of the approaches presented in this chapter, we will assume that it is possible to achieve invariance to the considered image transforms independently, and in the following order: spatial, scale, affine, orientation, and illumination. This is not always true and we will explain when that case arises. In the next section, we will explore different properties that are essential to local interest point detectors, and in Section 3.1.2 we will give some examples of local interest point detectors.

3.1.1 Goals and properties

As previously stated, local interest point detectors have the task of extracting specific points and areas from images in a invariant way. There are many local interest point detectors. They are normally defined by the type of local structure they identify and their degree of invariance

to image transformations. The degree of invariance of a particular local interest point detector determines the range of applications in which it can be used. In the remainder of this section, we define the main types of invariance that can be currently achieved by local interest point detectors. We also explore different methodologies to achieve each invariance type.

Spatial Invariance

Spatial invariance refers to a local interest point detector's ability to detect the same local interest point before and after a translation of the image. Spatial invariance is the basis of all other types of invariance. Local interest detectors are operators which respond to characteristic locations in the image. The extraction of those characteristic locations, is the result of extrema detection process of the detector's response over the image. The choice of which characteristic points to search in the image is done so that we can perform such extrema search based on the smallest local information possible, to ensure the locality of our detector. Since local interest points are defined by the information contained in a small area of the image, we can assume that a translation of the image will not affect the detection and therefore the local interest points' extraction. This is a direct result of the fact that an image translation will not affect the local information of the image. However, if the image is cropped, due to the translation, local interest points defined over the cropped region will be lost.

Scale invariance

Scale invariance deals with the recovery of the same local interest point and associated area after a change in camera zoom or image resizing. Scale invariance is achieved by determining the scale at which the local structure is best detected in the image, i.e. the scale for which our detector has the highest response. In the early eighties, Witkin (1983) proposed to consider scale as a continuous parameter for image representation, opening the way for the most commonly used approach to deal with scale changes in image representations, *scale-space theory*. With scale-space theory, we can analyze the response of local interest point detector across scales. In the domain of digital images, the scale space parameter σ is discretized. Thus, the scale-space representation of an image is defined as a set of images at different discrete scale levels $(\sigma_j, j = \{1, \ldots, s\})$.

An image's scale-space representation can be constructed by applying a smoothing kernel to the original image either followed by a re-sampling of the image, creating a pyramid like representation (cf. Figure 3.3(a)), or without changing its size (cf. Figure 3.3(b)). Creating a scale-space representation by smoothing without re-sampling is less efficient, as we retain redundant information (maintain the full size of the image). However, by keeping the image size constant we simplify the correspondence between point at different scales (levels of the scale-space representation), since the point position in the image doesn't change across scales. Lindeberg (1994) has shown that under some rather general assumptions on scale invariance, the Gaussian ker-

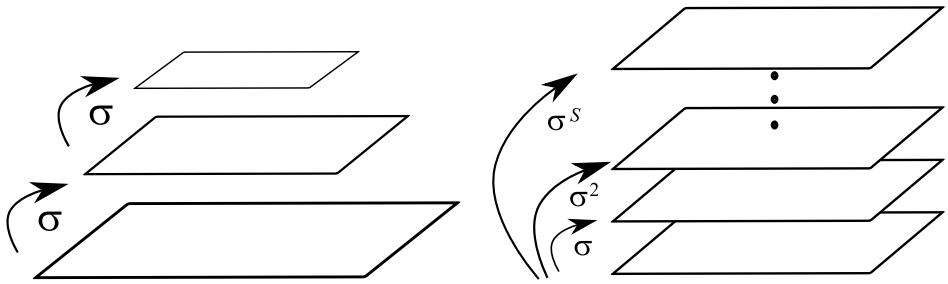


Figure 3.3. Scale space representations. (a) Pyramid representation constructed using sub-sampling. (b) Scale-space representation constructed by successive Gaussian Smoothing of the high resolution image.

nel and its derivatives are the only possible smoothing kernels for scale space analysis. The bi-dimensional isotropic Gaussian kernel parameterized by the scale factor σ is defined by:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp - \frac{x^2 + y^2}{2\sigma^2}. \tag{3.2}$$

As explored by Sporring *et al.* (1997), the Gaussian kernel has many properties which makes it interesting for image computations and feature extraction. Separability is one of the most attractive properties of the isotropic Gaussian kernel as it allows us to obtain a multi-dimensional Gaussian kernel as the product of one-dimensional kernels:

$$G(x, y, \sigma) = G(x, \sigma)G(y, \sigma). \tag{3.3}$$

This can greatly accelerate the computation of a Gaussian kernel convolution with an image, in the case of considerably large Gaussian kernels (i.e. $\sigma >> \sqrt{2}$).

Another interesting property for the Gaussian function is the commutative semi-group property, which states that successive n Gaussian smoothings of an image are equivalent to one smoothing with a kernel of σ equal to the square root of the sum of the square of all individual kernels' σ,

$$G(x, y, \sigma_1) * \cdots * G(x, y, \sigma_n) * I(x, y) = G(x, y, \sqrt{\sigma_1^2 + \cdots + \sigma_n^2}) * I(x, y), \tag{3.4}$$

where * denotes the convolution operator.

The different levels of an image's Gaussian scale-space representation are created by convolution with the Gaussian kernel:

$$L(\mathbf{x}, \sigma) = G(\mathbf{x}, \sigma) * I(\mathbf{x}). \tag{3.5}$$

where I is the image, $\mathbf{x} = (x, y)$ is the position of a point in the image, and $G(\mathbf{x}, \sigma)$ is the isotropic Gaussian kernel parameterized by the scale factor σ, see equation 3.2.

For a given local interest point detector, we can define a minimum and a maximum scale of interest. The minimum scale of interest is defined by the noise in the image and by the minimal size of the point neighborhood which represents the characteristic local structure. The maximum scale of interest is defined by the locality constrains of our detector and the image size. These limits are usually set to realistic limits, obtained from empirical testing on typical images. Given a minimum and maximum scale limits to performs our search, we can create a scale-space representation of the image, divided into n discretized steps.

Using scale-space theory we now know how to represent our image in a scale invariant way. However, in general, the spatial derivatives from the scale-space representation decrease with scale. In other words, if an image is smoothed, then the magnitude of the spatial derivatives computed from the smoothed data can be expected to decrease. To obtain true scale invariance we need to introduce a normalizing scale factor into the spatial derivatives which define our local interest point detector. Lindeberg (1998) defined the normalizing scale factor to be σ_D^2 in which σ_D is the scale of the Gaussian smoothing of the image. Using this normalization we can now design detectors in a scale invariant way.

As an example, we can explore the Laplacian-of-Gaussian scale invariant local interest point detector introduced by Lindeberg (1998). The Laplacian-of-Gaussian can be used to detect blob like regions in the image. The Laplacian-of-Gaussian detector is defined as:

$$I_{Lap}(\mathbf{x}, \sigma_D) = \sigma_D^2 \mid L_{xx}(\mathbf{x}, \sigma_D) + L_{yy}(\mathbf{x}, \sigma_D) \mid \tag{3.6}$$

where $L_{xx}(\mathbf{x}, \sigma_D)$ and $L_{yy}(\mathbf{x}, \sigma_D)$ are the second order derivatives of the image at point $\mathbf{x}$ and scale σ_D. These derivatives are defined as:

$$L_{xx}(\mathbf{x}, \sigma_D) = \frac{\partial^2}{\partial x \partial x} * G(\sigma_D) * I(\mathbf{x}) \text{ , and } L_{yy}(\mathbf{x}, \sigma_D) = \frac{\partial^2}{\partial y \partial y} * G(\sigma_D) * I(\mathbf{x}) \tag{3.7}$$

Using this detector Lindeberg (1998) performed experiments regarding scale invariance, which we now analyze. Given a local interest point in two images at different scales, we can compare the response of the detector for that local interest point over different scales, for both images (Figure 3.4 bottom). We can see that the maximum of the detector occurs so that the local interest area overlaps with the same image content in both images (Figure 3.4 top). The ratio of the scales for which we obtain the maximum response of the detector in the two images is

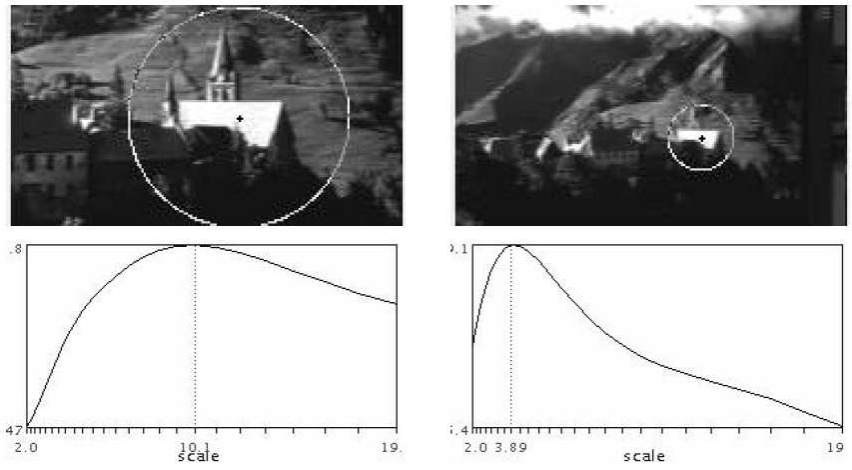


Figure 3.4. This image shows two images that contain the same object at different scales. The automatic scale detection retrieves the same area around the interest point. In the graph shown below the images we can see clearly that the scale response of the detector increases near the characteristic scale.(images obtained from Lindeberg (1998))

approximately the scale change that occurred between the two images. This enables us to define a local interest area around a local interest point that is invariant to scale. We will refer to the scale for which the detector obtain a maximum over scales as the *characteristic scale*.

Affine Invariance

Affine invariance can be seen as a generalization of scale invariance to handle cases where the scale change is not isotropic. In this case scale can change by different factors in different directions. This occurs in the case of a camera viewing angle change. The non-uniform scaling has an influence on the location, scale of the local interest point, and shape of the local interest area.

Affine transformation of the local image patch can be handled using the more general affine Gaussian kernel, to obtain a affine-space representation of the image. The affine Gaussian kernel is defined by,

$$G(\mathbf{x}, \Sigma) = \frac{1}{2\pi\sqrt{\det \Sigma}} \exp -\frac{x^T \Sigma^{-1} y}{2}, \tag{3.8}$$

where Σ defines the affine transformation of the image. Using this new kernel we can define an affine-space approach. However, this approach will now have four parameters to explore, instead of one in the non-affine case. This approach to affine image representation was used‘ by Lindeberg and Garding (1997). However, the increase in the dimensionality of the image

representation makes the exhaustive search unfeasible for the detection of local interest points in the whole image.

Most affine local interest point detectors use local information inside the local interest area (detected using scale invariance) to define the local affine transformation and obtain affine invariance (Mikolajczy and Schmid (2004); Mikolajczyk and Schmid (2003); Baumberg (2000); Tuytelaars and Gool (2000)). We will review an iterative approach to local affine invariance in Section 3.1.2, when we introduce the affine invariant Harris corner detector.

Rotation invariance

The locations detected by all the local interest point detectors used in this thesis are invariant to any arbitrary rotation of the image. This means that these detectors will recover the corresponding point in the image after an image rotation. However, this means that we don't know the specific orientation of the local interest area. To obtain a rotation invariant representation we either need an orientation invariant local descriptor or we need to compute a consistent orientation to the local interest area, which remains invariant with respect to a rotation of the local interest area's image content. By assigning a consistent orientation to each local interest point, based on local image properties, each point's descriptor can be represented relative to this orientation and therefore achieve invariance to image rotation.

The most straightforward approach to define an orientation at the local interest point $\mathbf{x}$ is to calculate the $arctang(L_x(\mathbf{x}, \sigma_D), L_y(\mathbf{x}, \sigma_D))$ at that point. This is however a very unstable approach since any small variation of the location of the point can affect the computed orientation considerably (Mikolajczyk and Schmid (2002)). A more stable approach was proposed by Lowe (1999), where the specific orientation is given through the analysis of the peak of a local histogram of the gradient orientations within a small region around the local interest point. The histogram used to determine the specific orientation has normally 36 orientation bins uniformly distributed in the full range of 360 degrees. The small area around the local interest point was defined by Lowe (1999) to be a third of the detected characteristic scale of the local interest area.

The problem with this approach is that it relies on the orientation information inside the local interest area. This may be reliable for local interest point around which the orientation content is high. However in the case of blob like regions, the center of which is usually uniform, this is not the case. In some cases the orientation detected inside local interest area is ambiguous, which means there may be more than one dominant gradient orientation inside that area. In those cases, it was proposed that if two orientations are dominant, two descriptors should be calculated for that location, one at each of the ambiguous orientations (Lowe (2004)). This improves the invariance to errors in the rotation estimation due to small errors in the local interest area scale and location, with the trade-off of creating more points. The increase in the number of extracted points leads to an increase in the calculations in both feature extraction and any operation related to the features in the image.

There are some alternative detectors that have a built-in dominant orientation detection. In these detectors edge information is used directly, based on a corner as initial point and finding the two edges that cross at that point. A model of the local interest point in fitted to the edge data and the characteristic orientation is found (Tuytelaars and Gool (2000)). These detectors are however a special case, all detectors presented here use the local histogram approach for dominant orientation estimation.

Illumination Invariance

Regarding illumination invariance there are two main concerns: the invariance of the interest point location, and the invariance of the local interest area content.

In terms of the local interest point's location, uniform illumination variations tend not to affect the location of interest points (or scale). This results, mainly, due to the maxima search in which the detection of local interest points is based. Even if the response of the detector, in some region of the image, changes, it should not alter the location of the maxima in that region. Another factor that contributes for such invariance is that most detector are, in some way, based on image gradient information, which is invariant with respect to uniform illumination changes.

In terms of the image content covered by the local interest area, changes will occur whenever illumination changes. Which, depending of the built-in invariance of the descriptor, may cause the local descriptor feature to change, reducing the chances of a good match to similar points. Thus, most descriptors perform some illumination normalization to reduce the influence of illumination changes. For grey-scale images, we can for instance apply a histogram equalization. For local descriptors based on local color pixel values, performing a general normalization is not trivial and these detector have normally a more complex way to achieve illumination invariance by combining information from the different channels (Tuytelaars and Gool (2000)).

3.1.2 Some local interest point detectors

As said in beginning of this section there are several local interest point detectors in the current literature. In the previous subsection we explored the general approaches to achieve certain types of invariance. In this section we will present some local interest point detectors, trying to cover the most significant examples in this field and the different approaches to invariance.

Harris Corner detector

One of the most popular approach for the detection of local interest points is the Harris corner detector (Harris and Stephens (1998)). Corners are advantageous local interest points, they are well defined in two directions and contain a good amount of edge information. Unfortunately,

corner can sometimes be created by occlusions, this can create 'virtual' corners which do not properly represent the content of the image. The original formulation of the Harris corner is based on the second moment matrix,

$$\mathcal{M}(\mathbf{x}, \sigma_I) = G(\mathbf{x}, \sigma_I) * \begin{bmatrix} \left(\frac{\partial I(\mathbf{x})}{\partial x}\right)^2 & \frac{\partial I(\mathbf{x})}{\partial x}\frac{\partial I(\mathbf{x})}{\partial y} \\ \frac{\partial I(\mathbf{x})}{\partial x}\frac{\partial I(\mathbf{x})}{\partial y} & \left(\frac{\partial I(\mathbf{x})}{\partial y}\right)^2 \end{bmatrix} \tag{3.9}$$

where $\frac{\partial I(\mathbf{x})}{\partial x}$ is the x derivative of image I at location $\mathbf{x}$, $\frac{\partial I(\mathbf{x})}{\partial y}$ is the y derivative of image I at location $\mathbf{x}$, and $G(\mathbf{x}, \sigma_I)$ a Gaussian of scale σ_I. The scale factor σ_I implicitly defines a neighborhood of the point.

The second moment matrix (Equation 3.9) characterizes the signal autocorrelation on the neighborhood of a point. By analyzing the second moment matrix we can obtain information about that neighborhood. Harris and Stephens (1998) analyzed the eigen decomposition of the second moment matrix and remarked that if the resulting decomposition contains two large eigenvalues, then there are two orthogonal directions in that point's neighborhood with a large gradient. This can be used as a measure of cornerness, and similarly saliency. However, computing the eigen decomposition of each location in the image is a computationally expensive task. As an alternative Harris and Stephens (1998) proposed a direct measure to detect points where the eigenvalues are large. This can be done by combining the trace and the determinant of the second moment matrix:

$$\Phi = \det(\mathcal{M}(\mathbf{x}, \sigma_I)) - \alpha trace^2(\mathcal{M}(\mathbf{x}, \sigma_I)), \tag{3.10}$$

where Φ is the cornerness measure and α is 0.04.

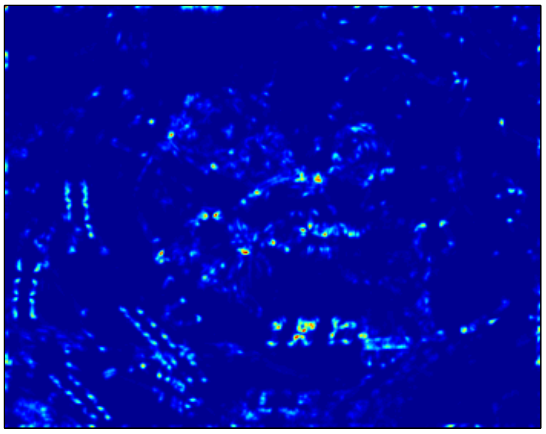
The cornerness factor Φ will have a high value at image locations where the image intensity values undergo a large change in two (approximately) orthogonal directions. To detect the locations where the corners are located we use a local maxima search. Figure 3.5 shows the original image, the cornerness map, and the detected corner point of an image.

In its originally formulation, as presented until now, the Harris corner detector was neither scale nor affine invariant. As an extension, to achieve scale invariance, a derivative scale σ_D is included in the second moment matrix computation according to:

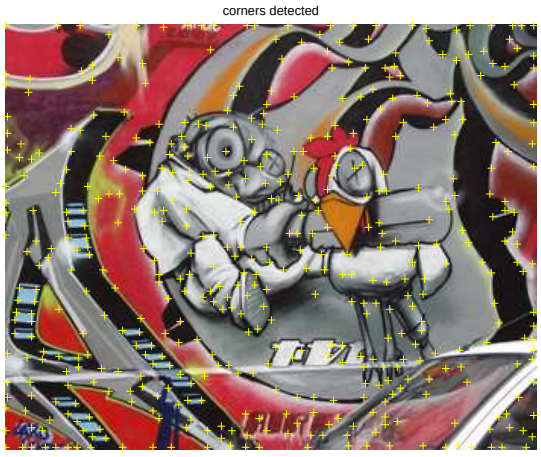
$$\mathcal{M}(\mathbf{x}, \sigma_I, \sigma_D) = \sigma_D^2 G(\mathbf{x}, \sigma_I) * \begin{bmatrix} L_x^2(\mathbf{x}, \sigma_D) & L_x L_y(\mathbf{x}, \sigma_D) \\ L_x L_y(\mathbf{x}, \sigma_D) & L_y^2(\mathbf{x}, \sigma_D) \end{bmatrix} \tag{3.11}$$

where $L_x = G(\mathbf{x}, \sigma_D) * \frac{\partial I}{\partial x}(\mathbf{x})$ and $L_y = G(\mathbf{x}, \sigma_D) * \frac{\partial I}{\partial y}(\mathbf{x})$, and σ_D is a derivation scale. The derivative scaling factor σ_D^2 is introduced to make the detector truly invariant to scale (see

(a)



(b)



(c)

Figure 3.5. Harris corner detector extraction process. (a) original image, (b) cornerness map, showing the intensity of the detector's response, and (c) the resulting local interest points (corners), which are local maxima of the cornerness map.

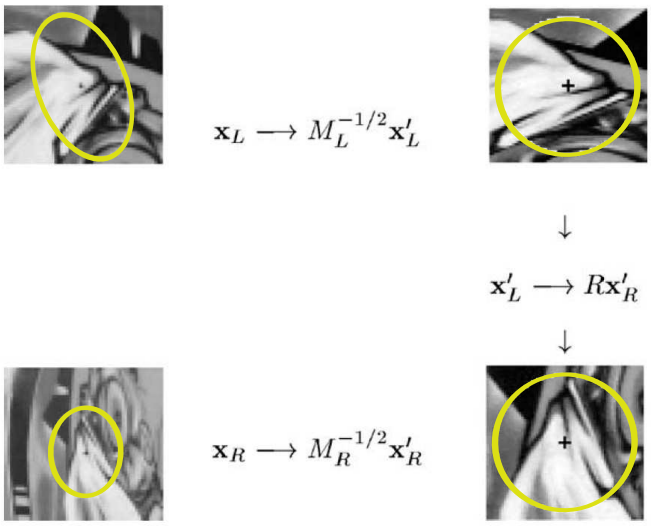


Figure 3.6. The relation between two different local affine transformation of a local interest area, and their representation after a isotropic projection. We can see that using the information contained in the second moment matrix we can obtain an affine invariant normalization of the area (with an unknown rotation).

Section 3.1.1). Using this new formulation we can apply a threshold to the cornerness measure over different scales, for each location in the image, and extract corners from an image in a scale invariant way.

Mikolajczyk and Schmid (2002) applied the Laplacian-of-Gaussian to the local interest point locations, extracted from different scales, to better refine the detected scale, we will call this approach the Harris-Laplacian corner detector.

To achieve affine invariance Mikolajczy and Schmid (2004) used an iterative approach in which the gradient information captured by the second moment matrix, computed at the local detected scale, is used to estimate the characteristic affine local area projection. The approach is summarized in Figure 3.7. Contrary to the affine invariance approach presented in Section 3.1.1, this approach does not model the viewing angle change that may have occurred. In a similar way to the characteristic scale detected in the scale-invariant approach, the characteristic affine transformation detected by this method is defined by the local image content. The resulting local affine area description is used to obtain invariance to affine transformations. This detector is normally entitled Harris-Affine (Mikolajczyk and Schmid (2002)).

Difference of Gaussians detector(DOG)

Lowe (1999) proposed to select key locations at maxima and minima of the difference of Gaussian (DOG) operator in scale space. This detector is scale, illumination, and orientation invariant.

Iterative affine invariance approach, based on the second moment matrix

1. detect the initial local interest point, using a scale-space invariant approach. Define a local interest area using the detected characteristic scale.
2. calculate the second moment matrix M at the detected location and scale.
3. use the second moment matrix to normalize the local interest area. This is done by applying the inverse of the second moment matrix as a local homography matrix.
4. Re-detect the local interest point, using a scale-space invariant approach, on the normalized local interest area.
5. Repeat step 2-4 until the local interest point location and area estimation remains stable.

Figure 3.7. Iterative approach to obtain affine invariance using the second moment matrix.

This technique uses scale-space peaks in the difference of Gaussian operator convolved with the image. The difference of Gaussian operator $DOG(I, \mathbf{x}, \sigma)$ is defined as:

$$DOG(I, \mathbf{x}, \sigma) = (G(\mathbf{x}, k\sigma) - G(\mathbf{x}, \sigma)) * I = L(\mathbf{x}, k\sigma) - L(\mathbf{x}, \sigma). \tag{3.12}$$

In Equation 3.12, we can observe that the DOG of an image can be computed from the difference of two Gaussian smoothed images, with a smoothing difference by a factor k. This allow the DOG interest points to be detected very efficiently by building an image pyramid with re-sampling between each octave. Furthermore, the DOG operator locates key points at regions and scales of high variation, making these locations particularly stable for characterizing the image. The DOG point detection can be divided into two part: the calculation of the DOG scale-space representation, and the search for the maxima in scale-space. We will start by describing the scale-space calculation.

The DOG detector is based on a Gaussian scale-space representation within each octave and a pyramid (re-sampling based) scale-space representation between octaves. Each octave is divided into s scale-space levels, the higher the s the more accurate the characteristic scale detection will be. Given a specific number of scale-space levels s per octave, we obtain the smoothing factor between levels inside an octave to be $k = 2^{1/s}$. However, with the increase of s the extraction

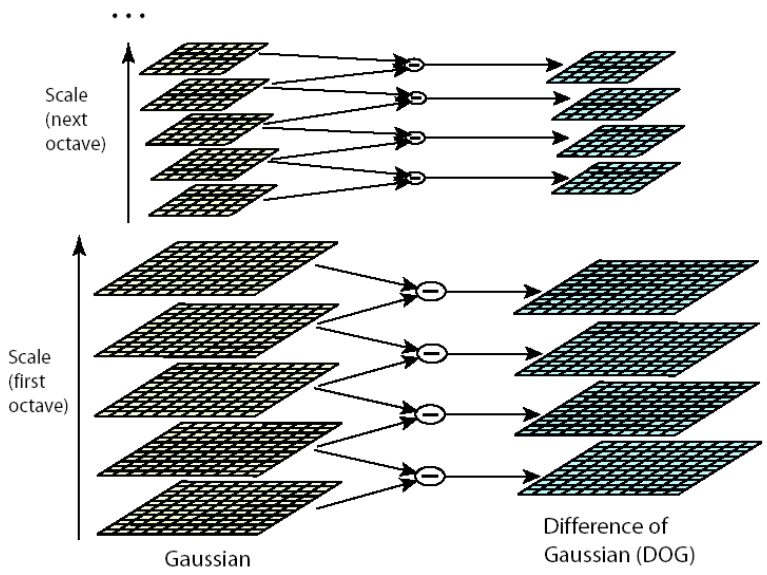


Figure 3.8. Illustration of the DOG scale space computation. The scale space is divided into octaves for which the image size stays the same.

process becomes slower as we need to compute more scale-space levels for each octave. In order to be able to compute s DOG detection levels, we need $s + 1$ scale-space levels. Also, since we want to perform a comparison of the DOG response with its neighbors at adjacent scale levels we need an extra DOG detection level. This leads to a total of $s + 3$ Gaussian levels in our scale-space representation, which spans slightly over one octave.

The detection of interest points using the DOG local interest point detector can be summarized as described in Figure 3.1.2.

Maxima and minima of the DOG scale-space operator are determined by comparing each pixel in the pyramid to its 26 neighbors, Figure 3.10 illustrates the neighborhood of the point in scale space. First, a pixel is compared to its 8 neighbors at the same level of the pyramid. If it is a maxima or minima at this level, then it is compared with the 9 pixels in the neighborhood of the same location at the lower level DOG and the 9 at higher level DOG. Since most pixels will be eliminated within a few comparisons, the cost of this detection is small and much lower than that of building the scale space representation. The points which are not eliminated are our local maxima/minima and constitute the resulting local interest points. For each point we also know the scale at which it was detected, which enables us to associate a characteristic scale to each point, achieving in this way scale invariance.

DOG detector's local interest point extraction process

1. Smooth original image $I_{initial}$ with Gaussian $G(\mathbf{x}, \sqrt{2})$, to create the first image of the scale space representation.
2. apply smoothing equal to smoothing step σ_k, to image $I_{initial}$.
3. subtract each Gaussian scale-space smoothed image with the image immediately lower in the scale-space representation.
4. perform DOG local interest point detection using a maxima detection procedure on the current Gaussian scale space representation.
5. set the initial image $I_{initial}$ to be the last Gaussian smooth image of the current octave.
6. re-sample the initial image $I_{initial}$ by taking each other pixel, creating an image of half the size of the original.
7. if image size larger that two times the size of the Gaussian kernel used to create the scale space representation, return to step 2.

Figure 3.9. Description of the algorithm for scene segmentation using model 1.

Saliency detector

Kadir and Brady (2001) have proposed a different approach for a scale invariant detector. In their work the authors used entropy to measure complexity and estimate local saliency. The may idea of this detector is that salient image regions exhibit unpredictability, or *surprise*, in their local attributes and over spatial scale. The method searches for scale localized features with high entropy. Search is based on a local image feature computed inside a circular search window. Exhaustive search using the circular window at different scales is performed, for each location and scale a local image feature is computed. The salient scale is selected at the entropy extrema of the local feature. The original implementation defined entropy over the grey level of the image intensity inside the circular window (local feature used was a histogram). However, other attributes like color or gradient may be used instead. The local interest point extraction process according to this method consists of three steps:

- Calculate Shannon entropy of local image grey-scale pixel values inside the area defined by the current position $\mathbf{x}$ and a range of scales σ, $H_D(s, \mathbf{x})$:

$$\mathcal{H}_D(\sigma, \mathbf{x}) \stackrel{\Delta}{=} -\int p(R_I, \sigma, \mathbf{x}) \log_2(p(R_I, \sigma, \mathbf{x}))dR_I \tag{3.13}$$

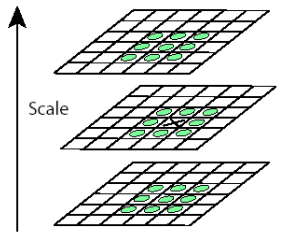


Figure 3.10. comparison with the neighbors

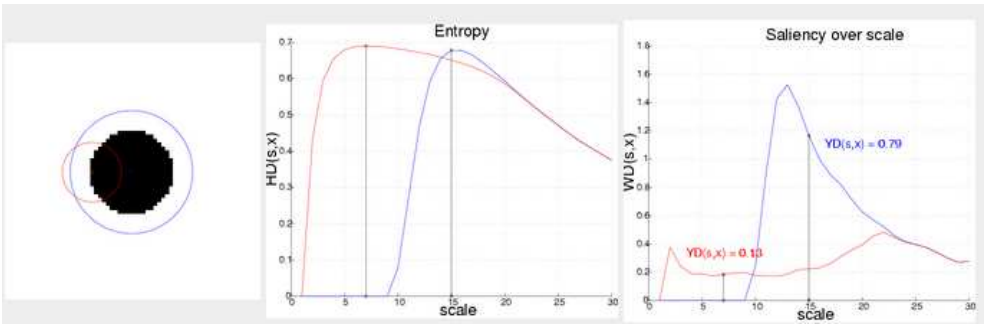


Figure 3.11. Illustration of the detector's scale detection of a local interest point at a local circular region. (left) detected location of maximum entropy for both circular windows (red -smaller scale, incorrect location-, and blue -larger scale, correct location-). (center) entropy graph showing the evolution over scale for both red and blue window locations (both entropy peaks occur at similar magnitudes). (right) saliency measure for both circles where we can observe the low response for the wrong detection, for the previous entropy maxima.

- Select scales at which the entropy over scale function exhibits a peak, σ_p,
- Calculate the magnitude change of the PDF as a function of scale at each peak, $W_D(\sigma, \mathbf{x})$.

$$\mathcal{W}_D(\sigma, \mathbf{x}) \triangleq \frac{\sigma^2}{2\sigma - 1} \int \mid \frac{\partial}{\partial \sigma} p(R_I, \sigma, \mathbf{x}) \mid dI \tag{3.14}$$

where R_I is the image representation we use to describe the local interest area The saliency is defined by the product of $H_D(s, x)$ and $W_D(s, x)$ at each peak.

Originally this method was limited to isotropic scale but was later extended (Kadir *et al.* (2004)). To achieve affine invariance using the same framework the circular sampling window is replaced with an ellipse (an affine transformation maps circles to ellipses). Scalar σ is replaced by vector $t = (\sigma, e, o)$ describing scale, ellipse eccentricity and orientation respectively. The local interest points are now found through a search over three parameters (more details in Kadir *et al.* (2004)).

Figure 3.12. Region contours detected using the MSER detector.

Maximally stable extremum regions (MSER)

Matas *et al.* (2002b) defined a scale and affine invariant local interest area detector, based on the intensity landscape of the grey-scale image, the maximally stable extremum regions (MSER). Instead of detecting a local interest point and then defining a local interest area around it, this detector obtains the area directly. This detector is based on regions in the image that have the same stable shape over a range of thresholds of the image intensities. The process of detection of these regions can be described by considering all the possible thresholds of a given grey-scale image. At each possible threshold there will be a part of the pixel in the image which are above that threshold, those will form binarized regions. The detected local interest areas will be a subset of all the regions detected at all thresholds, which shape remains stable across a range of thresholds. These regions are of interest since they are invariant to affine transformations of the image intensity (Matas *et al.* (2002b)). The resulting regions detected in the images will correspond to what we perceive as uniform regions, in figure 3.12 we show images with MSER detected regions.

This approach shares some principles with the watershed algorithm in that both algorithms explore the binarization of images at several thresholds to define regions, however in the MSER case the focus is on the thresholds where regions remain stable. The watershed algorithm focuses on the thresholds where regions merge, and two watersheds touch (Vincent and Soille (1991)).

3.2 Local descriptors

After obtaining all the interest locations $\mathcal{L}_{\mathbf{x}} = \{(\mathbf{x}_i, \mathcal{A}_i), i = 1, \ldots, N_{\mathbf{x}}\}$ in an image, we need to compute the local descriptors. The features $f_i = \mathcal{F}(I, \mathbf{x}_i, \mathcal{A}_i)$ for each of those areas in order to describe them. Local descriptors are designed to capture a compact and complete description of the local area in order to allow for a similarity measure to be applied between interest points. These features must be highly distinctive and yet as invariant as possible to remaining invariance

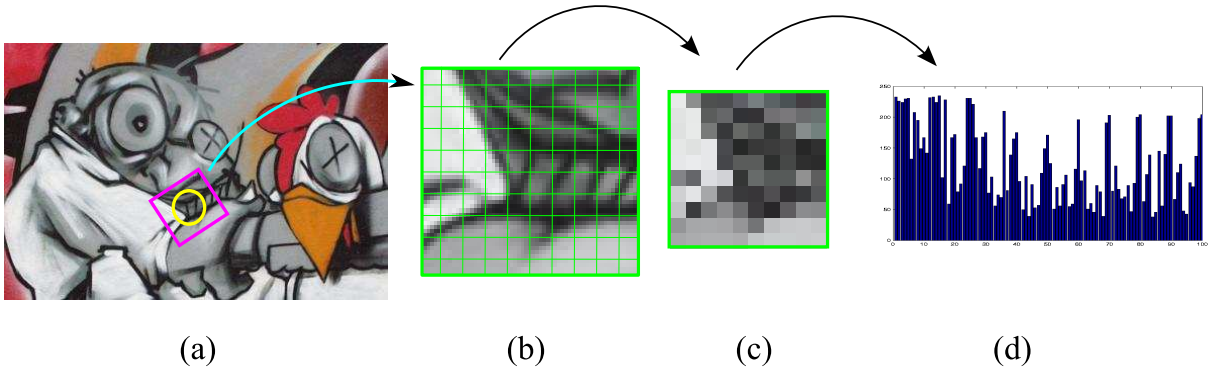


Figure 3.13. Illustration of the extraction of a rescaled pixel sampling based feature. (a) original image with detected area and patch to sample. (b) orientation normalized patch with sampling grid. (c) resulting sampled grey scale patch. (d) feature vector obtained from the grey scale patch.

issues and possible noise. These features are what we entitle local interest point descriptors (even if in fact they describe the local interest area).

Invariance and distinctiveness of local descriptors are inversely dependent, we normally lose invariance to obtain a more specific descriptor. In general, the more the description is invariant the less information it conveys. Due to the increase of the local interest point's invariance to image transformations, local interest areas are increasingly more stable, which makes the local descriptor's invariance less influential in the systems performance, allowing for more specific descriptors. The problem is to compute a complete representation that is simultaneously compact and specific. In the next subsection we will present some local descriptors explaining their feature extraction process and their possible invariance to the variability of the local interest area.

3.2.1 Grey-scale image sampling

The simplest feature that we can use to describe a local interest area is the image pixels' intensity inside that area. Cross-correlation of grey-scale image patches, defined by the local interest area, is the simplest implementation to achieve matching between points (Mikolajczyk and Schmid (2003); Schaffalitzky and Zisserman (2002)). To use the grey-scale values of the local interest area as a local interest descriptor, we sample $N \times N$ pixel values from that area and create a N^2 dimensional vector. Figure 3.13 illustrates the steps of the creation of a grey-scale feature from a detected local interest area. This approach has no invariance to noise or illumination changes. Nevertheless, its great simplicity makes this approach still useful in some cases (Fei-Fei and Perona (2005)).

As we seen before local interest point detectors provide a good degree of invariance. The resulting local interest area is invariant to geometric and illumination transformations. This allows us to relax the need for robustness on the part of the local descriptor feature. Researchers in recent

work have reported that the use of the image patch sampling is enough to describe the local area for the task related to quantized local descriptors (Fei-Fei and Perona (2005); Quelhas *et al.* (2005); Sivic *et al.* (2005)).

3.2.2 Differential invariants

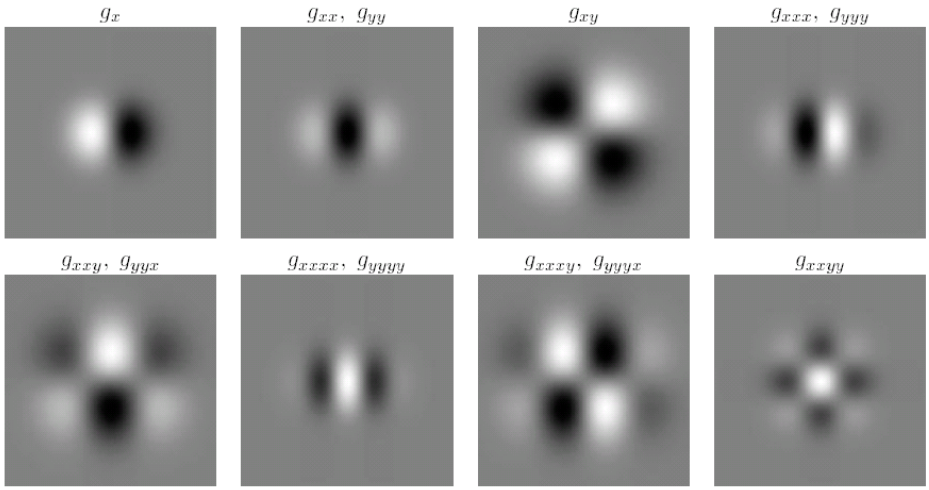


Figure 3.14. Uniform Gaussian derivatives up to fourth order. The derivative kernel g_{yy} is equal to the g_{xx} kernel rotated by 90 degree. Similarly, we can obtain g_{yyy}, g_{yyx}, g_{yyyy}, and g_{yyyx} kernels.

Differential invariants based descriptors obtain a decomposition of the local structure surrounding a local interest point based on the convolution of that points neighborhood with a set of Gaussian derivative basis (see Figure 3.14). The response to different Gaussian derivatives are combined to obtain the description of the local structure. In most cases the authors propose a combination of the allowing also to obtain some invariance if required (Schmid and Mohr (1997); Schaffalitzky and Zisserman (2002); Baumberg (2000)).

Koenderink and van Doorn (1987) proposed to describe the local structure around a point $\mathbf{x}$ by using a set of *local jet* of order p. The full range of *local jets* is defined by:

$$J_p(I, \mathbf{x}, \sigma) = \{L_{i_1 \ldots i_n}(\mathbf{x}, \sigma)\}, \forall i \in \{x, y\}, \forall n = [1, \ldots, p], \tag{3.15}$$

where $L_{i_n}(\mathbf{x}, \sigma)$ corresponds to the convolution of the image with the Gaussian derivative kernel $G_{i_n}(\mathbf{x}, \sigma)$. See Figure 3.14 for images of the Gaussian derivative kernels.

Local Jets descriptors are one of several approaches that base the local interest area description in a decomposition of the gradient information inside that area into a predefined gradient basis.

Baumberg (2000) and later on Schaffalitzky and Zisserman (2002) proposed to use a family of Gaussian filters which can be combined to represent a Gaussian derivative in any orientation. These family of Gaussian filters is entitled steerable filters. Using steerable filters, Baumberg (2000) created a combination of several individual filter in a orientation invariant way.

3.2.3 Generalized color moments

Tuytelaars and Gool (2000) presented a local descriptor where the local interest area is characterized by color moment invariants. They used generalized color moments, introduced in Mindru *et al.* (1999) to better exploit the multi-spectral nature of the data. These moments contain powers of the image coordinates and of the intensities of the different color channels. The general equation of these moments is as follows:

$$M_{pq}^{abc} = \int_{\Omega} \int x^p y^q \left[R(x,y)\right]^a \left[G(x,y)\right]^b \left[B(x,y)\right]^c dxdy \tag{3.16}$$

where the moment will have order $p+q$ and degree $a+b+c$.

For the invariant features the authors use 18 moment invariants. These are invariant functions of moments up to the first order and second degree (i.e. moments that use up to second order powers of intensities (R,G,B) and first order powers of (x,y) coordinates). These 18 moments form a basis for all geometric/photometric invariants involving this kind of moments (Mindru *et al.* (1999)).

3.2.4 SIFT

The SIFT feature describes the local interest area using a concatenation of local histograms of edge orientation computed over the a grid sub-division of the local interest area's gradient map (Lowe (1999, 2004)), see Figure 3.15. SIFT features have become widely used for both wide-baseline matching and quantized local descriptor approaches and have been found to perform best for many tasks by several authors (Mikolajczyk and Schmid (2005); Lowe (2004); Fei-Fei and Perona (2005); Quelhas *et al.* (2005); Bosch *et al.* (2006)). This was the main local descriptor used throughout this thesis.

The SIFT extraction process is based on the extraction of gradient samples from the image at the scale of the local interest point to describe. In its original formulation (Lowe (1999)), SIFT feature extraction was coupled with the scale-space representation of the DOG interest point detector. To increase the speed of the feature extraction process, the author used the pre-computed scale-space smoothed images to compute the SIFT descriptor feature. If we apply the SIFT descriptor together with some other local interest point detector, which does not have a scale-space representation, we must Gaussian smooth (or re-sample) the image to the scale of

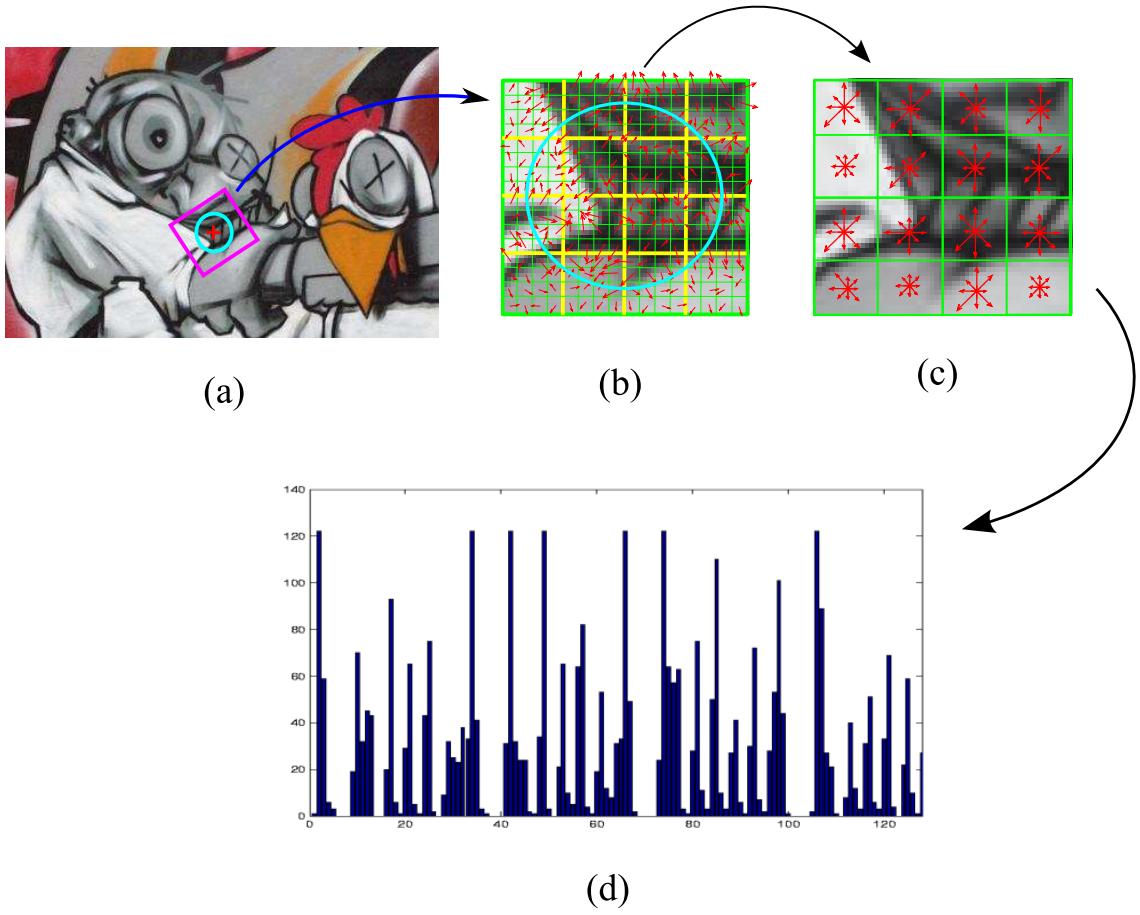


Figure 3.15. Illustration of the SIFT feature extraction process. (a) original image with the local interest point to describe, showing the detected location, scale and area used for sampling. (b) local interest area with gradient samples at each grid point, blue circle illustrates the Gaussian weighting window. (c) local individual orientation histograms which result of accumulating each sample into the corresponding bin of its local histogram. (d) final 128 dimensional SIFT feature (before normalization).

the detected point. We consider the case where we have access the scale-space representation which was used to extract the local interest point.

The SIFT feature extraction process, as illustrated in Figure 3.15, is summarized in Figure 3.2.4. In short, the image's gradient is sampled and its orientation is quantized. Using a grid division of the local interest area, local gradient orientation histograms are created where the gradient magnitude is accumulated. The final feature is the concatenation of all the local gradient orientation histograms. A Gaussian weighting is introduce in the SIFT feature extraction process (Figure 3.2.4) to give more importance to samples closer to the center of the local interest area. This contributes to a greater invariance of the SIFT descriptor, since samples closer to the center of the local interest areas are more robust to errors in the local interest area estimation.

SIFT feature extraction procedure.

1. select the Gaussian smoothed image corresponding to the local interest point's characteristic scale (σ_D),
2. sample the image gradient based on the scale and orientation of the local interest point, using a regular grid around the local interest point location $\mathbf{x}_i$ (Figure 3.15(b)),
3. normalize the sampled gradient's orientation with relation to the local interest point's orientation,
4. apply a Gaussian weighting to the gradient's magnitude, with $\sigma = \sigma_D/2$. (light blue circle in Figure 3.15(b)),
5. quantize the gradients orientation into n orientations ($n = 8$ in Figure 3.15(c)),
6. create grid division orientation histograms in which to accumulate the magnitude of the previously quantized local gradient (yellow grid in Figure 3.15(b)),
7. form a vector by concatenating the grid histograms (Figure 3.15(c)) into one histogram (Figure 3.15(d)),
8. normalize the feature vector to further increase illumination invariance.

Figure 3.16. SIFT feature extraction process (more details in Lowe (2004)).

In Lowe (2004) it was found that the best compromise between performance an speed was obtained by using a 16×16 gradient sampling grid and a 4×4 sub-histogram grouping (cf. Figure 3.15). The final descriptor proposed in this formulations is 128 (4x4x8) dimensional.

As mentioned in the beginning of this subsection, the SIFT descriptors is one of most prominent local interest point descriptor. One of the main reasons for its success is its low complexity, which makes this detector fast and easy to implement. Another reason for the success of SIFT is the intrinsic invariance to small errors in the calculation of the position and area, resulting from representing the local image information with a histogram.

The Gradient Location Orientation Histogram (GLOH) has recently been proposed as an extension to the SIFT descriptor, to further increase its robustness and distinctiveness (Mikolajczyk and Schmid (2005)). This approach uses a log-polar location grid with 3 bins in radial direction and 8 in angular, instead of the regular $N \times N$ grid of the SIFT, which results 17 location bins. The gradient orientations are quantized in 16 bins. This gives a 272 bin histogram. When

applied to the task of wide-baseline matching this descriptors was found to provide a small improvements over the SIFT (Mikolajczyk and Schmid (2005)). This descriptor as been, until the time of the writing of this thesis, little explored. However, this descriptor may be a more stable alternative to the SIFT descriptor.

3.2.5 PCA-SIFT

PCA-SIFT was introduced in an attempt to create a more compact local descriptor, which could obtain a level of performance similar to that of SIFT on wide-baseline tasks (Ke and Sukthankar (2004)). PCA-SIFT makes use of a PCA projection of the gradient map in the local interest area to describe the local structure and texture. Similar to grey-scale image sampling (see Section 3.2.1), PCA-SIFT features are obtained by extracting a sub-sampled patch of the local interest area. PCA-SIFT can be summarized in the following steps:

- subsample the local interest area,
- calculate the dx and dy gradient maps,
- **training:** pre-compute an eigen-space to express the dx and dy gradient maps of the local interest areas using a set of training data.
- **testing:** project the gradient of the new local interest areas into the pre-computed eigen-space to obtain a decomposition of the gradient map, the eigen space coefficients of the obtained decomposition are the elements of the PCA-SIFT feature vector.

This feature is substantially more compact that SIFT (20 dimensions was found to perform well) and is reported to perform similar to SIFT or better in the task of wide-baseline matching(Ke and Sukthankar (2004)). However, little experiments have been performed using this feature in the framework of quantized local descriptors (Zhao *et al.* (2006)). This detector may in fact be an interesting alternative for both wide-baseline matching and quantized local descriptors frameworks.

3.3 Wide-baseline performance evaluation

As we explained in the beginning of this section, one of the main applications of local interest point detectors and descriptors is the point-to-point correspondence attribution between images, of a certain object or scene. These images can have a different point-of-view, orientation, and scale. We can compare the repeatability of detectors and the performance of descriptors on the task of point-to-point matching by using datasets for which we know the perspective transformations between images (ground-truth homography).

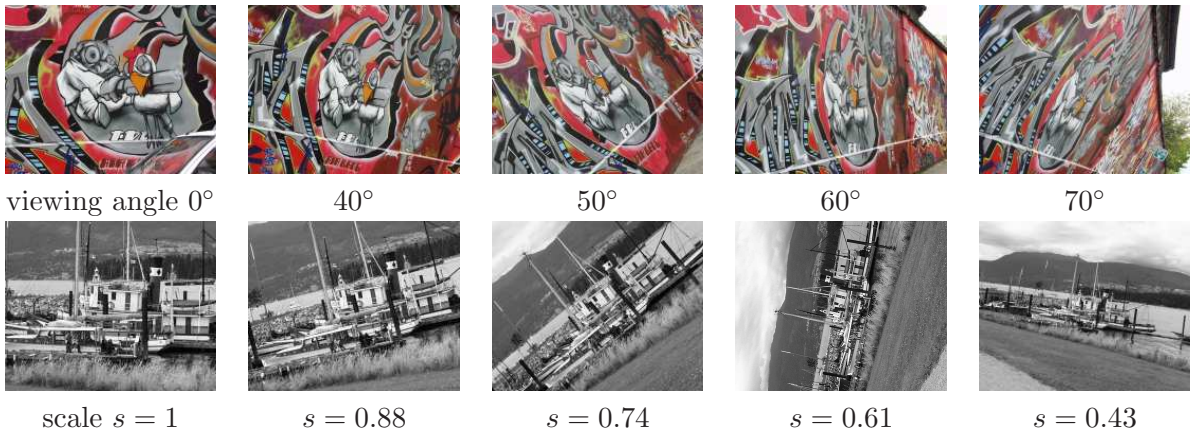


Figure 3.17. Wide-baseline testing datasets: view-point change (top) and scale change (bottom). Under each image we show the relative viewing angle change or scale change.

For these experiments, we use the image datasets introduced by Mikolajczy and Schmid (2004), which contains the images sets for viewing angle and scale changes (together with varying orientation). We consider all transformations relatively to a first reference image, which we will try to match to all other images of the same scene. Figure 3.17 shows both datasets with viewing point (top) and scale changes (bottom). The images are either of planar scenes or the camera position is fixed during acquisition, so that in all cases the images are related by homographies, i.e. plane projective transformations (Hartley and Zisserman (2000)). This means that the mapping relating images is known (or can be computed), and this mapping is used to determine ground truth matches for the affine covariant detectors. In the viewing point change test the camera varies from a frontal-parallel view to one with significant foreshortening at approximately 60 degrees to the camera. The scale change is acquired by varying the camera zoom, and varies from 1 to 0.43. Images have a resolution of 800×640 in the viewing point change dataset and 850×680 in the scale change dataset. We will not perform any evaluation of the rotation invariance of our detectors, since all used detectors obtain orientation invariance in a similar manner.

Implementation of detectors and descriptors

Before starting with performance experiments, we need to clarify one remaining issue regarding the use of local interest point detectors and descriptors, the issue of implementation. Different implementations of the same detector and descriptor may have different performances. In this chapter we use binaries available at `http://www.robots.ox.ac.uk/ṽgg/research/affine/`. In Chapter 5 we use both the binaries from `http://lear.inrialpes.fr/people/mikolajczyk/` and those available in `http://lear.inrialpes.fr/people/dorko/downloads.html`. As we will see in Chapter 5, for scene classification, performance varies considerably for different implementations.

The variations in performance have mainly to do with implementation choices, for instance some implementations may be designed to be faster while other may aim at being more accurate. As such, it is important to use similar implementations when comparing local interest point detectors and descriptors.

3.3.1 Local interest point detector evaluation

We will now compare the performance of some local interest point detectors, presented in Section 3.1.2, when applied to images that undergo scale and viewing angle changes (affine transformations). We will use as performance measures the accuracy and stability of the point detection result. The stability and accuracy of the detectors is evaluated using the repeatability criterion introduced in Schmid *et al.* (2000), see also Mikolajczyk and Schmid (2005). Other criteria exists in literature and could be equally used (Lowe (2004); Sebe and Lew (2003)).

The repeatability score for a method's performance in a point-to-point correspondence task between two images is given by the ratio between the number of correct point-to-point correspondences found and the number of detected points inside the area of the scene/object present in both images:

$$Repeatability = \frac{\#\text{ of correct correspondences}}{\#\text{ of detected points}} \tag{3.17}$$

To verify the correctness between any matches between two point we use the known planar homography between images H. We do not use a full 3D homography model since our scenes are approximately planar. This also enables us to obtain a direct point to point projection between images, contrary to a point to line in the case of a full 3D homography model (Hartley and Zisserman (2000)). Distances between points are in this way simple to calculate. In concrete, we consider two points to correspond to each other if:

1. The error in relative point location is less than 1.5 pixels: $||\mathbf{x}_a - H.\mathbf{x}_b|| < 1.5$, where H is the homography between images.

2. The error in the image surface covered by the local interest area is less that 40% of that same area ($\epsilon_\sigma < 0.4$). This has different implications for the case of a scale invariant detector or an affine invariant detector. For the case of a scale invariant detector the surface error is:

$$\epsilon_\sigma = \left| 1 - \sigma_h^2 \frac{\min(\sigma_a^2, \sigma_b^2)}{\max(\sigma_a^2, \sigma_b^2)} \right| \tag{3.18}$$

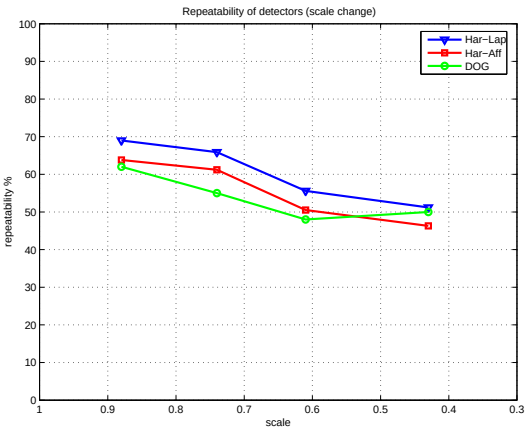


Figure 3.18. Repeatability of interest points with respect to scale changes.

where σ_a and σ_b are the scales selected by the detector for corresponding points x_a and xb, and σ_h is the actual scale variation between the two image I_a and I_b (know previously from the known homography).

In the case of a fully affine detector the surface error is defined as:

$$\epsilon_S = \left| 1 - \frac{(r_a \cap (H_l^T r_b H_l))}{(r_a \cup (H_l^T r_b H_l))} \right| \tag{3.19}$$

where r_a and r_b are the elliptic regions defined by $x^T r x = 1$. The union of the regions is $(r_a \cup (H_l^T r_b H_l))$ and $(r_a \cap (H_l^T r_b H_l))$ is their intersection. H_l is the locally linearized homography H in point x_b. In this calculation we ignored any possible error of position allowed in the previous verification (maximum 1.5 pixels).

Figures 3.18 and 3.19 show the repeatability for the cases of scale and viewing angle change. With regards to changes in scale we can see that scale invariant descriptors (DOG and Harris-Lap) obtain a high repeatability, and that the Harris-Affine detector performs slightly better than DOG but worse that Harris-Lap.

In the case on viewing angle change, shown in Figure 3.19, scale invariant detectors (DOG and Harris-Lap) have good repeatability for small viewing angle changes, but their performance degrades rapidly for higher viewing angle changes. On the other hand, Harris-affine, although having less repeatability for small viewing angle changes, maintains a good performance as the viewing angle change increases.

From these results we can conclude that if we need consistent point-to-point matching across images that may undergo affine transformation, then a fully affine detector will be the best choice. However if our viewing conditions vary only in scale, we may find it more advantageous to use a simpler scale invariant detector.

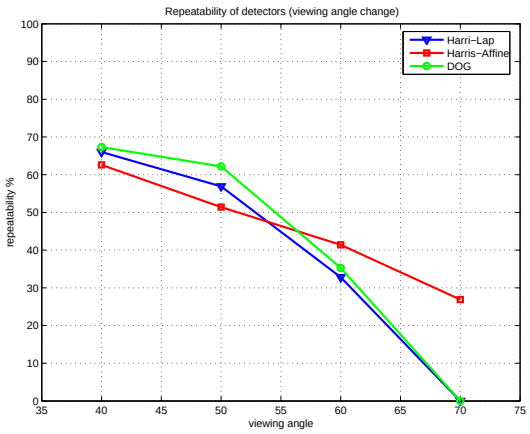


Figure 3.19. Repeatability of interest points with respect to viewing angle changes.

Detector	run time (in seconds)	number of detected points
DOG	0.7	1527
HAR-LAP	7	1438
HAR-AFF	12	1463

Table 3.1. Complexity of several detectors. (HARR-LAP stands for the Harris Laplace scale invariant detector and HARR-AFF stands for the Harris affine invariant detector)

3.3.2 Local interest point detector computational complexity

The computational complexity of a local interest point detector is an important factor for the usability of the detector in a practical system. As a local detector becomes more complex, its extraction speed is reduced. Either in the case of large datasets, where all images need to be processed, or user input images, where the user is waiting for the results, an increase in processing time may mean a decrease in usability of the system.

Table 3.1 shows the run time for the extraction of local interest points from an 800×640 image using several detectors. These results are reproduced from Mikolajczy and Schmid (2004), which were obtained using a Pentium II 500 MHz computer. We can observe that the DOG detector is considerably faster since it is based on the subtraction of images. It is, in most cases, possible to increase the local interest point detection speed at the cost of accuracy, if the application demands for more speed. On the other hand, we may be willing to reduce the detector speed to increase its accuracy. Even if the DOG detector is not always the most accurate detector, as we seen in the previous subsection, its speed makes it a very interesting detector for real systems.

3.3.3 Local interest point descriptor evaluation

To evaluate the performance of a local interest point descriptor, we must analyze the correctness of the matches obtained using that descriptor. The evaluation criterion we use is based on the number of correct matches and the number of false matches obtained between image pairs. This is similar to the criterion proposed by Ke and Sukthankar (2004) and also used in Mikolajczyk and Schmid (2005).

In this performance test we use two image separated by a viewing angle of 50°, images at 0° and 50° degrees in Figure 3.17, and we consider the task of matching local interest points from the first image (0°) with the local interest points from the second image (50°). For each image we extract local interest points using the Harris-affine local interest point detector. Using the framework presented in Section 3.3.1, we select the Harris-affine local interest points that correspond correctly between images, obtaining the ground-truth point-to-point correspondence between the points in both images. In this case we define a correct detection as one that has an error in relative point location lower than 1.5 pixels, and for which the error in the covered image surface is less that 50% of that same area ($\epsilon_\sigma < 0.5$) (see Equation 3.19). We consider the set of correct corresponding local interest point to extract our local descriptors.

Given the two sets of descriptors extracted from both images, we now obtain a point-to-point correspondence based on the distance between descriptors, which we will then compare with the ground-truth point-to-point correspondence. We attribute the point-to-point correspondences from the 0° image to the 50°, by making each point in the first image match the point in the second image with the smallest distance between the respective descriptors, subject to a threshold. The smallest the threshold on the distance, the more demanding we are on the similarity of the descriptors.

To analyze the point-to-point correspondence performance we use $1 - precision$ and $recall$ defined as:

$$Recall = \frac{\#\text{ of correct matches}}{\#\text{ of ground-truth correspondences}} \tag{3.20}$$

$$1 - Precision = \frac{\#\text{ of false matches}}{\#\text{ of correct matches} + \#\text{ of false matches}} \tag{3.21}$$

A perfect descriptor would allow us to obtain a recall equal to 1 for any precision. However, as we would expect, low distance thresholds will provide high precision, by pruning most bad results, but will cause a final low recall value. On the other hand, high threshold values will provide lower precision but higher recall.

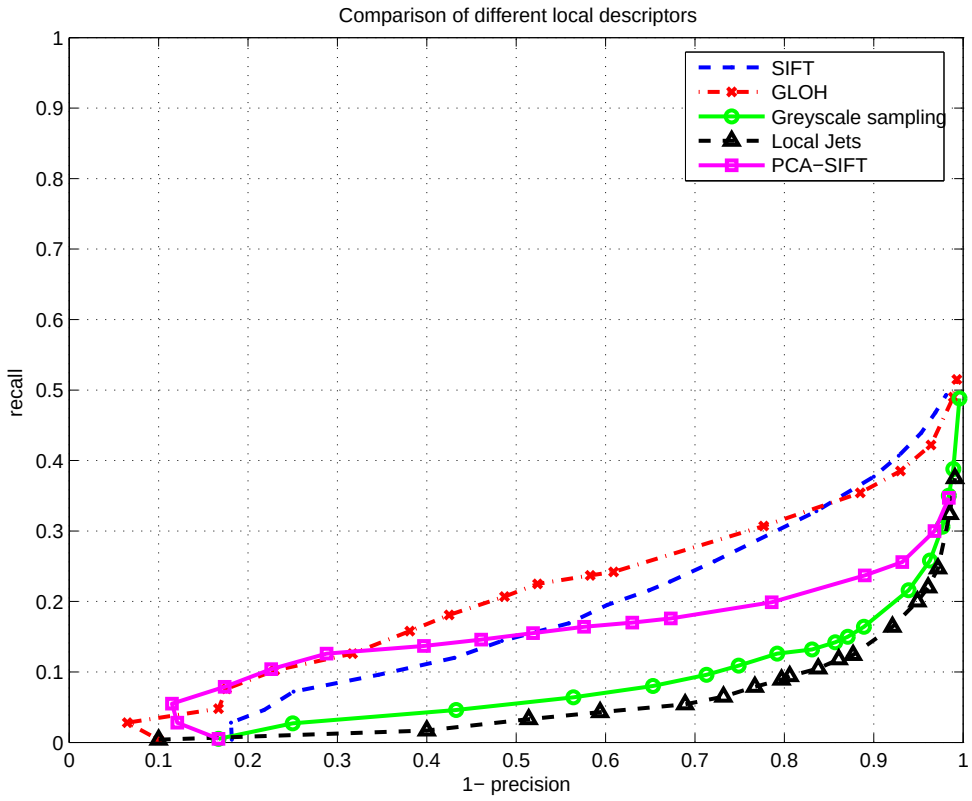


Figure 3.20. Graph for the local interest point descriptors evaluation, with viewing point angle change. These results are based on Harris affine detected regions

Figure 3.20 shows the matching performance for several descriptors as we change the matching criteria threshold. We can see that more recent approaches to local interest point description (SIFT, GLOH, PCA-SIFT) outperform more classical approaches (grey-scale sampling, local Jets). Overall, the GLOH descriptor is the best performing descriptor we tested, a close second is the SIFT descriptor. However, among the highest performing approaches there are advantages in using different descriptors depending if we are interested in a higher precision or recall. PCA-SIFT may be interesting in a task that requires high precision, or when we need a lower dimensional descriptor (more compact representation).

3.4 Chapter conclusion

In this chapter, we presented an overview of several representative examples of local interest point detectors and descriptors. We explored the most widely used approaches to obtain invariance to image transformations. Finally, we compared several detectors and descriptors, in the task of point-to-point matching across image with varying scale and viewing angle.

Given the results obtained in the task of point-to-point matching we observed that in the case of an image scale change, scale invariant obtain a similar performance to affine invariant methods. On the other hand, in the presence of an affine image transformation, scale invariant methods were not capable of handling large changes in viewing angle. These results motivate the use of detector which are adequate for the task to solve.

Chapter 4

Image representation

IMAGE representation is a very important element for image classification, annotation, segmentation or retrieval. Nearly all the methods in computer vision which deal with image content representation resort to features capable of representing image content in a compact way. In this chapter, we explore several models for image representation and the issues associated with them. The first image representation model is the bag-of-visterms (BOV), built from automatically extracted and quantized local descriptors refereed to as *visterms* in the remainder of this thesis. Alternative representations to BOV are then proposed. First, we introduce a soft clustering based BOV representation, where instead of quantizing the local descriptors a Gaussian Mixture Model (GMM) is used to model the distribution of local descriptors in the feature space. Secondly, to account for the potential discrimination power of the visterm, a visterm weighting procedure is applied to the BOV representation. Finally, a novel representation is obtained through a high-level abstraction of the BOV into a multinomial distribution over aspects using latent aspect modeling. In addition, we expand the BOV framework to handle several feature inputs using fusion of local features. Fusion will be introduced at two alternative stages of our approach: before quantization, combining information at the local descriptor level or after quantization, at the visterm level by concatenating the resulting BOVs.

Visterms are sometimes referred to as *visual words*, an analogy used by some authors. However, there has been little analysis on the possible differences or similarities between visterms in images and words in text. To explore the validity of this analogy, we will investigate the properties of the BOV representation and compare those properties with its text analog representation, the bag-of-words (BOW).

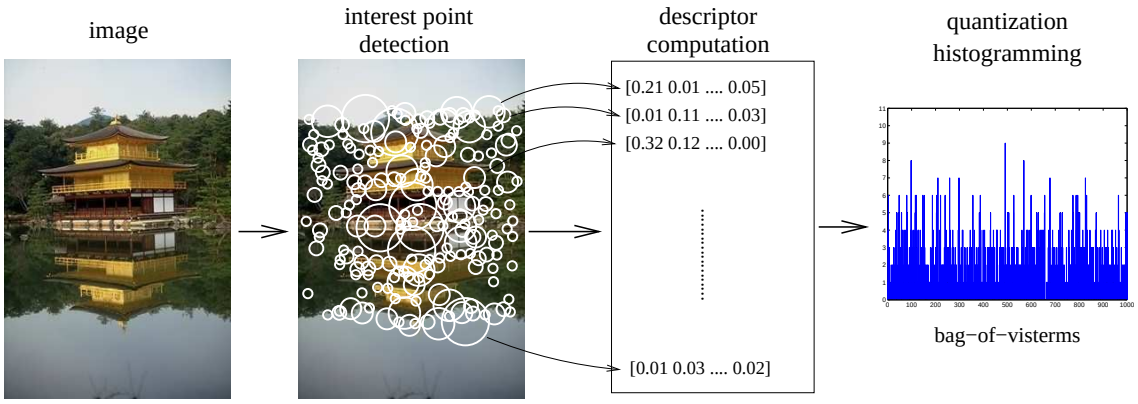


Figure 4.1. Computation steps of an image's BOV representation. From left to right: the original image, the image with the extracted local interest areas plotted, the feature extraction, and finally the resulting bag-of-visterms.

4.1 BOV representation

As previously discussed in Chapter 2 local features based representation can produce a versatile and robust image representation capable of representing global and local content at the same time. Describing an object or scene using local features computed at interest locations makes the description robust to partial occlusion and image transformation. This results from the local character of the features and their invariance to image transformations (see previous chapter). The BOV representation, which is derived from these local features, has been shown to be one of the best image representations in several tasks (as discussed in Chapter 2).

The BOV representation was first used by Willamowski *et al.* (2004) as an image representation for an object recognition system. In the BOV representation, local descriptors f_j are quantized into their respective visterms $v_i = Q(f_j)$ and used to represent the images from which they were extracted. The quantization process groups similar descriptors together, with the aim that the descriptors in each resulting group arise from local patterns with similar visual appearance. The number of occurrences of each group/visterm in a given image is the elementary feature of the BOV representation. More precisely, the BOV representation is the histogram of the various visterms' occurrences.

To construct the BOV feature vector h from an image I four steps are required. They are illustrated in Figure 4.1. In brief, local interest points are automatically detected in the image, then local descriptors are computed over the regions defined around those local interest points. This first step of local point extraction and descriptor computation is similar to the feature extraction process used in wide-baseline matching and described in the previous chapter. After this extraction step, the descriptors are quantized into visterms, and all occurrences of each visterm of the vocabulary are counted to build the BOV representation of the image. In the next subsection we detail the steps of the BOV construction, and explain some of the choices we made in our implementation.

4.1.1 BOV representation design

The BOV construction requires two main design decisions: the choice of local interest point detector/descriptors that we apply on our images to extract local features, and the choice of which method we use to obtain the visterms' vocabulary. Both these choices can influence the resulting system's performance. In the rest of this section we will first explore the choice of the local interest point detector/descriptor for our system and then tackle the problem of the vocabulary construction. All the choices were made within the scope of scene image classification, and were justified by several experiments done in this context. However, given a new task, the optimal choices could be different. Nevertheless, as we will see from the results of many experiments presented in this thesis, BOV is a robust image representation, which retains its good performance over a large range of parameter choices.

Local interest point detectors/descriptors

As we seen before in the previous chapter there exist several interest point detectors D in the current literature. They vary mostly by the amount of invariance they theoretically ensure, the image property they exploit to achieve invariance, and the type of image structures they are designed to detect (Mikolajczy and Schmid (2004); Tuytelaars and Gool (1999); Lowe (2004); Mikolajczyk *et al.* (2005)). For our local interest point detection task, we used the difference of Gaussians (DOG) point detector Lowe (2004), see Section 3.1.2 for details. This detector identifies blob-like regions where a maximum or minimum of intensity occurs in the image, and is invariant to translation, scale, rotation and constant illumination variations. We chose this detector since it was shown to perform well for the task of wide-baseline matching when compared to other detectors, as published in (Mikolajczyk and Schmid (2003); Mikolajczy and Schmid (2004)). This was also confirmed by our wide-baseline performance comparison presented in the previous chapter. This detector is also faster than similar performing detectors. An additional motivation to prefer the DOG detector over fully affine-invariant detectors, is that an increase of the degree of invariance removes information about the local image content, which may be valuable for classification. On the other hand we need some invariance so that we can recover a similar description from some typical transformed versions of the same image.

As for the description of the local interest area $\mathcal{A}$ we use the SIFT (Scale Invariant Feature Transform) feature introduced by Lowe (2004). The SIFT descriptor is a histogram based representation of the gradient orientations of the gray-scale image patch, see Section 3.2.4 for details on the SIFT feature construction and properties . This feature performs well when used in wide-baseline tasks, as seen in the previous chapter and as reported in related literature (Mikolajczyk and Schmid (2005)). SIFT was also found to work best for the task of object classification (Monay *et al.* (2005); Sivic *et al.* (2005)) and scene classification (Fergus *et al.* (2005a); Fei-Fei and Perona (2005); Quelhas *et al.* (2005)). In Section 5.6.2, we compare the performance of different local interest point detectors and descriptors when applied to scene image classification task.

Visual Vocabulary

Our visual vocabulary is defined by a set of visterms $V = \{v_i, i = 1 \dots N_V\}$ and an assignment function $Q(f)$ which assigns a visterm v_i to a feature f according to a nearest neighbor rule:

$$f \longmapsto Q(f) = v_i \Longleftrightarrow \text{dist}_Q(f, v_i) \leq \text{dist}_Q(f, v_j) \tag{4.1}$$

Given the set of visterms V, the feature space is divided into N_V separated regions $S = \{S_i, \dots, S_{N_V}\}$ according to:

$$S_i = \{f \mid Q(f) = v_i\} \tag{4.2}$$

where S_i if the region of the space where f resides. Our goal, in the vocabulary construction step, is to define the function $Q(f)$ or equivalently, the space division S, so that we can obtain a good BOV representation. To guaranty this a quantization method is normally used since such a methods split the feature space while minimizing the representation error.

When building a visual vocabulary from local features we need to choose which data to extract the vocabulary from, what quantization method to use, and how many visterms to have in our vocabulary. Straightforward reasoning motivates the extraction of the vocabulary from images that are as similar as possible to those on which we will test our system. However, we hope that the use of a universal vocabulary built from a large variety of images could perform as well. We will explore the dependence of our system's performance on the data we train our visual vocabulary in the next chapter.

Choosing the vocabulary quantization scheme remains an important task. In our work we rely on the most widely used approach, K-means clustering. This is a standard approach found across many works in the literature, and considered to be good performing approach (Sivic and Zisserman (2003); Sivic *et al.* (2005); Bosch *et al.* (2006); Quelhas *et al.* (2005); Willamowski *et al.* (2004); Fergus *et al.* (2005a); Fei-Fei and Perona (2005)). Recent work introduce novel ways to create the visterm vocabulary, such as hierarchical clustering (Leibe *et al.* (2006); Jurie and Triggs (2005); Mikolajczyk *et al.* (2006); Grauman and Darrell (2005)). However, due to their implementation complexity, we did not compare them to K-means clustering in our task.

K-means clustering is one of the simplest clustering algorithm (Bishop (1995)). This algorithm performs a partitioning (or clustering) of a set of data points $\mathcal{F} = \{f_i, i = 1, \dots, N_\mathcal{F}\}$ into disjoint subsets S_i [1] containing data points so as to minimize the sum-of-squared error function:

[1] With an abuse of notation we will also denote by S_i the set of features of $\mathcal{F}$ which belong to the region S_i.

$$J(\mathcal{F}) = \sum_{i=1}^{N_V} \sum_{f_j \in S_i} \mid f_j - \mu_i \mid^2, \tag{4.3}$$

where μ_i is the geometric centroid of the data point subset S_i. The K-means algorithm searches to partition the feature space into N_V regions, represented by a centroid μ. The algorithm proceeds by iterating two steps. The first step consists of assigning each data point (feature) to its closest centroid. In the second step each region's center is updated by computing the mean of the features that were assigned to region i. It can be shown that iterating between these two steps contributes to the minimization of the criteria J. The algorithm for K-means clustering is presented in Figure 4.2.

K-means clustering algorithm

1. randomly seed the K centroids using data points.
2. assign each data point to the group S_i that has the closest centroid μ_i, according to Equation 4.1.
3. recalculate the value of each centroid μ_i, assigning it the average of all data points in subset S_i.
4. repeat 2 and 3 until the values of the centroids μ_i do not change.

Figure 4.2. K-means clustering algorithm.

At the end of the K-means optimization, the resulting centroids define a polygon tessellation of the feature space, enabling the attribution of a label v_i to any new feature, according to Equation 4.1.

Given an image I with a set of features $\mathcal{F}(I) = \{f_j, j = 1 \dots N_{\mathcal{F}(I)}\}$, we can use the K-means model to attribute a label v_i to each local descriptors feature f_j, where i is the index of the closest centroid μ_i to feature f_j. By performing this attribution to every feature in that image we obtain the BOV representation $h(I)$ of that image:

$$h(I) = (h_j(I))_{j=1..N_V}, \text{ with } h_j(I) = \sum_{i=1}^{N_f(I)} \delta_j(Q(f_j)) \tag{4.4}$$

where $\delta_j(x)$ is equal to 1 for $x = j$ and 0 otherwise.

The hyper-parameter K denotes the size of our vocabulary, which defines the visual coherence and the diversity of our visterms, and consequently of the BOV image representation. Increasing

K will result in a finer division of the feature space, meaning that the visterms will be more specific. In principle this would produce a more precise representation of our image. However, there is the danger that this may result in an over-segmentation of the feature space, i.e. with several visterms representing the same local content. On the other hand, a small K will have visterms representing larger regions of the feature space, making the visterms less specific but making the image representation more stable across similar images. The trade-off between discriminative power and robustness of the BOV representation is in this way controlled by K, and will be discussed in the next chapter.

The BOV representation of an image contains no information about the spatial relationship between visterms. The standard BOW text representation results in a very similar 'simplification' of text data: even though word ordering contains a significant amount of information about the original data, it is completely removed from the final document representation. The bag-of-words approach has the advantage of producing a simple data representation, but potentially introduces the well known *synonymy* and *polysemy* ambiguities, as will be shown in Section 4.4.

4.2 BOV representation alternatives

In the previous section we presented the BOV representation. We will now explore some refinements to some of the steps in the BOV methodology and enhance the BOV representation. First, we will explore the usage of a GMM to replace the K-means vocabulary construction. Then we will consider the association of weights to the visterms that were obtained using K-means.

4.2.1 GMM based representation

Applying K-means clustering to the visterm modeling removes all knowledge about the distance of each particular feature f_j to the corresponding cluster center μ_i. This can be problematic in the case of two similar features, representing the same local structure/texture, that are assigned to two different clusters due to being close to the border between those two clusters. In this case our system could benefit from knowing that both features were in fact similar, by keeping the knowledge that both features were close to *b*oth clusters. This is an issue with any quantization based visual vocabulary construction. One way to address this issue is to perform soft clustering and assignment, in which we do not attribute a single label to each local descriptor but instead allow for multiple assignments with membership probabilities. For instance, we can attribute to a label v_i the probability that the local descriptor has been generated by the "cluster" center μ_i using a Gaussian generation process. In other words, for this modeling we use a Gaussian Mixture Model (GMM).

GMMs provide good flexibility and precision in modeling the underlying statistics of sample data. The main assumption in GMM modeling is that each data points has been generated by one Gaussian out of all the Gaussians in our mixture that generated each data point. The

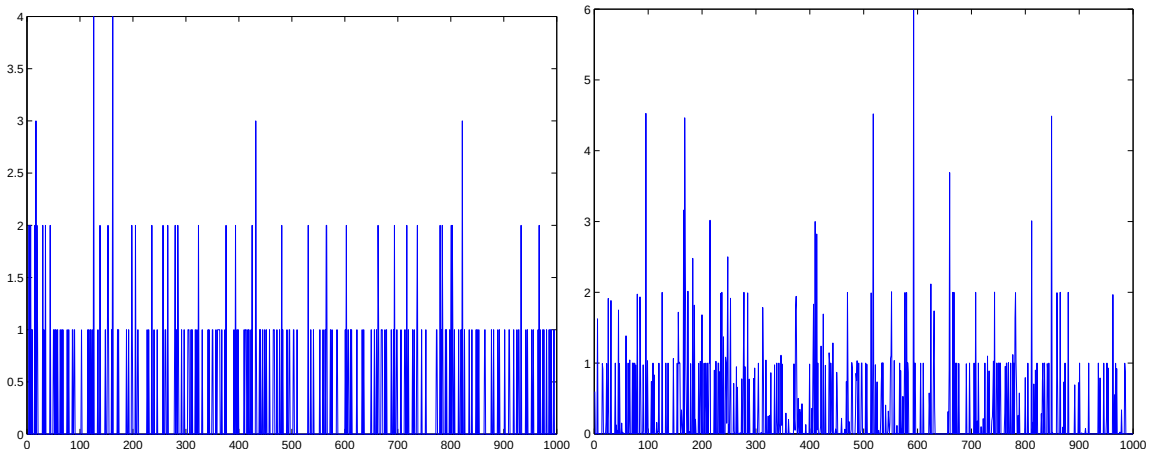


Figure 4.3. BOV representation of an image of our database using K-means clustering (left) and GMM modeling (right). In both the GMM and the K-means models we use 1000 clusters/Gaussians to model our features. We can see that the GMM modeling results in a more smooth BOV representation, without changing the basic shape of the histogram.

likelihood of a feature for a Gaussian distribution is given by:

$$\mathcal{N}(f;\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{|f|}{2}}\sqrt{\mid \Sigma \mid}} \exp\left(-\frac{1}{2}(f-\mu)^T\Sigma^{-1}(f-\mu)\right) \tag{4.5}$$

where μ denotes the mean, Σ the covariance matrix of the Gaussian distribution, $\mid f \mid$ is the feature dimension, and $\mid \Sigma \mid$ is the determinant of the Gaussian's covariance matrix. We will use diagonal covariances in our experiments. Let us denote by $g_i \in G = \{g_1, \ldots, g_{N_V}\}$ the latent variable indicating index of the Gaussian that generated a given feature. The likelihood of a feature for the GMM modeling is given by:

$$p(f_j) = \sum_{i=1}^{N_V} p(f_j, g_i) = \sum_{i=1}^{N_V} p(g_i)p(f_j \mid g_i) = \sum_{i=1}^{N_V} w_i.p(f_j \mid g_i), \tag{4.6}$$

where N_V is the number of Gaussians in the mixture, $p(g_i) = w_i$ denotes the probability of Gaussian i in the mixture, and $p(f_j \mid g_i) = \mathcal{N}(f_j;\mu_i,\Sigma_i)$. By definition, we have:

$$\sum_i w_i = 1, \text{ and } \forall i : w_i \geq 0. \tag{4.7}$$

We used the Expectation-Maximization (EM) with K-means initialization to train our GMM mixture model (Bishop (1995)). Given the complete mixture model, we can now define the new

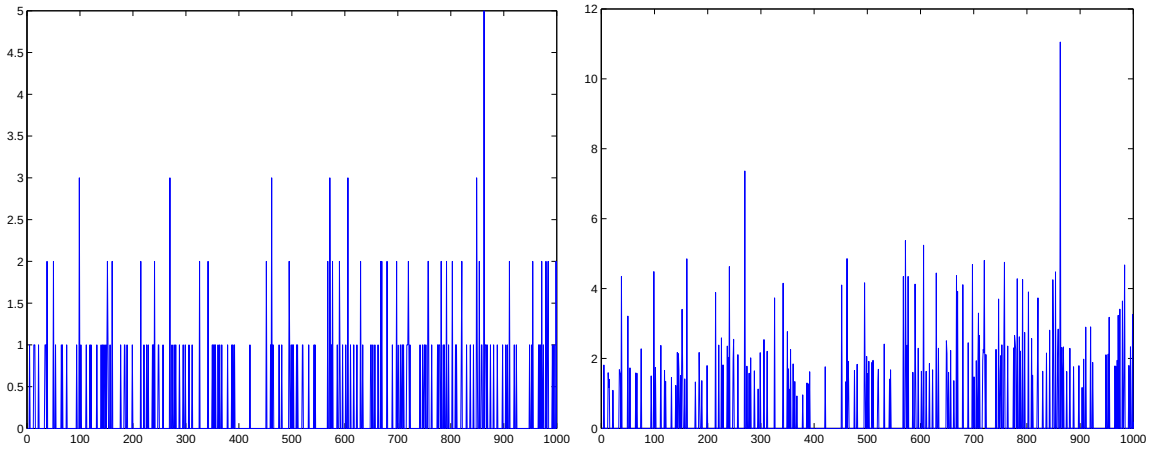


Figure 4.4. Plot of the original BOV representation of an image from dataset D^O (left) and its *tf-idf* weighted version (right).

soft BOV image representation, by using the probability of each feature in the image f_j being generated by each Gaussian g_i in our model. The new representation is defined as:

$$m(I) = (m_i(I))_{i=1..N_V}, \text{ with } m_i(I) = \sum_{j=1}^{N_f(I)} p(g_i \mid f_j) \tag{4.8}$$

where $N_f(I)$ is the total number of local descriptors in the image I and g_i is the ith Gaussian in the Mixture model. The posterior probability $p(g_i \mid f_j)$, denotes the probability of the Gaussian g_i having generated the local descriptor f_j, and can be calculated by:

$$p(g_i \mid f_j) = \frac{p(f_j \mid g_i)p(g_i)}{p(f_j)} \tag{4.9}$$

where $p(f_j)$ is defined by Equation 4.6

In figure 4.3 we can see both the K-means based hard BOV representation (left) and the GMM soft BOV representation (right) of the same image. The GMM based BOV shows a smoother binning distribution, while keeping the same overall distribution. Both these approaches are evaluated in the next section on scene image classification tasks.

4.2.2 TF-IDF weighting

When constructing vocabularies to represent text documents several step are taken to improve the BOW representation. Often, a weight is associated with each word in the vocabulary. If

this weight is low, the corresponding word can even be eliminated. Several pruning heuristics exist that try to give more importance to terms that should be, based on their occurrence, more characteristic of the documents' content (Salton and Buckley (1987); Baeza-Yates and Ribeiro-Neto (1999)). In this subsection we propose to apply such a weighting scheme to our visterm vocabularies.

One of the most often used weighting strategies in information retrieval is tf-idf (term frequency - inverse document frequency). This weight is a statistical measure used to evaluate how important a word is for a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is weighted accordingly to the frequency of the word in the corpus. Let us denote by $tfidf(I) = (t_j(I))_{j=1,\ldots,N_V}$, the representation vector associated with this strategy. $t_j(I)$ is defined by:

$$t_j(I) = tf(I, v_j) \times idf(v_j) \text{ with: } idf(v_j) = \log\left(\frac{N_D}{N_{D,v_j}}\right), \text{ and } tf(I, v_j) = n(I, v_j) \quad (4.10)$$

where N_D is the total number of documents in the dataset and N_{D,v_j} is the number of documents where the visterm v_j appears. In this equation we can see that $\mathrm{n}(I, v_j)$ is the BOV representation as defined in equation 4.4, to which we apply the weighting $idf(v_j)$. Figure 4.4 shows the original BOV representation of an image (left) and the tf-idf weighted version of the representation for the same image. We can see that the weighting affects the representation, however we expect that the changes can improve the representation by giving more weight to visterms that are more important.

4.3 BOV representation with fusion

Until now we have presented the BOV framework as a representation based on only a single feature. There are however cases where different information can be extracted from the image. In the BOV framework we can either define new features to extract the set of information we want from the image, or we can use a fusion framework to gather the different types of information independently and then combine them to build the image representation. The latter scheme allows us to extract local features with different local properties and then combine them depending of the problem at hand. In this section we investigate different BOV image representation fusion schemes. We will explore feature fusion at two different levels of the BOV representation. Note that, it is also possible to explore fusion at the decision level. It is however beyond the scope of our work.

The construction of the fusion based BOV feature vector h of an image I from two different features involves the steps illustrated in Figure 4.5. First, similarly to the single feature case, interest points are automatically detected in the image. However, in the next step we compute more than one local descriptors over those regions (two in our example). After the local descriptor extraction we must choose between two alternative ways of performing feature fusion. In the

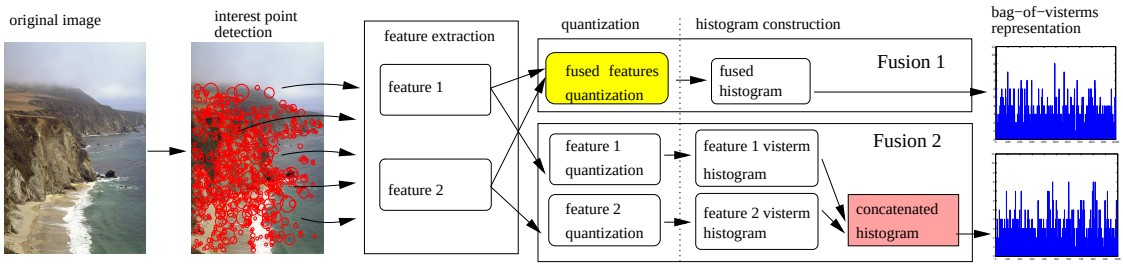


Figure 4.5. Schematic representation of BOV fusion system illustrating the two alternative fusion approaches: fusion between feature 1 and feature 2 is done either at the feature level before quantization (yellow box) or after quantization at the BOV level (pink box).

first case, the extracted features are concatenated in an appropriate fashion, and the resulting feature is then quantized to produce the BOV representation. In the second case, we quantize the features independently, create one BOV histogram for each, and then concatenate the two resulting histograms. Lets call fusion at the local descriptor level *Fusion 1* and at the BOV histogram level *Fusion 2*. In the following we describe in more detail each of these approaches.

Fusion 1

In this approach, local features are concatenated prior to clustering, resulting in a joint feature vocabulary. The fusion occurs by concatenating feature f_1 and feature f_2, after normalization, and weighted by a mixing value α according to:

$$f = \begin{bmatrix} \alpha f_1^\star \\ (1-\alpha) f_2^\star \end{bmatrix} \text{ with } f_1^\star = \beta_1 f_1 \text{ and } f_2^\star = \beta_2 f_2 \tag{4.11}$$

where β_1 and β_2 are normalization factors which are set to the inverse of the average Euclidean distance between a significant number of f_1 pairs and f_2 pairs respectively. As a consequence of this concatenation, denoting by dist_Q the Euclidean distance, in the K-means algorithm, the distance between two concatenated features f^a and f^b corresponds to a weighted distance:

$$\text{dist}_\text{Q}(f^a, f^b) = \alpha \, \text{dist}_\text{Q}(f_1^{\star,a}, f_1^{\star,b}) + (1-\alpha) \, \text{dist}_\text{Q}(f_1^{\star,a}, f_2^{\star,b}) \tag{4.12}$$

where the distance $\text{dist}_\text{Q}(f_1^{\star,a}, f_1^{\star,b})$ and $\text{dist}_\text{Q}(f_2^{\star,a}, f_2^{\star,b})$ are approximately of the same order of magnitude due to the normalization step. The fusion mixing value α is learned through cross-validation on training data. It is the mixing value α which will determine the relative importance of each feature f_1 and f_2 to create the fused feature f.

Fusion 2

In this second strategy, it is assumed that, at the interest point level, information gathered from feature 1 is independent from the information from feature 2. This strategy works by concatenating the BOV representation of feature 1 and feature 2, weighted by a factor α :

$$h = (\alpha h_1, (1 - \alpha) h_2) \tag{4.13}$$

The use of the mixing value is necessary to balance the relevance of the BOV representation before applying a classifier. Once again this mixing value must be found through cross-validation.

The choice between these two fusion approaches dependents on the task. Fusion 2 approach has the advantage of being simple since the feature normalization step in Fusion 1 is not necessary and also because we do not need to create several vocabularies in the search for the best mixing value. However, Fusion 1 allows us to model the joint feature co-occurrence, which may be interesting when there exists some correlations between the two feature types. One advantage of Fusion 2 is that we can re-utilize existing BOV representations since fusion is obtained by BOV concatenation. The fact that Fusion 2 is based on independent BOV representations also allows for the use of local features computed at different locations in the images. In Chapter 5 we will apply both fusion frameworks to the task of natural scene image classification by implementing fusion between texture and color local features.

4.3.1 Fusion of vocabularies from the same feature

Until now, we have proposed the fusion of different features into one BOV representation. However, as we described in Section 4.1 one of the important steps in the construction of the BOV representation is the vocabulary construction. However, it is possible to obtain better results by using more that one vocabulary of the same feature to represent our images. This fusion system is equivalent to the Fusion approach 2 presented in the previous section, but in the case where the extracted features are the same and what differs is the vocabulary construction. In short this approach involves the concatenation of different BOV representations. In the next chapter we will consider the fusion of vocabularies with different sizes and vocabularies extracted from features at certain scales.

4.4 Analogy with text

In this thesis we apply techniques to visterms that are similar to those commonly applied to text. The use of such techniques are sometimes motivated by the fact that a visterm vocabulary can be

consider similar to a text vocabulary. The analogy between visterms in images and words in text documents was originally proposed by Sivic and Zisserman (2003), and later on further explored by Willamowski *et al.* (2004). We find the analogy interesting and feel the need to explore to what extent does this analogy hold, if it holds at all. To that end we compare properties of the bag-of-words (BOW) representation in text documents with bag-of-visterms representation of images. We first discuss the *sparsity* of the document representation, an important characteristic of text documents. We then consider issues related to the semantic of terms, namely *synonymy* and *polysemy*.

4.4.1 Representation sparsity

To compare the BOV image representation sparsity to that of a BOW text document representation, we extracted a visual vocabulary of an image dataset and a text vocabulary of a text dataset. This allows us to compare the behavior between the BOV representation of an image dataset and the BOW representation of a standard text categorization dataset.

For our experiments we chose to use the REUTERS-21578[2] dataset as our text dataset and our city/landscape scene images (D_1^O) as our image dataset (see Appendix A for more details and sample images). The REUTERS-21578 database contains 9600 training and 3300 testing documents. The standard word stopping and stemming processes applied to the training documents produces a vocabulary of 17900 words. As previously observed in natural language statistics, the frequency of each word across the text database follows the Zipf's law:$P_r = r^{-b}$, where r is the word rank according to its frequency and b is close to unity (see Figure 4.6 (left)). This distribution results in an average number of 45 non-zero elements per document, which corresponds to an average sparseness of 0.25%. Out of the 17900 words in the dictionary, 35% occur once in the dataset and 14% occur twice. Only 33% of the words appear in more than five documents. Our dataset D_1^O contains 6680 city and landscape images. We extracted a 1000 dimensional vocabulary by applying the K-means algorithm to this database and generated the BOV representation for each image document of this database, see Section 5.6.1. Since the visterm vocabulary is created by the K-means clustering of SIFT descriptors extracted from a set of representative images, the resulting vocabulary exhibits different properties than in text. The K-means algorithm identifies regions in the feature space containing clusters of points, which prevents the low frequency effect observed in text data. The visterm with the lowest occurrence frequency still occurs in 117 images of the full dataset (0.017 relative frequency). In our experiments, given a vocabulary of 1000 visterms, we observed an average of 175 non-zero elements per image, which corresponds to a data sparseness of 17.5%. As shown in Figure 4.6 (right), the frequency distribution of visterms differs from the Zipf's law behavior usually observed in text.

The main reason for the sparsity difference in the visterm vocabulary when compared with the text vocabulary is derived from the way we construct the visual vocabulary. By using clustering to construct the visual vocabulary we intrinsically produce a "flatter" distribution for visterms

[2] www.daviddlewis.com/resources/testcollections/reuters21578

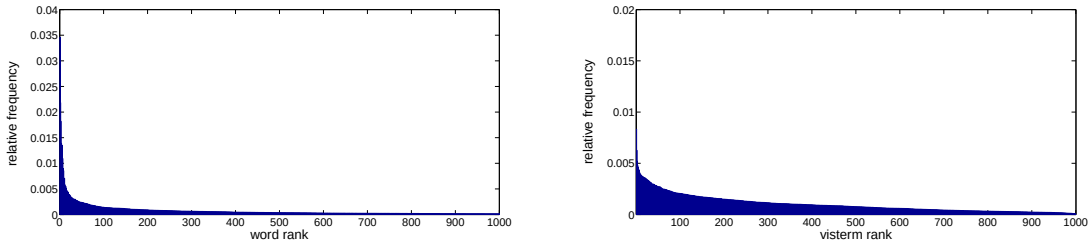


Figure 4.6. Word frequency distributions from text and image vocabularies. Left: Relative frequency distribution of the words extracted from REUTERS-21578, first 1000 words. Right: Relative frequency distribution of the visterms in the city/landscape dataset D_1^O for a 1000 dimensional vocabulary.

than for words. This is caused by the error minimization in the clustering that favors the creation of visterms in high density areas. This means that the clustering process focuses in representing the most occurring local descriptors regardless of their potential discriminative power.

On one hand, the sparsity difference between visterm and text vocabularies can be considered as an advantage, as the data sparseness observed in the text bag-of-words representation is indeed one of the main problems encountered in text retrieval and categorization. Documents belonging to the same topic may have very different bag-of-words representations because specific words that relate to that topic do not appear in both descriptions. On the other hand, a flatter distribution of the features might imply that, on average, visterms in the visual vocabulary provide less discriminant information. In other words, the semantic content captured by individual visterms is not as specific as the one of words. We address this issue in the next section.

4.4.2 Polysemy and synonymy with visterms

One known characteristic of text vocabularies is the existence of *polysemy* -a single word may represent different meanings- and *synonymy* -several words may characterize the same meaning. This is likely to happen as well in the case of visterms. In this section we will explore to which extent is our visual vocabulary affected by *synonymy* and *polysemy*. We will first look at the visterm occurrence rate per class and then we will try to find specific examples of *polysemy* and *synonymy* by direct inspection of visterms from our vocabulary.

To study the "semantic" nature of the visterms, we first considered the class conditional average of the BOV representation. Figure 4.7 (top) shows the average of visterms for the city and landscape scene categories, computed over the first split of dataset D_1^O (see Appendix A for details). We display the results when using a vocabulary of 100 visterms. Nevertheless, the behavior is similar for other vocabulary sizes. We first notice that there is a large majority of terms that appear in both classes: all the terms are substantially present in the city class; only a few of them do not appear in the landscape class. This contrasts with text documents, in which words are in general more specifically tied to a given category. Furthermore, we can observe that

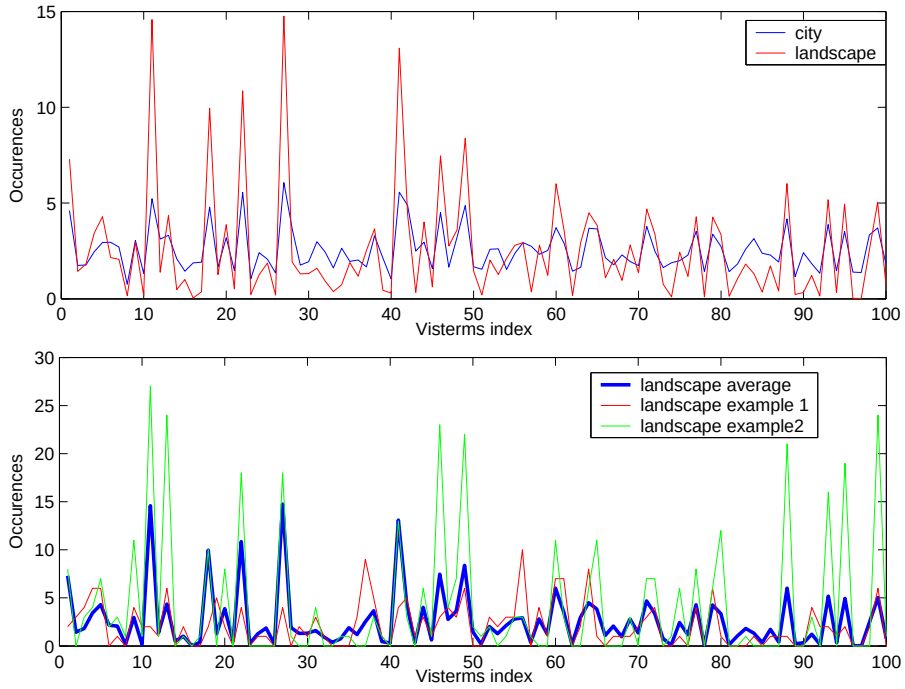


Figure 4.7. Visterm occurrence rate per class in the BOV representation. Top: average of the BOV representation with respect to city (blue) and landscape (red) computed over the first split of dataset D_1^O. Bottom: landscape average (blue) compared with individual samples (red and green).

the major peaks in the two class averages coincide. Thus, when using the BOV representation, one part of the discriminant information with respect to the classification task seems to lie in the difference of average word occurrences. It is worth noticing that this is not due to a bias in the average in visterm numbers, since the difference in the average amount of visterm per class is only in the order of 4% (city 268/ landscape 259, for dataset D_1^O). Additionally, these average curves hide the fact that there exists a large variability between samples, as illustrated in Figure 4.7 (bottom), where two random examples are plotted along with the average of the landscape class. Overall, all the above considerations indicate that visterms, taken in isolation, are not class specific, which in some sense advocates against feature selection based on analysis of the total occurrence of individual features (e.g. Dorko and Schmid (2003)), and reflects the fact that the semantic content carried by visterms, if any, is strongly related to polysemy and synonymy issues.

To illustrate that visterms are subject to *polysemy* and *synonymy*, we consider samples from three different visterms shown in Figure 4.8 obtained when building the vocabulary V_{1000} (see Subsection 5.6.1 for details). As can be seen, the top visterm (first two rows in Figure 4.8) represents mostly eyes. However, windows and publicity patches get also indexed by this visterm, which provides an indication of the polysemic nature of that visterm, which means here that although this visterm will mostly occur on faces, it can also occur in city environments. The second two rows in Figure 4.8 present samples from another visterm, this visterm also represents

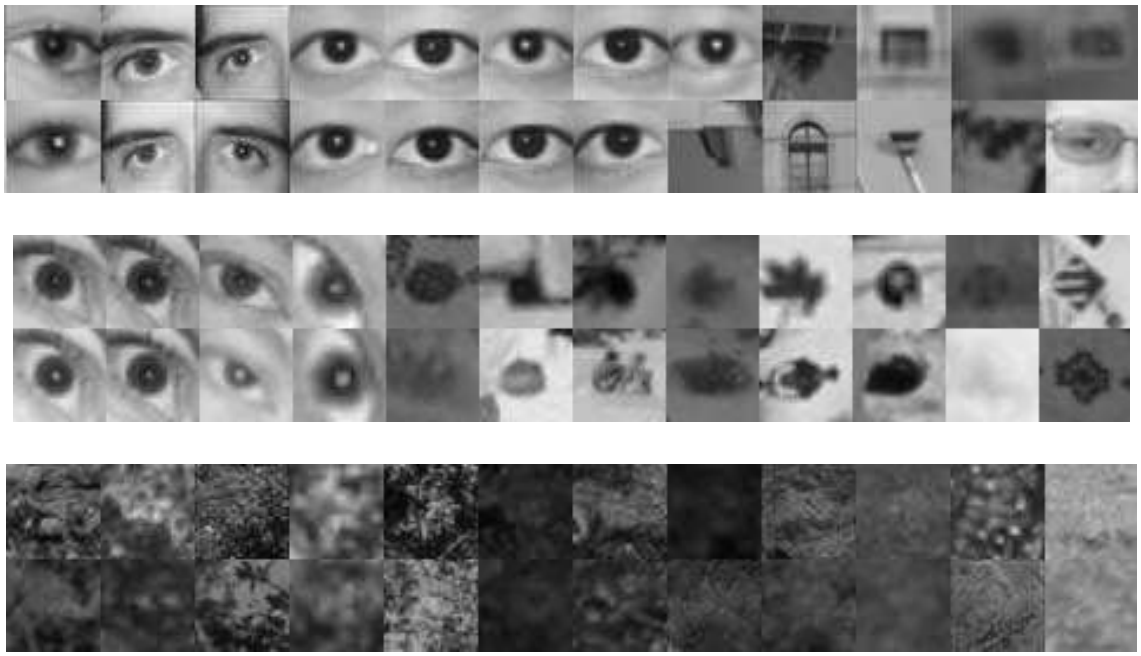


Figure 4.8. Image patch samples from randomly selected feature clusters corresponding to three visterms from a vocabulary of 1000 visterms. We can see the occurrence of both synonymy and polysemy in these images: both the top and middle image display a high relation to image patches of human eyes (synonymy) but contain also other image structures like windows or publicity (polysemy). The bottom samples are related to fine grain texture with different origins (rock, trees, road or wall texture...), which can appear in many contexts.

eyes, which makes it a synonym of the first displayed visterm. Finally, the samples of a third visterm (last two rows of Figure 4.8) indicate that this visterm captures a certain fine grain texture that has different origins (rock, trees, road or wall texture...), which illustrates that not all visterms have a clear semantic interpretation.

One factor that may affect the polysemy and synonymy issue is the vocabulary size: the polysemy of visterms may be more important when using a small vocabulary size than when using a large vocabulary. Conversely, with a large vocabulary, there are more chances to find many synonyms than with a small one. Latent aspect modeling has been introduced in the field of text document representation to deal with both synonymy and polysemy issues. A latent aspect representation could in principle lead to a more stable representation for different vocabulary sizes. We will explore this kind of representation in the next section.

4.5 Latent aspect representation

Latent aspect modeling is a methodological framework that enables us to extract and represent the contextual usage of words by statistical computations applied to a large corpus of documents. Latent aspect modeling used in text documents has been found to be capable of

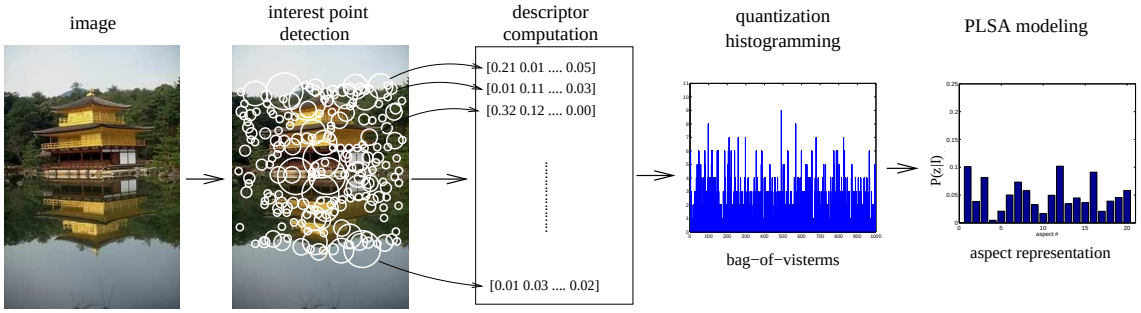


Figure 4.9. Computation steps of an image's PLSA based representation.

handling synonymy and polysemy issues by analyzing the co-occurrence of terms in documents.

As seen in the previous section, even if the BOV image representation statistics differ from those of the BOW representation of text, we found that synonymy and polysemy ambiguities still occur in the BOV image representation in a similar way. In the field of text document retrieval and categorization probabilistic latent aspect models (Hofmann (2001); Blei *et al.* (2003); Keller and Bengio (2004); Buntine (2002)) have been proposed to capture co-occurrence information between elements in a collection of discrete data in order to disambiguate the bag-of-words representation. In a similar way we use, in this thesis, the Probabilistic Latent Semantic Analysis (PLSA) model, introduced by Hofmann (2001), to analyze visterm co-occurrences in order to clarify synonymy and polysemy issues and produce a more stable representation. Though PLSA suffers from a non-fully generative formulation, its tractable likelihood maximization makes it an interesting alternative to fully generative models (Blei *et al.* (2003); Buntine (2002)) with comparative performance (Sivic *et al.* (2005)).

4.5.1 Probabilistic Latent Semantic Analysis (PLSA)

Probabilistic Latent Semantic Analysis (PLSA) was introduced by Hofmann (2001). PLSA is a statistical model that associates a latent variable $z_l \in \mathcal{Z} = \{z_1, \ldots, z_{N_A}\}$ with each observation (occurrence of a word in a document). These variables, usually called aspects, are then used to build a joint probability model over images and visterms, defined as the mixture:

$$P(v_j, I_i) = P(I_i)P(v_j \mid I_i) = P(I_i) \sum_{l=1}^{N_A} P(z_l \mid I_i)P(v_j \mid z_l). \tag{4.14}$$

PLSA introduces a conditional independence assumption, namely that the occurrence of a visterm v_j is independent of the image I_i it belongs to, given an aspect z_l (see Figure 4.11). The model in Equation 4.14 is defined by the probability of an image $P(I_i)$, the conditional probabilities $P(v_j|z_l)$ which represent the probability of observing the visterm v_j given the aspect z_l, and

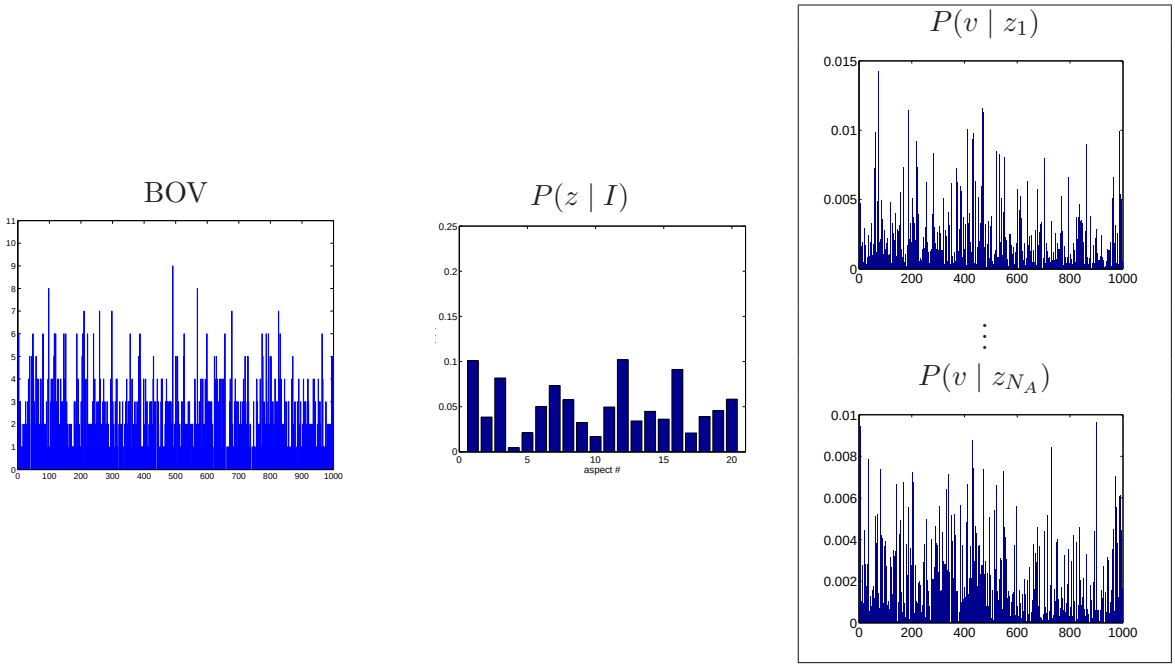


Figure 4.10. Illustration of PLSA's multinomial representation. For a given BOV of an image we obtain the corresponding $P(z \mid I)$ which is a multinomial decomposition of the BOV representation into a set of basis aspect functions characterized by a multinomial visterm distribution $P(v \mid z)$.

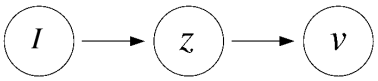


Figure 4.11. PLSA graphical model. This graph makes explicit the independence assumption between v and image I given an aspect z.

by the image-specific conditional multinomial probabilities $P(z_l|I_i)$. The aspect model expresses the conditional probabilities $P(v_j|I_i)$ as a convex combination of the aspect specific distributions $P(v_j|z_l)$. The parameters of the model are estimated using the maximum likelihood principle. More precisely, given a set of training images D_{test}, the likelihood of the model parameters Θ can be expressed by:

$$\mathcal{L}(\Theta|D_{test}) = \prod_{I \in D_{test}} \prod_{j=1}^{N_V} p(v_j, I)^{\mathrm{n}(I,v_j)} \tag{4.15}$$

where the probability model is given by equation. 4.14. The optimization is conducted using the Expectation-Maximization (EM) algorithm (Hofmann (2001)).

The parameters are estimated by the Expectation-Maximization (EM) procedure described in Hofmann (2001) which maximizes the likelihood of the observation pairs (v_j, I_i). The E-step

algorithm for the training of PLSA

1. random initialization of the $P(z \mid I)$ and $P(v \mid z)$ probability tables
2. E-step: calculate $P(z_k \mid I_i, v_j)$ according to Equation 4.16,
3. M-step: calculate $P(v_j \mid z_k)$ and $P(z_k \mid I_i)$ according to Equations 4.17 and 4.18 respectively
4. if stopping condition not yet achieved, go to step 2.

Figure 4.12. Description of the algorithm for scene segmentation using model 1.

estimates the probability of the aspect z_k given the element v_j in the image I_i (Equation 4.16).

$$P(z_k \mid I_i, v_j) = \frac{P(v_j \mid z_k)P(z_k \mid I_i)}{\sum_{k=1}^{N_z} P(v_j \mid z_k)P(z_k \mid I_i)} \tag{4.16}$$

The M-step then derives the conditional probability distributions $P(v_j \mid z_k)$ (Equation 4.17) and $P(z_k \mid I_i)$ (Equation 4.18) from the estimated conditional distribution of aspects $P(z_k \mid I_i, v_j)$ and the frequency count $n(I_i, v_j)$ of the element v_j in image I_i.

$$P(vj \mid z_k) = \frac{\sum_{i=1}^{N_I} n(I_i, v_j)P(z_k \mid I_i, v_j)}{\sum_{i=m}^{N_v}\sum_{i=1}^{N_I} n(I_i, v_m)P(z_k \mid I_i, v_m)} \tag{4.17}$$

$$P(z_k \mid I_i) = \frac{\sum_{j=1}^{N_v} n(I_i, v_j)P(z_k \mid I_i, v_j)}{n(I_i)} \tag{4.18}$$

To prevent over-fitting, the number of EM iterations is controlled by an early stopping criterion based on a validation data likelihood. Starting from a random initialization of the model parameters, the likelihood maximization is stopped when the criterion is reached. The corresponding latent aspect structure defined by the current conditional probability $P(v_j \mid z_k)$ is saved. Derived from the vector-space representation, the inference of $P(z_k \mid d_i)$ can be seen as a feature extraction process and used for classification. It can also be used to rank images with respect to a given latent aspect z_k, which illustrates the latent structure learned from the data. This

estimation procedure allows to learn the aspect distributions $P(v_j|z_l)$. These image independent parameters can then be used to infer the aspect mixture parameters $P(z_l|I)$ of any image I given its BOV representation $h(I)$. In Figure 4.10 we can see the original BOV representation of an image and the corresponding $P(z_l \mid I)$ we obtain. We also display some $P(v \mid z)$ which represent the co-occurrence captured by PLSA which allow for the multinomial decomposition.

PLSA features

Using PLSA as described before we obtain a model that gives us a possible decomposition of the information in the image into several aspects that are themselves a combination of visterms, as illustrated in Figure 4.10. This leads us to the last image representation that we will use in our work:

$$a(I) = \begin{bmatrix} a_1 \\ \vdots \\ a_{N_A} \end{bmatrix}, \text{ with } a_l = P(z_l|I). \tag{4.19}$$

This representation will be used as an input to a scene classifier in the next chapter and then used to perform ranking of images in an image retrieval setup in Chapter 6. We will further extend the latent aspect modeling in Chapter 6 by proposing to use the co-occurrence information captured by latent aspect models as a source of contextual information to improve image segmentation.

4.6 Chapter Conclusion

In this chapter we have introduced the fundamentals of the image representations explored in this thesis. We described the standard BOV approach and introduced both latent aspect modeling and fusion to create alternatives to the regular BOV approach. To investigate the analogy between visterms in images and words in text we presented a study of the characteristics of both the BOV and the BOW representations. In the next chapters we will apply these representations to several tasks using both scene and object images.

Chapter 5

Scene Classification

SCENE classification is an important task in computer vision. It is a difficult problem, interesting in its own right, but also as a means to provide contextual information to guide other processes such as object recognition (Torralba *et al.* (2003)). Scene classification is different from image retrieval and object categorization. In image retrieval several searches can retrieve the same image, while in scene classification we search for one only, most adequate, class label for each image. Scene classification also differs from object recognition in terms of image content. Images of a given object are usually characterized by the presence of a limited set of specific visual parts, tightly organized into different view-dependent geometrical configurations. On the other hand, scene images are generally composed of several entities (e.g. car, house, building, face, wall, door, tree, forest, rocks), organized in often unpredictable layout. The visual content (entities, layout) of a specific scene class exhibits a large variability, characterized by the presence of a large number of different visual descriptors. In view of this, while the specificity of an object strongly relies on the geometrical configuration of a relatively limited number of visual descriptors (Sivic and Zisserman (2003); Fergus *et al.* (2005b)), the specificity of a scene class greatly rests on the particular patterns of co-occurrence of a large number of visual descriptors. Since the image representations we are exploring in this thesis, as presented in the previous chapter, retain no information about the location of the local descriptors, scenes are a very interesting test set for those representations. We will explored object classification in Chapter 7. Scene representation is a problem that has been widely explored in the context of content-based image retrieval (Szummer and Picard (1998); Smeulders *et al.* (2000); Vailaya *et al.* (2001, 1998); Vogel and Schiele (2004b)), but existing approaches have traditionally been based on global features extracted from the whole image, on fixed spatial layouts, or using image segmentation methods whose results are often difficult to predict and control (Boutell *et al.* (2004); Smeulders *et al.* (2000); Vailaya *et al.* (2001, 1998); Serrano *et al.* (2002); Kumar and Herbert (2003b); Vogel and Schiele (2004b)).

In this chapter, we present scene classification results using the image representations introduced in the previous chapter. First, we provide the results obtained using our main representations:

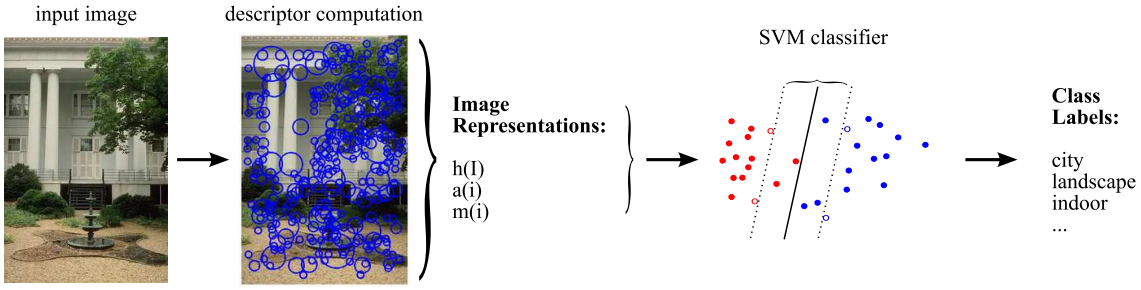


Figure 5.1. Illustration of our scene image classification approach, based on local interest point descriptors.

the K-means based BOV representation, and the latent aspect representation. These experiments are performed on our own dataset and datasets from Fei-Fei and Perona (2005) and Vogel and Schiele (2004b). The performance of the methods under different conditions including vocabulary size, number of latent aspects, and amount of training data is presented and discussed. Afterwords, we explore several variants of the BOV representation: the GMM based BOV representation and a color/texture fusion based BOV. In addition, since our choices for local detectors and descriptors from which to construct our representation were mainly based on the task of wide-baseline matching, we will also present the evaluation of our choice. Finally, we will also explore different ways to create modified versions of our vocabularies to improve the BOV representation. Before considering the results, we will first present the main setup and protocol considered in the experiments.

5.1 Task and approach

In this chapter we will explore the use of the image representations presented in the previous chapter for the task of scene classification. Figure 5.1 illustrates our approach to scene image classification. In short, we extract the local invariant interest points and descriptors, construct our image representation based on the extracted descriptors (cf. previous chapter), and obtain the resulting class label attribution using an SVM classifier. In the next subsections we will describe in more detail the datasets, experimental setup, classifiers, and baselines used in this chapter.

5.1.1 Datasets

In this chapter we use several datasets, for different tasks. In this subsection we present a concise description of the datasets considered for each task, for more details on all the presented datasets see Appendix A.

Due to the lack of a large scene image dataset, at the time of the start of the experiments presented here. We decided to create our own dataset of scene images D^O, this dataset is used

for experiments presented in Sections 5.3 and 5.4. This dataset consists of image from indoor, city and landscape images, from which we defined two binary classification tasks (city vs landscape using dataset D_1^O and indoor vs outdoor using dataset D_3^O) and one 3-class classification task (city/landscape/indoor using dataset D_3^O). Most image from dataset D^O were obtained from the COREL database, which has a large amount of images for most of the city and landscape scenes. Due to the lack of indoor images in the COREL database we collected indoor images from the Internet using the Google image search engine. The final size of dataset D^O is 9457 images. Selecting some image from the full dataset, we obtained a 5-class dataset D_4^O (mountain, forest, city street view , city panoramic view, and indoor), with a total of 6364 images.

For the training of the vocabulary and PLSA models we use two datasets. The first dataset is D_{3v}^O, this dataset consists of images from D^O and allows us to train models in images which are similar to those used to test our representation. The second dataset is D^A and consists of images from a mixture of several images from several sources and was designed to give us the possibility to learn our models in generic data (data not specifically selected for our classification tasks).

In Sections 5.5 and 5.8.2, we use the natural scene database D^V collected by Vogel and Schiele (2004b), which is constituted of images from 6 different types of natural scenes. This dataset contains a total of 700 images, distributed over the 6 natural scene classes: coasts, river/lakes, forests, plains, mountains, and sky/clouds. We chose this data because of its good resolution and color. Additionally, it is the only available public database we found which contained several natural classes. A drawback, however, is that the database has some non negligible overlap between classes (e.g. an image belonging to a given class could also easily belong to another class given its content). This overlap between categories was originally introduced as an important property of the database to evaluate human classification performance/confusion.

Also in Section 5.5, we perform scene classification on the 13-class dataset D^F introduced by Fei-Fei and Perona (2005). This dataset contains a total of 3859 images. The images are distributed over 13 scene classes: bedroom, coast, forest, highway, inside city, kitchen, living room, mountain, open country, office, street, suburb, and tall buildings.

5.2 Experimental setup

The protocol we followed for each of the classification experiments in this chapter was the following. The full dataset of a given experiment was divided into 10 parts, thus defining 10 different splits of the full dataset. One split corresponds to keeping one part of the data for testing, while using the other nine parts for training (hence the amount of training data is 90% of the full dataset). In this way, we obtain 10 different classification results.

The values we report for all experiments correspond to the average error over all splits, and standard deviations of the errors are provided in parentheses after the mean value.

5.2.1 SVM classifier

To perform classification we employed Support Vector Machines (SVMs) (Burges (1998); Vapnik (1995)). While there exist other alternatives we decided to only consider one classifier, since the focus of this work is on image representation. SVMs have proven to be successful in solving machine learning problems in computer vision and text categorization applications, especially those involving large dimensional input spaces. In the current work, we used Gaussian kernel SVMs, whose bandwidth was chosen based on a 5-fold cross-validation procedure, on the training set.

Standard SVMs are binary classifiers, which learn a decision function through margin optimization, such that function is large (and positive) for any input which belongs to the target class, and negative otherwise. For multi-class classification, we adopt a one-against-all approach (Weston and Watkins (1998)). Given a n-class problem, we train n SVMs, where each SVM learns to differentiate images of one class from images of all other classes. In the testing phase, each test image is assigned to the class of the SVM that delivers the highest output of its decision function.

We also perform experiments in a hierarchical way for our 3-class classification task (see Section 5.3.2). In that case, we use two binary SVM classifiers, applied to the decisions in the hierarchy. More generally using a hierarchical approach need $n-1$ SVMs for a n-class problem.

5.2.2 Baselines

In this chapter we perform several scene classification tasks, using different datasets. For each dataset/task we compare our results with the most adequate baseline.

city/landscape/indoor scene classification task

For the scene classification tasks of city/landscape and indoor/city/landscape we compare the performance of our image representation with the image representations proposed by Vailaya *et al.* (2001), using SVM classifiers in both cases. We selected this approach, as it reports some of the best results from all scene classification approaches for datasets with landscape, city and indoor images and since it has already been proven to work on a significant enough dataset. Thus, it can be regarded as a good representative of the state-of-the-art.

In the approach proposed by Vailaya *et al.* (2001) two different representations are used for each binary classification tasks: color features are used to classify images as indoor or outdoor, and edge features are used to classify outdoor images as city or landscape. Color features are based on the LUV first- and second-order moments computed over a 10×10 spatial grid of the image, resulting in a 600-dimensional feature space. Edge features are based on edge coherence his-

tograms calculated on the whole image. Edge coherence histograms are computed by extracting edges in only those neighborhoods exhibiting some edge direction coherence, eliminating in this way areas where edges are noisy. Directions are then discretized into 72 directions, and their histogram is computed. An extra non-edge pixels bin is added to the histogram, leading to a feature space of 73 dimensions. In the three-class problem this approach applies both methods in a hierarchical way. Images are first classified as indoor or outdoor given their color representation. All correctly classified outdoor images are further classified as either city or landscape, according to their edge direction histogram representation.

6 natural class scene classification task

We considered as first baseline the approach originally introduced at the same time as the database by Vogel and Schiele (2004a). In that work, the image was divided into a grid of 10×10 blocks, and on each block, a feature vector composed of a 84-bin HSI histogram, a 72-bin edge histogram, and a 24 features grey-level co-occurrence matrix was computed. These features were concatenated after normalization and weighting, and used to classify (with an SVM) each block into one of 9 local semantic classes (water, sand, foliage, grass,...). In a second stage, the 9 dimensional vector containing the image occurrence percentage of each regional concept was used as input to an SVM classifier to classify images into one of the 6 scene classes. The reported performance of that approach were good: 67,2. Note however that the approach of Vogel and Schiele (2004a) requires much more work than ours, as labeled data (image blocks with labels) to train the intermediate regional concept classifier are necessary. In that work, approximately 70000 blocks were manually labeled!

As an additional baseline we consider a more traditional color histogram approach, a concatenated Luv 96-bins linear histogram (32 bins for each dimension: L, u, and v).

13 class scene classification task

In the case of the 13-class dataset scene classification task we compare our results with those obtained by Fei-Fei and Perona (2005). In their system, Fei-Fei and Perona (2005) propose to model a scene category as a mixtures of aspects, and each aspect is defined by a multinomial distribution over the quantized local descriptors. The authors proposed the use of two variations of LDA Blei *et al.* (2003) to model the scene categories.

5.3 BOV classification results

We have seen in the previous chapter the methodology to obtain the BOV representation h of an image I. We now explore the performance of the BOV representation on the task of scene

Method	indoor/outdoor		city/landscape	
baseline	10.4	(0.8)	8.3	(1.5)
BOV V_{100}	8.5	(1.0)	5.5	(0.8)
BOV V_{300}	7.4	(0.8)	5.2	(1.1)
BOV V_{600}	7.6	(0.9)	5.0	(0.8)
BOV V_{1000}	7.6	(1.0)	5.3	(1.1)
BOV V'_{100}	8.1	(0.5)	5.5	(0.9)
BOV V'_{300}	7.6	(0.9)	5.1	(1.2)
BOV V'_{600}	7.3	(0.8)	5.1	(0.7)
BOV V'_{1000}	7.2	(1.0)	5.4	(0.9)

Table 5.1. Table with the classification error for the baseline model (Vailaya *et al.* (2001)) and the BOV representation for two binary classification tasks indoor/outdoor and city/landscape in database D^O. We cross-validate both the size and source of 8 different vocabularies. Means and standard deviations (in parentheses) of the resulting classification error are shown.

image classification. For the experiments presented in this section, several different vocabularies are used, these are discussed in Section 5.6.

5.3.1 Binary scene classification experiments

The D^O dataset contains two binary subdivisions: city/landscape (D_1^O) and indoor/outdoor (D_3^O). Table 5.1 provides the classification error for those two binary classification tasks. We can observe that the BOV approach consistently outperforms the baseline method (Vailaya *et al.* (2001)). This is confirmed in all cases by a Paired T-test, for $p = 0.05$. It is important to remind that contrarily to the baseline methods, the BOV representation uses the same features for both tasks and no color information. As mentioned before the fact that we can, with the same feature representation, tackle more problems illustrates the versatility of the BOV representation.

Regarding vocabulary size, overall we can see that for vocabularies of 300 visterms or more the classification errors are equivalent. This contrasts with the work in Willamowski *et al.* (2004), where the 'flattening' of the classification performance was observed only for vocabularies of 1000 visterms or more. A possible explanation may come from the difference in task (object classification) and from the use of the Harris-Affine point detector (Mikolajczy and Schmid (2004)), known to be less stable than DOG (Mikolajczyk and Schmid (2003)).

The comparison between rows 2-5 and 6-9 in Table 5.1 shows that using a vocabulary constructed from a dataset different than the one used for the classification experiments does not affect the results (error rates differences are within random fluctuation values). This result confirms the observations made in Willamowski *et al.* (2004), and suggests that it might be feasible to build a generic visterm vocabulary that can be used for different tasks. Based on these results, we use the vocabularies built from D^A in all the remaining experiments, unless stated otherwise,

Method	indoor/city/landscape	
baseline	15.9	(1.0)
BOV V'_{100}	12.3	(0.9)
BOV V'_{300}	11.6	(1.0)
BOV V'_{600}	11.5	(0.9)
BOV V'_{1000}	11.1	(0.8)
BOV V'_{1000} hierarchical	11.1	(1.1)

Table 5.2. Indoor/city/landscape three-class classification results for baseline and BOV models. The baseline model system is hierarchical, using color features for the indoor/outdoor classification and then edges to differentiate between city and landscape (cf Section 5.2.2).

since it simplifies the BOV computation process.

There is a considerable computational advantage in using a vocabulary built from an auxiliary dataset (D^A). Given a new dataset on which to perform a classification task, we can use the previously trained vocabulary and in this way avoid the time overhead from training the K-means vocabulary model in the new data.

5.3.2 Three-class classification

For further analysis of the BOV classification performance, the two binary classification tasks were merged to obtain a three-class problem *indoor* vs. *city* vs. *landscape*), see Section 5.1.1. Table 5.2 shows the classification results of the BOV approach for this three-class classification task. Classification results were obtained using either a multi-class SVM, or two binary SVMs, similarly to the baseline approach in the hierarchical case. First, we can see that once again the BOV representation outperforms the baseline approach with statistically significant differences. This is confirmed in all cases by a Paired T-test, with p=0.05. Secondly, we observe the stability of the results with vocabularies of 300 or more visterms, the vocabulary of 1000 visterms giving slightly better performance. Based on these results, we assume V'_{1000} to be an adequate choice and use V'_{1000} for all experiments in the rest of this chapter. Finally, we can observe that the classification strategy, hierarchical or multi-class SVM (one against the others), has little impact on the results for this task.

We can better analyze the classification results by looking at the confusion matrix shown in Table 5.3. We can see that landscape images are very well classified. However, there exists some confusion between the indoor and city classes. This can be explained by the fact that both classes share not only similar local image structures (which will be reflected in the same visterms appearing in both cases), but also similar visterm distributions, due to the resemblance between some more general patterns (e.g. doors or windows). The two images on the left in Figure 5.2 illustrate some typical errors made in this case, when city images contain a majority of geometric shapes and little texture. Some city images are also misclassified as landscape.

Total classification error				11.1 (0.8)	
Ground Truth	Resulting Classification (%)			Classification Error (%)	Total number of images
	indoor	city	landscape		
indoor	89.7	9.0	1.3	10.3	2777
city	14.5	74.8	10.7	25.2	2505
landscape	1.2	2.0	96.8	3.1	4175

Table 5.3. Confusion matrix for the city/landscape/indoor classification problem (using dataset D_3^O). The BOV representation built using the vocabulary V'_{1000}. Percentage of correctly classified and misclassified images is presented, along with the class dependent error-rates.

Total classification error rate				20.8 (2.1)		(Baseline:30.1 (1.1))	
	mountain	forest	indoor	panorama	street	class error (%)	number of images
mountain	85.8	8.6	2.5	0.5	2.6	14.2	590
forest	8.9	80.3	1.6	2.4	6.7	19.7	492
indoor	0.4	0	91.1	0.4	8.1	8.9	2777
city-panorama	3.5	1.8	8.0	46.9	39.8	53.1	549
city-street	2.0	2.2	20.8	6.0	68.9	31.1	1957

Table 5.4. Classification error rate and confusion matrix for the five-class classification problem, using the BOV approach with vocabulary V'_{1000}.

The main explanation for this is that city images often contain natural elements (vegetation like trees or flowers, or natural textures), and specific structures which produce many visterms. The images in Figure 5.2(right) illustrate typical mistakes in this case.

5.3.3 Five-class classification

As explained in Section 5.1.1, we created a five-class dataset D_4^O from our main dataset D^O. This dataset contains images from: mountain, forest, city street view city panoramic view, and indoor.

Table 5.4 presents the overall error rate and the confusion matrix obtained with the BOV approach in the five-class experiment, along with the baseline overall error rate. The latter number was obtained using the edge coherence histogram global feature from Vailaya *et al.* (2001). Note than in Vailaya *et al.* (2001) no five-class experiment was reported.

The BOV representation performs much better than the global features in this task, and the results show that we can apply the BOV approach to a larger number of scene classes and obtain good results. Note that a random class attribution would lead to an 80% error rate, and a majority class attribution (indoor in this case) to a 56% error rate. Analyzing the confusion matrix, we observe that some mistakes are made between the forest and mountain classes,

Figure 5.2. Typical error in the classification of city images in the three-class problem: city images classified as indoor(left), and city images classified as landscape (right).

reflecting their sharing of similar textures and the presence of forest in some mountain images. Another observation is that city-panorama images are often confused with city-street images. This result is not surprising because of the somewhat ambiguous definition of the classes (see Figure 5.3) which was already perceived during the human annotation process. The errors can be further explained by the scale-invariant nature of the interest point detector, which makes no distinction between some far-field street views in the city-panoramic images, and close-to middle-view similar structures in the city-street images. Another explanation is the unbalanced dataset, with almost four times as many city-street images than panoramic ones. Finally, we observe that the main source of confusion lays between the indoor images and the city-street images, for similar reasons as those described in the three-class task.

5.4 PLSA results

In this section we use the PLSA representation as presented in the previous chapter to perform scene image classification. This representation is defined by a N_A dimensional feature vector $a(I)$ representing the probability distribution $P(z_l|I_i)$ of latent aspects l given each specific document i (Equation 4.19). PLSA should provide us with a more stable representation which is less influenced by ambiguity problems like synonymy and polysemy. However, given that PLSA is an unsupervised approach, where no reference to the class label is used during the

Figure 5.3. Four example images from the city-panorama class from dataset D^O. We can clearly see the class ambiguity shown in these examples. This ambiguity causes confusion with the city-street class, as we can see in Table 5.4.

aspect model learning, we may wonder how much discriminant information remains in the aspect representation. To answer this question, we compare the classification errors obtained with the PLSA and BOV representations. Furthermore, to test the influence of the training data on the aspect model, we conducted two experiments which only differ in the data used to estimate the $P(v_j|z_l)$ multinomial probabilities defining the aspects. More precisely, we defined two cases:

PLSA-I for each dataset split, the training data part (which is used to train the SVM classifier, cf Section 5.2) was also used to learn the aspect models.

PLSA-O the aspect models are trained only once on the auxiliary dataset D^A.

As the dataset D^A comprises city, outdoor, and city-landscape overlap images, PLSA learned on this set should capture valid latent aspects for all the classification tasks simultaneously. Such a scheme presents the clear advantage of constructing a unique N_A- dimensional representation for each image that can be tested on all classification tasks. As with the cross-validation of the choices in the creation of the visterm vocabulary we perform the evaluation of the several choices for our PLSA representation in the context of the binary classification tasks: city/landscape and indoor/outdoor and in the context of the three-class classification task city/landscape/indoor.

Method	A	indoor/outdoor	city/landscape	indoor/city/landscape
BOV		7.6 (1.0)	5.3 (1.1)	11.1 (0.8)
PLSA-I	20	9.5 (1.0)	5.5 (0.9)	12.6 (0.8)
PLSA-I	60	8.3 (0.8)	4.7 (0.9)	11.2 (1.3)
PLSA-O	20	8.9 (1.4)	5.6 (0.9)	12.3 (1.2)
PLSA-O	60	7.8 (1.2)	4.9 (0.9)	11.9 (1.0)

Table 5.5. Comparison of BOV, PLSA-I and PLSA-O strategies on the indoor/outdoor, city/landscape and indoor/city/landscape scene classification tasks, using 20 and 60 aspects. All experiments were based on a visterm representation using vocabulary V'_{1000}.

PLSA-O					
N_A	20	40	60	80	100
Error	5.6 (0.9)	4.9 (0.8)	4.9 (0.9)	4.8 (1.0)	5.0 (0.9)

Table 5.6. Classification errors when performing city/landscape scene classification task, using different number of aspects for our PLSA-O representation.

5.4.1 Classification results : two and three-class cases

We performed the same classification experiments than those conducted with the BOV representation in the two binary and the three-class problems (previously explored by BOV). Table 5.5 shows the classification performance of the latent space representation for 20 and 60 aspects for the two strategies PLSA-I and PLSA-O, based on the V'_{1000} vocabulary. The corresponding results for BOV with the same vocabulary are re-displayed for comparison purposes.

Discussing first the PLSA training data issue, we observe that the performance of both strategies is comparable for the city/landscape scene classification, while PLSA-O is better than PLSA-I for the indoor/outdoor (paired T-test, with $p = 0.05$) task. However, in the indoor/city/landscape case PLSA-I performs better than PLSA-O, although not significantly. This might suggest that aspect models learned on the same set used for SVM training may cause some over-fitting in the indoor/outdoor case. Since using PLSA-O allows to learn one single model for all tasks, we chose this approach for the rest of the experiments. Of course, the dataset from which the aspects are learned should be sufficiently representative of the collection to be classified in order to obtain a valid aspect-based representation.

Comparing the 60-aspect PLSA-O model with the BOV approach, we remark that their performance is similar overall, with a slight advantage for PLSA in the city/landscape case (although not significantly), while the opposite holds for the three-class task. Learning visual co-occurrences with 60 aspects in PLSA allows for dimensionality reduction by a factor of 17 while keeping the discriminant information contained in the original BOV representation. Note that PLSA with 60 aspects performs better than the BOV representation with the vocabulary V_{100} in all cases (see Tables 5.5 and 5.2).

Total classification error				11.9(1.0)	
	indoor	city	landscape	class error(%)	total images
indoor	86.6	11.8	1.6	13.4	2777
city	14.8	75.4	9.8	24.5	2505
landscape	1.3	1.9	96.8	3.1	4175

Table 5.7. Classification error and confusion matrix for the three-class problem using PLSA-O. PLSA-O was trained on V'_{1000} using 60 aspects.

Total classification error rate			23.1 (1.1)	(BOV 20.8 (2.1),	Baseline: (30.1 (1.1)))	
	mountain	forest	indoor	city-panorama	city-street	class error (%)
mountain	85.5	12.2	0.8	0.3	1.2	14.5
forest	12.8	78.3	0.8	0.4	7.7	21.7
indoor	0.3	0.1	88.9	0.2	10.5	11.1
city-panorama	3.6	4.9	8.8	12.6	70.1	87.4
city-street	1.6	1.4	20.4	1.7	74.9	25.1

Table 5.8. Classification error and confusion matrix for the five-class classification problem using PLSA-O with 60 aspects.

We also conducted experiments to study the importance of the number of aspects on the classification performance. Table 5.6 displays the evolution of the error with the number of aspects for the city/landscape classification task. The results show that the performance is relatively independent of the number of aspects in the range [40,100]. For the rest of this chapter we will use a PLSA model with $N_A = 60$ aspects.

For comparison purposes, we present in Table 5.7 the confusion matrix in the three-class classification task. The errors are similar to those obtained with the BOV (Table 5.3). The only noticeable difference is that more indoor images were misclassified in the city class.

5.4.2 Classification results: five-class case

We can once again further analyze the performance of our representation by performing classification in the five-class task. Table 5.8 reports the overall error rate and the confusion matrix obtained with PLSA-O in the five-class problem. As can be seen, PLSA performs slightly worse than BOV, but still better than the baseline. By comparing the confusion matrix with that of the BOV case (Table 5.4), we can see that, while the forest, mountain and indoor classification behavior remains almost unchanged, the results for the two city classes were significantly altered. The main explanation comes from the rather loose definition of the city-panorama class, which contains many more images from landmark buildings in the middle distance than 'cityscape' images. Due to this fact, combined with the visterm scale invariance, the PLSA modeling generates a representation for the city-panorama images which clearly contains building-related aspects,

and introduces confusion with the city-street class. In this case, the abstraction or compression level of the PLSA modeling loses some of the discriminative elements of the BOV representation. Due to the unbalanced dataset, the city-street class benefits from this confusion, as shown by its reduced misclassification rate with respect to the city-panorama class. Furthermore, aspects are learned on the D^A dataset, which contains a relatively small amount of city-panorama images compared to city-street images. This imbalance can explain the ambiguous aspect representation of the city-panorama class and the resulting poor classification performance.

The five-class experiment raises a more general issue. As we introduce more classes or labels, the possibility of defining clear-cut scenes and of finding images that belong to only one class diminishes, while as a consequence the number of images whose content belongs to several concepts increases. This means that with more classes, the task could be better formulated as an annotation problem rather than a classification one. Using other image representations, PLSA-based approaches have shown promising performance for this task (Monay and Gatica-Perez (2004)).

5.4.3 Results with Reduced Amount of Labeled Training Data

Since PLSA captures co-occurrence information from the data it is learned from, it can provide a more stable image representation. We expect this to help in the case of lack of sufficient labeled training data for the classifier. Table 5.9 compares classification errors for the BOV and the PLSA representations for the different tasks when using less data to train the SVMs. In this case, for each of the splits, images were chosen randomly from the training part of the split to create a reduced training set. Care was taken to keep the same class proportions in the reduced set as in the original set, and to use the same reduced training set in those experiments involving two different representation models. The amount of training data is given both in proportion to the full data set size, and as the total number of training images. The test data of each split was left unchanged.

Several comments can be made from this table. A general one is that for all methods, the larger the training set, the better the results, showing the need for building large and representative datasets for training (and evaluation) purposes. Qualitatively, with the PLSA and BOV approaches, performance degrades smoothly initially, and degrades sharply when using 1% of training data. With the baseline approach, on the other hand, performance degrades more steadily. Comparing methods, we can first notice that PLSA with 10% of training data outperforms the baseline approach with full training set (i.e. 90%), this is confirmed in all cases by a Paired T-test with $p = 0.05$. BOV with 10% of training still outperforms the baseline approach with full training set (i.e. 90%) for Indoor/Outdoor (paired T-test with $p = 0.05$). In both the city/landscape and 3-class case, BOV still outperforms the baseline method, but not significantly. More generally, we observe that both PLSA and BOV perform not worse than the baseline for -almost- all cases of reduced training set. An exception is the city/landscape classification case, where the baseline is better than the BOV when using 2.5% and 1% training data, and better than the PLSA model for 1%. This can be explained by the fact that edge

Method	Amount of training data				
	90%	10%	5%	2.5%	1%
Indoor/Outdoor					
# of training images	8511	945	472	236	90
PLSA	7.8(1.2)	9.1(1.3)	10.0(1.2)	11.4(1.1)	13.9(1.0)
BOV	7.6(1.0)	9.7(1.4)	10.4(0.9)	12.2(1.0)	14.3(2.4)
Baseline	10.4(0.8)	15.9(0.4)	19.0(1.4)	23.0(1.9)	26.0(1.9)
City/Landscape					
# of training images	6012	668	334	167	67
PLSA	4.9(0.9)	5.8(0.9)	6.6(0.8)	8.1(0.9)	17.1(1.2)
BOV	5.3(1.1)	7.4(0.9)	8.6(1.0)	12.4(0.9)	30.8(1.1)
Baseline	8.3(1.5)	9.5(0.8)	10.0(1.1)	11.5(0.9)	13.9(1.3)
Indoor/City/Landscape					
# of training images	8511	945	472	236	90
PLSA	11.9(1.0)	14.6(1.1)	15.1(1.4)	16.7(1.8)	22.5(4.5)
BOV	11.1(0.8)	15.4(1.1)	16.6(1.3)	20.7(1.3)	31.7(3.4)
Baseline	15.9(1.0)	19.7(1.4)	24.1(1.4)	29.0(1.6)	33.9(2.1)

Table 5.9. Comparison of classification performance for PLSA-O with 60 aspects, BOV with vocabulary V'_{1000}, and baseline approaches, when using an SVM classifier trained with progressively less data. The amount of training data is first given in proportion of the full dataset, and then for each task, as the actual number of training images.

orientation features are particularly well adapted for this task, and that with only 25 city and 42 landscape images for training, global features are competitive.

Furthermore, we can notice from Table 5.9 that PLSA deteriorates less as the training set is reduced, producing better results than the BOV approach for all reduced training set experiments (although not always significantly better). Previous work on probabilistic latent space modeling has reported similar behavior for text data (Blei *et al.* (2003)). PLSA's better performance in this case is due to its ability to capture aspects that contain general information about visual co-occurrence. Thus, while the lack of data impairs the simple BOV representation in covering the manifold of documents belonging to a specific scene class (eg. due to the synonymy and polysemy issues) the PLSA-based representation is less affected.

Table 5.10 presents the evolution, in the five-class experiments, of the classification error when less labeled training data is available. It shows that the loss of discriminative power between the city-panorama and city-street classes continue to affect the PLSA representation, and that, in this task, the BOV approach outperforms the PLSA model for reduced training data. Both methods, however, perform better than the baseline based on global features.

Method	Amount of training data				
	90%	10%	5%	2.5%	1%
five-class					
# of training images	5727	636	318	159	64
PLSA	23.1(1.2)	27.9(2.2)	29.7(2.0)	33.1(2.5)	38.5(2.6)
BOV	20.8(2.1)	25.5(1.7)	28.3(1.3)	30.8(1.6)	37.2(3.4)
Baseline	30.1(1.1)	36.8 (1.4)	39.3 (1.4)	42.8 (1.6)	49.9 (3)

Table 5.10. Comparison between BOV, PLSA-O, and the baseline, when using an SVM classifier trained with progressively less data on the 5-class problem.

5.5 Results on other datasets

Throughout the time in which the research work presented in this thesis was performed, other researchers presented work in scene classification and at the same time introduced new datasets into the field (Fei-Fei and Perona (2005); Vogel and Schiele (2004b)). Given the appearance of those new datasets, we have also compared our framework when applied to those datasets. In Fei-Fei and Perona (2005), the authors tackle the classification of 13 different scene types, from multiple contexts (indoor/outdoors/city areas). In Vogel and Schiele (2004b), the authors tackle the classification of 6 different natural scenes types. For details on the constitution of these datasets and example images see Section 5.1.1 and Appendix A. These two datasets are challenging given their respective number of classes and the intrinsic ambiguities that arise from their definition. In this 13-class dataset for example, images from the inside city and street categories share a very similar scene configuration. Similarly, the differences between bedroom and living room examples can be very subtle. In the 6-class dataset, some examples of the coasts and waterscapes classes are hard to distinguish, even for a human. Note that the same trend of an ambiguous class definition was observed for our five-class classification task. We will now explore the 13-class dataset presented by Fei-Fei and Perona (2005) and we will later explore the dataset presented by Vogel and Schiele (2004b) in a color/texture fusion framework in Section 5.8.2.

In this chapter, we evaluated different visterm vocabularies built from different data sources, and conducted a comparison of aspect representations learned from extra data (PLSA-O) or learned on the same data used to learn the SVM classifier (PLSA-I). Given that we have no extra set of representative images for the 13-class classification task, we can not present the same range of experiments for these datasets. In the following, we only consider the vocabulary V'_{1000} learned from D^A, and one of the PLSA-based aspect representation.

As a first approach, we classify the images based on their BOV representation. We extracted features using DOG+SIFT and constructed the BOV. Classification results were obtained by training a multi-class SVM using a 10-split protocol as before. Please note that no parameter tuning on the vocabulary was done in this case, as we directly apply the vocabulary V'_{1000} used in Section (5.3), (e.g. the vocabulary size remained the same as before).

Class	confusion matrix													perf.
bedroom	30.6	7.9	0.0	1.9	3.7	6.9	21.3	3.7	1.4	11.6	2.8	1.4	6.9	30.6
coast	0.3	78.3	1.4	5.8	0.0	0.0	0.0	4.7	6.4	0.0	1.4	0.6	1.1	78.3
forest	0.3	0.0	89.0	0.0	0.0	0.0	0.0	4.3	5.2	0.0	0.0	0.6	0.6	89.0
highway	0.0	14.2	0.0	67.7	2.3	1.5	0.8	1.9	4.2	0.8	3.1	1.5	1.9	67.7
in. city	1.0	1.0	1.0	2.3	64.6	4.5	2.6	0.0	0.3	2.3	6.8	1.3	12.3	64.6
kitchen	6.2	1.0	0.0	1.4	16.7	40.0	11.4	0.0	0.5	14.8	2.9	1.4	3.8	40.0
liv. room	10.7	0.7	0.0	1.4	5.9	6.6	45.7	0.7	0.3	9.7	5.9	5.2	7.3	45.7
mountain	0.3	2.4	3.7	1.3	0.0	0.0	0.0	82.1	7.5	0.0	1.1	0.8	0.8	82.1
o. country	0.7	12.2	9.0	2.4	0.5	0.0	0.0	11.0	60.5	0.0	1.5	1.2	1.0	60.5
office	1.4	1.4	0.0	1.9	1.9	6.5	6.5	0.0	0.0	77.2	0.5	0.9	1.9	77.2
street	0.7	0.7	0.3	3.8	8.9	1.0	1.0	1.7	0.3	0.3	72.3	0.7	8.2	72.3
suburb	0.0	0.4	0.4	0.0	0.8	0.8	2.1	0.4	2.1	0.8	0.8	89.2	2.1	89.2
t. buildings	1.4	4.2	1.7	2.5	7.9	2.2	1.4	1.4	2.0	0.8	6.5	1.1	66.9	66.9
overall														66.5

Table 5.11. Classification results for the BOV representation, in the 13-class problem presented in Fei-Fei and Perona (2005). Overall performance is obtained by averaging over all classes.

The confusion matrix for the 13 classes and the classification performance per class are presented in Table 5.11. The classification performance is substantially higher than the one presented by Fei-Fei and Perona (2005), which reported an overall classification performance of 52.5% when using the same combination of detector/descriptors we adopted here (DOG+SIFT) for learning their LDA model. The performance of our method is also slightly higher than the *best* performance reported in Fei-Fei and Perona (2005), 65.2%, which was obtained with a different detector/descriptor combination: GRID/SIFT. Given that we do not have access to the individual per-image results of Fei-Fei and Perona (2005), we cannot assess the statistical significance of these results, but we can nevertheless consider that the BOV approach is competitive with Fei-Fei and Perona (2005).

We also applied the PLSA-I approach to solve the same classification problem, as described in Section 4.5.1. We learned the PLSA model with 40 aspects, since this is the number of aspects used by Fei-Fei and Perona (2005) to produce results. Classification results were obtained, as before, by using a multi-class SVM classifier and a 10-split protocol.

Table 5.12 shows the performance of the PLSA-I aspect representation. The classification accuracy is higher than the one obtained in Fei-Fei and Perona (2005) when using the (DOG+SIFT) combination, but is lower than the *best* performance reported in Fei-Fei and Perona (2005), and also lower that the result we obtained with the BOV representation. The performance degradation between the BOV and the PLSA representations results from the same phenomena observed for the five-class experiments in Section 4.5.1. In the presence of a high number of classes, the PLSA decomposition tends to result in a loss of important details for the distinction of ambiguous classes. As with the BOV case, we can also say that the PLSA approach remains competitive with respect to Fei-Fei and Perona (2005).

Class	confusion matrix													perf.
bedroom	31.9	1.9	0.0	0.5	5.1	8.8	23.1	0.9	0.5	13.9	4.2	4.2	5.1	31.9
coast	0.6	65.3	1.4	9.7	0.0	0.0	0.3	6.4	13.3	0.3	0.6	0.8	1.4	65.3
forest	0.0	0.6	86.3	0.0	0.0	0.0	0.0	8.5	4.0	0.0	0.3	0.3	0.0	86.3
highway	1.2	12.3	0.0	58.8	2.7	0.8	0.0	1.9	4.2	0.8	6.2	2.3	8.8	58.8
in. city	1.6	0.0	0.3	2.3	63.6	3.9	2.3	0.0	0.0	3.2	9.4	2.6	10.7	63.6
kitchen	7.1	0.0	0.0	1.0	21.0	15.7	18.1	0.0	0.0	31.0	1.9	0.5	3.8	15.7
l. room	13.5	0.3	0.0	2.4	2.4	4.2	45.0	0.0	0.3	14.2	4.5	4.8	8.3	45.0
mountain	0.3	4.5	7.8	0.5	0.0	0.3	0.0	73.8	10.7	0.0	0.8	1.1	0.3	73.8
o. country	0.7	9.0	7.6	1.2	0.2	0.0	0.0	13.9	64.4	0.0	1.5	1.2	0.2	64.4
office	4.7	0.5	0.0	0.9	2.8	7.9	13.0	0.0	0.5	67.0	0.5	0.9	1.4	67.0
street	0.3	0.3	0.0	4.5	9.9	0.7	2.1	0.3	0.3	0.0	68.2	1.7	11.6	68.2
suburb	1.7	0.4	0.0	1.7	0.8	0.8	3.3	0.4	1.2	0.0	0.4	88.4	0.8	88.4
t. buildings	2.0	3.1	1.1	3.4	9.6	1.4	4.5	1.4	1.4	1.1	5.6	3.4	62.1	62.1
overall														60.8

Table 5.12. Classification results for the PLSA-I representation, in the 13-class problem presented in Fei-Fei and Perona (2005). Overall performance is obtained by averaging over all classes.

Class	confusion matrix						performance per class
coasts	59.9	9.9	2.1	8.5	18.3	1.4	59.8
river/lakes	1.6	24.3	10.8	10.8	27.0	5.4	24.3
forests	2.9	5.8	81.6	4.9	4.9	0.0	81.6
plains	18.3	6.1	8.4	52.7	11.5	3.1	52.7
mountains	11.2	8.9	2.2	2.8	73.7	1.1	73.7
sky/clouds	5.9	2.9	0.0	5.9	5.9	79.4	79.4
overall							61.9

Table 5.13. Classification results for the BOV representation, in the 6-class problem presented in Vogel and Schiele (2004b). Overall performance is obtained by averaging over all classes.

5.5.1 Classification results: 6-class

The dataset presented by Vogel and Schiele (2004b) is composed of less classes than Fei-Fei and Perona (2005), with a total of six natural scene types. The ambiguity between class definitions is however more important, and some images are difficult to classify in only one scene type. The number of examples per class is significantly smaller than the number of examples per class in Fei-Fei and Perona (2005) and than our five-class dataset.

The six natural scene types are on several images quite confusing, and are even challenging for human labeling. To perform classification, we first apply the BOV using the V_{1000} vocabulary which was learned on D^A, like presented in Section 5.6.1. We did no extra tuning of the vocabulary for this task. Classification results were obtained by using a multi-class SVM trained

Class	confusion matrix						performance per class
coasts	40.1	9.9	9.2	12.0	25.4	3.5	40.1
river/lakes	20.7	21.6	11.7	12.6	30.6	2.7	21.6
forests	1.9	3.9	78.6	7.8	7.8	0.0	78.6
plains	20.6	6.9	11.5	35.9	21.4	3.8	35.9
mountains	8.4	7.3	11.7	5.6	65.9	1.1	65.9
sky/clouds	14.7	0.0	0.0	8.8	5.9	70.6	70.6
overall							52.1

Table 5.14. Classification results for the PLSA-O representation, in the 6-class problem presented in Vogel and Schiele (2004b). Overall performance is obtained by averaging over all classes.

using a 10-split training protocol, as described in Section 5.2.

The classification results using the BOV representations (V_{1000} vocabulary learned on D^A) are presented as a confusion matrix and its corresponding per class classification accuracy in Table 5.13. In this case, our system has a slightly reduced classification accuracy (61.9%) when compared with the performance presented in Vogel and Schiele (2004b)(67.2%). Note, however, that these results have not been obtained using identical features: Vogel and Schiele (2004b) relies on a fixed grid, where a texture and color features are extracted. We believe that the difference in performance with respect to our work arises from the fact that natural scene discrimination can benefit greatly from the use of color, something we have not made use of, but which in light of these results constitutes an issue to investigate in future work. Moreover, the intermediate classification step proposed in Vogel and Schiele (2004b) requires the manual labeling of hundreds of regional descriptors, which is not needed in our case.

Given the reduced set of examples per class, and the need for a large number of representative examples to train a PLSA model, we could not perform the PLSA-I approach for this 6-class problem. However, in order to evaluate the performance of the aspect representation for these data, we derived the aspect representation for these images based on the previous PLSA model with 60 aspects learned on the D^A dataset (see Section 4.5.1). The corresponding classification results, as shown in Table 5.14, indicate a decrease in performance (52.1%) with respect to both the BOV classification and the results reported in Vogel and Schiele (2004b). The fact that the PLSA model has been learned on the D^A dataset, which does not contain any *coasts*, *river/lakes*, or *plain* examples likely explains the poor discrimination between the 6-classes when the corresponding aspect representation is used.

Also, the uncertainty introduced by ambiguous classes co-occurring in the same image results in a PLSA model that does not provide a discriminant representation for the various classes. Another factor to have into account is that PLSA-O was trained on data that is more general, and not very significant of this particular set of classes (sky/clouds, river/lakes). Although PLSA-O has proven to have a good performance in a variety of case, that performance degrades when the nature of the testing data diverges from the one used for learning.

Overall, these experiments support some of the findings obtained in Section 5.3, namely that modeling scenes as a *bag-of-visterms* performs well even in problems with a large amount of classes, and that PLSA modeling can find limitations in cases of large amount of overlapping classes. At the same time, these experiments offer other insights. Our framework is competitive with recent approaches, and that feature fusion mechanisms (adding color) have a potential for an increased classification performance.

5.6 Vocabulary issues

As we have seen in the previous chapter there are several steps in the creation of the BOV image representation. Each of those steps requires some design choices. These choices are all cross-validated, however it is impossible to search all possibilities for all combined parameters. In this section we explore some choices in the creation of our visterm vocabulary.

First, we comment on the performance of the different size vocabularies used until now. We also comment on the results obtained with vocabularies from different datasets, regarding the influence that the specificity of the vocabulary training data can have in the final classification performance. Secondly, we test different local interest points and descriptors to evaluated their influence in the final system's performance. Finally, we explore the influence of the amount of data used to learn the vocabulary on the final classification performance.

5.6.1 Scene vocabulary construction

The vocabulary on which we base our BOV representation is of great importance. The two main choices to make when creating a vocabulary are: how large to should the vocabulary be, and which data to collect the features needed to build the vocabulary . To address the first point, we considered four vocabularies of 100, 300, 600, and 1000 visterms, denoted by V_{100}, V_{300}, V_{600}, and V_{1000}, respectively, and constructed these using the training set images contained in D^O_{3v} through K-means clustering as described in Chapter 4. Dataset D^O_{3v} is a subset of our training dataset for this problem. By comparing the classification performance of our system while using different size vocabularies we can investigate the best vocabulary size for our problem. Additionally, we constructed four more vocabularies V'_{100}, V'_{300}, V'_{600}, and V'_{1000} from an external image database D^A . Database D^A contains data from various sources all different from those of the test set (and also different from the classifier training set). It is a mix of several different datasets and images from the Internet (see Appendix A for details). Comparing the classification performance of our system using the vocabularies obtained from D^O_{3v} and D^A allows us to analyze indirectly the dependence of the BOV representation on the source of the vocabulary training data.

With the experiments presented in Section 5.3 we have shown that the classification results are good and remain stable for any of the vocabularies tested with 300 or more visterms. A small

Method	SIFT	CF	PATCH	Average number of points per image
DOG	11.1 (0.8)	22.5 (1.1)	22.1 (0.9)	271
MHA	11.9 (1.1)	18.4 (1.1)	20.6 (1.3)	424
MH	11.8 (1.0)	19.3 (0.9)	-	580
GRID	19.9 (0.9)	-	19.8 (0.8)	300

Table 5.15. Comparison of different combinations of detector/descriptors for the task of indoor/city/landscape classification. Results are presented for several detector/feature combinations. The average number of point per image detected by each point detector is also given.

gain is obtained by using the vocabulary of 1000 visterms. We also found, based on the same experiments, that using vocabularies learned from D^A result in a similar performance, which motivated the usage of general vocabularies.

5.6.2 Comparison of different detectors/descriptors

As we explained in the previous chapter our choice of local interest point detector and descriptors was based on the good performance of these detectors/descriptors applied to the wide-baseline matching task (Mikolajczyk and Schmid (2005); Mikolajczy and Schmid (2004)). To support that choice in the context of scene image classification we now take a small step back, and explore different combinations of point detectors/descriptors for the specific task of scene image classification. We choose to perform this study on the city/indoor/landscape problem since we believe that a multi-class classification problem is more representative for this data, but at the same time it is not obscured by many of the additional issues of a many-class problem. Four point detection methods: DOG (Lowe (2004)), multi-scale Harris affine (MHA) (Mikolajczy and Schmid (2004)), multi-scale Harris-Laplace (MH) (Mikolajczy and Schmid (2004)), and a fixed 15x20 grid (GRID), and three descriptor methods: SIFT (Lowe (2004)), complex filters (CF) (Schaffalitzky and Zisserman (2002)), and a 11×11 pixel patch sample of the area defined by the detector (PATCH) were used in paired combinations. The results for different detectors/descriptors combinations are shown in Table 5.15.

In Table 5.15, we can see that the combination DOG+SIFT is the best performing one. This result is confirmed by a Paired T-test, with p=0.05, when compared with all other results. However, MHA+SIFT and MH+SIFT produce similar results. This confirms that SIFT is a powerful local image representation, as already known in the field of local descriptors (Fei-Fei and Perona (2005); Mikolajczyk and Schmid (2003)). As for detectors, it is important to note that, although the multi-scale Harris and multi-scale Harris affine detectors Mikolajczy and Schmid (2004) allow for similar performance, DOG is computationally more efficient and more compact (less feature points per image). Although Table 5.15 shows DOG+SIFT to be the best choice for this particular task, it is possible that other combinations may perform better for other tasks/data. Based on these results, however, we have confirmed in practice that DOG+SIFT constitutes a reasonable choice.

5.6.3 Training the Vocabulary with less samples

To build our vocabulary, we have until now used what is considered sufficient amount of training data. For a maximum of 1000 clusters we have used approximately 1 million data points. Taking into account the dimensionality of our features (128 dimensional), the resulting vocabulary should be appropriated for the task of representing scene images. However, we would like to know if there is really a necessity for the use of such a large amount of data point, or if a smaller amount would suffice.

To test the dependence of our system's performance on the size of the dataset used to learn the vocabulary we perform some experiments in which we reduce the number of images from which we extract the features to cluster. The results are presented in Table 5.16.

dataset size	100%	50%	30%	10%
	3805	1902	1141	380
performance	5.3 (1.1)	5.2 (1.0)	5.4 (9.8)	5.4 (1.2)

Table 5.16. Table with the classification error for the task of city/landscape scene classification using D_1^O dataset, using vocabularies learned with variable amount of data. Note that the vocabulary constructed with 100% of the dataset is $V_1'000$

As we can see, there is no significant change of the performance results with the reduction of the amount of training examples. There is only a small trend that can indicate a small increase in error. We can conclude that we can obtain a valid vocabulary even with a small amount of training data.

5.7 Related image representations

In this section we present representation which are related to the BOV representation. We first present a GMM based representation in which we use a GMM to model our data and extract our bag-of-words like representation. Afterwords, we explore the use of tf-idf visterm weighting for our image representation in the task of scene classification.

5.7.1 GMM based BOV representation

As we seen Chapter 4, GMMs can be used to create a BOV like histogram representation which is smoother than the plain BOV representation, and should account for local descriptors falling near the feature boundaries separating two visterms. This should be in principle a more stable representation w.r.t. noise in the calculations of the local descriptors. We now evaluate this image representation on the scene image classification task, on both binary and the 3-class problems defined in Section 5.3.

problem	city/landscape	indoor/outdoor	city/landscape/indoor
BOV	5.3 (1.1)	7.6 (1.0)	11.1 (0.8)
GMM	5.5 (1.1)	6.6 (0.8)	10.1 (0.8)

Table 5.17. Comparison of classification performance for several scene classification tasks between the BOV representation and the GMM based representation.

problem	city/landscape	indoor/outdoor	city/landscape/indoor
standard BOV	5.3 (1.1)	7.6 (1.0)	11.1 (0.8)
tf-idf weighted BOV	5.1 (1.2)	7.8 (1.1)	10.8 (0.9)

Table 5.18. Comparison of classification performance for several scene classification tasks between the BOV representation and the tf-idf weighted representation.

As we can see in Table 5.17, using the GMM based representation we obtain an increase in the classification performance when compared with those obtained using the K-means based BOV, for the tasks of indoor vs. outdoor and 3-class scene classification. In the case of the city vs. landscape task the performance is similar. The obtained increase in performance is small but significant, and indicates that as the classification task becomes more complex, the GMM based representation becomes more advantageous. This results motivates the possible use of GMM based representations for future work.

5.7.2 Results using tf-idf visterm weighting

As already mentioned in Section 4.2.2, one of the most often used weighting strategies in information retrieval is tf-idf (term frequency - inverse document frequency). The motivation and formulation of this weighting strategy is presented and explained in the previous chapter, see Section 4.2.2. We now perform experiments with tf-idf weighted BOV representations for binary and 3-class scene classification tasks, as defined in Section 5.3. Table 5.18 shows the resulting performance, compared with the standard BOV approach, As we can see there is little difference in performance between standard BOV and the tf-idf weighted BOV. Given these results there is not motivation to use this approach, however there may be other problems where this approach may be advantageous.

5.8 Fusion schemes

In this section we explore fusion approaches aimed at improving our vocabulary towards a better image representation.

multi-level vocabularies	$V_1'000$	V_{ML_1}	V_{ML_2}
classification performance	66.5	70.2	71.8

Table 5.19. Results using multi-level vocabularies for the 13-class scene classification problem.

5.8.1 Multi-level Vocabularies

The BOV representation is dependent on the vocabulary of visterms on which it is based. Phenomena like many synonymy and polysemy may reduce the performance of the resulting representation. Until now we have discuss only vocabularies where all local descriptors are clustered independently of the scale at which the local interest point was detected in the image. We want to cluster descriptors from different scales together since we want our vocabulary to be scale invariant, however there is the risk of clustering different local structures together because of that same scale invariance. In this subsection we explore ways of combining information of different vocabularies to try to obtain a higher performance from a K-means based vocabulary (see Section 4.3.1 for details).

From previous experiments we found that, from all sizes for our vocabularies we tested, the best vocabulary size for our experiments was of dimension 1000. However we have not yet explored the power of more than one vocabulary at the same time. This is possible if we use more than one of the previously tested vocabularies. This can possibly allow us to use at the same time specific visterms from the vocabularies with a large range of visterms and more general visterms from smaller vocabularies. We will call this type of vocabulary a multi-level vocabulary.

To construct a multi-level vocabulary we create several individual vocabularies of different sizes and use each vocabulary to extract several sets of visterms from each image. This will create several *bag-of-visterms* for the same image, we then concatenate all the obtained *bag-of-visterms* into one single BOV representation. This creates a histogram that contains all the information of the different learned vocabularies.

Table 5.19 shows the results of the multi-level vocabulary representations for the 13-class scene classification problem. In this case we consider two multilevel vocabularies V_{ML_1} and V_{ML_2}, where V_{ML_1} results from the concatenation of the BOV representation obtained with vocabularies V'_{100}, V'_{300} and V'_{600}, and V_{ML_2} results from the concatenation of the BOV representation obtained with vocabularies V'_{100}, V'_{300}, V'_{600}, and V'_{1000}. As we can see the use of a multi-level vocabulary can improve the classification results, even when the final dimension is the same as the previously best vocabulary. The different, although small, is statistically significant for a paired t-test with p=0.05.

5.8.2 Color and texture BOV fusion for scene classification

We have shown in this chapter that the BOV framework is a valid approach to scene image classification. However, the BOV representation of scenes based on SIFT features makes no use of color information. Although this may be acceptable for some classes it is obvious that for some natural scenes, color is an important feature and it can provide us with a more complete visterm representation. We know that in the field of scene image modeling, most works use color and texture information to perform classification and retrieval (Szummer and Picard (1998); Vailaya *et al.* (2001); Serrano *et al.* (2002); Vogel and Schiele (2004a); Bosch *et al.* (2006)).

Vailaya *et al.* (2001) used histograms of different low-level cues to perform scene classification. Different sets of cues were used depending on the two-class problem at hand: global edge features were used for city vs landscape classification, while local color features were used in the indoor vs outdoor case. More generally scene recognition methods tend to fuse color and texture information. Both Serrano *et al.* (2002) and Szummer and Picard (1998) propose a two-stage classification of indoor/outdoor scenes, where color and texture features of individual image blocks are computed over a spatial grid layout are first independently classified into indoor or outdoor. The local classification outputs are then further combined to create the global scene representation used in the final image classification stage. Vogel and Schiele (2004a) propose a similar two-stage approach based on a fixed grid layout to perform scene retrieval and classification. Several local features (color and edge histograms, grey-level co-occurrence matrix) are concatenated after normalization and weighting into one feature vector. Finally, Boutell *et al.* (2004) use only Luv color moments in a 7x7 block layout to perform scene multi-label scene classification. More recently, Bosch *et al.* (2006) have extended the SIFT feature to capture color from RGB images by applying the SIFT feature extractor to all three channels and then combine the resulting descriptors into one descriptor of dimension 384.

To address the lack of color information we apply the BOV representation using fusion framework presented in the previous chapter. Using this approach we can combine color and texture local descriptors to improve our BOV representation for natural scenes, see Figure 5.4. We investigate the different ways to fuse together local information from texture and color in order to provide a better visterm representation. We develop and test our methods on the task of image classification using a 6-class natural D^V scene database introduced by Vogel and Schiele (2004a), see Appendix A for details. In the remainder of this section we show classification results using the *bag-of-visterms* (BOV) representation (histogram of quantized local descriptors), extracted from both texture and color features. We investigate two different fusion approaches at the feature level: fusing local descriptors together and creating one representation of joint texture-color visterms, or concatenating the histogram representation of both color and texture, obtained independently from each local feature. On our classification task we show that the appropriate use of color improves the results w.r.t. a texture only representation.

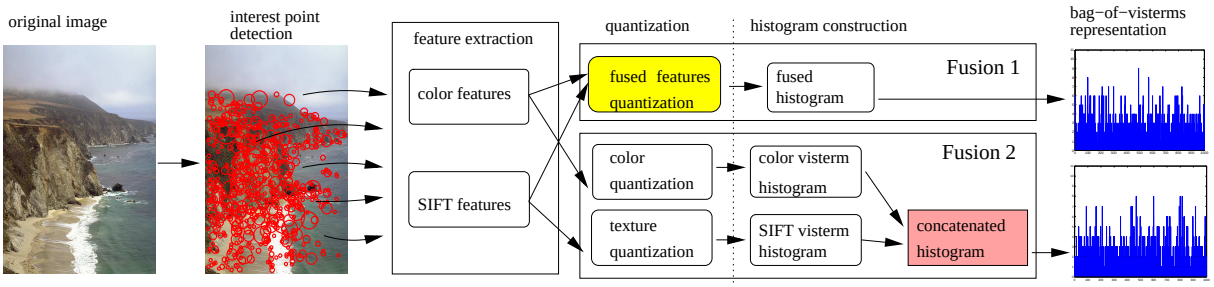


Figure 5.4. Schematic representation of our color and texture fusion system and of the two alternative fusion approaches: fusion between texture and color information is done either at the feature level before quantization (yellow box) or at the bag-of-visterm level(pink box). This is a specific implementation of the system presented in Section 4.3 in Chapter 4.

Color visterm modeling

To create a visterm methodology that uses color from local areas as an input for the visual vocabulary creation we need to gather local color information from those areas. In order to obtain local color information we designed a local color feature. Our local color feature is based on 121 Luv values computed on a 11×11 grid normalized to cover the local area given by the interest point detector. From these values, we calculate the mean and standard deviation for each dimension and concatenate the result into a 6-dimensional vector. Each dimension of this vector is then normalized to unit variance so that L (luminance) does not dominate the distance metric. The use of the Luv color space was motivated by the fact that it is a perceptual color space (it was designed to linearize the perception of color distances) and that it has also been known to perform well in both retrieval and recognition applications (Vailaya *et al.* (2001); Boutell *et al.* (2004)).

For the local texture/structure information we continue to use the SIFT local descriptor (Lowe (2004)). Given the scene classification performance results obtained until now, we assume that this is a good choice.

For a more compact representation, we applied a principal component analysis (PCA) decomposition on this features using training data. By keeping 95% of the energy, we obtain a 44-dimensional feature vector. The PCA step did not increase or decrease performance, but allowed for faster clustering.

5.8.3 SIFT and color visterm BOV classification performance

In this section we present the results obtained by our several approaches for this classification problem. We compare out system's performance with the baseline introduced in Section 5.2.2. Let us first explore the classification results obtained using the BOV representation constructed from each feature type separately.

Class	confusion matrix						performance
coasts	61.3	9.9	1.4	9.2	17.6	0.7	61.3
river/lakes	18.0	30.6	9.9	12.6	24.3	4.5	30.6
forests	0.0	0.0	90.3	2.9	6.8	0.0	90.3
plains	15.3	11.5	6.1	55.7	7.6	3.8	55.7
mountains	10.1	6.1	2.8	6.1	73.7	1.1	73.7
sky/clouds	14.7	2.9	0.0	14.7	0.0	67.6	67.6
overall							63.2
Baseline							67.2

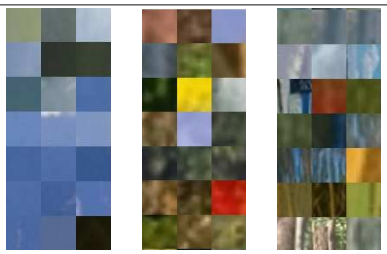


Table 5.20. Classification performance for the 6-class dataset D^V using the texture only, SIFT based, BOV representation (top). We can also see some samples of patches belonging to three random visterms (bottom). note: these results were obtained using a different implementation of the DOG and SIFT local interest point extractor/descriptor, and as such cannot be compare to the results presented in Table 5.11 (see Section 3.3 for a discussion on local interest point detectors/descriptors' implementations).

SIFT features

Table 5.20 provides the result obtained with the SIFT based BOV representation. While being slightly lower than the baseline, this approach performs surprisingly well given that no color information is used. This is illustrated by the sample patches belonging to 3 different visterms (Table 5.20, bottom). As can be seen (and as expected), visterm patches have no coherence in terms of color.

Color features

Table 5.21 shows the results obtained using the Luv color visterms. Although significantly smaller than the performance obtain with SIFT visterms, the result is still relatively good given the features' simplicity (6 dimensions). Overall, all classes are affected by the performance degradation. Surprisingly, the forest class gets the most degradation, indicating that there is more reliable information in the local structure than in the color. When observing samples associated to some visterms (Table 5.21, right), we can see that the goal of color coherence is achieved, but that coherence in terms of texture/structure is mainly lost (there remain some coherence due to the specific interest point detector employed). To further analyze the performance of our BOV

Class	confusion matrix						performance
coasts	49.3	16.9	2.8	12.7	15.5	2.8	49.3
river/lakes	21.6	31.5	9.0	7.2	30.6	0.0	31.5
forests	4.9	8.7	70.9	7.8	7.8	0.0	70.9
plains	9.2	9.2	6.9	53.4	16.8	4.6	53.4
mountains	12.3	12.3	1.7	14.0	59.2	0.6	59.2
sky/clouds	14.7	11.8	0.0	14.7	0.0	58.8	58.8
overall							53.9

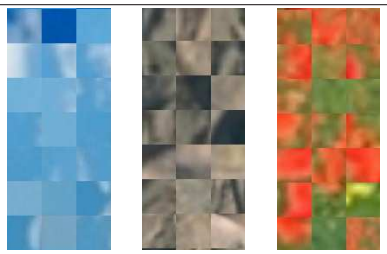


Table 5.21. Classification results for the Luv color space based BOV representation. Sample patches belonging to three random visterms.

approach, we compared it with a simple Luv color histogram (see Section 5.2.2). The system performance in this latter case exhibited a strong performance drop, achieving a 34.1% recognition rate. This illustrates the necessity for both a data-driven and local approach, embedded in out BOV representation, as compared to global approaches based on more arbitrary color representations.

5.8.4 Fusion classification performance

In this section we will now test the hypotheses that combining the two previously presented sources of local color and texture information we can improve over the use of any of the individual sources alone. We will explore both feature fusion approaches introduce in the previous chapter. To evaluate the performance of our proposed fusion approach we present the classification results combining color and texture information in the BOV representation.

Fusion 1:

As we seen in the previous chapter we call Fusion 1 to the approach in which local features are concatenated prior to clustering. This results in a joint texture/color vocabulary of 2000 visterms. The average mixing value α obtained through cross-validation was 0.8, indicating that more importance was given to the SIFT feature. Table 5.22 displays the results obtained in this case. These results shows an overall improvement w.r.t. those based on the SIFT feature

Class	confusion matrix						performance
coasts	69.0	8.5	2.1	7.7	10.6	2.1	69.0
river/lakes	21.6	28.8	9.0	11.7	26.1	2.7	28.8
forests	1.9	1.9	85.4	2.9	7.8	0.0	85.4
plains	9.2	9.2	2.3	62.6	12.2	4.6	62.6
mountains	8.4	5.6	1.1	5.6	77.7	1.7	77.7
sky/clouds	5.9	0.0	0.0	14.7	2.9	76.5	76.5
overall							66.7

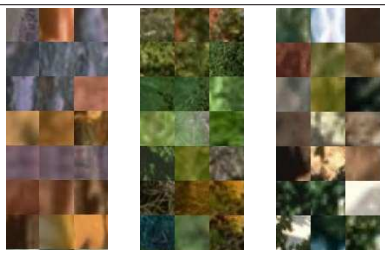


Table 5.22. Classification results with the first fusion strategy: joint texture/color visterms. Sample patches belonging to three visterms.

alone, and are very close to the baseline results (67.2%). The sky/clouds class is the one that beneficiate mostly from the improvement, with a reduction of its overlap with the coasts class.

When looking at the constructed vocabulary, we observe that visterms have coherence in both texture and color, as illustrated in Table 5.22. However, since now both features influence the clustering process, we notice an increase of the noise level in both color and texture coherence within the clusters.

Fusion 2:

In this second strategy, it is assumed that, at the interest point level, information gathered from color is independent from texture/structure information. This strategy thus works by concatenating the BOV representation of color and texture, after having them weighted by the factor α. Interestingly enough, the optimal α value was again found to be 0.8, showing again an emphasis on information arising from the SIFT features. Table 5.23 shows the obtained results. These are nearly identical to those obtained with the first fusion strategy, and again very close to those of the baseline.

Overall, the results are encouraging, and demonstrate that the two approaches are valid for the scene classification task. Both fusion approaches performed significantly better than grey-scale BOV representation. The fact that both approaches reach similar results to the baseline may indicate that we are reaching the performance limit that may be obtained in this data when not

Class	confusion matrix						performance
coasts	58.5	13.4	1.4	13.4	10.6	2.8	58.5
river/lakes	20.7	36.0	7.2	9.9	23.4	2.7	36.0
forests	1.9	1.0	89.3	2.9	4.9	0.0	89.3
plains	12.2	6.1	6.9	64.1	7.6	3.1	64.1
mountains	6.1	7.3	3.4	6.7	76.0	0.6	76.0
sky/clouds	14.7	0.0	0.0	11.8	0.0	73.5	73.5
overall							66.2

Table 5.23. Classification results with the second fusion strategy: concatenation of the texture and color BOV representation.

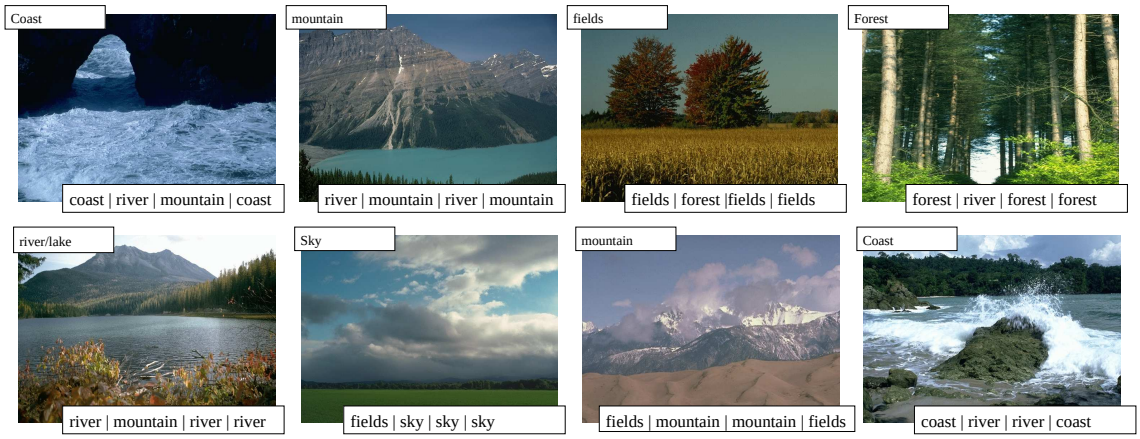


Figure 5.5. Images illustrating the resulting classification of the evaluated systems. Ground-truth is shown on the top left corner. On the bottom are all attributed labels: SIFT BOV, Color BOV, feature fusion and histogram fusion (from left to right).

using any spatial information.

Figure 5.5 shows some images with the labels attributed by each systems we tested. We can see that some labels are subjective. For some images several possible labels could be considered correct. As such some of the errors that the systems produce seem logical. This indicates that the BOV representation captures valid scene properties, however this dataset does not supply a clear enough class definition for the training of the systems.

5.9 Chapter conclusion

In this chapter we have presented experimental results that show that the BOV representation of scene images allows for a good performance in the task of classification, performing best or at least as good as the baseline methods in all tested datasets. In addition we have showed that

PLSA, when applied to the BOV representation, allows for a more robust system that performs better when we have less labeled training data. This is a matter of great importance for most modern systems where annotation of images is an expensive task. We have also explored tf-idf BOV weighting and found that little change in performance was obtained.

As an alternative to the standard BOV approach we also introduced a GMM based representation. This approach showed some improvements over the standard BOV approach, specially for more complex tasks. This motivates this approach for tasks where a more powerful representation may be desired. However, this representation is based on a GMM training which is more computationally expensive than the standard BOV.

To better represent natural scene images we explored the fusion of greyscale local descriptor with color features, as proposed in the previous chapter. The resulting representation allowed for an improvement in performance. Using BOV feature concatenation (cf. Fusion 2 defined in the previous chapter) we proposed the use of several SIFT based greyscale vocabularies. This resulted in an small, but promising increase in performance.

It is important to note that although we apply all image representations to classification of our images, these same representations can be potentially useful for other tasks, like for instance image auto-annotation.

Chapter 6

Scene ranking and segmentation

WE have seen in the previous chapter that by using PLSA we can obtain a more robust image representation, which performs better than BOV in the case of less training data. This increase in robustness is obtained due to the co-occurrence information that PLSA captures from unlabeled data. As we described in Chapter 4, PLSA captures the co-occurrence of visterms across all images in our dataset and attempts to disambiguate the possible polysemy and synonymy issues in our vocabulary. In the modeling of our data by PLSA, $P(z_l|I_i)$ gives the mixture weights of each aspect z_l for a particular image I_i, and $P(v_j|z_l)$ provides the multinomial probabilities of each visterm v_j for z_l. If we assume that each aspect captures specific visterm co-occurrences distributions in our images, then we may expect that the aspects will be characteristic of some semantic image content. Then if a certain aspect z_l has a large weight $P(z_l|I_i)$ for a particular image I_i, this should mean that the image contains the semantic elements captured by aspect z_l. We can explore this connection to relate image content with the image aspect decomposition $P(z|I)$. In the next section, we will explore the use of $P(z_l|I_i)$ to create a ranking of the images in our dataset, with respect to the semantic elements captured by each aspect. Some of the resulting aspect based rankings show a close affinity with the classes in our dataset, motivating the use of this approach for the task of image retrieval and browsing. This has been studied, in parallel to our work, by other authors to automatically produce a classification result by using PLSA to cluster images, based on prior knowledge about the number of classes (Sivic *et al.* (2005)). In Section 6.2, we will further explore the relation between the semantic content in the image and the aspects learned by the applied latent aspect modeling. Using the most significant aspect in the aspect decomposition of several image we will show that different aspects capture different semantic content which relate to different image areas, in a coherent way across images.

In Section 6.3 we will propose the use of the co-occurrence captured by $P(z_l|I_i)$ as a novel form of contextual information to improve segmentation of scene images. In our proposed approach we will use context in two ways: by using the fact that specific learned aspects correlate with the semantic classes, which resolves some cases of visual polysemy, and by formalizing the notion

that scene context is image-specific -what an individual visterm represents depends on what the rest of the visterms in the same bag represent too. We will demonstrate the validity of our approach on a man-made vs. natural visterm classification problem.

6.1 Scene image ranking using PLSA

Since we expect images with similar visterm co-occurrences to have similar visual content, we would like to perform image ranking based on the co-occurrence information captured by PLSA. Indeed, aspects can be conveniently illustrated by their most probable images in a dataset. Given an aspect z, images can be ranked according to:

$$P(I|z) = \frac{P(z|I)P(I)}{P(z)} \propto P(z \mid I) \tag{6.1}$$

where $P(I)$ is considered as uniform. Figure 6.1 displays the 10 most probable images from the 668 test images of the first split of the D^O dataset, for seven out of 20 aspects learned on the D^A dataset. The top-ranked images for a given aspect illustrate its potential 'semantic meaning'.

The top-ranked images representing aspect 1, 6, 8, and 16 all clearly belong to the landscape class. Note that the aspect indices have no intrinsic relevance to a specific class, given the unsupervised nature of the PLSA model learning. More precisely, aspect 1 seems to be mainly related to horizon/panoramic scenes, aspect 6 and 8 to forest/vegetation, and aspect 16 to rocks and foliage. Conversely, aspect 4 and 12 are related to the city class. However, as aspects are identified by analyzing the co-occurrence of local visual patterns, they may be consistent from this point of view (e.g. aspect 19 is consistent in terms of texture) without allowing for a direct semantic interpretation.

Considering the aspect-based image ranking as an information retrieval system, the correspondence between aspects and scene classes can be measured objectively. Defining the Precision and Recall paired values by:

$$Precision(r) = \frac{RelRet}{Ret} \quad Recall(r) = \frac{RelRet}{Rel},$$

where Ret is the number of retrieved images, Rel is the total number of relevant images and $RelRet$ is the number of retrieved images that are relevant, we can compute the precision/recall curves associated with each aspect-based image ranking considering either City and Landscape queries, as illustrated in Figure 6.2.

Those curves demonstrate that some aspects are clearly related to such concepts, and confirm observations made previously with respect to aspects 4, 6, 8, 12, and 16. As expected, aspect 19

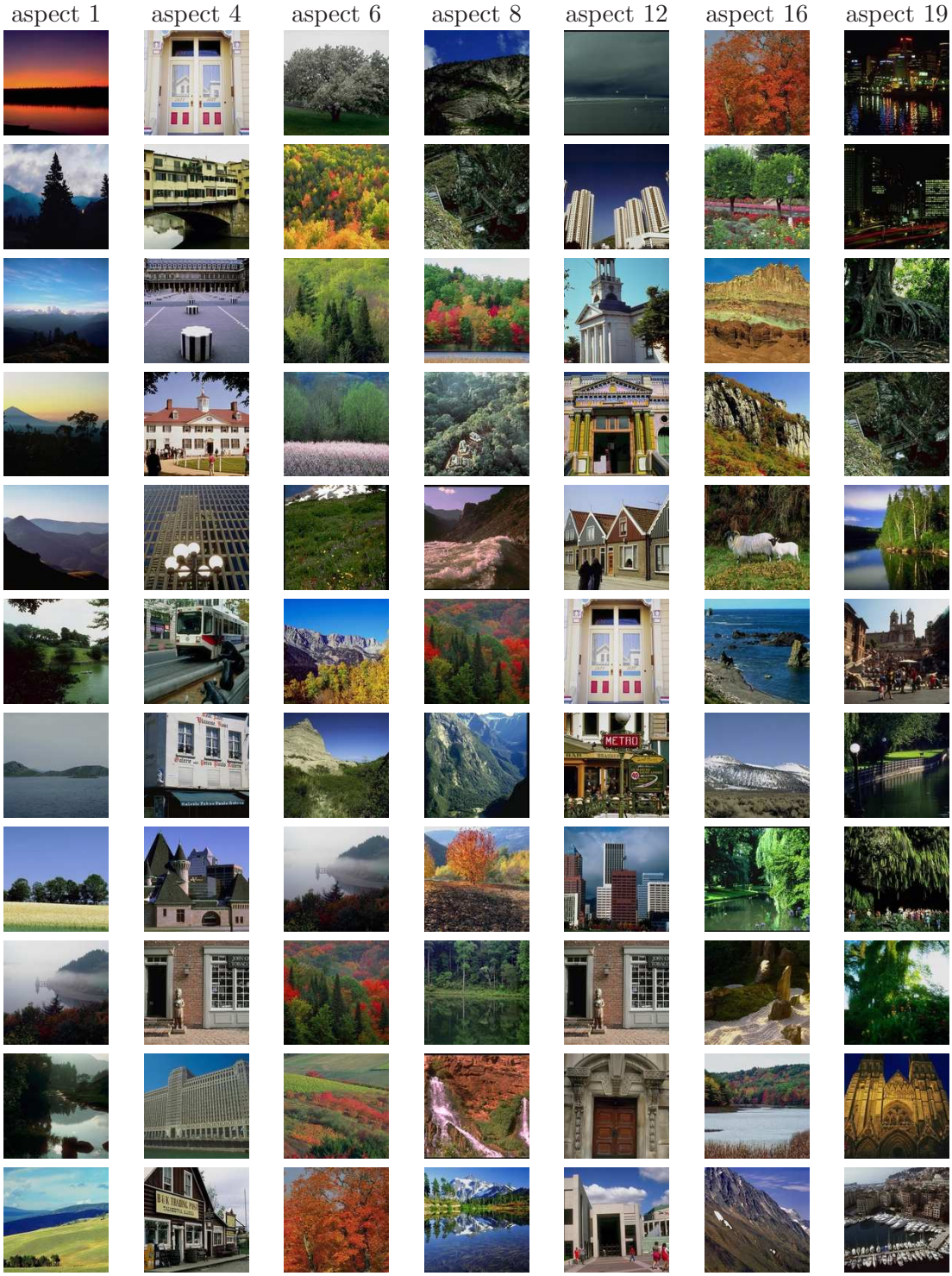


Figure 6.1. The 11 most probable images from the D^O dataset for six aspects (out of 20) learned on the D^A dataset. Images have been cropped to square size for convenient display.

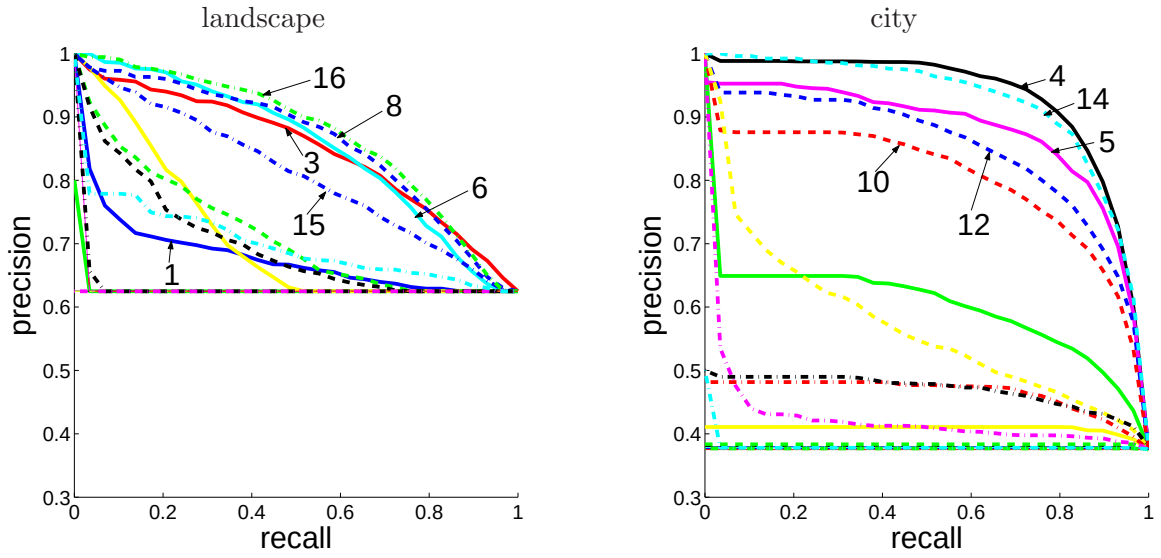


Figure 6.2. Precision/recall curves for the image ranking based on each of the 20 individual aspects, relative to the landscape (left) and city (right) query. Each curve represents a different aspect. Floor precision values correspond to the proportion of landscape(resp. city) images in the dataset.

does not appear in either the City or Landscape top precision/recall curves. The 'Landscape' related ranking from aspect 1 does not hold as clearly for higher recall values, because the co-occurrences of the visterm patterns appearing in horizons that it captures is not exclusive to the landscape class. Overall, these results illustrate that the latent structure identified by PLSA is highly correlated with the semantic structure of our data.

With the experiments we presented in this section we showed that there is, in fact, a relationship between the aspect obtained from PLSA modeling of our dataset and the dominant image content in our dataset. The experiments presented in this section demonstrate the potential of PLSA as a very attractive tool for browsing/annotating and exploring the content of unlabeled image collections.

6.2 Images as mixtures of aspects

As we have seen in Figure 6.1 some aspects from our model relate to specific scene categories. In this section we further study the relationship between aspects and scene content, by investigating which are the visterms which contribute to a given aspect. To this end, we will exploit the fact that PLSA explicitly decomposes the BOV of an image as a mixture of latent aspects expressed by the $P(z \mid I)$ distribution learned from PLSA. In this section we will consider the same latent structure with 20 aspects which was used for the aspect-based image ranking, presented in the previous section.

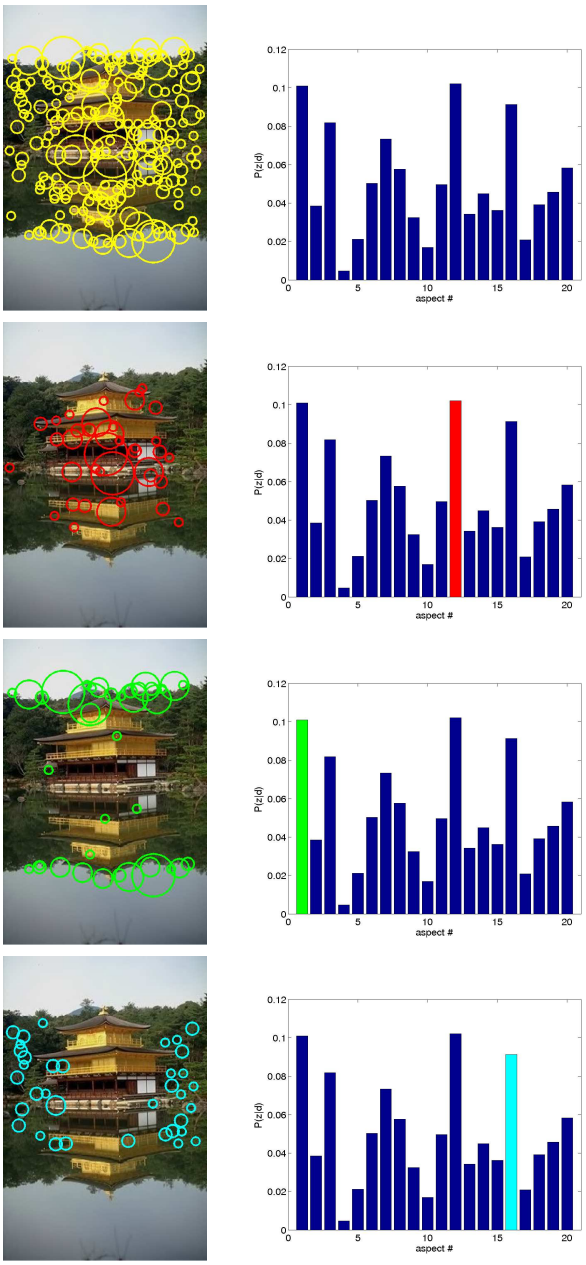


Figure 6.3. Illustration of the relation between the image's aspect distribution and the different visterm regions in the image.

We can assess the relevance of the PLSA modeling by evaluating whether the aspect's individual visterms themselves match the aspect scene type. This can be achieved by mapping each visterm of an image to its most probable aspect and displaying the resulting visterm labeling to generate a segmentation-like image. Accordingly, the mapping can be computed by:

$$\begin{aligned} z_{v_i} &= \arg\max_z(P(z \mid v_i, I)) \\ &= \arg\max_z(\frac{P(v_i \mid z)P(z \mid I)}{\sum_z P(v_i \mid z)P(z \mid I)}). \end{aligned} \tag{6.2}$$

Using this mapping we can now perform a simple segmentation of the image, by selecting the visterms v_i which have the same aspect mapping z_{v_i}. This is what is done in Figures 6.3 and 6.4. As we can see,by selecting the visterms which are mapped to the most significant aspects in the image, we obtain groups of visterms which define sparse regions of the image. For a given aspect, there regions have, most of the time, a consistent visual semantic content across images (for instance, look at visterms belonging to aspect 4 and 6 in images from Figure 6.4).

In Figure 6.4 we can see that it is possible to obtain a semantic content classification of the visterms in an image based on the two most important aspects of an image.

In the next section, we will exploit this property and more formally and in more detail the classification of all visterms into a region label, rather then into an aspect index.

6.3 Scene segmentation using latent space modeling

Associating semantic class labels to image regions is a fundamental task in computer vision, useful for image, video indexing and retrieval, and as an intermediate step for higher-level scene analysis (Kumar and Herbert (2003b,a); Lazebnik *et al.* (2003); Vogel and Schiele (2004a)). This task is intrinsically related to the task of scene classification presented in the previous chapter, however it differs from image classification in the fact that we classify part of the image as belonging to a class type.

While most segmentation approaches segment image pixels or blocks based on their luminance, color or texture, we use the same representation as used in the previous chapter, and consider local image regions characterized by viewpoint invariant descriptors (Lowe (2004)). As we have seen previously, this region representation is robust with respect to partial occlusion, clutter, and changes in viewpoint and illumination, (see Chapter 3). The use of local invariant descriptors for object image segmentation has been used by Dorko and Schmid (2003). In this setup all region descriptors that belong to the object class in the training set are modeled with a

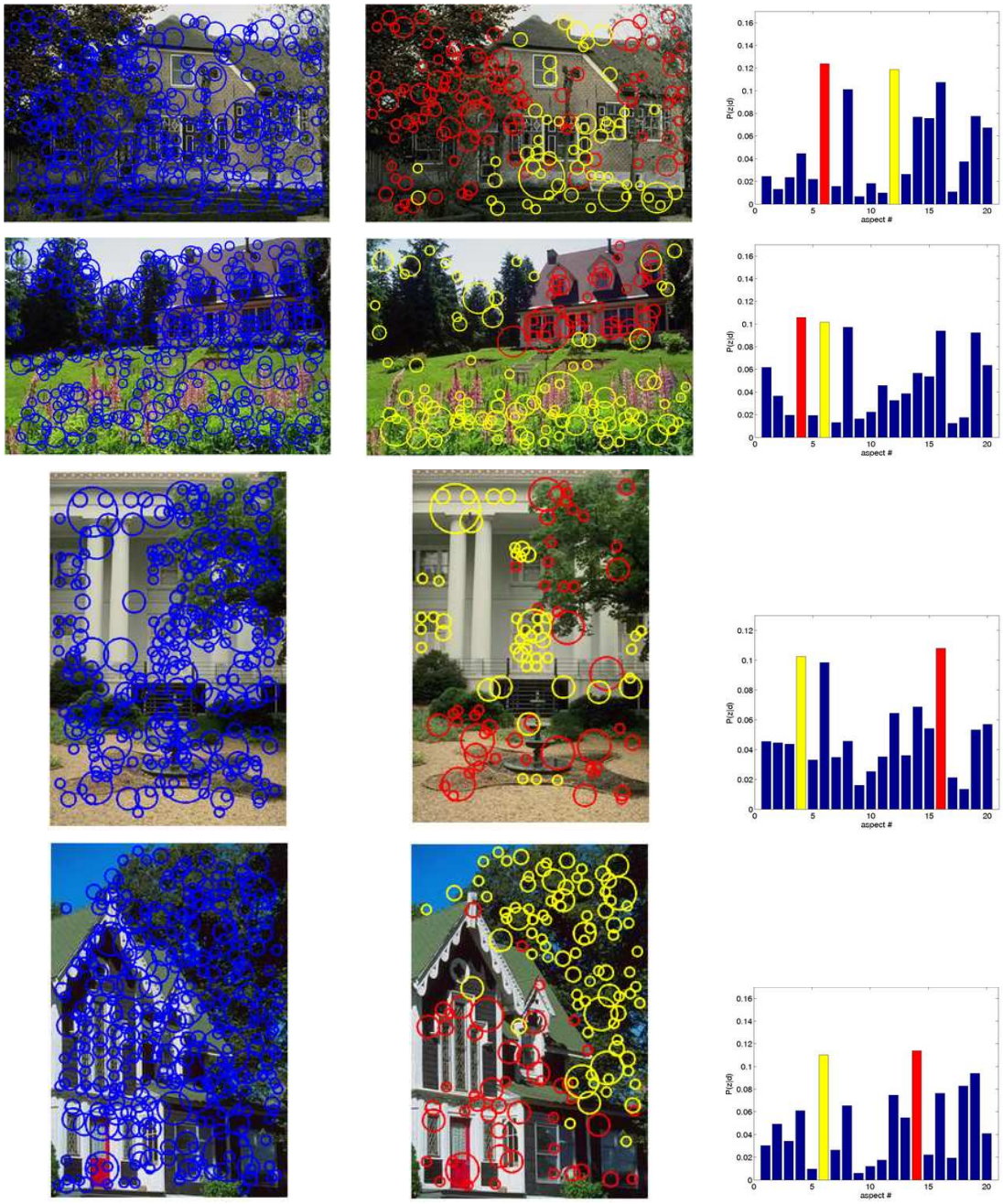


Figure 6.4. Images displaying aspect based semantic visterm classification. As we can see, using the two most important aspect in each image's aspect distribution $P(z \mid d)$, we obtain a segmentation of the image into semantic coherent regions. Also we can notice that the relation between aspects and classes that was exhibited when performing ranking of images still holds (see Figure 6.2). Aspects 6, and 16 relate to image regions overlapping with landscape and aspects 4, 12 and 14 are related to image regions overlapping with city.

Figure 6.5. Scene segmentation using local invariant regions (yellow). Regions are classified either as man-made (blue) or nature (not shown), and superimposed on a manual segmentation (white).

Gaussian Mixture Model (GMM), and a second GMM is trained on non-object regions. In this non-contextual approach, new regions are independently classified depending on their relative likelihood with respect to the object and non-object models. A similar approach introducing spatial contextual information through neighborhood statistics of the GMM components collected on training images is proposed in Lazebnik *et al.* (2003), where the learned prior statistics are used for relaxation of the original region classification.

In image segmentation, quantized local descriptors -referred to as *textons*- have also been used to build local BOV representations of windowed image regions (Malik *et al.* (2001)). The similarity between these regions is then defined based on the BOV histogram representation, and segmentation is conducted for each individual image using spectral clustering. Using a similar grid layout, Vogel and Schiele recently presented a two-stage framework to perform scene retrieval (Vogel and Schiele (2004a)) and scene classification (Vogel and Schiele (2004a)). This work involves an intermediate image block classification step, that can be seen as scene segmentation. Exploring spatial dependencies, Kumar and Herbert (2003a) apply a random field model to segment image areas that represent man-made scene structures. Their approach is based on the extraction of features from a grid of blocks that fully cover the image. Local invariant regions do not provide a full segmentation of an image, but they often occupy a considerable part of the scene and thus can define a "sparse" segmentation, as shown in Figure 6.5.

In this section we propose a novel approach for contextual segmentation of complex visual scenes, based on the exploitation of the visterm co-occurrence captured by the aspect distribution $P(z_l|I)$. The latent semantic analysis will allow us to model the context in two ways, as detailed in the next section: by using the fact that specific learned aspects correlate with the semantic classes, which resolves some cases of visual polysemy, and by formalizing the notion that scene context is image-specific -what an individual visterm represents depends on what the rest of the visterms in the same bag represent too. In addition, we show that the classical notion of context based on spatial proximity can be integrated within the framework. We demonstrate the validity of our approach on a man-made vs. natural visterm classification problem.

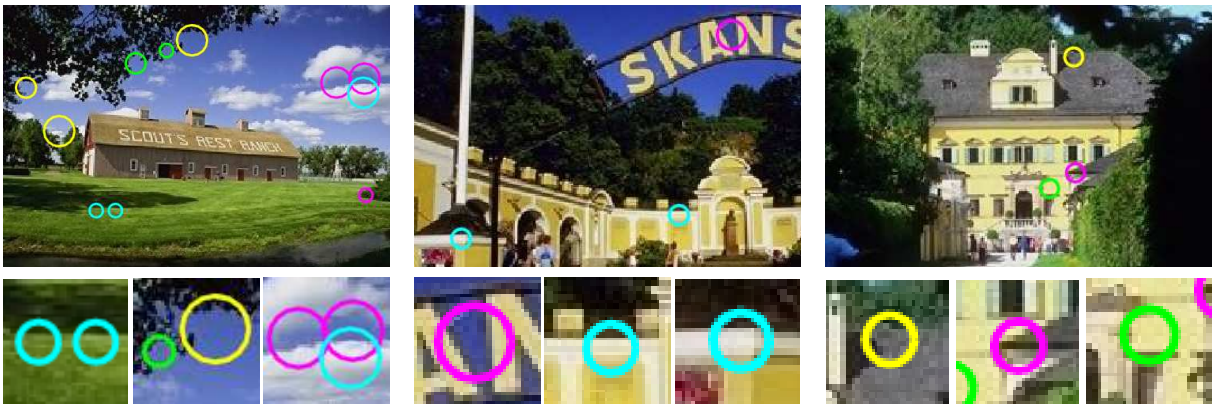


Figure 6.6. Regions (represented by visterms) can have different class labels depending of the images where they are found. Left: various regions (4 different colors, same color means same visterm) that occur on *natural* parts of an image. Center and right: the same visterms occur in man-made structures. All these regions were correctly classified by our approach, switching the class label for the same visterms depending on the context.

6.3.1 Approach

In general, the constituent parts of a scene do not exist in isolation, and the visual context -the spatial dependencies between scene parts- can be used to improve region classification (Li (1995); Kumar and Herbert (2003b,a); Murphy *et al.* (2003)). Two regions, indistinguishable from each other when analyzed independently, might be discriminated as belonging to the correct class with the help of context knowledge. Broadly speaking, there exists a continuum of contextual models for image segmentation. At one end, one would find explicit models like Markov Random Fields, where spatial constraints are defined via local statistical dependencies between class region labels (Geman and Geman (1984); Li (1995)), and between observations and labels (Kumar and Herbert (2003a)). The other end would correspond to context-free models, where regions are classified assuming statistical independence between the region labels, and using only local observations (Dorko and Schmid (2003)).

Lying between these two extremes is the BOV representation which we have until now explored for classification. On one hand, unlike explicit contextual models, spatial neighborhood information are discarded in this representation, and any ordering between the descriptors disappears. On the other hand, unlike point-wise models, although the descriptors are still local, the scene is represented collectively. This can explain why, despite the loss of "strong" spatial contextual information, BOVs have been successfully used in a number of problems, including object matching (Sivic and Zisserman (2003)) and categorization (Willamowski *et al.* (2004); Sivic *et al.* (2005)), and, as we have demonstrated in the previous chapters, scene classification (Quelhas *et al.* (2005); Fei-Fei and Perona (2005); Bosch *et al.* (2006)). In addition, as a collection of discrete data, the BOV representation is suitable for building probabilistic models where a new form of context is implicitly captured through visterm co-occurrence.

The main issue with classifying regions using visterms is that visterms are not class-specific. As shown in Figure 6.6, the same visterms commonly appear both in man-made and nature views. This is not unexpected, as we have seen in Chapter 4 that the majority of visterms occur in both man-made and natural scenes (see Figure 4.7). This is due to the visterm vocabulary construction which does not make use of class label information. This class independent occurrence constitutes a problematic form of visual polysemy.

In this section, we show that aspect models can also be used for region classification. We propose probabilistic models that exploit two ways of using context. In the first place, we use the fact that specific aspects correlate with the semantic classes, which implicitly helps in cases of polysemy (Hofmann (2001)). In the second place, scene context is image-specific: the "meaning" of a particular visterm depends on what the "meaning" of the other visterms occurrences in the same bag is. We show that this relation can be formally expressed in the probabilistic model, so that even though visterms occur in both classes, the information about the other visterms in the same bag can be used to influence the classification of a given visterm in a specific class, and hence improve visterm discrimination.

To illustrate our approach, we present results on a man-made vs. natural region classification task, and show that the contextual information learned from co-occurrence improves the performance compared to a non-contextual approach. In our view, the proposed approach constitutes an interesting way to model visual context that could be applicable to other problems in computer vision. Furthermore, we show, through the use of a Markov Random Field model, that standard spatial context can be integrated, resulting in an improvement of the final segmentation.

6.3.2 Scene segmentation problem formulation

The perspective on image segmentation that we consider in this work differs from the traditional notion of homogeneous region partition of the image. We perform segmentation of the image based on class labels defined in our dataset, and we base our segmentation on the classification of local patches that do not cover the whole image. Our segmentation task can be formulated as the automatic extraction of visterms from the image followed by the classification of each visterm v into a class c, where c stands either for $m$an-made structures or *natural* regions. Regarding the feature extraction and visterm attribution we maintain the choices presented in the previous chapter. For our local interest point detectors we use the DOG detector (see Section 3.1.2), and as our local descriptor we use the SIFT descriptor (see Section 3.2.4). The vocabulary used in the experiments presented in this section was trained on the dataset D^A using K-means clustering.

Visterm segmentation

Assume a discrete set of visterms, corresponding to the quantization of the local descriptors. We rely on a likelihood ratio computation to classify each visterm v of a given image I into a class c. The ratio is defined by

$$LR(v) = \frac{P(v|c = \text{man-made})}{P(v|c = \text{natural})}, \tag{6.3}$$

where the probabilities will be estimated using different models of the data, as described in the next subsections.

Empirical distribution

Given a set of training data, the term in Equation 6.3 can be simply estimated using the empirical distribution of visterms, as was done in Dorko and Schmid (2003). More precisely, given a set of manually segmented images D_s^S, into man-made and natural regions (e.g. Figure 6.5 (c)), $P(v|c)$ is simply estimated as the number of times the visterm v appears in regions of class c, divided by the number of occurrences of v in the training set.

6.3.3 Aspect modeling

In this section we propose two probabilistic aspect models that exploit the co-occurrence information captured by the PLSA aspects for visterm classification.

Empirical estimation of probabilities is simple but may suffer from several drawbacks. Firstly, a significantly large amount of training data might be necessary to avoid noisy estimates, especially when using large vocabulary sizes. Secondly, such estimation only reflects the individual visterm occurrences, and does not account for any kind of relationship between them. We propose to exploit aspect models that capture visterm co-occurrences to classify visterms (Hofmann (2001); Blei *et al.* (2003)). These models, through the identification of latent aspects, enable the classification of the visterms of an image based on the occurrence of other visterms in the same image. The histogram of visterms in image I, the BOV, contains this information. Even if the BOV representation discards all spatial neighborhood relationships, we expect the co-occurrence context (i.e. the other visterms) to provide some of the necessary information for the disambiguation of the meaning of a polysemic visterm. To this end, we propose two models.

Aspect model 1

The first model associates a hidden variable representing our aspects $z \in \mathcal{Z} = \{z_1, \dots z_{N_A}\}$ with each observation according to the graphical model of Figure 6.7, leading to the joint probability defined by:

$$P(c, I, z, v) = P(c)P(I|c)P(z|I)P(v|z). \tag{6.4}$$

This model introduces several conditional independence assumptions. The first one, traditionally encountered in aspects models, is that the occurrence of a visterm v is independent of the image I it belongs to, given an aspect z. The second assumption is that the occurrence of aspects is independent of the class the document belongs to. The parameters of this model are learned using the maximum likelihood (ML) principle (Hofmann (2001)). The optimization is conducted using the Expectation-Maximization (EM) algorithm, allowing us to learn the aspect distributions $P(v|z)$ and the mixture parameters $P(z|I)$.

Notice that, given our model, the EM equations do not depend on the class label. Besides, the estimation of the class-conditional probabilities $P(I|c)$ do not require the use of the EM algorithm. We will exploit these points to train the aspect models on a large dataset (denoted D^S) where only a small part of it has been manually labeled according to the class (we denote this subset by D_l^S. This allows for the estimation of a precise aspect model, while alleviating the need for tedious manual labeling. Regarding the class-conditional probabilities, as the labeled set will be composed of man-made-only or natural-only images, we simply estimate them according to:

$$P(I|c) = \begin{cases} 1/N_c & \text{if } I \text{ belongs to class } c \\ 0 & \text{otherwise,} \end{cases} \tag{6.5}$$

where N_c denotes the number of images belonging to class c in the labeled set D_l^S. Given this model, the likelihood we are looking for can be expressed as

$$P(v|c) = \sum_{l=1}^{N_A} P(v, z_l|c) = \sum_{l=1}^{N_A} P(v|z_l)P(z_l|c), \tag{6.6}$$

where the conditional probabilities $P(z_l|c)$ can in turn be estimated through marginalization over labeled images,

$$P(z_l|c) = \sum_{I \in D_l^S} P(z_l, I|c) = \sum_{I \in D_l^S} P(z_l|I)P(I|c). \tag{6.7}$$

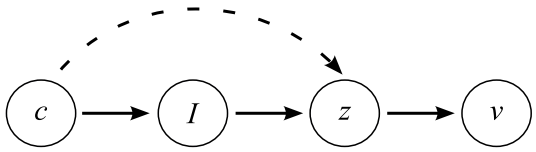


Figure 6.7. Aspect model 1 and aspect model 2 (if including dashed line).

These equations allow us to estimate the likelihood ratio as defined by Equation 6.3. Note that this model extends PLSA by introducing the class variable (Hofmann (2001)).

algorithm for the training of model 1.

1. learn the aspect model parameters $P(v \mid z)$ and $P(z|I)$ in an unsupervised way on D^S. This is equivalent to applying the PLSA learning procedure on D^S (see Figure 4.5.1).
2. compute the $p(v|c)$ from Equation 6.6, using the labeled dataset D_l^S.
3. compute the likelihood ratio $LR(v)$ (Equation 6.3) using the result from step 2.

Figure 6.8. Description of the algorithm for scene segmentation using model 1.

Aspect model 2

From Equation 6.6, we see that, despite the fact that the above model captures co-occurrence of the visterms in the distributions $P(v|z)$, the context provided by the specific image I has no direct impact on the likelihood. To explicitly introduce this contextual knowledge, we propose to evaluate the likelihood ratio of visterms conditioned on the observed image I,

$$LR(v, I) = \frac{P(v|I, c = \text{man-made})}{P(v|I, c = \text{natural})}. \tag{6.8}$$

The evaluation of $P(v|I, c)$ can be obtained by marginalizing over the aspects,

$$P(v|I, c) = \sum_{l=1}^{N_A} P(v, z_l|I, c) = \sum_{l=1}^{N_A} P(v|z_l)P(z_l|I, c), \tag{6.9}$$

where we have exploited the conditional independence of visterm occurrence given the aspect variable. Under model 1 assumptions, $P(z_l|I,c)$ reduces to $P(z_l|I)$, which clearly shows the limitation of this model to introduce both context and class information. To overcome this, we assume that the aspects are also dependent on the class label (cf dashed link in Figure 6.7). The parameters of this new model are the aspect multinomial $P(v|z)$ and the mixture multinomial $P(z|I,c)$, which could be estimated by an EM algorithm, as with PLSA, but using only labeled data this time. However, as our model is not fully generative (Blei *et al.* (2003)), from the learned model only $P(v|z)$ could be directly used and we would have to estimate $P(z|I_{new},c)$ for each new image I_{new}. Since the class is obviously unknown for new images, this means that in practice all the dependencies between aspects and labels observed in the training data and learned in the $P(z \mid I_{train},c)$ multinomial would be lost. To avoid this, we propose to separate the contributions to the aspect likelihood due to the class-aspect dependencies, from the contributions due to the image-aspect dependencies. Thus, we propose to approximate $P(z_l|I,c)$ as:

$$P(z_l|I,c) \propto P(z_l|I)P(z_l|c), \tag{6.10}$$

where $P(z_l|c)$ is still obtained using Equation 6.7. The complete expression for the contextual visterm probability modeling is thus given by:

$$P(v|I,c) \propto \sum_{l=1}^{N_A} P(v|z_l)P(z_l|c)P(z_l|I). \tag{6.11}$$

The main difference with Equation 6.6 is the introduction of the contextual term $P(z_l|I)$, which means that visterms will not only be classified based on them being associated to class-likely aspects, but also on the specific occurrence of these aspects in the given image.

Inference on new images

With aspect model 1 (and also with empirical distribution), visterm classification is done once for all at training time, through the visterm co-occurrence analysis on the training images. Thus, for a new image I_{new}, the extracted visterms are directly assigned to their corresponding most likely label. For aspect model 2, however, the likelihood-ratio $LR(v,I_{new})$ (Equation 6.8) involves the aspect parameters $P(z|I_{new})$ (Equation 6.11). Given our approximation (Equation 6.10), these parameters have to be inferred for each new image, in a similar fashion as for PLSA (Hofmann (2001)). $P(z_l|I_{new})$ is estimated by maximizing the likelihood of the BOV representation of I_{new}, fixing the learned $P(v|z_l)$ parameters in the Maximization step. Figure 6.3.3 summarizes the approach.

algorithm for the training of model 2.

1. learn $p(v \mid z)$ by applying the standard PLSA procedure to the unlabeled data D^S (cf. algorithm described in Figure 4.5.1).
2. learn the $p(z_l|c)$ using Equation 6.7 applied to the labeled data D_l^S.
3. for a new image I_{new}:
   - compute $p(z_l|I_{new})$ using the PLSA procedure (cf. algorithm described in Figure 4.5.1).
   - compute $p(v|I_{new}, c)$ using Equation 6.11 .
   - compute the likelihood ratio $LR(v)$ (Equation 6.3).

Figure 6.9. Description of the algorithm for scene segmentation using model 2.

6.3.4 Experimental setup

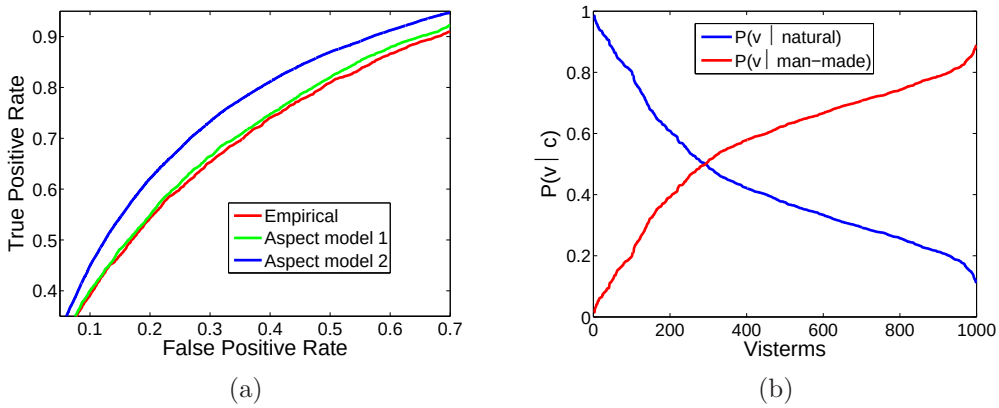
We validate our proposed models on the segmentation of scenes into natural vs. man-made structures. This Section first presents our setup. It is followed by detailed objective performance evaluation illustrated with segmentation results on a few test images. Finally, we study the integration of a regularization strategy to further improve results.

Datasets

In our experiments we used three image datasets. The first set, D^O, contains 6680 photos depicting mountains, forests, buildings, and cities. This is the same dataset used in Chapter 5 for scene classification. From this set, 6000 images are used with no associated label D_t^S, while the remaining subset D_l^S is composed of 600 images, whose content mainly belonged to one of the two classes, and were hand-labeled with a single class label leading to approximately 300 images of each class. Dataset D_t^S was used to construct the vocabulary and learn the aspect models, while D_l^S was used to estimate the likelihoods for each class (cf. Equation 6.7). A third dataset D_s^S, containing 485 images of man-made structures in natural landscapes, which were hand-segmented with polygonal shapes (Figure 6.5), was used to test the methods.

Performance evaluation

Given that we are segmenting the image based on local descriptors, but have the segmentation ground-truth based on segmentation mask which are uniform areas, we need to attribute the



(a) (b)

Figure 6.10. (a) True Positive Rate vs. False Positive Rate for the three methods. (b) $P(v \mid c)$ for man-made and natural structures, estimated on the test set.

ground-truth labels to the local descriptors. This is done by attributing to each local descriptor the label of the pixel of its position $\mathbf{x}$. This approach for the attribution of labels to each visterm was chosen due to its simplicity. A possibly more appropriate approach could be based on the overlap of the local interest area with the segmented regions. Also, the elimination of points near the boundaries of the ground-truth segmentation would eliminate ambiguous visterms which define an area which covers both man-made and natural areas of the image. The global performance of the algorithm was assessed using the True Positive Rate (TPR, number of correct positive visterms retrieved over the total number of positive visterms in the test set), False Positive Rate (FPR, number of false positives over the total number of negative visterms) and True Negative Rate (TNR=1-FPR), where man-made structure is the positive class. The FPR, TPR and TNR values vary with the threshold applied to each model's likelihood ratio.

Parameter setting

In the previous section we found that the use of a vocabulary with 1000 visterms and an aspect model with 60 aspects performed well for the task of scene classification. Motivated by those results we use the same 1000 visterm vocabulary V'_{1000} (cf. Section 5.6.1) and we choose to learn 60 aspects for both aspect model 1 and 2.

6.3.5 Results

Figure 6.10(a) displays the Receiver Operating Curve (ROC, TPR vs. FPR) of the two aspect models and the empirical distribution (baseline). As can be seen, the aspect model 1 performs slightly better than the empirical distribution method (although not significantly), while aspect model 2 outperforms significantly the two other methods (confirmed with paired t-test with

p=0.04).

To further validate our approach, Table 6.1 reports the Half-Total-Recognition Rate (HTRR) measured by 10-fold cross-validation. For each of the folds, 90% of the test data D_s^S is used to estimate the likelihood threshold T_{EER} leading to Equal Error Rate (EER, obtained when TPR=TNR) on this data. This threshold is then applied on the remaining 10% (unseen images) of D_s^S, from which the HTRR (HTRR=(TPR+TNR)/2) is computed. This table shows that the ranking observed on the ROC curve is clearly maintained, and that aspect model 2 results in a 7.5% performance relative increase w.r.t. the baseline approach.

	Emp. distribution	Aspect mod. 1	Aspect mod. 2	Ideal (no context)
HTRR (%)	67.5	68.5	72.4	71.0

Table 6.1. Half Total Recognition Rate (in percent).

Ideal case (no context)

As mentioned in Section 6.3.3, aspect model 1 and the empirical distribution method assign specific visterms to the man-made or natural class independently of the individual documents in which those visterms occur. This sets a common limit on the maximum performance of both systems, which is referred here as the *ideal case*. This limit is given by attributing to each visterm the class label corresponding to the class in which that visterm occurs the most in the *test data*. On our data, this *ideal case* provides an HTRR of 71.0%, showing that the visterm class attribution from empirical distribution and aspect model 1 is already close to the best achievable performance. Indeed, the visterm class conditional probabilities shown in Figure 6.10b indicate that there is a substantial amount of polysemy. The class conditional probabilities are obtained by dividing the number of visterm occurrences in one class by the number of that visterm occurrences in both classes. Polysemy is indicated by the simultaneously quite high probabilities in both classes (e.g. for instance note that all visterms appear at least 15% in the *natural* class). Thus, in order to have a chance of performing better than the *ideal case*, visterms must be labeled differently depending on the specific image that is being segmented. This is the case with the aspect model 2 which, due to its ability to address the polysemy and synonymy ambiguities, is able to outperform the *ideal case*. More precisely, aspect model 2 switches visterm class labels according to the contextual information gathered through the identification of image-specific latent aspects. Indeed, in our data, successful class label switching occurs at least once for 727 out of the 1000 visterms of our vocabulary.

The impact of the contextual model can also be observed on individual images. Figure 6.11 displays examples of man-made structure segmentation, where likelihood thresholds were estimated at EER value. As we can observe in those images, aspect model 2 improves the segmentation with respect to the two other methods in two different ways. On one hand, in the first three examples, aspect model 2 increases the precision of the man-made segmentation, producing a slight decrease in the corresponding recall (some points in the man-made areas are lost). On the

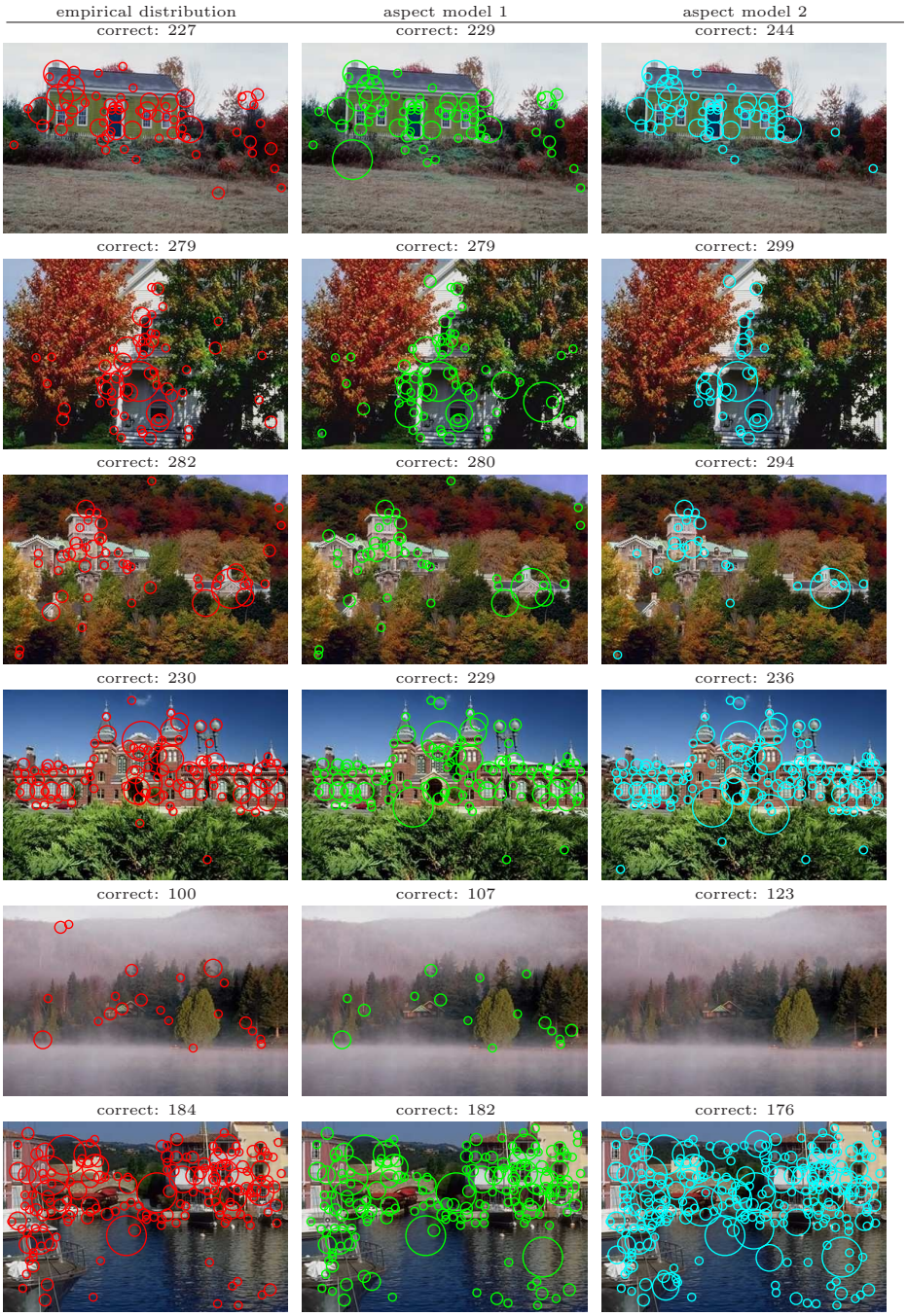


Figure 6.11. Image segmentation examples for a likelihood threshold set at the Equal Error Rate on the training data. Images show the visterms that were classified as belonging to a man-made structure. Results provided by: first column, empirical distribution; second column, aspect model 1; third column, aspect model 2. The total number of correctly classified regions (man-made + natural) is given per image. The first five rows illustrate cases where aspect model 2 outperforms the other approaches. In the fifth row, an extreme example of a strong natural context that is correctly identified by aspect model 2 leads to the classification of all regions as natural (though some should be labeled as man-made). The last row shows the confusion of the region classification, when the context is not correctly identified (in this case, overestimated) by aspect model 2.

other hand, the fourth example shows aspect model 2 producing a higher recall of man-made visterms while maintaining a stable precision. In the fifth example, the occurrence of a strong context causes the whole image to be taken as natural scene. In the sixth example, however, the overestimation of the man-made related aspects leads to visterms that are dominantly classified as man-made. Nevertheless, overall, as indicated in Figure 6.10 and Table 6.1, the introduction of context by co-occurrence is beneficial.

6.3.6 Markov Random Field (MRF) regularization

The contextual modeling with latent aspects that we present in this section can be conveniently integrated with traditional spatial regularization schemes. To investigate this we present the embedding of our contextual model within the MRF framework (Geman and Geman (1984)), though other schemes could be similarly employed (Kumar and Herbert (2003b); Lazebnik *et al.* (2003)).

Problem formulation

Let us denote by $\mathcal{S}$ the set of sites s, and by $\mathcal{Q}$ the set of cliques of two elements associated with a second-order neighborhood system $\mathcal{G}$ defined over $\mathcal{S}$. The segmentation can be classically formulated using the Maximum A Posteriori (MAP) criterion as the estimation of the label field $C = \{c_s, s \in \mathcal{S}\}$ which is most likely to have produced the observation field $\mathcal{V} = \{v_s, s \in \mathcal{S}\}$. In our case, the set of sites is given by the set of interest points, the observations v_s take their value in the set of visterms V, and the labels c_s belong to the class set $\{man-made, natural\}$. Assuming that the observations are conditionally independent given the label field (i.e. $p(\mathcal{V}|C) = \prod_s p(v_s|c_s)$), and that the label field is an MRF over the graph $(\mathcal{S}, \mathcal{G})$, then due to the equivalence between MRF and Gibbs distribution ($p(x) = \frac{1}{Z}e^{-U(x)}$), the MAP formulation is equivalent to minimizing an energy function (Geman and Geman (1984)):

$$U(C,\mathcal{V}) = \underbrace{\sum_{s\in\mathcal{S}} V_1(c_s) + \sum_{\{t,r\}\in\mathcal{Q}} V_1'(c_t,c_r)}_{U_1(C)} + \underbrace{\sum_{s\in\mathcal{S}} V_2(v_s,c_s)}_{U_2(C,\mathcal{V})}, \tag{6.12}$$

where U_1 is the regularization term which accounts for the prior spatial properties (homogeneity) of the label field, whose local potentials are defined by:

$$\begin{aligned} V_1(\text{man-made}) = \beta_p \ \text{ and } \ V_1(\text{natural}) = 0, \\ V_1'(c_t,c_r) = \begin{cases} \beta_d & \text{if } c_t \neq c_r, \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \tag{6.13}$$

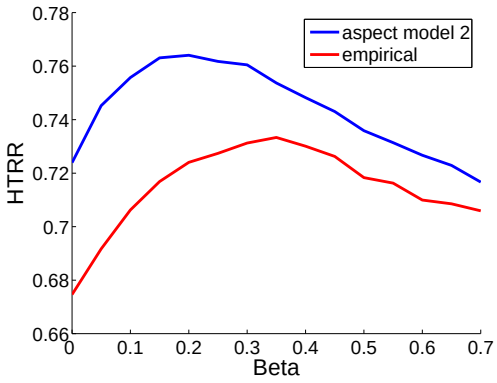


Figure 6.12. Plot of the Half Total Recognition Rate as we alter the value of the regularization parameter β_d.

β_d is the cost of having neighbors with different labels, while β_p is a potential that will favor the man-made class label (if $\beta_p < 0$) or the natural one (if $\beta_p > 0$), and U_2 is the data-driven term for which the local potential are defined by:

$$V_2(v_s, c_s) = -\log(p(v_s|c_s)). \tag{6.14}$$

where the probability $p(v_s|c_s)$ is given by any of the models studied in Section 6.3.2 (empirical distribution) or Section 6.3.3 (aspect model 1 - Equation 6.6, or aspect model 2 - Equation 6.11).

Experimental setup

To implement the above regularization scheme, we need to specify a neighborhood system. Several alternatives could be employed, exploiting for instance the scale of the invariant detector (e.g. see Lazebnik *et al.* (2003)). Here we used a simpler scheme: two points t and r are defined to be neighbors if r is one of the $N_{\mathcal{N}}$ nearest neighbors of t, and vice-versa. For this set of experiments we defined the neighborhood to be constituted by the five nearest neighbors. Finally, in the experiments, the minimization of the energy function of Equation 6.12 was conducted using simulated annealing (Li (1995)).

The performance is evaluated using the Half Total Recognition Rate, as defined in Section 6.3.5. We investigate the impact of the regularization on the segmentation. The level of regularization is defined by β_d (a larger value implies a larger effect). The regularization is conducted by starting at the Equal Error Rate point, as defined in the 10-fold cross-validation experiments described in Section 6.3.5. More precisely, for each of the folds, the threshold T_{EER} is used to set the prior on the labels by setting $\beta_p = -\log(T_{EER})$. Thus, in the experiments, when $\beta_d = 0$ (i.e. no spatial regularization is enforced), we obtain the same results as in Table 6.1.

Figure 6.13. Effect of the MRF regularization on the man-made structure segmentation. The first three rows illustrate the benefit of the MRF regularization where wrongly classified isolated points are removed. The last row shows the deletion of all man-made classified regions from an image when natural regions dominate the scene.

Results

Figure 6.12 displays the evolution of the HTRR in function of the regularization parameter β_b. On that plot we can see that the best segmentation performance corresponds to an HTRR of 73.1% and a β_d of 0.35 with the empirical modeling, and an HTTR of 76.3% for a β_d of 0.2 and aspect model 2. This latter value of β_d is chosen for all the MRF illustrations reported in Figure 6.13 and 6.14.

The inclusion of the MRF relaxation boosted the performance of both aspect model 2 and empirical distribution. However, even if boosting benefited most to the empirical distribution modeling, it is important to point out that aspect model 2 still outperforms the empirical

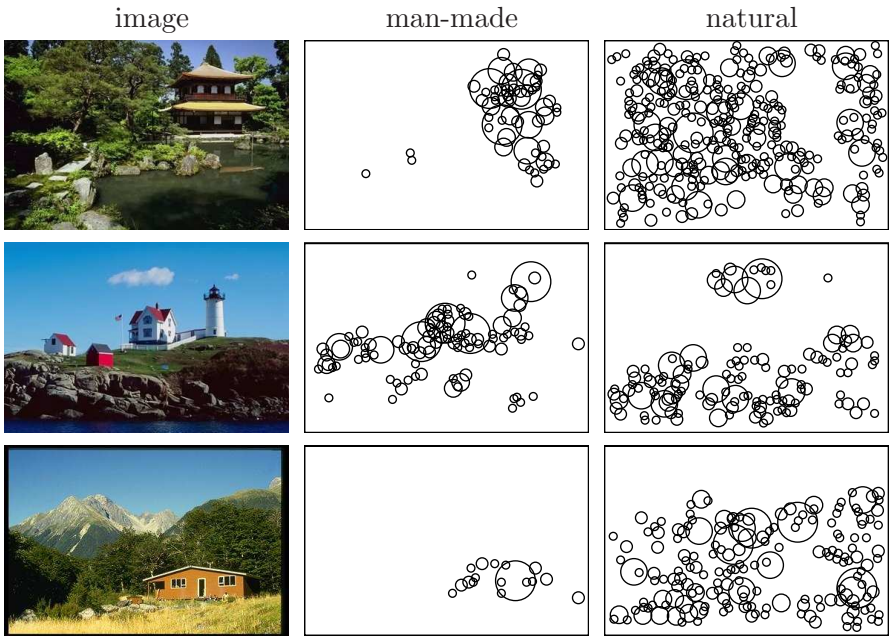


Figure 6.14. Three other examples that illustrate the final segmentation obtained with aspect model 2 and MRF regularization. The display is different than in previous figures to avoid image clutter.

distribution model. This was to be expected, as aspect model 2 was already capturing some of the contextual information that the spatial regularization can provide (notice also that the maximum is achieved for a smaller value of β_d in aspect model 2).

Besides obtaining an increase of the HTRR value, we can visually notice a better spatial coherence of the segmentation, as can be seen in Figure 6.13 and 6.14. We can observe in the images that the MRF relaxation process reduces the occurrence of isolated points, and tends to increase the density of points within segmented regions. We show on the last row of Figure 6.13 that as can be expected when using prior modeling, on certain occasions the MRF step can over-regularize the segmentation, causing the attribution of only one label to the whole image.

6.4 Chapter conclusion

In this chapter we explored the PLSA-based ranking provided by the PLSA modeling. We found that this unsupervised methodology allows us to browse the data according to different visual themes that are extracted from the BOV representation and which directly relate to the type of content that are captured by the visterms (local structure in the current case). This relation strongly motivates the use of this approach for the design of browsing and exploration tools for image collection.

Also in this chapter, we have shown the decomposition of the BOV representation as a mixture of aspects learned by the PLSA modeling. We illustrated that this mixture decomposition can translate into an aspect-based segmentation of an image. To further validate this idea, we introduced new computational models to perform a contextual segmentation of images. These models enable us to exploit a different form of visual context, based on the co-occurrence analysis of visterms in the whole image rather than on the more traditional spatial relationships. Visterm co-occurrence is summarized into aspects models, whose relevance is estimated for any new image, and used to evaluate class-dependent visterm likelihoods. We have tested and validated these models on a man-made vs. natural scene image segmentation task. One model has clearly shown to help in disambiguating polysemic visterms based on the context they appear in. Producing satisfactory segmentation results, it outperforms a state-of-the-art likelihood ratio method (Dorko and Schmid (2003)). Moreover, we investigated the use of Markov Random Field models to introduce spatial coherence in the final segmentation and show that the two types of context models can be integrated successfully. This additional information enables to overcome some visterm classification errors from the likelihood ratio and aspect models methods, increasing the final segmentation performance. The results we obtained motivate this new contextual co-occurrence modeling approach as a promising tool for image segmentation.

Chapter 7

Object Recognition

IN this chapter we will explore the use of local descriptors for the task of object recognition, classification and ranking. In the previous chapters we have presented and discussed several local descriptor based approaches, but these had only been applied on scene recognition and segmentation tasks. In this chapter we extend our experiments to images of objects. We will start this chapter by addressing the object recognition task, within a one shot learning framework, i.e., a framework in which a specific object is learned using only one image of that object. The models learned for each object of a predefined set are then used to recognize the identity of an object appearing in one image, where the object is seen from any viewpoint and image resolution. The methodology for this approach relates to the methodology we used in Chapter 3 to perform wide-baseline matching. Within a fusion framework, we propose to use color bi-modes (Matas *et al.* (2002a)) local descriptors in addition to local structure descriptors to improve the object modeling and to increase the recognition performance. We tested this method on household planar objects (boxes). In a second work, we evaluate the BOV representation presented in Chapter 4 in a 7-class object classification task, where the goal is to classify each image as containing an object of a certain class. Finally, we will verify if by applying latent aspect modeling to our representation we can find aspects that correlate to object classes and obtain a more robust representation w.r.t. smaller amount of training data, as observed for scene classification in the previous chapter.

7.1 Object modeling from one training image

Object recognition can be based on several aspects of the object's representation in an image: shape (Malik *et al.* (2001)), color (Koubaroulis *et al.* (2002); Chang and J.Krumm (1999)), parts organization (Schneiderman and Kanade (1998)) among others. All these representations have strong points and weaknesses making them appropriate for different tasks. One technique that has shown successful results consists in the use of collections of local descriptors computed

at interest points. This technique has been used in the past few years to perform recognition tasks such as image retrieval (Schaffalitzky and Zisserman (2002); Tuytelaars and Gool (1999)) and location identification (Mikolajczy and Schmid (2004)). Through the introduction of more geometric and scale invariance they have later been adapted for object recognition (Lowe (1999); Mikolajczy and Schmid (2004)). As we seen in Chapter 3, in this approach partial occlusion of an object is handled as far as enough detected locations are left un-occluded so that a positive match is possible. These methods are invariant to viewing conditions like pose, lighting and scale.

In this section, we present the methodology and results of experiments performed early in the research line which lead to this thesis. We address the task of object recognition using object models based on local interest point detectors/descriptors and learned from one single training image. We study the recognition performance of these models with respect to view angle and image resolution variations. In the studied approach the object is modeled as the set of local interest point locations $\mathbf{x}_j$ and the associated descriptors f_j, automatically extracted on the single training image. Given a new image I, the same representation is extracted and matched against each object model. The recognized object is the one whose model has the highest number of matches. The main issue is the matching process, which is not easy and is normally based only on local greyscale information (Lowe (2004); Mikolajczy and Schmid (2004); Schaffalitzky and Zisserman (2002)), that may be ambiguous. However, color is a powerful cue for recognizing man-made objects. To improve the results obtained using only grey-scale information we will introduce local interest descriptors based on color.

7.1.1 Object modeling

Interest Point Detectors and Neighborhood Definition

As we seen before in Chapter 3, local descriptor methods rely on the automatic detection of specific image locations $\mathbf{x}$ surrounded by specific image areas $\mathcal{A}(\mathbf{x})$. The specification and extraction of both the locations and areas must be reproducible, that is, invariant under geometric and photometric transformations. If this is the case, areas around the same object point detected in different images will always "cover" the same 3D content, as illustrated in Figure 7.1.

In the context of the one shot learning experiments we exploited the Harris-Affine detector (Mikolajczyk and Schmid (2002)). This choice was motivated by analysis that have shown that this detector is the most repeatable and stable in the presence of strong geometric and photometric transforms (Mikolajczyk and Schmid (2003)). It is important for this application that our local interest points are as invariant as possible, since we are learning the object model from only one image.

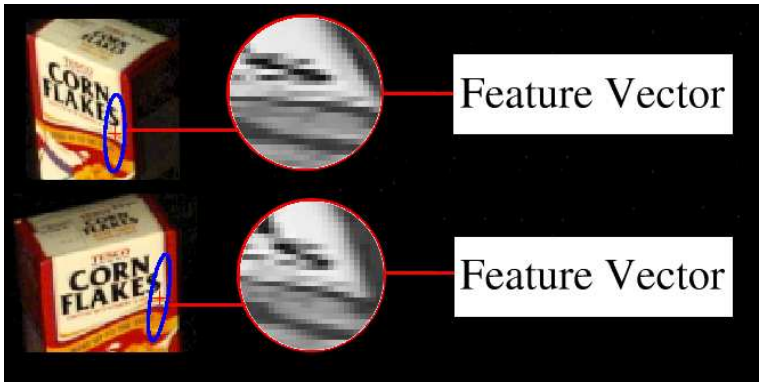


Figure 7.1. Illustration of the stability of the local interest area across image viewing angles.

Structural Features

We used steerable filter introduced by Freeman and Adelson (1991), as our structural information descriptors. These descriptors were found to be a good compromise between robustness and dimensionality (Mikolajczyk and Schmid (2003)). Steerable filters are a class of filters in which a filter of arbitrary orientation is synthesized as a linear combination of a set of basis filters (Freeman and Adelson (1991)).

To be used as local structural descriptors, steerable filters are applied to the local area of the image extracted in the previous step. The responses at a given number of orientations are combined into a set of differential invariants with respect to rotation and illumination, as described in Chapter 3.

The choice of this descriptor over more specific ones, like the SIFT descriptor, was motivated by the extra rotation and illumination invariance obtained by using this descriptor. In the following, we will denote by f_j the structural descriptor extracted around the interest point $\mathbf{x}_i$.

Color Features

Koubaroulis *et al.* (2002) defined interest locations for their method as the locations whose color density distribution exhibit several modes. In their implementation local modes were extracted using a mean-shift algorithm. This method is not invariant to scale changes since the width of the kernel in the mean-shift algorithm was set a priori. In our case, we rely on the extracted points $\mathbf{x}$ and the associated area $\mathcal{A}$, and we assume that these areas contain at least two modes in the color density function distribution. This happens to be a reasonable assumption since our local interest points are corners, which often correspond to color change, in particular on man-made objects. The color modes are collected using K-means clustering in RGB space. Several experiments that we have conducted on a training set of the SOIL-47A database, have shown that estimating reliably the number of modes is difficult, and that the

majority of neighborhoods have a bi-modal color content. Thus in the following we assume that each interest point neighborhood has exactly 2 color modes.

Color descriptor

We must now use the RGB modes to define the local color descriptors, in an way which is invariant to possible changes of geometric and lighting conditions. It is well known that these two factors, description and invariance, oppose each other, since increasing invariance results in information loss (Jan-Mark and Boomgaard (2001)). In a controlled environment the (R, G, B) color values would be the most effective feature. This is however not the case in the presence of illumination changes.

We thus opted for an affine invariant illumination model, where we assume that local illumination changes are similar for each mode in the local area. This is a reasonable assumptions in most applications. We adopted the model proposed by Koubaroulis *et al.* (2002). This model makes use of a chromatic color representation, referred to as rg space:

$$r_m\left(R_m, G_m, B_m\right) = \frac{R_m}{R_m + G_m + B_m}, \tag{7.1}$$

$$g_m\left(R_m, G_m, B_m\right) = \frac{G_m}{R_m + G_m + B_m} \tag{7.2}$$

where $m = 1, 2$ denotes the color mode index

For each mode we compute these features and combined them with the intensity ratio between modes to obtain the local color descriptor c. This ratio is invariant since we assume that both modes undergo the same multiplicative illumination changes. Thus, for an interest point $\mathbf{x}_i$, the complete color descriptor is given by:

$$c_i = \left[r_1^i, g_1^i, V_{12}^i, r_2^i, g_2^i\right), \text{ with } V_{12}^i = \left(\frac{R_1^i + G_1^i + B_1^i}{R_2^i + G_2^i + B_2^i}\right) \tag{7.3}$$

Object model

Given an initial image I of an object o, we build the object representation by extracting the interest points of that image, and compute the structural and color descriptors around those points. The object is thus represented by $\mathcal{R}_o = (\mathbf{x}_i^o, t_i^o), i = 1 \ldots N_o$, where $t_i^o = [f_i^T, c_i^T]$

7.1.2 Object recognition

The goal here is to recognize the object contained in an image, and is is assumed that the object occupies the main part of the image. Thus the problem is the following:

Given a query image characterized by its set of interest points and features $R_q = (\mathbf{x}_i^q, t_i^q)_{i=1,...,N_q}$, find the object model o, characterized by its list of features $R_o = (\mathbf{x}_i^o, t_i^o)_{i=1,...,N_o}$, which has the greatest similarity $S(q, o)$ with the query image.

The similarity $S(q, o)$ that we use is defined by the number of descriptors of the query R_q that have a match in the object model R_o. The similarity is computed in two steps, that we now describe

1- Matches are gathered based on a distance between the descriptors. The object feature t_j^o corresponding to a query feature t_i^q is the object feature that has the closest distance to the query feature, as far as these corresponding features are not to distant. More formally:

$$\begin{array}{l} j^i = \arg\min_{j=1,...,N_o} \left(d^2 \left(t_i^q, t_j^o \right) \right) \\ \\ \left\{ \begin{array}{l} Match\left(t_i^q\right) = t_{j^i}^o,\ if\ d^2\left(t_i^q, t_j^o\right) < T \\ Match\left(t_i^q\right) = 0\ otherwise \end{array} \right. \end{array} \tag{7.4}$$

where $d^2(t_i^q, t_i^o)$ is a feature distance that we will describe later.

Thus, at the end of this step, we are left with a set of pairs $((\mathbf{x}_i^q, t_i^q), (\mathbf{x}_{j^i}^o, t_{j^i}^o), i \in M(q, o))$, where $M(q, o)$ denotes the set of features of R_q which have a match in R_o.

2- Validation:geometric consensus

It is easy to notice that the simple counting of feature matches can be dominated by false correspondences that must somehow be pruned. This is done here, as proposed by most state-of-the-art methods, by the use of geometric model constraints (Hartley and Zisserman (2000)). A set of correspondences is validated if there exists a valid geometric transformation (homography) between the locations of the points in the query image and their matches in the model image. This solution makes use of the assumption that the object is rigid and planar. Although very effective is is a very computer intensive approach.

Distance computation:

The matching step of the algorithm relies on a feature distance. In our case, a feature t_i comprises the structural components f_i and the color components c_i.

While the fusion of descriptors can be made in multiple ways and at several stages of a classi-

fication process (Kittler *et al.* (1998)), given our modeling, the fusion is done through feature concatenation

Accordingly, the distance in the concatenated feature space is then defined as:

$$d_M^2(t_i, t_j) = d_M^2(f_i, f_j) + \alpha d_C^2(c_i, c_j) \tag{7.5}$$

where α is a mixing factor, and d_M and d_C are distance functions defined in the structural and color feature space, respectively. The mixing factor α allows to control the influence of each source on the distance and thus on the final recognition result. The exact value of the mixing parameter α must be trained since its value depends on the unknown importance and reliability of each of the fused features on the definition of an object model. With respect to d_M and d_C we use the following distances:

In the case of structural features, the distance used by state-of-the-art methods is the Mahalanobis distance:

$$d_M^2(f_i, f_j) = [f_i - f_j]^T \Lambda^{-1} [f_i - f_j] \tag{7.6}$$

where Λ is a covariance matrix calculated on a set of training images as explained in (Mikolajczyk and Schmid (2003)).

In the case of the color features used in this work we use the following distance (Koubaroulis *et al.* (2002)):

$$d_C^2(c_a, c_b) = \min\{||c_a - c_b||^2, ||c_a - c_{b'}||^2\} \tag{7.7}$$

where $c_{b'}$ represents the color feature vector with the order of the indexes (i, j) switched. This is necessary due to possible variations in the order in which the modes are stored.

7.1.3 Results

Setup

The described method was tested on the SOIL-24A database which is a subset of the SOIL-47A database described in Koubaroulis *et al.* (2002). This subset is composed of 24 images of colorful, household objects; see Figure 7.2 for sample images. Object are represented by images of approximately 220x220 pixels at full size. This database was created with the purpose of evaluating the degradation of object recognition methods with respect to the change of viewing angle. In this way we obtain the overall behavior of the system to a possible random positioning

Figure 7.2. Examples of Soil-24A object images at different angles (0 and 45 degrees) and different resolutions (100% and 50%).

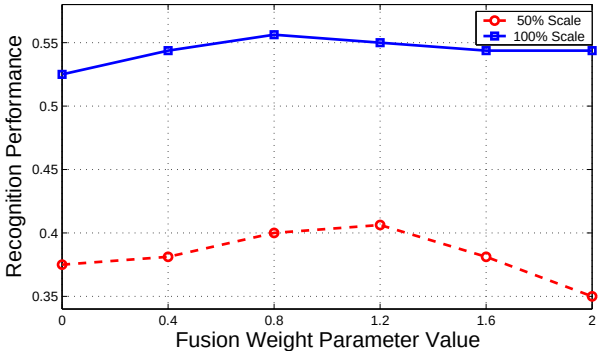
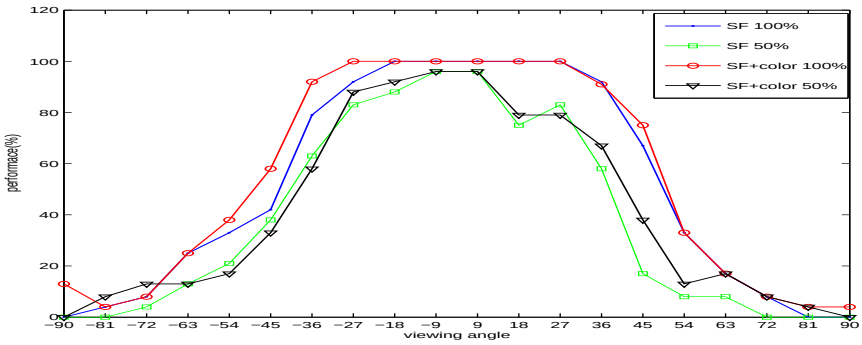


Figure 7.3. Fusion weight training curves for both full and half resolution object representation.

of objects. We have sub-sampled the original database to half resolution since images of indoor scenes will quite probably occupy a smaller image region than in the original database.

For training, images of 4 extra objects not belonging to the SOIL-24A database were selected from the SOIL-47A database. All hyper-parameters were estimated using this training set. In Figure 7.3 we present the training graph for the parameter α that weights the descriptors' fusion. The optimal value that we will use in the testing is 0.8 for full resolution images and 1.2 for half resolution images. This curve shows that color influence is more accentuated in the low resolution case. This is illustrated by a greater relative improvement of the performance at the optimal value in the training set. However, at the same time, the performance drops faster as we



Angle	-90	-81	-72	-63	-54	-45	-36	-27	-18	-9	9	18	27	36	45	54	63	72	81	90	average
SF 100%	0	4	8	25	33	42	79	92	100	100	100	100	100	92	67	33	17	8	0	0	50.0
SF 50%	0	0	4	13	21	38	63	83	88	96	96	75	83	58	17	8	8	0	0	0	37.5
SF + color 100%	13	4	8	25	38	58	92	100	100	100	100	100	100	91	75	33	17	8	4	4	53.5
SF + color 50%	0	8	13	13	17	33	58	88	92	96	96	79	79	67	38	13	17	8	4	0	41.1

Table 7.1. Retrieval performance on the SOIL-24A database, at full and half resolution. (SF stands for Steerable filters).

move away from this value. This phenomenon can be explained by the fact that, the database is known to contain several objects with very similar colors (Koubaroulis *et al.* (2002)). This results also in an unexpected greater confusion in the matching process (Equation 7.4) with the correct object model in the low resolution case. Indeed, as the color contribution to the distance increases, and since color is intrinsically less discriminative than the structural information, the distance of a local feature to the correct match becomes similar to the distance to other features of th object model. Since we only consider the best match, the method becomes sensitive to noise and prone to errors.

Results

Table 7.1 shows the results of the method when applied to the SOIL-24A database. Structural features produce very good results on near frontal angles but start to break down at high angles. In this case, at angles higher than 45 degrees degradation is very high. Unlike previously reported, in our experiments, the use of the structural features did not allow to hold above 60% matching performance up to 60 degrees of viewing angle change (Mikolajczyk and Schmid (2003)). A probable explanation is the higher image resolution of the images used in Mikolajczyk and Schmid (2003), where objects were represented by images of 800x640 pixels size.

When applied to the SOIL-24A, the proposed color fusion scheme produces overall better results than gradient based features alone, giving 7% and 10% of relative improvement in relation to the use of only steerable filters for full and half resolution respectively. At 100% resolution, the addition of color allows to broaden the range of angles over which the performance is good or reasonable, i.e., we observe a performance increase at angles near $36° - 54°$. For the lower

resolution case, the improvement is more spread among angle values, and for same viewing angles, we can notice some deterioration. This can be due to the combined influence of the introduced matching confusion mentioned above and the fact that the weighting parameter was optimized for the training set might have been too high for the test set.

7.2 Object classification using PLSA

In this section we investigate the use of the BOV and latent space representations, introduced in Chapter 4 for object representation and classification. More precisely, we investigate whether PLSA enables to learn consistent patterns of visual co-occurrence and if the learned visual models improve performance when less labeled data are available, and produce a soft clustering of the images in our dataset related to the classes contained in that same dataset. The resulting compact representation retains sufficient discriminative information for accurate object classification, and improves the classification accuracy through the use of unlabeled data when less labeled training data are available. We perform experiments on a 7-class object database containing 1776 images, see Appendix A for details.

For both the BOV and PLSA representation we maintain the choices from Chapter 5. We use the V'_{1000} vocabulary created using K-means from the D^A dataset, see Section 5.6.1. The PLSA aspects we learned on the object database using 60 aspects for the recognition experiments and 20 aspects for ranking (cf. Section 4.5.1).

7.2.1 Image soft clustering

As we seen in the previous chapter, the latent structure learned by PLSA can be illustrated by the top-ranked images in a dataset with respect to the posterior probabilities $P(z_k \mid d_i)$. Figure 7.4 shows a ranking of seven out of 20 aspects identified by PLSA on the 7-class dataset. We selected $N_z = 20$ for a cleaner ranking visualization. From Fig. 7.4, we observe that aspects 3 and 17 seem closely related to face images. The first ten images ranked with respect to aspect 8 are all bike images, while top-ranked images for aspect 10 mostly contain phones. Buildings are present in aspect 5, and all images related to aspect 7 are tree images. Aspect 12 does not seem to be related to any specific object category.

As discussed in the previous chapter, we can analyze the ranking more objectively by using precision and recall curves. The precision and recall curves for the retrieval of faces, cars, bikes, and trees are shown in Fig. 7.5. The top left graph shows that the homogeneous ranking holds on for more than 10 retrieved images in aspect 3 and 17, confirming the observations made on Figure 7.4. We see that another aspect (13) is closely related to face images. The top right graph from Figure 7.5 shows that aspect number 12 is related to car images if looking deeper in the ranking, a fact which is not obvious from the observation of Figure 7.4. Note however that the precision/recall values are not as high as for the faces case. The bottom left graph confirms

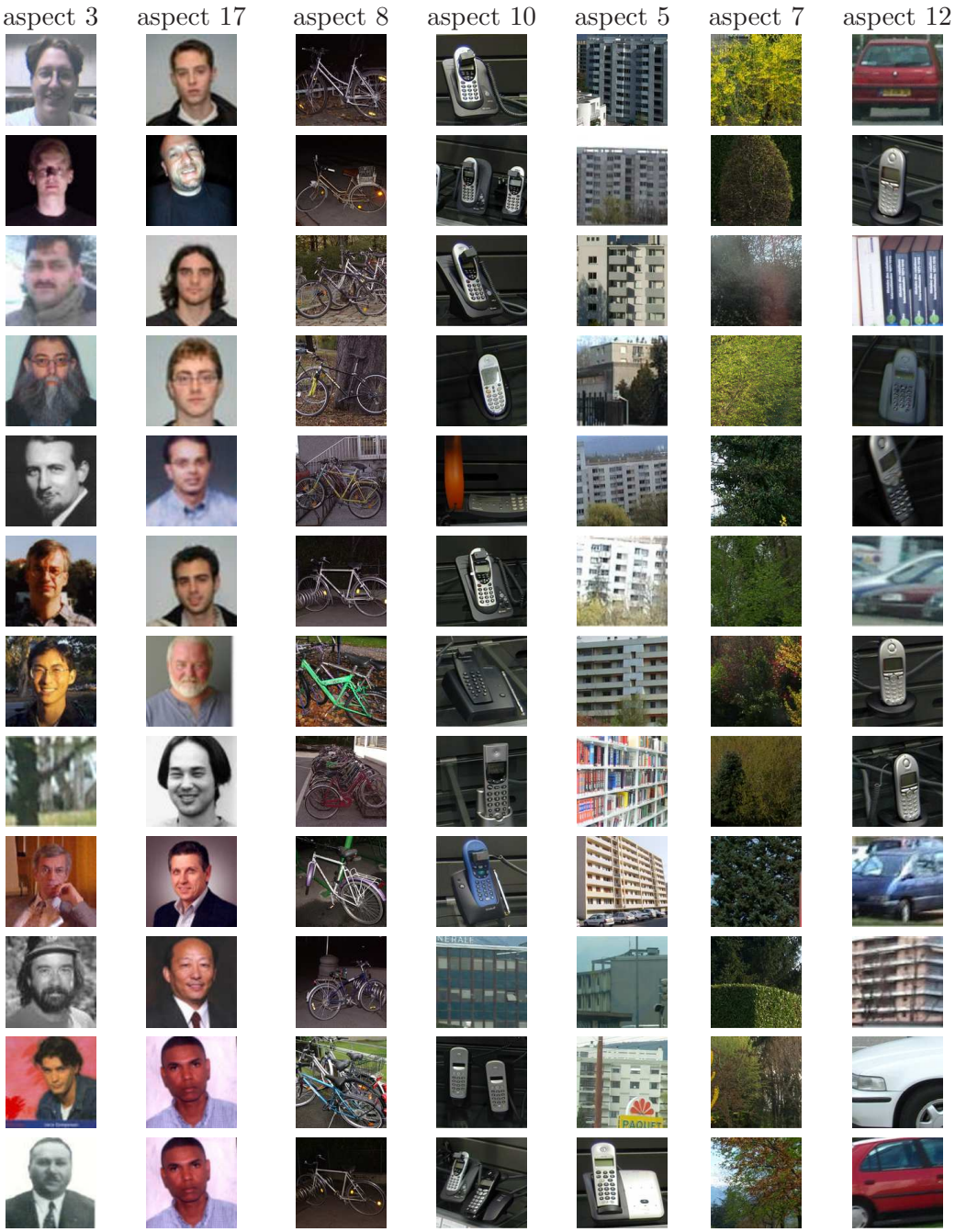


Figure 7.4. Top 10 ranked images with respect to $P(I_i \mid z_k)$ for seven selected aspects. Images are cropped for a convenient display. We observe that aspects 3 and 17 are related to face images, aspect 8 relates to bikes, aspect 5 relates to buildings, aspect 10 relates to phones and that aspect 7 relates to trees.

that aspect 8 is linked to bike images, as well as aspect 1 even if less obvious. The bottom right graph shows that top-ranked images with respect to aspect 7 are mainly tree images. These results confirm that PLSA can capture class-related information in an unsupervised manner.

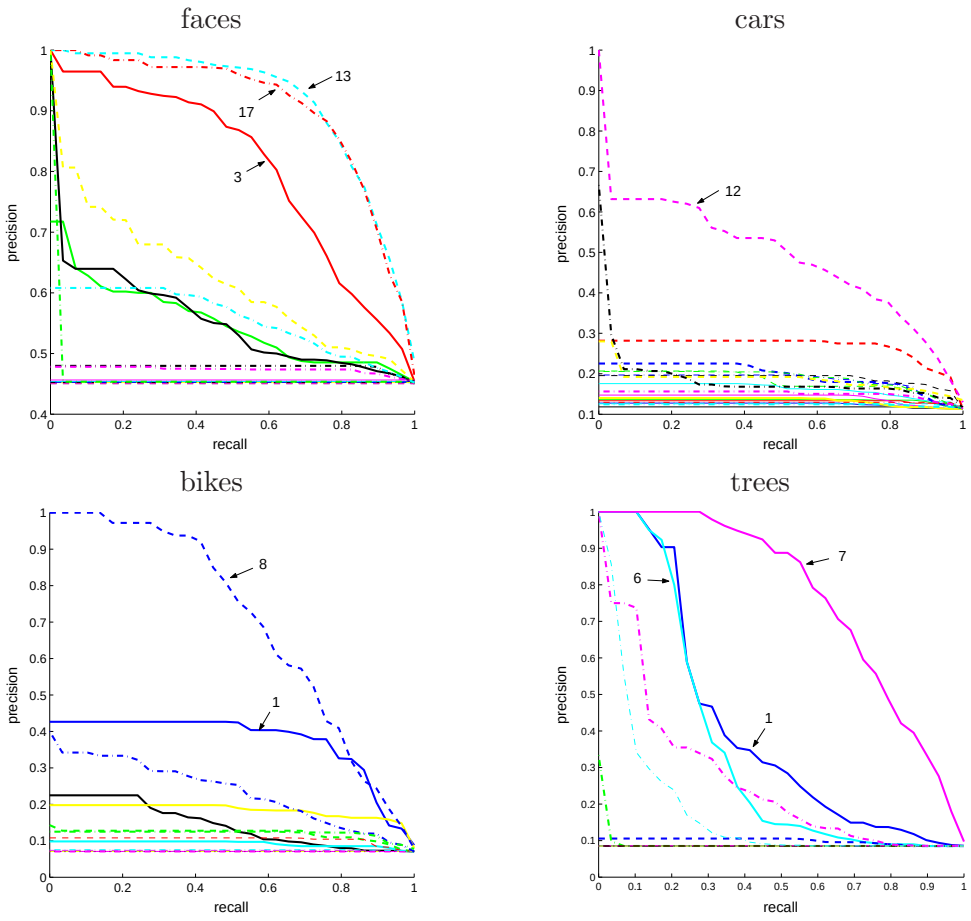


Figure 7.5. Precision and recall curves for the 'face', 'car', 'bike' and 'tree' categories, according to an aspect-based unsupervised image ranking. The lowest precision values on the graph correspond to the class prior distribution.

7.2.2 Images as mixtures of aspects

Our PLSA modeling explicitly considers an image as a mixture of latent aspects expressed by the $P(z \mid d)$ distribution learned from PLSA. The same latent structure with $N_z = 20$ aspects used for the aspect-based image ranking is considered here. As illustrated by the aspect-based image ranking from Figure 7.4, some identified aspects relate to specific object categories. Within the dataset, different examples of aspect mixtures can be observed. In Figure 7.6 (a) the aspect distribution is mainly concentrated on the aspect related to 'building' images. The image only contains building structures, therefore the aspect distribution seems coherent. On the contrary,

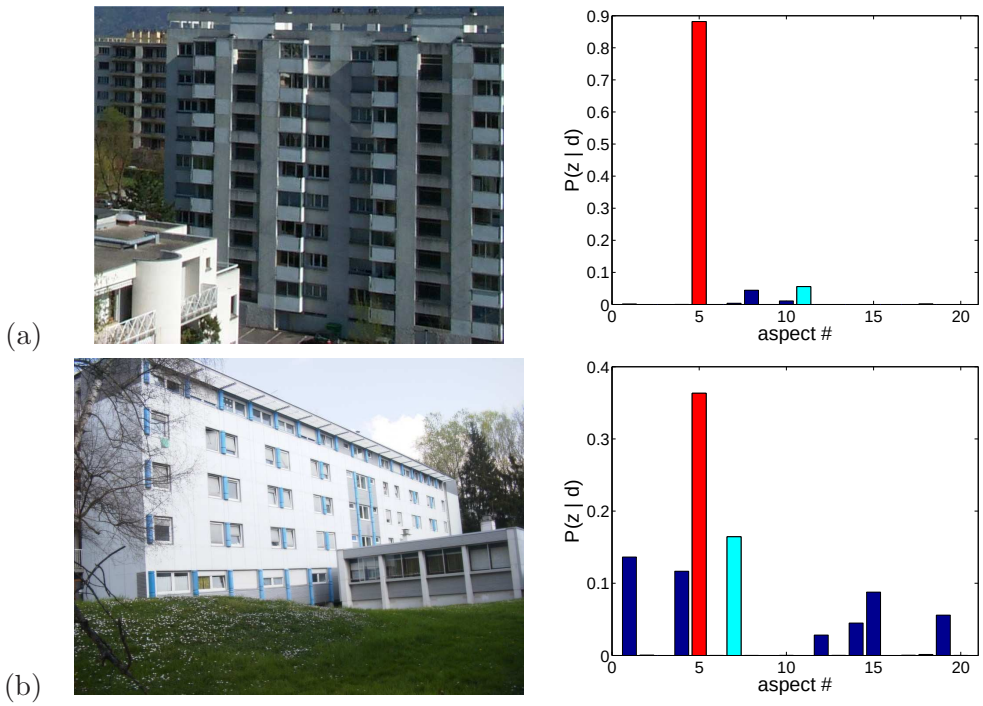


Figure 7.6. Images and their corresponding aspect distribution $P(z \mid d)$ for $N_z = 20$: (a) is concentrated on aspect 5 (building), while (b) is a mixture of aspects 5 (building), 7 (tree) and 1.

the image from Figure 7.6 (b) is composed of both 'building' and 'tree'-related structures. The corresponding aspect distribution interestingly reflects this image composition with the most probable aspects related to 'building' and 'tree'.

It is important to point out that there are cases where the aspect distribution does not clearly correspond to the image semantic. Figure 7.7 (a) shows the close-up of a bike, but the aspect distribution is not concentrated on aspect 8, previously related to 'bike' images. The aspect distribution $P(z \mid d)$ rather describes the image as a mixture of several aspects with no specific dominance. This ambiguous aspect representation could derive from the fact that only a few examples of this type of close-up appear in the database. In Fig. 7.7 (b), the image is identified as a mixture of aspect 8 and 7, which perfectly reflects the image composition. Bikes are located in the image on a tree/vegetation background.

7.2.3 Classification results

Here we propose to compare the aspect-based PLSA and the BOV representations on the 7-class supervised classification task. To evaluate the quality of the feature extraction, we compare the classification based on the BOV representation with the aspect-based representation with the same classification setup as presented in Chapter 5 (multi-class Gaussian Kernel SVM, with one

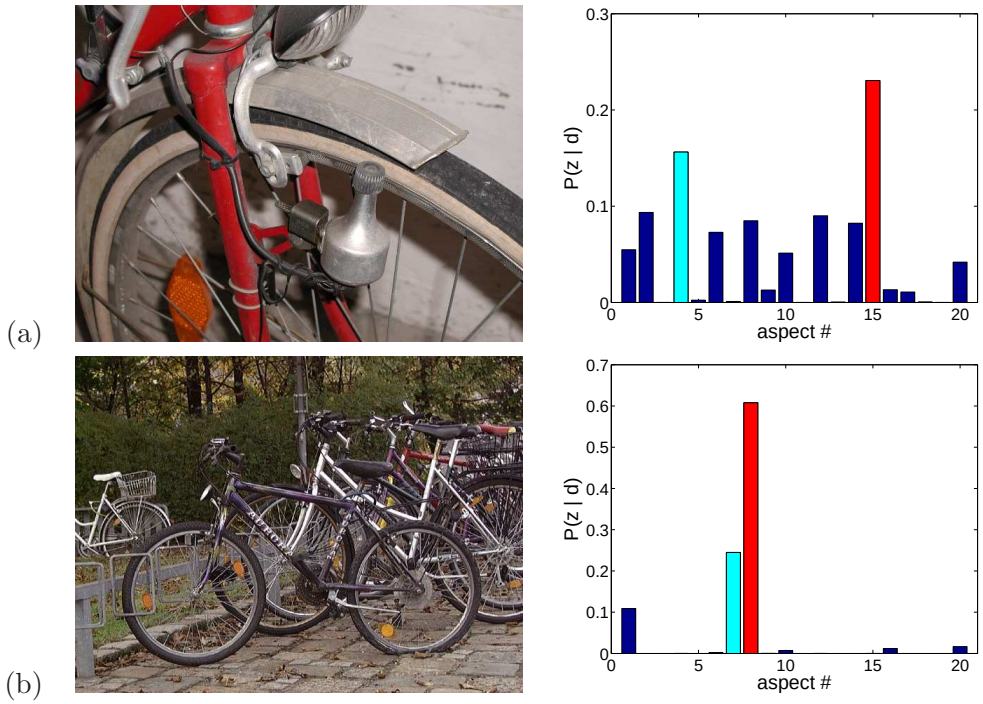


Figure 7.7. Images and their corresponding aspect distribution $P(z \mid d)$ for $N_z = 20$. (a) is a mixture of different aspects, (b) is a mixture of aspect 8 (bikes) and 7 (trees).

	faces	buildings	trees	phones	cars	bikes	books	error
faces	772	2	7	3	3	2	3	2.5(0.04)
buildings	6	100	6	5	12	5	16	33.3(1.70)
trees	1	3	141	1	3	1	0	6.0(0.60)
phones	14	0	0	187	6	2	7	13.4(1.20)
cars	18	1	2	12	162	3	3	19.4(1.46)
bikes	0	3	3	1	2	116	0	7.2(0.38)
books	13	8	0	9	9	1	102	28.2(1.86)

Table 7.2. Confusion matrix for the 7-class object classification problem using the bag-of-visterms features, summed over 10 runs, and average classification error with the variance over ten runs in brackets.

SVM per class trained one against the other). In particular, the PLSA model, with $N_z = 60$, is trained on all non-test images of a given split and the resulting model is used to extract the aspect-based representation on the test images.

Table 7.2 and Table 7.3 show the confusion matrix for the BOV and the PLSA-based classification with $N_z = 60$ aspects. The last column is the per class error. We see that the classification performance greatly depends on the object class for both the BOV and the PLSA representations. These differences are caused by diverse factors. For instance 'trees' is a well defined class

	faces	buildings	trees	phones	cars	bikes	books	error
faces	772	2	5	1	10	1	1	2.5(0.02)
buildings	2	113	3	3	18	5	6	24.6(1.40)
trees	3	3	140	0	2	2	0	6.7(0.40)
phones	9	5	0	166	23	2	11	23.1(0.60)
cars	14	5	0	3	172	4	3	14.4(0.67)
bikes	0	3	4	0	4	113	1	9.6(0.69)
books	7	13	0	6	14	0	102	28.2(1.54)

Table 7.3. Confusion matrix for the 7-class object classification problem using PLSA with $N_z = 60$ aspects as a feature extraction process, summed over 10 runs, and average classification error with the variance over ten runs indicated in brackets.

that is dominated by high frequency texture visterms, and therefore does not get confused with other classes. Similarly, most 'face' images have an homogeneous background and consistent layout which will not create ambiguities with other classes in the BOV representation. This explains the good performance of these two categories.

On the contrary, 'car' images present a large variability in appearance within the database. Front, side and rear car views on different types of background can be found, which makes it a highly complex category for object classification, generating an important confusion with other classes. 'Phones', 'books' and 'buildings' are therefore confused with 'cars' in both the BOV and the PLSA case. The 'bike' class is well classified despite a variability in appearance comparable to the 'car' images, because the bike structure generates a discriminative BOV representation.

Method	90%	50%	10%	5%
PLSA ($N_z = 60$)	11.1(1.6)	12.5(1.5)	18.1(2.7)	21.7(1.7)
BOV	11.1(2.0)	13.5(2.0)	21.8(3.6)	26.7(2.8)

Table 7.4. Comparison between the bag-of-visterms (BOV) and the PLSA-based representation (PLSA) for classification with an SVM classifier trained with progressively less training data on the 7-class problem. The number in brackets is the variance over the different data splits.

Table 7.4 summarizes the whole set of experiments when we gradually train the SVM classifiers with less training data. When using all the training data (90% of all data) for PLSA learning and classifier training, BOV and PLSA achieve a similar total error score. This proves that while achieving a dimensionality reduction from 1000 visterms to $N_z = 60$ aspects, PLSA keeps sufficient discriminative information for the classification task.

The case in which PLSA is trained on all the training data, while the SVMs are trained on a reduced data portion of it corresponds to a partially labeled data problem. Since PLSA is completely unsupervised, it can take advantage of any unlabeled data and build an aspect-based representation from it. This advantage with respect to supervised strategies is shown in Table 7.4 for 50%, 10% and 5% SVM training data. Here the comparison between BOV and PLSA is done for the same reduced number of labeled images to train the SVM classifiers, while the PLSA

model is still trained on the full 90% training data. The total classification errors show that the features extracted by PLSA outperform the raw BOV representations for the same amount of labeled data. Note also that the variance over the splits is smaller for PLSA, which suggests that the model is more stable than BOV given the reduced dimensionality.

7.3 Chapter Conclusion

In this chapter we tested the use of local interest descriptors to model household objects, from one single image. We tested this representation in the task of object recognition in the presence of varying viewing angle and reduced resolution, with good results. To improve the recognition performance we proposed to use color local descriptors in a fusion framework to aid in the recognition task. The inclusion of color lead to an increase of performance for large view angle changes. Nevertheless, the overall performance increase was not as high as expected. Even though the inclusion of color was able to substantially increase the performance at high viewing angles, it also decreased the performance at near frontal views. This is a result from the noise that is present in the color local features, which create confusion althrough the range of viewing angles.

Using the BOV and PLSA approaches presented in the previous chapter we have also tested those representation for the task of object classification. We showed that, like in the case of scene images classification, using PLSA on a bag-of-visterms representation (BOV) produces a compact, discriminative representation of the data, outperforming the standard BOV approach in the case of small amount of training data. Also, we showed that PLSA can capture semantics in the BOV representation allowing for both unsupervised ranking of object images and description of images as a mixture of aspects.

Chapter 8

Conclusion and future directions

IN this chapter we provide a summary of the major contributions and findings of the work presented in this thesis. However, there are some issues that were either supercicially treated, or only mentioned, or simply unaddressed by our work. Accordingly, we describe some of those open areas for future research.

8.1 Summary and contributions

The central theme of this thesis was the modeling of images using local interest point descriptors. The main image representation we proposed is based on the histograms of quantized local interest descriptors, the bag-of-visterm representation (BOV). As an important extension, we studied the use of latent aspect modeling when applied to the BOV representation for image representation (using probabilistic latent semantic analysis, PLSA). Scene classification and segmentation, as well as object recognition were the main tasks and applications addressed in this thesis. Several contributions have resulted from our work:

- Through extensive experiments, we demonstrated that the bag-of-visterms approach is adequate for scene classification, consistently outperforming state-of-the-art methods which rely on a suite of hand-picked features. Furthermore, we have also shown that it is able to handle different data and classes, without any redesign of the features.
- To demonstrate the versatility of the BOV representations, we also applied it to the task of object image classification. Based on the obtained results, we believe that the bag-of-visterms modeling methodology is a very effective approach for solving scene and object classification problems.
- To address the analogy between visterms in images and words in text, we explored the visterm vocabulary co-occurrence properties, and compared them to those of words in

text documents. The results of such comparison showed that on one hand similarly to what occurs in text, there exist a good level of synonymy and polysemy in our visterm vocabulary, but on the other hand, other statistical properties, such as sparsity, are quite different than those encountered in text documents.

- To increase performance in object and scene classification, we proposed several schemes to fuse local color descriptors with, grey-scale based local interest point descriptors. Given the obtained results, we can say that it is possible to construct a visterm representation based on color that improves the standard bag-of-visterm representation.

- Using PLSA, we also showed that a latent aspect modeling provides a compact representation, competitive with the bag-of-visterm representation in terms of performance and results, in general. Importantly, the aspect representation exhibits a more graceful performance degradation with decreasing amount of training data. This result is potentially relevant for the portability and re-usability of future systems, since it allows to reuse a classification system for a new problem using less training data. Similar results and behaviour were obtained in both the task of scene and object image classification

- Finally, we demonstrated that the aspects learned by the latent aspect modeling capture visterm co-occurrences which are semantically meaningfull and relate to the classes of objects and scenes present in the images of the training dataset. This is a property valuable for aspect-based image ranking. Using this property of latent aspect modeling we proposed computational models to perform contextual segmentation of images. These models enable us to exploit a different form of visual context, based on the co-occurrence analysis of visterms in the whole image rather than on the more traditional spatial relationships. We also investigated the use of Markov Random Field models to introduce further spatial coherence into the image segmentation and show that the two types of context models can be integrated successfully.

Given the promising results we obtained using the different proposed image representations based on quantized local descriptors, and the versatility of the proposed model to handle different problems, we believe that these approaches, will be influential in both scene and object classification and will contribute to future progress in related domains. In addition, the results we obtained in ranking and segmentation of scene images using latent aspect modeling, as a local descriptor based image representation, have shown to be a promising research area, not yet fully explored.

8.2 Future research directions

In addition to the contributions presented in this thesis, a number of open questions were raised which suggest further investigation. In the following sections, we describe some directions of future research and possible methodologies which may be used to address these issues.

Vocabulary enhancement

In Chapter 5 we studied the construction of our visterm vocabulary. As we have shown, the choices we make in our vocabulary construction can greatly influence the performance of our system. Different choices of descriptors, detectors, and vocabulary size can change the overall performance of the system. Recently, some author have found that using a regular spatial sampling of the image, either using the greyscale values of the image or with a descriptor, provides good results (Fergus *et al.* (2005a)), which are sometimes better than those obtained using local interest point detectors (Bosch *et al.* (2006)). The choice of descriptors also limits the information we capture from the image, As we seen in some application using color can produce significant improvments. There is overall a lack of variety in the available descriptors, which are mostly texture based descriptors. Local structure defined by lines or edges is normally not properly captured.

The choice of the quantizer is another issue. K-means, although acceptable, is not the most adequate choice for the creation of our visual vocabulary. By using K-means we obtain visterms which represent almost exclusively the few densest regions of our feature space. In this way, K-means fails to capture information about the distribution of data on other, possibly informative, regions of our feature space. A possible path for future research would be to use other methodologies which can extract a vocabulary that is more informative and less noisy, or in other words, in which individual visterms would be more stable in terms of their visual semantics.

Spatial information modeling

The BOV representation does not retain any information about the original spatial position of the local descriptors which gave origin to the visterms. This representation has the advantage of simplicity and compactness. However, the position of patches is an important part of the information contained in the image. The importance of the spatial information of the local interest point descriptors is evident on the results obtained in Chapter 6, where by applying an MRF regularization we improved segmentation results. One possible path of future research is to define fixed regions in the image, possibly a grid, from which we can group our visterms into a neighborhood representation. Another possibility is to use adjacent visterms to define the spatial co-occurrence of visual content. Sivic *et al.* (2005) explore such a principle to create binary relationships, increasing the discriminative power of the obtained visterms.

Latent aspect modeling can, as we have shown, capture the co-occurrence of visterms in an image. It is possible that this modeling could be extended to learn some general spatial co-occurrence patterns in addition to feature co-occurrence. A modification, which may lead to that objective, would be the inclusion of a spatial regularization step in the training of the aspects. This would lead to a more consistent spatial distribution of the aspect in the image, capturing regional co-occurrence characteristics. This is, we believe, a major direction of future

research.

Browsing and retrieval

As mentioned in Chapter 6, the aspects learned by using PLSA modeling exhibits a strong visual relationship with semantic image content. This could be explored for browsing and retrieval of images, without the need for supervised labeling. The co-occurrences capture by latent aspect modeling can be used to relate the content of two different images or even of a part of an image with another (hyperlinking). Using such an approach, we can introduce browsing capabilities into any system without having any knowledge about the image's content. However, due to the unsupervised nature of latent aspect modeling, we cannot be sure of any discriminative power. This means that two images with different representations may actually have similar content. This is shown in Chapter 6 where several aspects model the same semantic content. Improvement in both the way we apply latent aspect modeling to our images, and the applications we create with the resulting models, are both of great importance for future applications dealing with generic image modeling.

Appendix A

Databases

In this appendix we present and discuss the databases used in this work. We present the original source and purpose of the database as presented by the authors. We also explain our usage of each database in the scope of this work. For further details about the experiments performed on each database we give the respective sections where those experiments can be found.

The databases used in our experiments are split into three groups: scene image databases, object image databases, and auxiliary databases. In the following sections we discuss each of those databases.

A.1 Scene Databases

Most of our thesis work is related to systems that deal with scene images. Scene images contain more variance in their structure and content than object images, scenes are also characterized by some overlap between classes. Scenes present in this way a challenge to any categorization system. We use different databases to test different aspects of our systems, like color or the behavior with large number classes.

A.1.1 CARTER Scene Database - D^O

We found that publicly available databases failed to meet our main requirements: large amount of data for each category, mutually exclusive classes and good image quality. Existing databases lacked at least one of those requirements. As such we created a new scene database that we present here.

The COREL image corpus is by far the most used database in tasks ranging from scene image

classification and ranking to more specific problem like location recognition and annotation. Several authors had already used the COREL database to perform several tasks related to scenes: Vailaya *et al.* (2001); Kumar and Herbert (2003b); Vogel and Schiele (2004a). We gathered the part of our scene database that relates to outdoor scenes from the COREL corpus. However the COREL corpus lacks enough indoor images to construct a balanced database. In order to gather extra indoor images we used the Google image search engine, and retrieved images from the Internet based on indoor related query words (room, bedroom, library, office...). A total of 6680 images were collected from COREL And 2777 from the Internet (annotations are available at http://carter.idiap.ch/databases.html). Besides the division between indoor and outdoor these images have other different subclasses. For our experiments we slitted the database into different sets. The database D^O is divided into:

D_1^O **:** this dataset of 6680 images contains a subset of the Corel database, and is composed of 2505 city and 4175 landscape images of 384×256 pixels. This dataset was further divided into sub-classes for some experiments:

- D_{1s}^O **:** this dataset is composed of 1957 city images, taken from a street level.
- D_{1p}^O **:** this dataset is composed of 548 city images taken from a panoramic view. This dataset and the D_{1s}^O dataset are similar, having scale as the main difference between them.
- D_{1m}^O **:** this dataset is composed of 590 images with mountains as the main theme in the scene.
- D_{1f}^O **:** this dataset is composed of 492 images with forest as the main theme in the scene.

D_2^O **:** this set is composed of 2777 indoor images retrieved from the Internet. The size of these images is 384×256 pixels on average. Original images with larger dimensions were resized using bilinear interpolation. The image size in the dataset was kept approximately constant to avoid a potential bias in the BOV representation, since it is known that the number of detected interest points is dependent on the image resolution.

D_3^O **:** this dataset is composed of all images from the datasets D_1^O and D_3^O. The total number of images in this dataset is 9457. This dataset has all the scene classes and sub-classes. This dataset was created to test our scene classification method on a multi-class problem.

D_{3v}^O **:** this is a subset of D_3^O composed of 3805 randomly chosen images. The purpose of this dataset was the creation of a vocabulary that was formed from the same data as the test images (this amount of images equal approx. 1 million data points for the methods presented in this thesis).

D_4^O **:** this is the result of joining the sub-class datasets from D_1^O (D_{1s}^O :, D_{1p}^O, D_{1m}^O, D_{1f}^O) and the indoor dataset D_2^O together in a five-class dataset. The datasets contains a total of 6364 images.

Each dataset part of D_1^O was used for different task in our experiments. We used the dataset D_1^O for binary classification of scenes as city or landscape, and dataset D_3^O for indoor/outdoor scene classification. We also tackled the three-class scene classification problem using D_3^O (indoor/city /landscape classification), and the five-class problem (indoor/city street /city panorama /forest /mountain) with dataset D_4^O is employed in the five-class problem.

A.1.2 Vogel and Schiele (2004b) Database - D^V

This database contains 6 natural scene classes. The images in the database are in color and with resolution of 720×480 pixels. The six classes in the database are: coasts (142), river/lakes (111), forests (103), plains (131), mountains (179), and sky/clouds (34) - where the number in parenthesis is the number of images per class. These images are part of the COREL image library and were manually selected by the authors. This database contains a significant class overlap, which was introduced by the authors to test the human capacity to classify natural scenes. In Fig. A.1 we show some images representing each of the 6 natural classes.

We used this database with two main objectives. The first was to test our systems in a database that contains large overlap between classes, this is a particular challenge in the training of latent space models since co-occurring characteristics from different classes make the modeling more difficult. The second was to test the inclusion of color in the local descriptors from which we create the *bag-of-visterm* representation.

A.1.3 Fei-Fei and Perona (2005) Database - D^F

This database is made of 3859 images in total, collected from other databases and the Internet, see Fei-Fei and Perona (2005) for details. This database is divided into a total of 13 classes. In comparison with most scene databases where the class number is between 2 and 6, 13 is a considerable large number of classes. Although 3859 is not a small amount of images for a scene database, due to the high number of classes, the resulting number of images per class is small (less than 220 in some cases). The database detailed description follows:

D^F: This data set contains a total of 3859 images of resolution 60000 pixels (approx.), varying in exact size and XY ratio. The images are distributed over 13 scene classes as follows (the number in parenthesis indicates the number of images in each class): bedroom (216), coast (360), forest (328), highway (260), inside city (308), kitchen (210), living room (289), mountain (374), open country (410), office (215), street (292), suburb (241), and tall buildings (356).
(available for download at: http://faculty.ece.uiuc.edu/feifeili/datasets.html)

We used this database to test our systems in the case of a higher number of classes. This database has the particularity that some of its classes are characterized mainly by the object in

t

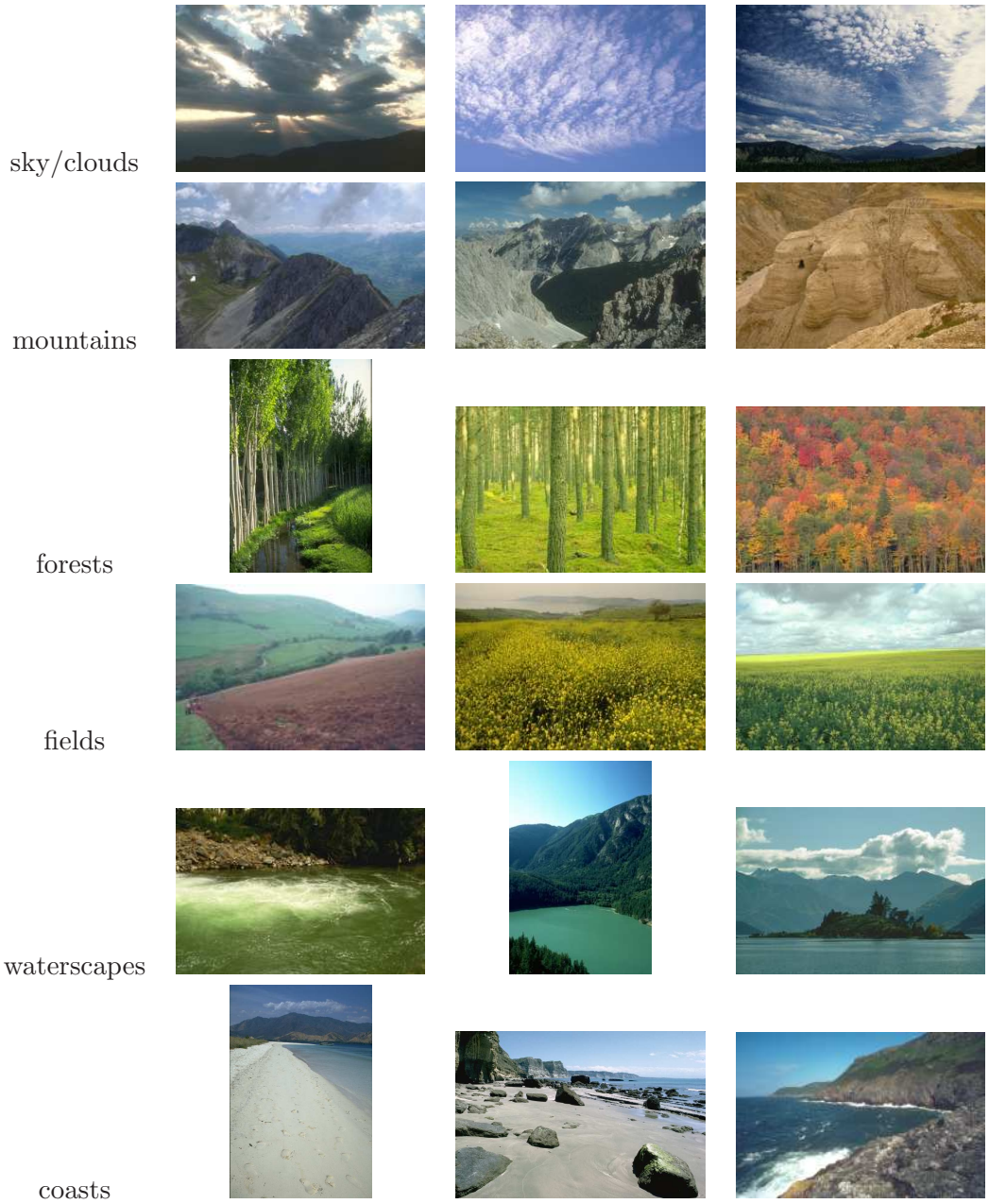


Figure A.1. Some images from the natural scenes in database D^V. From top to bottom: sky/clouds, mountains, forests, fields, waterscapes and coasts.

Figure A.2. Images from Database D^F. Left column from top to bottom: mountain, coast, forest, highway, inside city, and street. Right column from top to bottom: living room, kitchen, suburb, bedroom, office, tall buildings, and open country

Figure A.3. The 24 planar surface object in the SOIL-47A database.

the scene (kitchen, office, bedroom ...). Scene detection in this database can be associated with object recognition to some extent, and in the same way this database can be considered a mix between a scene and object database.

A.2 Object Databases

The databases that follow were design to test object recognition, either from the same class or the same instance of the object. We used these databases for either the test of local point detectors/descriptors or to verify that our system's performance when dealing with object images.

A.2.1 SOIL-24A - D^S

The SOIL-24A database is a subset of the Surrey Object Image Library SOIL-47. The SOIL-47 database was created to study the invariance of object recognition methods with the variation of the point-of-view. The major objective was to evaluate the performance of object recognition methods with respect to a large range of affine/projective transformations, using only a single frontal view to build the object model. The SOIL-47 database is slitted into the SOIL-47A and SOIL-47B datasets. SOIL-47A differs from SOIL-47B for the fact that the latter has illumination variations. The SOIL-24A is the subset of SOIL-47A which is constituted only by box objects.

In Fig. A.3 we can see the frontal image for each object in the SOIL-24A. In Fig. A.4 we show the point-of-view variation that were introduced in the database.

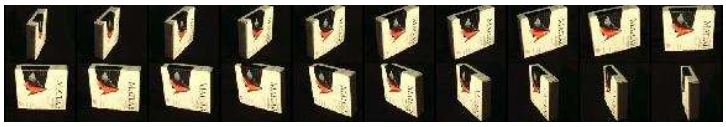


Figure A.4. An object of the SOIL-24A dataset, presented in all the 20 existing variations of the point-of-view.

A.2.2 LAVA 7-class Object Database - D^L

This database was created by Willamowski *et al.* (2004). It was used between all the member of the European Project LAVA to evaluate different bag-of-visterm models in a joint internal challenge. This database contains: faces (792), buildings (150), trees (150), cars (201), phones (216), bikes (125) and books (142), adding up to a total of 1776 images.

The database was created to evaluate the capabilities of generic image recognition systems. It is listed here as an object recognition database since all its classes are based on particular objects, but some of its classes closely relate to scenes.

The size of the images varies considerably on the original data: images can have between 10k and 1,2M pixels while most image sizes are around 100-150k pixels. We resized all images to 100k pixels since the local invariant feature extraction process is dependent of the image size. This ensures that no class-dependent image size information is included in the representation.

A.3 CARTER auxiliary Dataset D^A

In this section we present a group of datasets that were combined to provide data independent to our test datasets so that we could train our vocabularies and latent space models. This database contains a set of images similar to those in our test datasets, but which were not contained in those datasets. This allows us to test the variation in performance of our system between the usage of generic models and those created on the test datasets. This data was never used to test our system in terms of classification performance.

This dataset is a grouping of several different datasets and is the corpus in which we train our generic models. The datasets contain images first gathered for object recognition, local recognition or just random images. The database was created by gathering the following 3805 images from different databases

D^A **:** this dataset is constituted by 3805 images from several sources:

- 1002 building images from the ZuBud D^Z database (Shao *et al.* (2003)),
- 144 images of people and outdoors, part of the Gratz object database D^G (Opelt *et al.* (2004)),

Figure A.5. Example images from each of the 7 classes in the D^L database. From top to bottom: bikes, cars, faces, books, trees, buildings, and phones.

Figure A.6. Images from the Zurich Building Image Database - ZuBud. On the first row we show 3 out of the 5 different view for one building. The second row shows 3 examples of images from different buildings.

- 435 indoor human faces part of the D^L database (Willamowski *et al.* (2004)),
- 490 indoor images from the COREL image library, manually selected,
- 1516 city/landscape overlap images from the COREL image library, manually selected,
- 267 Internet photographic images, collected from *images.google.com*.

The images from the COREL corpus that are integrated into this database were those that contained city/landscape overlap and the few ones that we found from indoor scenes, examples of these images can be seen in Fig. . The are non-overlapping with database D_1^O. The Internet retrieved images are those of outdoor scenes and do not overlap with the images from the indoor scenes in D_2^O We gathered this number of images since it provides approx. 1 million data points in most our experiments.

A.3.1 Zurich Building Image Database (ZuBud) - D^Z

The ZuBud image database was created to test the recognition of specific location within a city, given images from those locations. The database contains 1005 images of Zurich city building. In this database there are 201 building each of which has 5 different viewing images. These different images are taken from different viewing angles and distances to the buildings. More details about the database can be found in Shao *et al.* (2003). Some images from this database can be seen in Fig. A.6.

We used this database as a source of city images, to contribute to the set of images that contain buildings but are not from COREL.

A.3.2 Gratz People and Bikes - D^G

This database was originally created to test detection of people (pedestrians) and bikes in images. The database is split into four different classes, depending on the contents of the images: bikes, persons,bikes and persons, and background images with no bikes and not persons. Typical images representing the classes in this database are shown in Fig. A.7.

We include this database in D^A to obtain more images of city streets (with or without people). The images of bikes are not used since they are too specific and normally contain closeups that are not representative of any scene.

A.3.3 Internet Images - D^I

All INTERNET images collected for the experiments in this thesis were obtained using *http://images.google.com* search engine. Using keywords that related to outdoor scenes,(mountain, forest, lake, city...), we obtained outdoor related images. The results of our image searches were very noisy, as such we manually selected the images that more correlated with the keywords and obtained 267 filtered images that we used as our auxiliary Internet database D^I.

This database is used to provide some more diversity to our model construction. Since it has no annotation, this database is not usable for any experimental results.

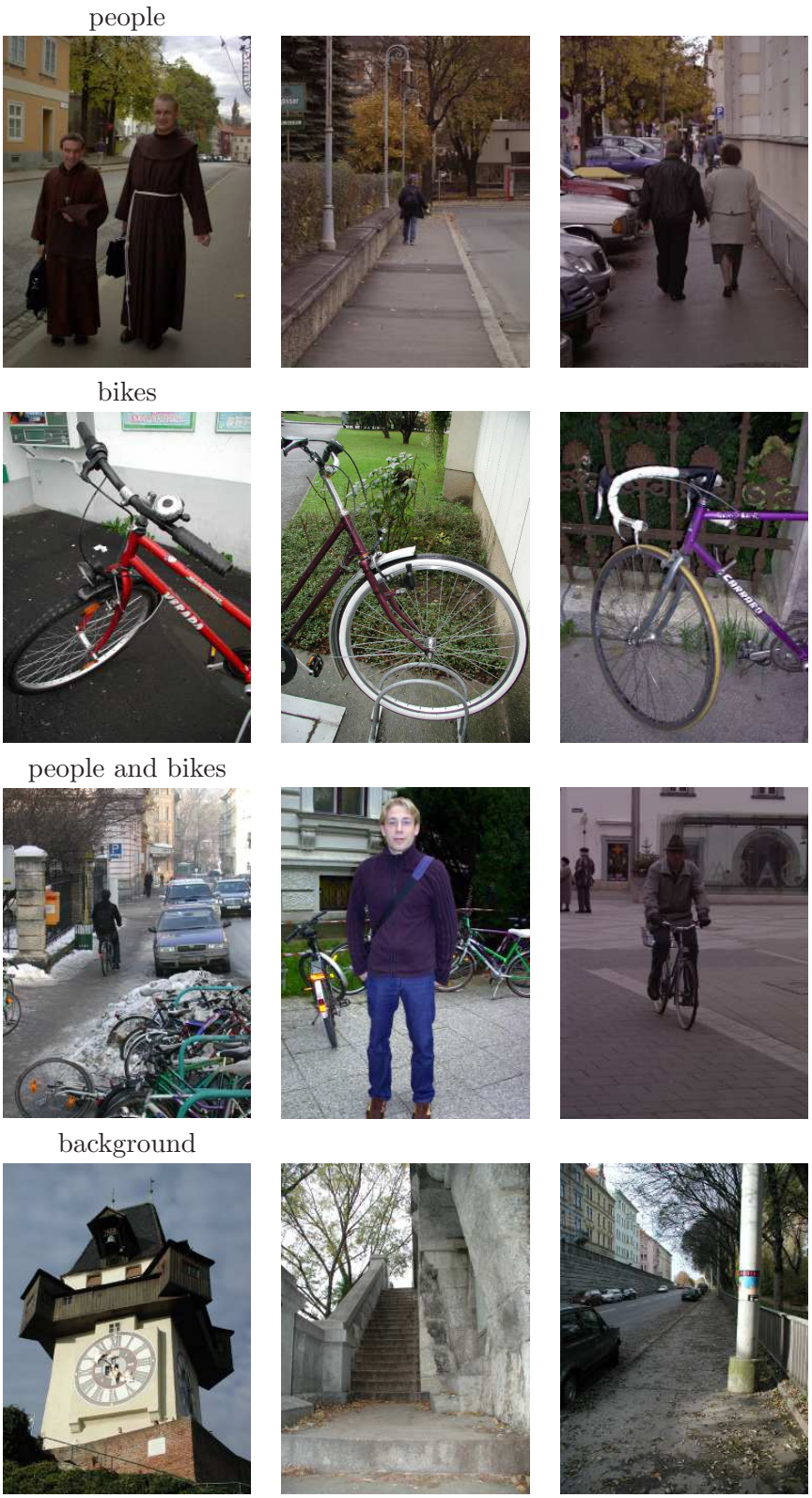


Figure A.7. Examples from the Database of persons and bikes from Gratz D^G. From top to bottom: images of people, bikes, images containing both people and bikes, and background images that contain neither.

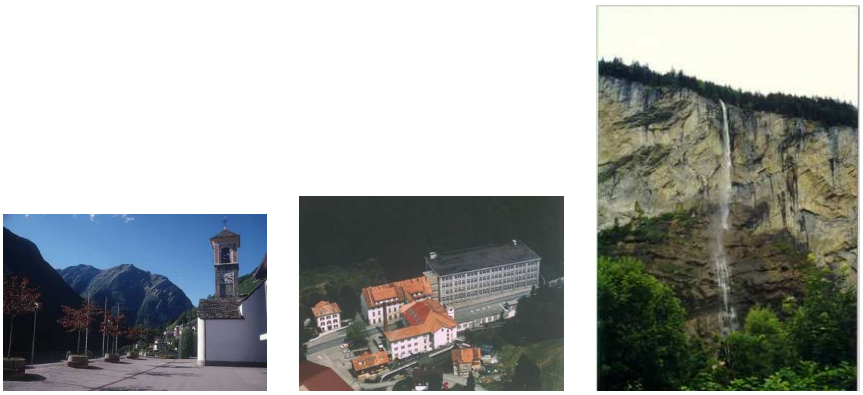


Figure A.8. Example images from those retrieved from the *http://images.google.com* search engine. These images represent generic scenes and have no particular annotation.

Bibliography

Agarwal, S. and Roth, D. (2002). Learning a sparse representation for object detection. In *In Proceedings of the European Conference on Computer Vision*, pages 113–130.

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. ACM Press.

Barecke, T., Kijak, E., Nurnberger, A., and Detyniecki, M. (2006). Video navigation based on self-organizing maps. In *In Proceedings of Conference in Image and Video Retrieval (CIVR)*, pages 340–349.

Barnard, K., Duygulu, P., Freitas, N., Forsyth, D., Blei, D., and Jordan, M. (2003). Matching words and pictures. *Journal of Machine Learning Research*, **3**, 1107–1135.

Baumberg, A. (2000). Reliable feature matching across widely separated views. In *In Proceedings of IEEE Conference in Computer Vision and Pattern Recognition (CVPR)*, pages 774–781.

Biederman, I. (1987). Recognition by components: A theory of human image understanding. *Psychological Review*, **94**, 115–147.

Bimbo, A. D. and Pala, P. (1997). Visual image retrieval by elastic matching of user sketches. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**(2), 121–132.

Bishop, C. (1995). *Neural Networks for Pattern Recognition*. Oxford University.

Blake, A. and Isard, M. (1998). *Active Contours*. Springer.

Blei, D. and Jordan, M. (2003). Modeling Annotated Data. In *In Proceedings of International ACM SIGIR Conference*, Toronto.

Blei, D., Andrew, Y., and Jordan, M. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, **3**, 993–1020.

Bosch, A., Zisserman, A., and Munoz, X. (2006). Scene classification via PLSA. In *In Proceedings of the European Conference on Computer Vision (ECCV)*, Graz, Austria.

Boutell, M., Luo, J., Shen, X., and C.M.Brown (2004). Learning multi-label scene classification. *Pattern Recognition*, **37**(9), 1757–1771.

Buntine, W. (2002). Variational extensions to EM and multinomial PCA. In *In Proceedings of the European Conference on Machine Learning*, Helsinki.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, **2**(2), 121–167.

Carson, C., Thomas, M., Belongie, S., Hellerstein, J. M., and Malik, J. (1999). Blobworld: A system for region-based image indexing and retrieval. In *In Proceedings of the International Conference on Visual Information Systems*.

Chang, P. and J.Krumm (1999). Object recognition with color cooccurence histograms. In *In Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Cootes, T., Cooper, D., Taylor, C., and Graham, J. (1992). A Trainable Method of Parametric Shape Description. *Image and Vision Computing*, **10**(5), 289–294.

Cootes, T. F., Edwards, G. J., and Taylor, C. J. (1998). Active appearance models. *Lecture Notes in Computer Science*, **1407**.

Dorko, G. and Schmid, C. (2003). Selection of scale invariant parts for object class recognition. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, Nice.

Dornaika, F. and Davoine, F. (2004). Head and facial animation tracking using appearance-adaptive models and particle filters. In *In Proceedings of IEEE Conference in Computer Vision and Pattern Recognition (CVPR)*.

Duygulu, P., Barnard, K., de Fretias, N., and Forsyth, D. (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *In Proceedings of the European Conference on Computer Vision (ECCV)*.

Fauqueur, J. and Boujemaa, N. (2003). New image retrieval paradigm: logical composition of region categories. In *In Proceedings of IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain.

Fei-Fei, L. and Perona, P. (2005). A Bayesian hierarchical model for learning natural scene categories. In *In Proceedings of IEEE Conference in Computer Vision and Pattern Recognition (CVPR)*, San Diego.

Fei-Fei, L., Fergus, R., and Perona, P. (2003). A bayesian approach to unsupervised one-shot learning of object categories. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, Nice.

Fei-Fei, L., Fergus, R., and Perona, P. (2004). Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, Washington DC.

Fergus, R., Perona, P., and Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. In *In Proceedings of IEEE Conference in Computer Vision and Pattern Recognition (CVPR)*, Toronto.

Fergus, R., Fei-Fei, L., Perona, P., and Zisserman, A. (2005a). Learning object categories from google's image search. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, Beijing.

Fergus, R., Perona, P., and Zisserman, A. (2005b). A sparse object category model for efficient learning and exhaustive recognition. In *In Proceedings of IEEE Conference in Computer Vision and Pattern Recognition (CVPR)*, San Diego.

Freeman, W. T. and Adelson, E. H. (1991). The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**(9), 891–906.

Gatica-Perez, D., Zhou, Z., Sun, M.-T., and Hsu, V. (2002). Video object hyper-lins for streaming applications. In *Proceedings of VISUAL Conference*.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6**, 721–741.

Gorkani, M. and Picard, R. (1994). Texture orientation for sorting photos at glance. In *In Proceedings of the International Conference on Pattern Recognition (ICPR)*, Jerusalem.

Grauman, K. and Darrell, T. (2005). The pyramid match kernel: Discriminative classification with sets of image features. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*.

Harris, C. and Stephens, M. (1998). A combined corner and edge detector. In *In Proceedings of the Alvey Vision Conference*, pages 147–151.

Hartley, R. and Zisserman, A. (2000). *Multiple view geometry in computer visiond*. Cambridge University Press, New York, NY, USA.

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, **42**, 177–196.

H.Yu and W.Wolf (1995). Scenic classification methods for image and video databases. In *In Proceedings of the SPIE International Conference on Digital Image Storage*, volume 2606, pages 363–371.

Jan-Mark, G. and Boomgaard, R. (2001). Color invariance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**(12), 1338–1349.

Jeon, J., Lavrenko, V., and Manmatha, R. (2003). Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of ACM SIGIR International Symposium on Information Retrieval*, Toronto.

Jurie, F. and Triggs, B. (2005). Creating efficient codebooks for visual recognition. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 604–610.

Kadir, T. and Brady, M. (2001). Scale, saliency and image description. *International Journal of Computer Vision*, **45**(2), 83–105.

Kadir, T., Zisserman, A., and Brady, M. (2004). An affine invariant salient region detector. In *In Proceedings of the European Conference on Computer Vision*.

Kass, M., Witkin, A., and Terzopoulos, D. (1987). Snakes: Active Contour Models. *International Journal of Computer Vision (IJCV)*, **1**(4), 321–331.

Ke, Y. and Sukthankar, R. (2004). PCA-SIFT: A more distinctive representation for local image descriptors. In *In Proceedings of IEEE Conference in Computer Vision and Pattern Recognition (CVPR)*.

Keller, M. and Bengio, S. (2004). Theme topic mixture model: A graphical model for document representation. Technical Report IDIAP-RR-04-05, IDIAP Research Institute, Martigny.

Kittler, J., Hatef, M., Duin, R., and Matas, J. (1998). On Combining Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(3), 226–239.

Koenderink, J. and van Doorn, A. (1987). Representation of local geometry in the visual system. In *Biological Cybernetics*, volume 55, pages 367–375.

Koubaroulis, D., Matas, J., and Kittler, J. (2002). Evaluating colour-based object recognition algorithms using the SOIL-47. In *In Proceedings of the Asian Conference on Computer Vision (ACCV)*, Melbourne, Australia.

Kumar, S. and Herbert, M. (2003a). Discriminative random fields: A discriminative framework for contextual interaction in classification. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, Nice.

Kumar, S. and Herbert, M. (2003b). Man-made structure detection in natural images using a causal multiscale random field. In *In Proceedings of IEEE Conference in Computer Vision and Pattern Recognition (CVPR)*, Toronto.

Lazebnik, S., Schmid, C., and Ponce, J. (2003). Affine-invariant local descriptors and neighborhood statistics for texture recognition. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, Nice.

Leibe, B. and Schiele, B. (2004). Scale invariant object categorization using a scale-adaptive mean-shift search. In *In Proceedings of DAGM'04 Annual Pattern Recognition Symposium*, pages 145–153.

Leibe, B., Mikolajczyk, K., and Schiele, B. (2006). Efficient clustering and matching for object class recognition. In *In Proceedings of British Machine Vision Conference (BMVC)*.

Li, J. and Wang, J. Z. (2006). Real-time computerized annotation of pictures. In *Proceedings of the ACM International Conference on Multimedia*.

Li, S. Z. (1995). *Markov Random Field Modeling in Computer Vision*. Springer.

Lim, J.-H. and Jin, J. (2004). Semantics discovery for image indexing. In *European Conference on Computer Vision ECCV'04*, Prague, Czech Republic.

Lindeberg, T. (1994). *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers.

Lindeberg, T. (1998). Feature detection with automatic scale selection. *International Journal of Computer Vision*, **30**, 79–116.

Lindeberg, T. and Garding, J. (1997). Shape-adapted smoothing in estimation of 3-D shape cues from affine deformations of local 2-D brightness structure. *In Proceedings of the European Conference on Computer Vision (ECCV)*, **15**, 415–434.

Lowe, D. (1999). Object recognition from local scale-invariant features. In *In Proceedings of the 7th International Conference on Computer Vision*, pages 1150–1157.

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

Malik, J., Belongie, S., Leung, T., and Shi, J. (2001). Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, **43**, 7–27.

Marr, D. (1982). *Vision*. W.H. Freeman.

Matas, J., Koubaroulis, D., and Kittler, J. (2002a). The multi-modal neighborhood signature for modeling object color appearance and applications in object recognition and image retrieval. *Computer Vision and Image Understanding*, **88**, 1–23.

Matas, J., Chum, O., Martin, U., and Pajdla, T. (2002b). Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, Cardiff.

Mikolajczy, K. and Schmid, C. (2004). Scale and affine interest point detectors. *International Journal of Computer Vision*, **60**(1), 63–86.

Mikolajczyk, K. and Schmid, C. (2002). An affine invariant interest point detector. In *In Proceeding of European Conference Computer Vision*.

Mikolajczyk, K. and Schmid, C. (2003). A performance evaluation of local descriptors. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, Toronto.

Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **27**(10), 1615–1630.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Kadir, F. S. T., and Gool, L. (2005). A comparison of affine region detectors. *International Journal of Computer Vision*, **65**(1/2), 43–72.

Mikolajczyk, K., Leibe, B., and Schiele, B. (2006). Multiple object class detection with a generative model. In *Proceedings of IEEE CVPR*, New York.

Mindru, F., Moons, T., and Gool, L. V. (1999). Recognizing color patterns irrespective of viewpoint and illumination. In *In Proceedings of IEEE Conference in Computer Vision and Pattern Recognition (CVPR)*.

Monay, F. and Gatica-Perez, D. (2003). On image auto-annotation with latent space models. In *Proceedings ACM International Conference on Multimedia*, Berkeley.

Monay, F. and Gatica-Perez, D. (2004). PLSA-based image auto-annotation: Constraining the latent space. In *Proceedings of the ACM International Conference on Multimedia*, New York.

Monay, F., Quelhas, P., Gatica-Perez, D., and J.-M.Odobez (2005). Constructing visual models with a latent space approach. Technical Report IDIAP-RR-05-14, IDIAP Research Institute, Martigny.

Monay, F., Quelhas, P., Odobez, J.-M., and Gatica-Perez, D. (2006). Integrating co-occurrence and spatial contexts on patch-based scene segmentation. In *Proceedings of Beyond Patches Workshop, in conjunction with CVPR*.

Mori, Y., Takahashi, H., and Oka, R. (1999). Image-to-word transformation based on dividing and vector quantizing images with words. In *Proceedings of the International Workshop on Multimedia Intelligent Storage and Retrieval Management*, Orlando.

Murase, H. and Nayar, S. (1995). Visual learning and recognition of 3-d objects from appearance. *International Jrounal of Compututer Vision*, **14**(1), 5–24.

Murphy, K., Torralba, A., and Freeman, W. (2003). Using the forest to see the trees: A graphical model relating features, objects and scenes. In *Proceedings of Neural Information Processing Systems*, Vancouver.

Naphade, M. and Huang, T. (2001). A probabilistic framework for semantic video indexing, filtering and retrieval. *IEEE Trans. on Multimedia*, **3**(1), 141–151.

Nene, S., Nayar, S., and Murase, H. (1996). Columbia object image library (coil-100). Technical Report CUCS-006-96, Columbia University, New York.

Niyogi, S. and Freeman, W. T. (1996). Example-based head tracking. In *In Proceedings of International Conference on Automatic Face and Gesture Recognition*.

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

Opelt, A., Fussenegger, M., Pinz, A., and Auer, P. (2004). Weak hypotheses and boosting for generic object detection and recognition. In *Proceedings of IEEE European Conference on Computer Vision*, Prague.

Paek, S. and S.-F., C. (2000). A knowledge engineering approach for image classification based on probabilistic reasoning systems. In *Proceedings of IEEE International Conference on Multimedia and Expo*, New York.

Quelhas, P., Monay, F., Odobez, J.-M., Gatica-Perez, D., Tuytelaars, T., and Gool, L. V. (2005). Modeling scenes with local descriptors and latent aspects. In *In Proceedings of IEEE International Conference on Computer Vision (ICCV)*, Beijing.

Salton, G. and Buckley, C. (1987). Term weighting approaches in automatic text retrieval. Technical report, Cornell University.

Schaffalitzky, F. and Zisserman, A. (2002). Multi-view matching for unordered image sets. In *In Proceedings of the 7th European Conference on Computer Vision, Copenhagen*.

Schmid, C. and Mohr, R. (1997). Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**(5), 530–534.

Schmid, C., Mohr, R., and Bauckhage, C. (2000). Evaluation of interest point detectors. *International Journal of Computer Vision*, **37**(2), 151–172.

Schneiderman, H. and Kanade, T. (1998). Probabilistic modeling of local appearance and spatial relationships for object recognition. In *In Proceedings of IEEE Conference in Computer Vision and Pattern Recognition (CVPR)*, pages 45–51.

Sebe, N. and Lew, M. S. (2003). Comparing salient point detectors. *Pattern Recognition Letters*, **24**, 89–96.

Serrano, N., Savakis, A., and Luo, J. (2002). A computationally efficent approach to indoor/outdoor scene classification. In *International Conference on Pattern Recognition*, Quebec.

Shao, H., Svoboda, T., Ferrari, V., Tuytelaars, T., and Gool, L. V. (2003). Fast indexing for image retrieval based on local appearance with re-ranking. In *Proceedings of IEEE International Conference on Image Processing*, Barcelona.

Shotton, J., Winn, J., Rother, C., and Criminisi, A. (2006). Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European conference in computer vision*.

Sivic, J. and Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *Proceedings of IEEE International Conference on Computer Vision*, Nice.

Sivic, J., Russell, B. C., Efros, A. A., Zisserman, A., and Freeman, W. T. (2005). Discovering objects and their location in image collections. In *Proceedings of IEEE International Conference on Computer Vision*, Beijing.

Smeulders, A., Worring, M., Santini, S., Gupta, A., and Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(12), 1349–1380.

Sporring, J., Nielsen, M., Florack, L., and Johansen, P. (1997). *Gaussian Scale-Space Theory*. Springer-Verlag.

Swain, M. and D.Ballard (1991). Color Indexing. *International Journal of Computer Vision (IJCV)*, **7**, 11–32.

Szummer, M. and Picard, R. (1998). Indoor-outdoor image classification. In *IEEE International CAIVD Workshop caivd (part of ICCV'98)*, Bombay.

Torralba, A., Murphy, K., Freeman, W., and Rubin, M. (2003). Context-based vision system for place and object recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, Nice.

Tuytelaars, T. and Gool, L. V. (1999). Content-based image retrieval based on local affinely invariant regions. In *Proceedings Visual*, Amsterdam.

Tuytelaars, T. and Gool, L. V. (2000). Wide baseline stereo based on local, affinely invariant regions. In *Proceedings of the British Machine Vision Conference*, Bristol.

Vailaya, A., Jain, A., and Zhang, H. (1998). On image classification: City images vs. landscapes. *Pattern Recognition*, **31**, 1921–1935.

Vailaya, A., Figueiredo, M., Jain, A., and Zhang, H. (2001). Image classification for content-based indexing. *IEEE Transactions on Image Processing*, **10**(1), 117–130.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer.

Vincent, L. and Soille, P. (1991). Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**(6), 583–598.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *In Proceedings of IEEE Conference in Computer Vision and Pattern Recognition (CVPR)*.

Vogel, J. and Schiele, B. (2004a). Natural scene retrieval based on a semantic modeling step. In *Proceedings of International Conference on Image and Video Retrieval*, Dublin.

Vogel, J. and Schiele, B. (2004b). A semantic typicality measure for natural scene categorization. In *Pattern Recognition Symposium DAGM'04*, Tubingen, Germany.

Vogel, J. and Schiele, B. (2006). Semantic modeling of natural scenes for content-based image retrieval. *International Journal of Computer Vision, in print*.

Weston, J. and Watkins, C. (1998). Multi-class support vector machines. Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London.

Willamowski, J., Arregui, D., Csurka, G., Dance, C., and Fan, L. (2004). Categorizing nine visual classes using local appearance descriptors. In *Proceedings of LAVS Workshop, in ICPR'04*, Cambridge.

Witkin, A. (1983). Scale-space filtering. In *In Proceedings of the 8th International Joint Conference on Artificial Intelligence*.

Zhang, R. and Zhang, Z. (2004). Hidden semantic concept discovery in region based image retrieval. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, Washington, D.C.

Zhao, W., Jiang, Y.-G., and Ngo, C.-W. (2006). Keyframe retrieval by keypoints: Can point-to-point matching help? In *Proceedings of International Conference on Image and Video Retrieval*, pages 72–81.

Curriculum Vitae