Visual feature graphs and image recognition

Régis Behmo

ECOLE CENTRALE PARIS

P H D T H E S I S

to obtain the title of

PhD of Science

of Ecole Centrale Paris
Specialty : Applied Mathematics

Defended by

Régis Behmo

Visual Feature Graphs and Image Recognition

Thesis prepared at Ecole Centrale Paris,
Laboratoire de Mathématiques Appliquées

Jury:

President	Patrick Bouthemy	-	INRIA
Reviewers	Mihai Datcu	-	DLR - Télécom ParisTech
	Edwin Hancock	-	University of York
Examinator	Frédéric Jurie	-	Université de Caen
Co-directors	Nikos Paragios	-	Ecole Centrale Paris
	Véronique Prinet	-	Institute of Automation, Chinese Academy of Sciences

Cette thèse est dédiée à Gwilym Phillips, à sa mémoire,
à son affection, et à sa passion pour la Technique.

Acknowledgments

I like to think of thesis as the accomplishment of a four year long collaboration across two continents. Bridging the continental gap between France and China has not been an easy task, to say the least; for that, a few people deserve my warm thanks. First and foremost, I would like to express my deepest gratitude to Véronique Prinet whose careful eye and unwavering support has been the driving force behind my work for the past four years. I am also extremely grateful to Prof. Nikos Paragios, whose scientific expertise and profound humanity will remain as an example that I shall remember throughout my whole career.

The financial support of this thesis was granted by INRIA (Institut National pour la Recherche en Informatique et Automatique) and the Ecole Centrale Paris, which are the two French institutions at the forefront of scientific collaboration with China. The LIAMA (Laboratoire en Informatique, Automatique et Mathématiques Appliquées), from the Institute of Automation of the Chinese Academy of Sciences (中科院自化研究所), in Beijing, provided me with a fantastic working environment for almost two years, between 2006 and 2009.

The most fruitful results of my research work have been obtained in collaboration with a small handful of people, to whom I feel especially indebted. In particular, I have greatly benefited from the sharp intellectual insights of Jean-Baptiste Bordes, PhD; from the lively support of Cyril Cassisa, Yves Piriou, YuanFei and He LiangLiang; from rewarding exchanges with Martin de La Gorce, PhD, Ahmed Besbes and Radhouène Neji, PhD.

The second part of my thesis was mainly the result of a joint work with Paul Marcombes, then a Master student at LIAMA, and Prof. Arnak Dalalyan; I would like to thank them both for their level of commitment, as their contribution was immensely beneficial to me.

Research and wanderlust have this in common, that their raison d'être is the discovery and understanding of what is yet unknown. I would like to believe that what I have discovered and understood during these past four years has made me a slightly richer person, somehow.

Abstract

We are concerned in this thesis by the problem of automated 2D image classification and general object detection. Advances in this field of research contribute to the elaboration of intelligent systems such as, but not limited to, autonomous robots and the semantic web. In this context, designing adequate image representations and classifiers for these representations constitute challenging issues. Our work provides innovative solutions to both these problems: image representation and classification. In order to generate our image representation, we extract visual features from the image and build a graphical structure based on properties of spatial proximity between the feature points. We show that certain spectral properties of this graph constitute good invariants to rigid geometric transforms. Our representation is based on these invariant properties. Experiments show that this representation constitutes an improvement over other similar representations that do not integrate the spatial layout of visual features. However, a drawback of this method is that it requires a lossy quantisation of the visual feature space in order to be combined with a state-of-the-art support vector machine (SVM) classifier. We address this issue by designing a new classifier. This generic classifier relies on a nearest-neighbour distance to classify objects that can be assimilated to feature sets, i.e: point clouds. The linearity of this classifier allows us to perform object detection, in addition to image classification. Another interesting property is its ability to combine different types of visual features in an optimal manner. We take advantage of this property to produce a new formulation for the classification of visual feature graphs. Experiments are conducted on a wide variety of publicly available datasets to justify the benefits of our approach.

Résumé

Nous nous intéressons dans ce travail de thèse aux problèmes de la classification automatique d'images 2D et de la détection d'objet. Les avancées dans ce champ de recherche contribuent notamment à l'élaboration de systèmes intelligents, tels que des robots autonomes et des réseaux sémantiques. Dans ce contexte, la conception de représentations d'images et de classificateurs adéquats constituent des enjeux difficiles. Notre travail fournit des solution à ces deux problèmes : la représentation et la classification d'images. Pour générer nos représentations d'image, nous échantillonnons des points d'intérêts visuels dans l'image et construisons une structure de graphes basée sur les propriétés de proximité entre les points d'intérêt. Nous montrons que certaines propriétés spectrales de ce graphe constituent de bons invariants à des transformations géométriques rigides. Notre représentation d'image est fondée sur ces propriétés invariantes. Les expériences montrent que ce type de représentation constitue une amélioration par rapport à d'autres représentations similaires mais qui n'incluent pas l'agencement spatial des points d'intérêt. Cependant, un inconvénient de cette approche est qu'elle requiert une quantification avec pertes de l'espace des descripteurs visuels afin de pouvoir être combinée à un classificateur efficace, tel qu'un support vecteur machine (SVM). Nous résolvons ce problème grâce à un nouveau classificateur. Ce classificateur générique utilise une distance au plus proche voisin pour classifier des objets qui peuvent être assimilés à des ensembles (aussi appelés : nuages) de points. La linéarité de ce classificateur nous permet également de réaliser des détections d'objets, en plus de classification d'images. Une autre propriété intéressante est sa capacité à combiner différents types de points d'intérêt de manière optimale. Nous tirons parti de cette propriété en produisant une formulation nouvelle pour la classification de graphe de points d'intérêt. Les résultats expérimentaux, obtenus à partir d'une variété de jeux de données publics, nous permettent de justifier la valeur de notre approche.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The ultimate purpose of our research work is to endow automated systems with the ability to answer the question: "what does this image contain?". More precisely, we are working on two problems:

Image classification, or Image labelling is the task that consists in classifying image according to their content. Examples of such tasks include the separation of inside and outside photography shots and the classification of shots according to their subject: "this is a car", or "this is not a car".

Object detection is the more difficult problem of detecting and locating instances of a certain object class inside an image with the best possible accuracy. An example of such a query, formulated in natural language, could be: "Does this image contain a car, and if yes, where?".

Together, these two problems constitute the research field of *image recognition.*

In the following, we will first give in section 1.1 a few historical landmarks that will show why exactly image recognition is a problem that is important to addess. We will then highlight in section 1.2 the most serious challenges that make image recognition a difficult research topic. Our contributions to the resolution of these challenges will be described in section 1.3 and the general organisation of the manuscript will be given in section 1.4. Section 1.5 clarifies a few points regarding the general experimental protocol and the data that we employed in our work.

1.1 Historical background and motivations

Human intelligence has developed the faculty to store knowledge, data that is abstract by nature, inside a physical support. Such a physical support appeals to one or more of the five human senses and thus has the potential to be understood by individuals other than the author of the document. The document, which is the physical support that contains the information that the author wants to share, usually addresses the hearing and sight senses, which are the dominant human senses. In this thesis, we are concerned by visual documents. This corpus can be separated in two categories: textual

documents and images. In order to understand what is at stake in the understanding of visual documents, and thus in image recognition, we recapitulate here some of the historical milestones that have punctuated the history of text understanding.

In 1440, the first European moveable press was designed by Johannes Gutenberg[1] and sparked an exponential growth of the number of available books. One of the consequences of the invention of the printing press was the rapid acceleration of the transmission of ideas spawned by the Renaissance cultural movement. Soon appeared the need to index the content of the printed documents stored in the libraries of the European capitals: in 1595, the *Nomenclator* of the Leiden University Library was the first published catalogue of an institutional library in the world. The catalogue provided researchers to browse the content of libraries by subject. Thus, content-based text *indexing* opened to door to content-based text *search*.

The development of the world wide web (WWW) was compared by many to the invention of the moveable press, in that the WWW caused a sudden, considerable increase in the amount of documents shared across the world. The nature of these documents are mainly: texts, pictures and videos. An indication of the amount of these documents is listed in table 1.1. The comparison with the invention of the print press does not stop at the mere growth of exchanged data: the WWW has transfigured the way we communicate and spearheaded the information revolution. Moreover, the early years of the WWW will be remembered as the years that saw the emergence of large, corporate-owned textual content indexes, such as those of Google, Yahoo! and Microsoft. Without these indexes and the associated search engines, we would not be able to make good use of the colossal amount of information contained in the WWW and the relevant bits of information would in their great majority stay out of reach.

Type	Host	Date	Amount	Source
text	Google index	07/2008	10^{12}	[Alpert 2008]
image	facebook.com	11/2009	$10^{10} + 2.10^{9}$/month	[Fac 2009]
image	flickr.com	10/2009	4.10^{9}	[Champ 2009]
video	youtube.com	11/2009	20 h/minute	[You 2009]

Table 1.1: Amount of documents listed by type and host on the WWW.

However, the solution provided by textual search engines does not extend to visual content. Despite the growing production of visual content — thanks to the rapid development of cheap, personal digital cameras — there is, as of 2009, still no practical solution to the indexing of the more than 10^{10}

[1] It is only fair to note that the first moveable press worldwide was invented by Bi Sheng in China, around 1100 A.C. However, the large number of pieces required to print texts in Chinese characters limited its spread.

images on the WWW. The consequence of the lack of content-based image index is the practical impossibility to perform *content-based image retrieval* (CBIR).

In order to understand the requirements that a future visual index would have to satisfy, we need to specify the needs of a CBIR engine. Just as in textual content, a CBIR request would consist of one or more topics. The equivalent of topics for images is *object classes*: an object instance is the presence in the image of an object of the specified class. The user query may specify the pose and appearance of the object instance inside the image. Thus, a future CBIR engine would have to search an index of images in which object instances have been detected prior to the user search. The large amount of images produced per second prevents a manual annotation of the images. The annotation process must therefore be automated.

1.2 The challenges of image recognition

The main difficulties of image understanding are low inter-class variability and high intra-class variability. Low inter-class variability refers to the fact that objects from different classes may have similar appearances. On the other hand, the issue of high intra-class variability arises when the instances of a given class are widely different from one another. An illustration of a class in which objects present high levels of visual variability is given in figure 1.1.

The difficulty of linking visual representations of objects to their class is known as the *semantic gap*. A CBIR query is formulated using a powerful, contextual, ambiguous language, which is the natural language. However, the result of the query is expected to be an image i.e: a matrix of pixels, which is a representation that stems from a formal language. The fact that we are trying to provide an elementary, repeatable answer to a query that is essentially contextual and opinionated poses a difficulty.

The semantic gap appears in multiple scientific domains but it is most accurately felt in the field of computer vision. As we mentioned, objects from the same class might look very different, while objects from different classes might look very similar. This issue might possibly be solved if an infinite amount of perfectly labelled training data was made available, but the truth of the matter is that the actual amount of training data is too limited with respect to the dimensions of the image space. Strictly speaking, the cardinal of the space of 100×100 pixels grey-level value images is 256^{10000}. However, available training datasets seldom contain more than 10^3 instances per class. As of November 2009, ImageNet [Deng 2009], which is the most populated training dataset freely available today, contains an average of 675 images per synset. In contrast, speech recognition systems often train on more than 10^4 samples, while the dimensionality of the sampled space is much lower.

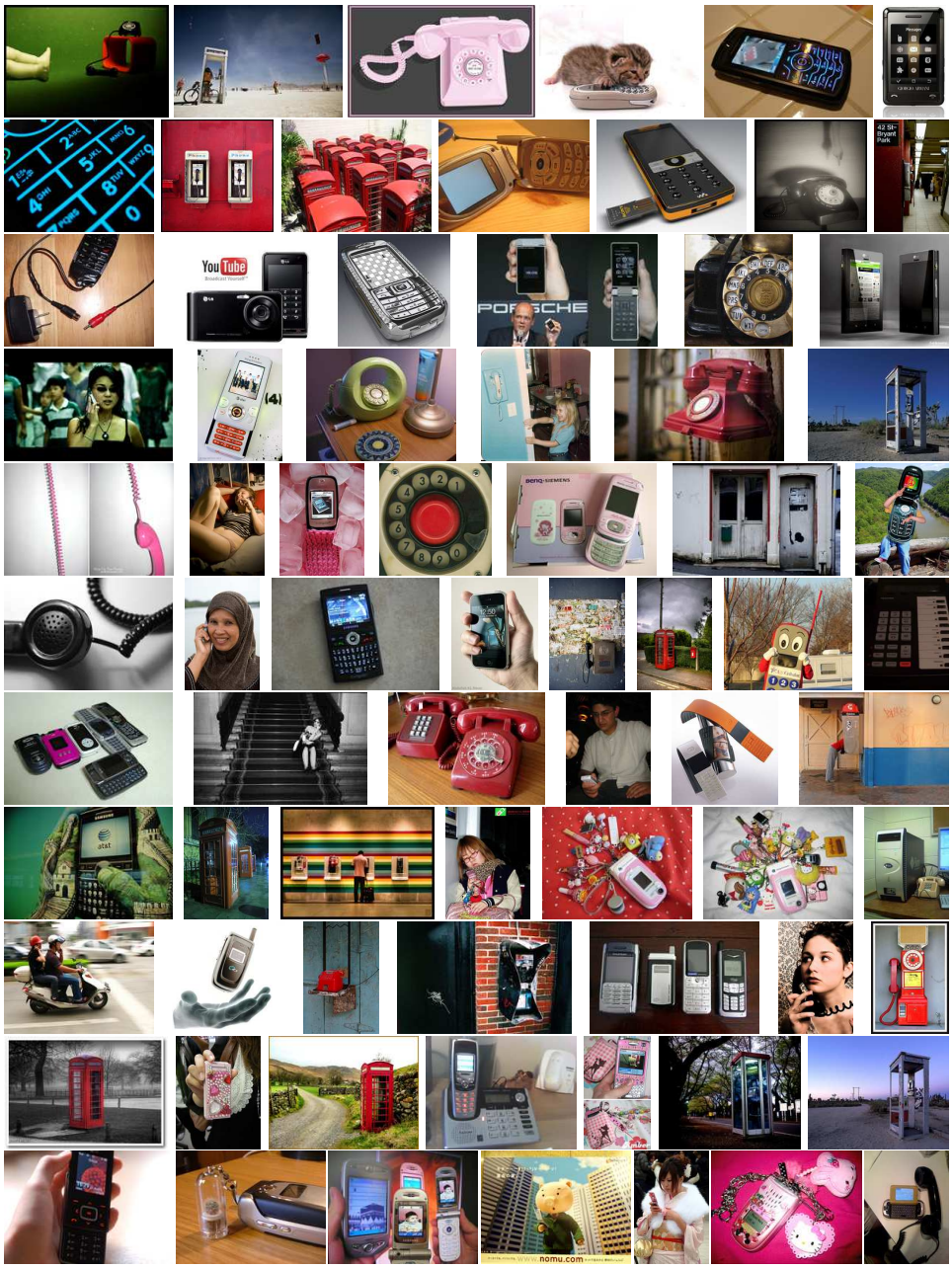

Figure 1.1: Phones: an example of large intra-class variability (images courtesy of Flickr.com)

Finally, a challenging issue is posed by the large number of visual classes: [Biederman 1987] gives a "liberal estimate" of 30.000 classes. Learning that many classes is equivalent to a human being learning 4.5 new classes every day for the first 18 years of is life. While humans are granted the faculty to perform this task at this speed, to endow computers with the same capabilities proves a challenging process.

The promises offered by the advent of effective image understanding are appealing. However, the difficulties posed by the semantic gap are also much harder to overcome because of the issues of low inter-class variability, high intra-class variability and the sparsity of training samples specific to the field of computer vision. The aim of this thesis is to provide certain theoretical and practical solutions to this problem.

1.3 Scientific contributions

Actual most competitive image recognition approaches are based on some variation of the bag-of-word (BoW) representation classified by support vector machine (SVM) or by some boosting method. Such methods have built their success on their ability to make use of a large number of visual features per image. Indeed, the simplicity of the BoW representation opens the door to large conceptual upgrades, while kernel and linear SVM are famous for their state-of-the-art classification performance.

Despite its overwhelming popularity, the BoW model suffers from two major flaws:

1. The BoW model makes it very difficult to take into account the layout of feature points in the image representation. Yet, the spatial configuration of an object has strong chances of being relevant to its class.

2. Building a histogram of features requires the definition of a quantisation of the feature space. However, it has been repeatedly shown that quantisation often strongly degrades the discriminative power of visual features, and thus the capacity to differentiate between classes.

This thesis provides principled solutions to both of these problems.

We first designed an image representation that integrates the layout of feature points in a natural way [Behmo 2008b, Behmo 2008a]. The method we describe consists in constructing an image-inferred graph of visual features and to produce a representation of this graph in a finite-dimensional space from its distance matrix. This is made possible by taking advantage of a quantisation of the visual feature space: similar visual features that are assigned to the same bin are grouped to produce a "feature-collased graph". The feature-collapsed graphs of different images can then be compared for classification. We observed that the distance matrix of the feature-collapsed

graph integrated the information related to the layout of points and constituted an image representation that was more discriminative. In combination with SVM, we studied the robustness of this representation with different graph construction processes that present different invariants to rigid geometric transforms. Moreover, we studied the impact of different distance measures inside the graph of features to produce the graph distance matrix.

In order to free ourselves from the feature quantisation step, we designed a generic classifier in the space of feature sets [Behmo 2010]. Feature sets are classified on the basis of the naive Bayes assumption by sums of affinely-corrected nearest neighbour distances. The affine parameters are computed as solutions to a minimisation of the hinge loss. The linearity of this classifier allows us to use it in combination with fast subwindow search to perform object detection. Moreover, an interesting property of this classifier is its ability to simultaneously take into account visual features of different types in an optimal manner, with respect to the hinge loss. This property allows us to formulate the problem of feature graph classification with unquantised features. We achieve this by grouping point pairs located at equal distances from one another.

1.4 Organisation of the manuscript

The remainder of this manuscript is organised as follows: in chapter 2 we describe the state of the art in the field of image understanding. We detail methods that were most inspirational to us. We also explain why some works, while formulating their starting problem in a way that is similar to us, end up with methods that are fundamentally different from ours.

A new image representation based on spectral properties of an image-inferred graph is given in chapter 3. We sample interest points from each image using an off-the-shelf interest point detector and descriptor and use these points as nodes of a graph. This graph is transformed by grouping some of its nodes and the distance matrix of the transformed graph is used as the image representation. Experiments on various datasets follow.

We consider that the major flaw of our graph-based representation is the feature quantisation step, which dismisses a large part of the information related to the image appearance but which is necessary to obtain a compact vector representation. This issue is addressed in chapter 4. We begin by evaluating the potential gain that could be obtained if we used unquantised features by actually implementing a classifier of feature sets. This classifier, which is actually a support vector machine kernel, is best suited for small sets of features and relies on an image-to-image distance. We then show why image-to-class distance is preferable and adopt a different formulation, called naive Bayes nearest neighbour (NBNN, [Boiman 2008]). We improve NBNN by exposing some of its profound theoretical and practical

limitations. In particular, we show that NBNN can be naturally adapted to the classification of images with different levels of description that we call *channels*. We manage to formulate the problem of graph classification in terms of combinations of channels, and thus achieve our goal of classifying graphs containing unquantised features. Chapter 5 summarises the main achievements of this thesis and gives some openings for future research.

1.5 Working context and experimental protocol

The practical problems that the algorithm we designed in this thesis address are image classification and object detection. Our objective should be distinguished from image retrieval, which is the problem of finding specific, unique objects in different contexts (e.g: landmark recognition). Therefore the object classes we are trying to index do not only present variations in pose and lighting, but also in appearance and shape. In the course of this thesis, we did not focus on just one dataset or one object class: for instance, we tried to address problems as widely different as the classification of high resolution satellite images and the detection of objects in natural photographs shot by personal cameras.

The algorithms we designed are supervised: in each experiment there is a set of annotated images that constitute the training dataset, and the remaining images constitute the testing (or: validation) dataset. Thus, the set of classes is always limited to the annotations found in the training dataset. In practice, we dealt with a number of classes of the order of 10^2 while the number of training images per class varied between 10^1 and 10^3.

Our algorithms were implemented and executed on personal machines with reasonable capabilities. The most powerful available computer was equipped with eight quadcore Intel Xeon processors running at a frequency of 3.20 Ghz and 16 Gb RAM.

In the course of our research, our objectives were set in terms of efficiency and possible gains over comparable methods, thus laying a strong emphasis on practical applications. By following the path of highest likely performance gain, we decided to set aside methods based on personal habits or current trends to focus on best working techniques. It is by adopting this strategy that we engineered our research axes.

Chapter 2

State of the art

Computer vision is a relatively recent field of research compared to other scientific domains and it is still possible to follow the hereditary trail from actual methods to the pioneer works. We date the seminal advances in modern image understanding to the early 90s: it is only after that date that methods began to be tested on large, challenging, publicly distributed datasets. Though prior work had been conducted for image segmentation and the representation of visual content, the necessity of image retrieval, and thus of automated image understanding, only emerged then. Initial research were mostly driven by industrial needs and security concerns: the first applications were designed to address the problems of material (i.e: texture) recognition, car detection and the joint problems of human face detection and recognition. With the ubiquitous expansion of the world wide web (WWW), concerns then shifted towards the more general problem of detecting object instances of many object in large image databases.

Image understanding is linked to several other scientific fields. Researchers have repeatedly turned to discoveries in the fields of brain sciences and psychology to understand how humans perceive and understand their environment. Ideas have been borrowed and adapted to the linearity of computer algorithms in order to get computers to perform tasks in a way that is similar to humans. The task of describing proper models for this purpose has required the help of tools drawn from all branches of mathematics, mainly: statistics, probabilities and machine learning. Ad hoc implementations must also take into account the practical limitations imposed by personal computers in terms of memory and processing speed; and while the capabilities of personal computers have followed an exponentially increasing curve, the computational requirements of algorithms as well as the size of visual datasets have taken a parallel, increasing path. Constraints on algorithm complexity have thus remained very strong; thus appropriate algorithmic data structures have been designed by the computer science community.

In this review of existing work related to our research, we will focus on computer vision innovations that have contributed to the advances of image recognition. Methods based on the sampling of local descriptors will be of more particular interest, as these methods have made the headlines of recent international image understanding competitions. We will draw comparisons with image representations based on the sampling of visual features, and

in particular: unquantised features. Our work also borrows much material from previous methods based on image-inferred graphs, although fundamental differences exist with most popular graph-based methods such as: the constellation model and methods that aim at solving the graph matching problem.

2.1 Image representations

An efficient way to decompose the problem of image recognition is to first construct a vector representation for each image and to build a class model from these representations. Most of the representations that we describe here are non-parametric, which greatly simplifies the construction of the image representation. We focus on representations based on global visual statistics, visual features, and in particular: quantised visual features.

2.1.1 Global descriptions

Some authors have argued in favour of representations based on the global aspect of images, notably for scene recognition. Indeed, a number of experimental observations demonstrate that human subjects are able to grasp several rather precise pieces of information relative to the meaning of a scene in a time lapse as short as 30 ms. In such a short time lapse, it is reasonable to assume that the human stare has not had time to focus on details of the visual scene, but only to capture certain visual statistical properties. This line of thought has inspired a number of methods to represent the content of a visual scene by the output of a set of statistical operators. In [Gorkani 1994], the authors argue that simple, coarse classification between two classes (city and suburbs) can be performed simply by considering the peaks of the edge orientations histogram of an image. Similarly, in [Szummer 1998] texture, colour and frequency properties are collected in a global image representation and classified by nearest neighbour. In [Oliva 2001, Oliva 2006], properties from the image spectrum as well as from a spatially coarse spatial histogram constitute the image representation at what the authors call the "basic" and "superordinate" levels.

Such approaches have their advantages. However, in our work we have focused on image descriptions based on local visual properties (what we shall from then on designate under the name of visual features). Though we admit that an ideal approach would probably combine both global and local representations for maximum efficiency, we believe that approaches based on local features provide more flexibility. Therefore, our methods do not aim at providing global representations of images, but representations based on local properties of visual features.

2.1.2 Visual features and image interest points

Whether we are facing a problem of image retrieval, wide baseline matching, stereovision, object categorisation, or practically any other computer vision task, we are at some point confronted to the problem of finding similarities and differences between images, or sets of images. This problem is easily tackled by the human brain, which has the experience necessary to easily understand physical external differences in shape, appearance or functionality between objects that are designed to essentially perform the same task. The deceptively simple vocabulary that we have developed to designate objects that perform the same task, or belong to the same *object class*, dismisses the potentially large differences that can occur between two instances of the same class. For example, English designates by the same term an ocean liner and a catamaran: "boat".

The ease with which we equally designate two things that look fundamentally different has no equivalent in the mathematical world, where two objects are called equal if and only if they are indeed equal. This problem becomes even more cumbersome when the only source of information available for an algorithm to understand the specificity of an object is its appearance. When the data that is available is just the 2D view of the object taken under a few viewing angles, the resulting comprehension is bound to be extremely limited. Nonetheless, the finiteness of the space of all possible images guarantees that, given a sufficiently large training dataset, image similarity should be sufficient to reach decisions concerning the presence or absence of an object in an image. The problem is that the space of possible images is too large to be sampled with sufficiently high frequency to ensure that a simple nearest neighbour method should provide us with correct labelling.

The combination of these two problems (the gap between the notions of equality in the human experience and in mathematics on one hand, and the large intrinsic dimensionality associated of an image as a mathematical object) is the guideline of the efforts of the computer vision community. First, we are bound to reason in terms of *proximity* (or equivalently: *distance*), and not equality between images: in other words, we should not try to determine if visual objects are equal, but if they are close to each other in a certain (yet unknown) space. Second, there is an imperative need to drastically reduce the dimensionality of images by dismissing all the non-relevant or redundant pieces of information they contain.

One common method to achieve these two goals is to consider finite-dimensional representations of sub-images, or image regions. The computer vision community has taken a strong turn during the past few years in the direction of *image features*, also called *visual features* or *points of interest*. The study of visual features focuses on two main aspects:

1. The detection, or sampling, of finite sets of points that are relevant to

the image.

2. The description of the visual neighbourhood of these points as the accumulation in a finite dimensional vector of certain local visual characteristics.

Taken together, the detection and the description of interest points consist in an effective "divide to conquer" strategy to analyse the content of images. In the work described in this thesis, we were not specifically concerned by the extraction of visual features: our methods are, in general, independent of the choice of feature detector and descriptor which is usually a replaceable brick of our methodology. Nonetheless, because the choice of visual features remains crucial from a performance point of view, we outline in the following few sections some major works related to the extraction of visual features.

In the following, we will first briefly describe the essential concepts underlying the detection of blobs and corners in an image. Then, we will give a more in-depth description of the concept of image scale-space, which has greatly contributed to the design of repetitive visual features. Lastly, we will list some influential feature descriptors that have made use of these concepts.

2.1.2.1 Blob detection

For an efficient sampling of uniform regions, also called *blobs*, the spatial centres of the regions need to be detected. They can be detected as points for which the intensity inside a Gaussian window of a certain scale widely varies in all directions. In other words, after convolution by a Gaussian filter $g(x, y, \cdot)$, the Laplacian of an image $f(x, y)$ will take strong absolute values at the centre of blobs:

$$\begin{aligned} g : \mathbb{R}^2 \times \mathbb{R}^+ &\rightarrow \mathbb{R} && (2.1)\\ (x, y, t) &\rightarrow \frac{1}{(2\pi t)} e^{-\frac{(x^2+y^2)}{2t}}, && (2.2)\\ L(x, y, t) &= g(x, y, t) * f(x, y), && (2.3)\\ \nabla^2 L &= \frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2} = L_{xx} + L_{yy}. && (2.4) \end{aligned}$$

Strongly negative (respectively: positive) values of $\nabla^2 L$ will correspond to light blobs (resp.: dark blobs). We are thus interested in sampling the extrema of $\nabla^2 L$ in the image.

2.1.2.2 Corner detection

Although useful to image comprehension, blobs are seldom the most essential source of information relatively to an image. In fact, psychological studies of

scene understanding by humans have shown early on how regions displaying high levels of change are the most informative to the human brain; most of blob regions can be dismissed for the sake of image representation sparsity, but the removal of regions such as corners and edges impedes scene understanding. The study of visual features thus focused early on on the discovery of edges and corners in images [Moravec 1980, Harris 1988]. The Moravec corner detector [Moravec 1980] samples regions that display large intensity changes in any direction. Indeed, the particularity of a corner region is that it displays a high image gradient magnitude (contrary to flat regions) and a displacement in any direction of a rectangular window over the corner will result in a strong intensity change.

In [Harris 1988], the authors adapt Moravec's corner detector to also shoot on edges; they also proceed to make Moravec's detector anisotropic, more robust to noise by adopting a Gaussian window instead of a rectangular one. Eventually, the method consists in studying maxima of the eigenvalues of the Harris matrix over all image pixels:

$$M = \begin{pmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{pmatrix}, \tag{2.5}$$

where I_x and I_y are the derivatives over the x and y axis respectively and $\langle \cdot \rangle$ denotes the averaging operator over a Gaussian window. If one of the eigenvalues is large while the other is small, this indicates the presence of an edge. If both eigenvalues are large, we are in presence of an corner. At the time, it was noted that the computation of the eigenvalues was too expensive and the evaluation of a "corner response" was proposed instead; observing that the trace of the Harris matrix was equal to the sum of the eigenvalues and that its determinant was equal to their product, the authors decided to employ the following formulation:

$$R = det(M) - k.trace(M)^2, \tag{2.6}$$

where k is a constant parameter, usually set to values in the range $[0.2, 0.4]$. The Harris response is positive on corners, negative on edges and close to zero on flat regions. Extrema of the Harris response over an 8-point neighbourhood thus indicate the presence of a visual point of interest.

2.1.2.3 The image scale space

The way in which the Harris corner and edge detector is described above hides the requirement of two scaling parameters: one for the Gaussian averaging operator and one for the derivation operator. The function of these scale parameters is to locate points in a robust fashion with respect to the scale at which the image is studied. The notion of image scale was first developed by [Witkin 1983, Koenderink 1984] and is understood as the relative

proximity from the camera to the object. As a matter of fact, there are two equivalent ways to consider image scale: we can talk about a reduction of the image scale by a factor σ in either cases, which are strictly equivalent:

1. Reduction of the image side length by a factor σ.

2. Convolution of the image with a Gaussian filter of smoothing parameter σ.

The scale space of an image can thus be viewed as a 3D matrix where the blurring factor varies along the third axis; since image blurring and size reduction are equivalent, the scale space can also be called scale pyramid. More precisely, it was shown (see e.g: [Koenderink 1984]) that given a continuous signal $f : \mathbb{R}^D \to \mathbb{R}$, the only so-called space-space linear representation $L : \mathbb{R}^D \times \mathbb{R}^+ \to \mathbb{R}$ verifying the diffusion equation

$$\partial_t L = \frac{1}{2}\nabla^2 L = \frac{1}{2}\sum_{i=1}^{D} \partial_{x_i,x_i} L, \tag{2.7}$$

with initial condition $L(\cdot, 0) = f(\cdot)$ was the family of Gaussian convolved signals:

$$L(\cdot, t) = g(\cdot, t) * f(\cdot), \tag{2.8}$$

where $g : \mathbb{R}^D \times \mathbb{R}^+ \to \mathbb{R}$ is given by:

$$g(x, t) = \frac{1}{(2\pi t)^{D/2}} e^{-\frac{(x_1^2+\dots x_D^2)}{2t}}. \tag{2.9}$$

One of the important contributions of [Lindeberg 1998] was to remark that similarly to the way interest point detectors work in the image space, it should be possible to detect image features that are cornerness extrema both over the two image dimensions and the scale dimension. It becomes then possible to talk about the characteristic scale of a visual feature point. In [Lindeberg 1998], the author argues that "image descriptors can be highly unstable if computed at inappropriately chosen scales", and selecting the descriptor scale becomes an absolute necessity for robust image representation. The key assumption here is that the characteristic scale of a point is indeed representative of a certain characteristic length; this assumption was formulated in the scale selection principle [Lindeberg 1998]:

Principle 1 (Scale selection principle) *In the absence of other evidence, assume that a scale level at which some (possibly non-linear) combination of normalised derivatives assumes a local maximum over scales, can be treated as reflecting a characteristic length of a corresponding structure in the data.*

In order to find interest points that are robust to scale change, one thus needs to take extrema of the corner or blob response both over space and scale; however, it should be noted that the response should be normalised by a scaling factor, since the amplitude of spatial derivatives in general decrease with scale [Lindeberg 1998]. Once response normalisation has been undertaken, scale-invariance feature points have been obtained. Here, we remind the warning raised in [Mikolajczyk 2005a] that the "scale invariant" formulation is misleading; in fact, the point detection process is covariant with scale change, but it is the point description that should remain invariant.

2.1.2.4 The description of visual features

Local visual content associated to sampled interest points can be described in a variety of manners. We list here some of the works associated to two main categories of descriptors: descriptors based on responses to filter banks and descriptors that consist of local distribution of some visual properties. We acknowledge the fact that other types of descriptors exist; here we limited ourselves to these two categories, as the description of local features is not the primary goal of this thesis. The reader is referred to [Mikolajczyk 2005b, Li 2008] for more extensive reviews of feature detectors and descriptors.

As we have emphasised multiple times, early computer vision methods were often concerned by the reproduction of the mechanisms orchestrated by the human brain to perform vision tasks. This also applies to the extraction of visual features. It was found in [Marčelja 1980] that the visual cortex cells respond more strongly to simple signals such as Gabor elementary filters; in fact, each cell of the visual cortex is tuned to a specific spatial frequency corresponding to a certain Gabor filter. A bank of Gabor filters with a certain number of orientations and scales can thus serve to efficiently describe the neighbourhood of visual features. Because of their direct connection with biology and human vision, descriptors based on the responses to elementary filters have enjoyed lasting popularity. Another example of filter-based descriptor is the descriptor that can be obtained as output of a steerable filter bank [Freeman 1991]: the steerable filter bank is a set of filters composed of the Gaussian filter and linear combinations of its derivatives, where the linear coefficients actually depend on the orientation.

Another, more recent, class of visual descriptors is made up of representations based on local histograms. Generally speaking, histograms are efficient representations of visual content as they are usually invariant to small local changes. Among them are the popular SIFT [Lowe 2003] and shape context [Belongie 2002] descriptors. SIFT descriptors are essentially the concatenation of several local rectangular gradient histograms computed around the local interest point. Despite the many technicalities involved in their production, the robustness of SIFT descriptors to small displacements and lighting changes have made them the *de facto* reference descriptor. The

SIFT descriptor has also provided the inspiration for several subsequent descriptors, such as the RIFT [Lazebnik 2005b], which consists of a set of elliptical gradient orientation histograms. The histogram of oriented gradients (HOG), employed in [Dalal 2005] for the detection of human silhouettes, can be considered as a simplified version of SIFT. The shape context descriptor [Belongie 2002], on the other hand, constructs for each interest point a histogram of relative position and orientation of other interest points. While the SIFT descriptors perform best on textured images, the shape context demonstrated impressive levels of performance on the recognition and matching of shapes, such as handwritten digits. Finally, we should mention the "speeded-up robust feature" (SURF, [Bay 2006]), which is a combination of feature detector and descriptor that proved quite efficient in our experiments: for SURF, detection is based on the Hessian matrix [Mikolajczyk 2005a], while the descriptor consists of a distribution of Haar wavelet responses in the neighbourhood of the detected point.

2.1.3 The bag of words representation

The bag of words representation can be seen as the practical proof of the effectiveness of visual feature points. It is inspired by methods initially defined for the text retrieval community (see for instance [Joachims 1998]): the original idea was to describe a text as an orderless accumulation of words, or, more precisely, as a word histogram. Given a text and a predefined dictionary (aka: corpus) of K words, the bag of words of the text is a vector of K dimensions, where the k^{th} entry indicates the number of times that word k appears in the text. The essential characteristic of this representation is that it dismisses any kind of information associated to the arrangement of words in the sentence, thereby getting rid of all grammar, rhetoric and text structure. The surprising fact is that despite the quantity of information ignored by the representation, it is quite often still possible to retain the gist of the text and to guess what was its general topic from this histogram. As an example, we represented in figure 2.1 the bag of words of the introduction of this thesis.

The same steps can be taken to represent an image, or a set of visual features, as a bag of words. The basics of the method remain the same, with one notable exception: since the space of visual features is continuous and a bag of words is computed given a corpus of finite size, the space of visual descriptors must be quantised in a finite number of bins prior to the computation of the bag of words (section 2.1.3.1). Once the bag of words of the images have been computed, images can be classified using just any machine learning algorithm; current state of the art results use support vector machine (section 2.2.4). In recent years, under the pressure of increased competition in the image classification field, we have observed a resurgence of information associated to the spatial localisation of interest

Figure 2.1: Bag of words of the introduction of this thesis (chapter 1). Larger height indicate a greater frequency of occurrence. Illustration produced using Wordle (`http://www.wordle.net`).

points in the bag of words representation, as a mean to increase performances (section 2.1.3.2).

As an illustration of why the bag of word representation can perform well in the context of image recognition, we provide in figure 2.2 a representation of an image in which local regions associated to a SIFT detector [Lowe 2003] have been mixed and all geographical organisation lost. What we can observe, is that it is still possible to guess some of the content of the original image from the bag of features; for instance, water and man-made construction features are clearly evident.

The first implementations of a bag of words for visual recognition had the same concern as part-based models, mentioned in section 2.4, to truthfully model physical structures. In [Leung 2001], "textons" are centres of visual features clusters. They are used as prototypes to represent 3D material textures. Bags of words constructed with textons are then classified by nearest-neighbour according to a χ^2 distance. Features consists of the responses to 48 filters and are quantised by k-means. Already, the authors of [Leung 2001] note that performances increase with the codebook size. Later approaches combined the nearest-neighbour classifier with PCA and obtained a consistent performance gain [Cula 2001a, Cula 2001b].

The problem of feature quantisation is considered from a different point of view by [Schmid 2001]: in this paper, clusters are characterised not only by a centre, but also by a covariance matrix. Thus, bags of words become histograms of probabilities as feature distances to cluster centres are con-

Figure 2.2: The Sydney Opera house: original image (top) and bag of features representation (below).

verted to Gaussian likelihoods.

We believe it is a sign of the times that these early developments of the bag of word were shortly followed by the publication of [Varma 2002]. In their work, Varma and Manik quantitatively evaluate the performances of cluster- versus probability density-based methods. The latter rely on a regular quantisation of the feature space along each coordinate. In that sense, probability density histograms are another version of a bag of words where each space bin is associated to a cluster of fixed size, and with cluster centre located at the centre of the bin. Results given by the authors are in favour of clustering-based methods. Even though the probability density estimation method can be contested, the performance gap indicated by [Varma 2002] will be widened by the introduction of parametric classifiers such as support vector machine (SVM) and Adaboost.

In [Dance 2004], experimental evidence shows that SVM on histograms of quantised features largely outperform other classifiers for image classification tasks. The use of SVM with BoW will be generalised to pyramids of features in [Grauman 2005, Lazebnik 2006]. In their work, Lazebnik et al. [Lazebnik 2006] improve the state of the art on image classification by re-introducing some amount of spatial information in the BoW. In section 4.3 we will draw our inspiration from their work to divide images in subre-

gions that constitute individual sources (aka: channels) of features.

Computing distances between histograms of quantised features requires the definition of an appropriate distance. L^1, L^2, χ^2 distances are the most frequently employed, but others, such as the Earth-Mover Distance [Zhang 2007], can also be of interest. In the case of L^1, L^2, χ^2 distances, codebooks are computed for the whole dataset. [Zhang 2007] show that, while these distances perform well on texture classification tasks, they are not sufficiently robust to clutter. In real-life datasets, such as Graz-02 [Marszałek 2007a], codebooks on whole datasets are too noisy. To alleviate this problem, [Zhang 2007] compute small codebooks containing 40 clusters per image. Each image has a signature over this codebook and distances between signatures is evaluated by the Earth Mover Distance (EMD, [Rubner 2000]).

2.1.3.1 Quantising the feature space

An important step in the construction of a bag of words is the quantisation of the feature space. While it is generally acknowledged that quantising a 100-dimensional space with just about 1000 cluster centres is undersampling, few actually take steps to address this issue. We outline here two approaches that aim at building dictionaries that are better suited to the classification task.

[van Gemert 2008] highlights two problems of traditional codebook construction methods: codeword uncertainty and codeword plausibility. Uncertainty refers to the problem of choosing between two or more codewords for a given feature point when they are located at approximately equal distances. Codeword plausibility (or the lack thereof) appears when the codeword that is nearest to a given feature is located at a too large distance, and thus does not truthfully represent the feature point. Naturally, both properties are particularly problematic in the case of feature spaces of large dimensions. In [van Gemert 2008], the authors argue that the impact of these issues can be alleviated opting for soft instead of hard quantisation. They then proceed to model the uncertainties associated to feature quantisation by a kernelised codeword assignment. The gain obtained over hard quantisation is consistent (between 4 and 9.3 percentage points, depending on the dataset). However, in section 4.1 we contest one of the the implicit assumptions of [van Gemert 2008]. Indeed, in order to quantify a feature over several codewords, we need to assume that there is a limited number of codewords in the close vicinity of the feature point. We show that this is not true for high-dimensional visual features extracted from large amount of images.

Another method to improve the relevance of codebook entries is to build dictionaries in a supervised manner. Results from [Zhang 2007] have already demonstrated that separating foreground and background features in differ-

ent codebooks improves the discriminative power of the bag of words. Lazebnik & Raginksy show in [Lazebnik 2007] how to build a codebook that minimises the loss of discriminative information. Similarly, in [Fulkerson 2008], codebooks are compressed by merging words in an iterative fashion. Merged word pairs are chosen so that the mutual information decreases as little as possible.

The common point between these approaches is that they build feature codebooks in order to improve the discriminative power of the bag of words representation. The problem of these approaches is that they do not cope well with the addition and removal of object classes, as new codebooks must be learned every time a class is added. Thus, we chose not to try to build optimised codebooks. Instead, we employed classical codebooks built by k-means to emphasise the benefit of our method instead of shifting the responsibility of the gain to codebook construction. Then, we proceeded to get rid entirely of the codebook, so as not to have to deal with quantisation issues.

2.1.3.2 The re-introduction of point layout

Although the essential characteristic of the highly competitive bag of words model is to dismiss the spatial coordinates of the feature points in the image, recent methods based on a histogram of features tend to re-introduce some level of information regarding the spatial layout of the interest points to attain high performances or to undertake other tasks, such as localisation or segmentation [Leibe 2004, Marszalek 2006]. There are several ways to do this. One of the possible steps to take is to address the issue of polysemy in the representation by using more discriminative features, for example groups of interest points. Indeed, in a feature codebook, one codebook entry may belong to different classes. However, a group of features is much more discriminative. This is the idea underlying [Sivic 2005, Puzicha 1998, Agarwal 2006, Lazebnik 2006, Ling 2007].

In [Sivic 2005], pairs of interest points, sometimes also called *doublets* are employed to extract more informative visual content. The next natural step is to group a greater number of interest points in "hyper features": in [Puzicha 1998, Agarwal 2006, Lazebnik 2006], interest points are recursively grouped together to produce pyramidal histogram representations; it is thus the combination of the information related to neighbouring points that produces the final representation.

In essence, the common point between these approaches is that they all rely on the co-occurrence of spatially close features of specific appearance to discriminate between classes. Similarly, the correlogram is an image representation that summarises these co-occurring events in a histogram structure; different definitions of a correlogram have been given, but in general it is a histogram of quantised features with distance, and sometimes angular

bins. Colour correlograms were one of the early methods designed to classify images: in [Huang 1997], bin (i, j, k) of the colour correlograms is the probability of obtaining colour j at a distance k of a pixel with colour i. The work of [Savarese 2006] is a generalisation to the case of textons; quantised correlograms then produce *correlatons*. This is the same idea that underlies the work of [Lazebnik 2005b] in which the spin image descriptors are in fact histograms of distance and pixel intensity. Similarly, in [Yang 2007], a quantised descriptor is computed for image regions and the *spatial keyton* is the histogram of distances between regions. Bin j delimited by r_i, r_{i+1} and θ_i, θ_{i+1} is the number of occurrences of feature j in the region at a distance $r \in [r_i, r_{i+1}]$ and an angle $\theta \in [\theta_i, \theta_{i+1}]$. The correlogram is usually centred on a particular interest point. A somehow simplified version of this kind of histogram is the bi-gram [Lazebnik 2005a] which counts the number of occurrences of adjacent textons, or quantified regions. Following the same current of thought, histograms of features can be used as "meta features", in the sense that they can be used to better describe the neighbourhood of a feature. In [Schmid 2004], certain local histograms are selected to describe image content and index images.

2.2 Feature-based image classifiers

Once an image representation based on visual features has been built, the second elementary brick in the design of an image classification system is an appropriate classifier. In this section, we review some of the existing classifiers adapted to feature-based image representations.

2.2.1 Naive Bayes classification

The naive Bayes classifier relies on an assumption is an assumption that allows us to considerably simplify the probabilistic formulation of image classification. In terms of probabilities, finding the most likely label $\hat{c}$ of an image I consists in maximising $P(c|I)$, which can be rewritten following the Bayes rule:

$$\hat{c} = \arg\max_c P(c|I) = \arg\max_c \frac{P(I|c)P(c)}{P(I)} = \arg\max_c P(I|c)P(c), \quad (2.10)$$

assuming $P(I)$ is a constant. Let us represent the content of image I by a finite set of visual features $\{x_k | 1 \leq k \leq K_I\}$. Computing a full model for $P(\{x_k\}|c)$ is a hard problem that requires many training samples. However, if we make the simplification assumption that all image features are independent from one another conditionally to the class, the likelihood at the

numerator of the RHS of equation 2.10 becomes:

$$P(I|c) = \prod_{k=1}^{K_I} P(x_k|c). \tag{2.11}$$

The evaluation of the most likely image label boils down to a problem of computing the product of the likelihoods of each image feature, independently from one another. This can be done in various ways: with Gaussian mixture models, by parametric estimation or by quantising the feature space and counting the number of quantised feature occurrence per class.

Despite the strength of the naive Bayes assumption, it has been successfully employed in a number of works. In [Schneiderman 2000], the image likelihood is decomposed over the various wavelet components of its representation and probabilities are estimated for quantised values of the descriptors. In the constellation model of [Fergus 2003], the appearance component is made of a product of the likelihoods of the various attributes, which implies a naive Bayes assumption (even though the correlation between the features is taken into account elsewhere in the probabilistic formulation).

2.2.2 Nearest neighbour classifiers

As we have seen in previous sections of this review of the state of the art, k-nearest neighbours (k-NN) is a popular classifier in the field of computer vision. k-NN, of which the nearest-neighbour classifier is a particular case, consists in assigning a data point to the class for which there are the most exemplars among the k nearest neighbours. k-NN thus requires the definition of an appropriate metric. In the simple nearest neighbour case (1-NN [Cover 1967]), it has been shown that the probability of error cannot excess twice the bayesian probability of error as the number of training samples becomes infinite. Thus a large amount of label information is contained in the nearest neighbour. This, along with the simplicity and versatility of k-NN, explains the popularity of this classifier.

k-NN, a non-parametric classifier, can be categorized among other methods based on *lazy learning* [Aha 1997]. In other words, the generalisation of the classifier is postponed until the first testing data point is known, as opposed to *eager learning* which tries to generalise the training data before seeing any test sample. Non-parametric methods have several advantages over their parametric counterparts. For instance, they require no training phase and they generally avoid the issue of overfitting. However, they often require all training data points to be stored in memory, which can be costly.

Steps have been taken to apply parametric methods to non-parametric classifiers. In particular, it is tempting to combine the benefits of SVM and k-NN. The authors of [Domeniconi 2005] formulate the interesting remark that the relevance of nearest neighbours vary depending on their direction

relatively to the test point. Let us take the example of two clouds of points that overlap in some region of the feature space. Training samples can be separated by SVM which draws a boundary between the two cloud points. Class assignment probability will greatly vary along the orthogonal direction of this boundary. Thus, a test point should have more "trust" in the neighbours that lie in the direction parallel to the boundary. This observation was made earlier by [Hastie 1994]: in their work, Hastie and Tibshirani also find a boundary between classes, though they use centroids in place of SVM. The resulting classifier is coined discriminant adaptive nearest neighbour (DANN). Finally, along the same line of thought, [Zhang 2006] also combine SVM and k-NN. They first perform k-NN and then resolve any possible ambiguity among the k nearest neighbours by SVM.

2.2.3 Metrics and kernels on feature sets

Once features and visual properties have been extracted from an image, it is only natural to try to compute distances between these same sets of features in order to differentiate, and thus classify images. However, there is no designated, straightforward metric on the space of feature sets and different methods can be employed at this point. The Hausdorff distance provides a good starting point: in a metric space $(\mathcal{E}, ||.||)$, the Hausdorff distance H between two subsets of $\mathcal{E}$ is defined as:

$$\forall X, Y \subset \mathcal{E},\ H(X,Y) = \max\left(\sup_{x\in X}\min_{y\in Y}||x-y||, \sup_{y\in Y}\min_{x\in X}||x-y||\right). \quad (2.12)$$

In other words, the Hausdorff distance between two sets of points is equal to the maximum distance between nearest neighbours from both sets. The Hausdorff distance has been used in computer vision with notable applications in stereovision and detection [Huttenlocher 1993]. Twenty-four modified versions of the Hausdorff distances have also been produced by [Dubuisson 1994] to match edge maps of objects; most subsequent distance measure and kernels between sets of features actually derive from one of these modified versions of the Hausdorff distance. [Odone 2001] develops a new version of the Hausdorff distance for matching of grey-level value images; in this work, images are considered as sets of pixels (a parallel can be drawn between the use of pixels as elementary descriptors of an image and the description of an image by a set of visual features).

The essential limitation of metrics based on the Hausdorff distance is that most of them are provably *not* positive semidefinite, and are thus not Mercer kernels; as a reminder, a continuous function $K : \mathcal{E} \times \mathcal{E} \to \mathbb{R}$ is said to be positive semidefinite if and only if:

$$\forall N \in \mathbb{N},\ \forall x_0, \cdots x_N \in \mathcal{E},\ \forall c_0, \cdots c_N \in \mathbb{R},\ \sum_{i=0}^{N}\sum_{j=0}^{N} c_i c_j K(x_i, x_j) \geq 0 \quad (2.13)$$

The positive semidefiniteness of a kernel is an essential property in numerous classifiers, notably for convergence of support vector machines (see section 2.2.4). Mercer's theorem gives us an indication why positive semidefiniteness is so important:

Theorem 1 (Mercer's Theorem) *Any positive semidefinite kernel* $K : \mathcal{E} \times \mathcal{E} \to \mathbb{R}$ *can be expressed as a dot product in a certain vector space:*

$$\exists \phi,\ \forall x, y \in \mathcal{E},\ K(x, y) = \phi(x).\phi(y). \tag{2.14}$$

Because of the possibility to decompose a kernel as a dot product, positive semidefiniteness becomes a very desirable property. Consequently, several authors have designed kernels for sets of features that verify the Mercer property of equation 2.13. In [Barla 2002], it is shown that the histogram intersection metric is positive definite. Moreover, the hypothesis required in support vector machine kernels and novelty detection are weakened and the Hausdorff kernel is adapted to the case of novelty detection. [Kondor 2003] introduces a Mercer kernel between sets of features based on the Bhattacharyya similarity.

On the other hand, [Boughorbel 2004] designs a non-Mercer kernel that closely resembles the Hausdorff kernel by selecting a matching strategy between point pairs. Even though the kernel does not verify the Mercer properties, the authors give inferior bounds on the probability that it is positive semidefinite and show that the kernel can be used for support vector machines nonetheless. In [Boughorbel 2005], this same kernel is adapted by selecting intermediate, fixed features to which distances are computed for all feature sets.

Finally, kernels between sets of features can make use of metrics different from the canonical metric. In [Caputo 2002], global information concerning the image colour distribution and shape are combined for recognition. In [Wolf 2003], the goal is to classify image sequences: each image is considered as a subspace of the original vector space and the principal angles between the subspaces (aka: the images) are computed by the kernel, which ultimately serves as a support vector machine kernel.

One of the main advantages of computing distances between images directly based on their visual features is that it does not require to modify the space of visual features. In particular, it does not need to be quantified.

2.2.4 Support Vector Machines (SVM)

Support vector machines are classifiers that possess several useful properties:

1. In the separable case, SVM select the best boundary by maximising the margin between the boundary and the nearest training samples.

2. In the non-separable case, it is still possible to train an SVM by minimising an upper bound of the predicted error.

3. SVM outputs prediction functions that are linear in the point coordinates, which allows for more flexibility.

4. A trained SVM is very sparse in the number of training points as the classifier only needs to remember the support vector training points. The better the separability between classes, the sparsest the classifier will be.

5. The SVM formulation can be adapted to kernelised distances as long as the kernel is a Mercer kernel (i.e: positive definite).

We are given a set of p training observations $(x_1, y_1), \dots (x_p, y_p)$. The $(x_k)_k$ refer to vectors in $\mathbb{R}^n$ and the $(y_k)_k$ are labels in $\{-1, 1\}$. y_k is equal to -1 or $+1$ if sample k belongs to the negative or positive class, respectively. Our goal is to design a decision function $D : \mathbb{R}^n \to \mathbb{R}$ such that the label of any sample x is given by the sign of $D(x)$. A decision function is of the form:

$$\begin{array}{rcl} D : \mathbb{R}^n & \longrightarrow & \mathbb{R} \\ x & \longmapsto & \displaystyle\sum_{k=1}^{p} \alpha_k K(x_k, x) + b, \end{array} \tag{2.15}$$

where K is a known kernel and values of parameters α_k, b must be found. Equation 2.15 is the formulation of the decision function in the dual space. This equation can also be written in the direct space:

$$\begin{array}{rcl} D : \mathbb{R}^n & \longrightarrow & \mathbb{R} \\ x & \longmapsto & \displaystyle\sum_{i}^{N} w_i \phi_i(x) + b. \end{array} \tag{2.16}$$

Provided that kernel K from the dual formulation possess a definite or infinite expansion of the form

$$\forall x, x', \; K(x, x') = \sum_{i=1}^{N} \phi_i(x).\phi_i(x'), \tag{2.17}$$

then the direct and dual formulations are equivalent. In this case, the direct parameters w_i can be expressed as a function of the dual parameters α_k:

$$w_i = \sum_{k=1}^{p} \alpha_k \phi_i(x_k). \tag{2.18}$$

Let us denote $w = (w_1, \dots w_N)$ and $\phi(x) = (\phi_1(x), \dots \phi_N(x))$. Any decision function $D(x) = w.\phi(x) + b$ defines a hyperplane in $\mathbb{R}^n$ of equation $D(x) = 0$. It can be shown that the distance between this hyperplane and any training

sample x_k is given by $y_k D(x_k)/||w||$ (see figure 2.3 for an illustration). If the training set is separable, then the margin M between the boundary and any given training sample verifies:

$$M \leq \frac{y_k D(x)}{||w||}. \tag{2.19}$$

With a normalised weight vector ($||w|| = 1$) the bound is reached for training samples called *support vectors*. Defining an appropriate boundary is then equivalent to maximising bound M. The solution to this problem can be found by transforming the dual problem by means of the Lagrangian (see [Boser 1992] for details). It is shown that optimal values of parameters α_k are non-zero only for support vectors, which drastically reduces the complexity of the final prediction function.

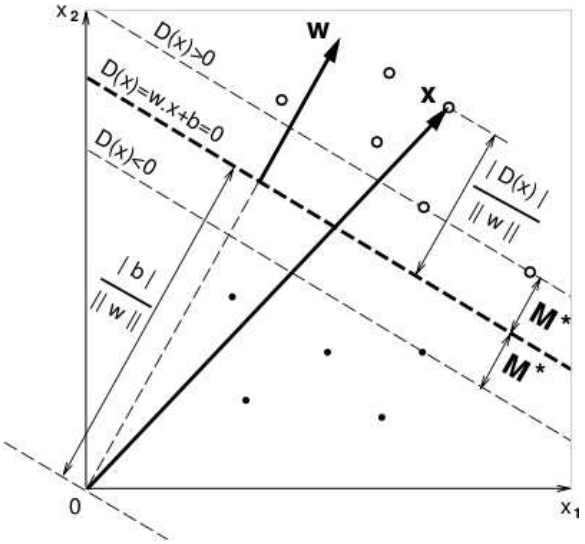

Figure 2.3: An illustration of the distance between a separating hyperplane and the training samples. Image courtesy of [Boser 1992].

When dealing with most real training datasets, the hypothesis of separability is seldom verified. In such cases, positive slack variables are introduced to make up for possible mis-classified training samples, and their weighted sum is added to the energy to be minimised. Because the weighting term can vary to the benefit of discriminativity or generalisation, SVM are a flexible training tool that can be adapted to a wide range of situations.

2.3 Graphical structures for computer vision

The use of graphical structures in computer vision can take a variety of meaning, depending on the specific scientific community. Here, we review

some uses of graphical structures related to our own work.

2.3.1 Graph matching

In our work, we infer graphical structures from images, produce a vector representation of these graphs and classify them with off-the-shelf parametric classifiers (see in particular chapter 3). However, a whole field of machine learnin is dedicated to the matching of graphs, and the computation of distances between graphs, which in turn enable graph classification. Here, we need to answer why existing graph matching solutions do not suit our own practical needs.

A subproblem of graph classification is the computation of distances between graphs. And because graphs are not ordered, linear structures, the problem of evaluating the quantitative difference between two graphs often boils down to establishing a match between the graph nodes; hence the problem of *graph matching*. Graph matching between two graphs $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ consists in finding a homomorphism $h : V_1 \to V_2 \cup \{\emptyset\}$ such that:

$$\forall v, v' \in V_1,\ (h(v), h(v')) \in E_2 \iff (v, v') \in E_1 \tag{2.20}$$

Because of the combinatorial structure of the problem, graph matching is considered very difficult. In our particular case, we are not looking for a bijective vertex-to-vertex matching, as image-inferred graphs can contain variable numbers of nodes. Moreover, as each graph node is characterised by an attribute spanning a continuous subset of $\mathbb{R}^d$, a cost function has to be added to the match of two graph nodes with different attributes. Our problem thus belongs to the category of *inexact* attribute graph matching. There are several broad categories of methods for solving this problem. Representing data under a relational form was done early on in the history of computer vision, for example in stereopsis [Boyer 1988]. One of the early solutions used for computing distances between graphs was to count the number of consistent cliques [Shapiro]. Of more particular interest is the concept of *graph edit distance*, introduced in [Sanfeliu 1983], which determines the distance between two attribute graphs as the cost of node and edge editing, deletion and addition required to transform one graph into the other. In a large sweeping movement, we can cite other methods based on tree search [Tsai 1979], genetic algorithms [Khoo 2002], probabilistic theory [Farmer 1999] and continuous optimisation [Luo 2001]. For a more detailed overview of the methods for inexact graph matching developed in the past three decades, we suggest the bibliographical reviews found in [Conte 2004, Bengoetxea 2002].

The crucial drawback common to all these methods is that they are not well suited to the kind of graphs we want to deal with. In practice, image-inferred graphs that we build from the image interest points contain

of the order of 10^3 nodes with attributes of dimensionality of the order of 10^2. This is an order of magnitude more than state-of-the art methods in graph matching. Indeed, graph matching is an NP-complete problem; and even though methods have been devised to reduce the matching cost to polynomial time [Myers 2000], existing solutions remain out of our reach if we want to include a too large number of nodes. Moreover, because of high intra-class variability, a graphical model understood in the classical sense would be extremely fuzzy. We do not claim that graph matching is not well suited to computer vision problems: a great number of challenges have been addressed with graph matching, including, but not limited to: optical character recognition, fingerprint identification, symbols recognition and image retrieval [Conte 2004]. However, available solutions based on graph matching simply do not scale to the number of graphs, nodes per graph and fuzziness of graph models required by object class recognition.

Even if most graph matching methods do not directly provide us with solutions for object class recognition, we believe that the field of spectral graph theory provides us with interesting investigation areas for the problem of graph classification. In particular, we are most interested in the representation of graphs by some of their spectral properties.

2.3.2 Spectral graph theory

The study of the spectral properties of graph takes its roots in the observation that we can represent a graph by its transition matrix and infer properties by the study of its spectrum. In order to understand the origins of this field of study, we first need to motivate and define the existence of random walks on graphs. As we are studying the properties of these random walks, we are confronted to formulations that include certain matrices related to the graph transition matrix. Eventually, the analytical evaluation of these properties require the computation of the spectrum of these matrices, and it is those results that are provided by spectral graph theory.

2.3.2.1 Random walks on graphs

We consider in this section the case of a general finite graph with weighted edges and no isolated node. We employ the same notations as in section 2.3.1. The degree function over the graph nodes is thus strictly positive. The product of the weight matrix $W = (w_{ij})_{i,j}$ and the inverse of the diagonal degree matrix T is a probability matrix $P = WD^{-1}$ that defines a Markov random walk over the graph nodes; the transition probability $p_{i,j}$ between two graph nodes i, j is proportional to the edge weight that connects them:

$$p_{i,j} = \frac{w_{i,j}}{d_i} = \frac{w_{i,j}}{\sum_k w_{i,k}} \tag{2.21}$$

In mathematical terms, we define a random walk $(Y_n)_{0\leq n}$ on graph $G = (V, E)$ started at node i_0 as follows:

$$\begin{aligned} & Y_0 = i_0, \\ \forall n > 0, \quad & P\left[Y_{n+1} = j \,|\, Y_n = i\right] = \begin{cases} \frac{w_{ij}}{\delta_i} & \text{if } e_{ij} \in E \\ 0 & \text{otherwise} \end{cases}, \end{aligned} \tag{2.22}$$

where $\delta_i = \sum_{j \in N(i)} w_{ij}$ is the degree of node i.

In order to understand the purpose of random walks in graph theory, one needs to consider the definition of a graph transition matrix as equivalent to the definition of the topology of a finite set of points. Indeed, defining the topology of a space consists in assigning to this space a distance (or similarity) measure, or, in the case of a finite set of points, a matrix of distances. Intuitively, the transition matrix of a graph can be considered as the similarity matrix of the set of graph nodes, and the operation of building a strong edge between two nodes can be viewed as moving those nodes closer to each other in a certain topological space. Walking randomly on the set of graph nodes following the transition probability matrix can thus be viewed as an exploratory process in a space of unknown topology; we shall see how properties of the random walk will serve to expose properties of a topological space, such as connectivity and the distance between any two points.

2.3.2.2 Random walk parameters: hitting time and commute time

Several interesting parameter values of a graph random walk can be distinguished: the hitting time, and the associated commute time, the cover time and the mixing rate [Lovász 1993]. We detail here the hitting and commute time parameters.

Definition 1 (Hitting time and commute time) *In a finite symmetric weighted graph, the* hitting time $H_T(i, j)$*, or* access time*, from node i to node j is the average number of steps of a random walk started at node i required to reach node j for the first time:*

$$H_T(i, j) = E\left[min\left\{n : Y_n = j\right\} \,|\, Y_0 = i\right]. \tag{2.23}$$

We call commute time *the "symmetrized" hitting time:*

$$C_T(i, j) = H_T(i, j) + H_T(j, i). \tag{2.24}$$

In other words, the commute time distance between nodes i and j is the average number of steps of a random walk on the graph nodes started at node i required to reach node j for the first time and return to node i for the first time.

Properties of hitting time matrices can be studied through the lens of probability theory and notable inequality results have been thus produced (see e.g. [Lovász 1993]), notably concerning bounds on the random walk parameters. However, the simple observation that the probability of a random walk started at i is in j after t steps is $P^t(i,j)$ bears the promise that the eigenvectors of the transition probability matrix will play a key role in the analytical evaluation of the random walk parameters.

2.3.2.3 Graph Laplacian and links to the graph spectrum

Since we consider a graph as a discrete space, we can generalise the definition of operators on scalar fields to operators on graphs. Of particular interest is the definition of the discrete Laplacian operator, which is one of the basic differential operators. Considering a function (or signal) $f : V \to \mathbb{R}$ on the graph nodes, the Laplacian operator applied to f at node i can be viewed as the distance of $f(i)$ to the average value of f on the nodes around i:

$$\Delta f(i) = \sum_j (f(i) - f(j))p_{ij} \tag{2.25}$$

$$= f(i) - \sum_j f(j)p_{ij}, \tag{2.26}$$

where $p_{i,j}$ is the transition probability from equation 2.21. In matrix form, we thus have the following formulation:

$$\Delta = D - W. \tag{2.27}$$

It can be shown that the Laplacian operator is symmetric positive definite and can be decomposed on an orthonormal basis of eigenvectors with positive eigenvalues (see e.g [Luxburg 2007]). It is the study of this decomposition that constitute the body of spectral graph theory.

An important result links the Laplacian operator to the hitting time matrix [Aldous 1995]:

$$\Delta H_T = J - volT^{-1}, \tag{2.28}$$

where $vol = \sum_i d_i$ is the volume of the graph, J is the all 1 matrix and T the diagonal matrix with entries δ_i.

2.3.2.4 Spectral clustering and image segmentation

The prime use of spectral graph theory is spectral clustering, which deals with the clustering of data that was arranged in a graph structure and then embedded in a space that reflects the topology of the graph structure [Luxburg 2007].

Applications to computer vision quickly appeared: these applications took advantage of the natural combination of locality preserving embedding and point clustering (usually k-means) for image segmentation: in

[Shi 2000], the normalised cut z, which is a cut that minimises a normalised sum of edge weights is found to be a solution to the equation:

$$D^{-1/2}\Delta D^{-1/2}z = \lambda z \tag{2.29}$$

with λ minimal. The solution to an optimal bipartition of the graph is thus the eigenvector of $D^{-1/2}\Delta D^{-1/2}$ that corresponds to the smallest, non-zero eigenvalue. [Meila 2000] later established the connection of this approach to random walks.

In [Belkin 2001], the eigenfunction corresponding to the lowest non-zero eigenvalue of the discrete Laplace-Beltrami operator on a graph outputs a one-dimensional embedding of the graph nodes that can then be used for clustering.

It was later found that although [Shi 2000, Belkin 2001] relied on just one eigenvector of the graph Laplacian to produce a cut or a clustering of the node set, it was also possible to make use of the whole set of eigenvectors to map the graph nodes to a space of arbitrary dimension [Ng 2001]. As we will later see in the case of commute time embedding, coordinate k (with k less than the number of graph nodes) of the mapping for node i can be set to coordinate i of eigenvector k, up to a normalisation factor. This observation will help us draw two different conclusions:

1. Eigenvectors of the graph Laplacian can be viewed as functions on the graph nodes, hence the denomination *eigenfunctions*.

2. The typical dimensionality of an undirected graph representation is of the order of $N(N-1)$, where N is the number of nodes in the graph. This very large dimensionality will lead us to the conclusion that a graph must be somehow simplified before it can be represented by a vector of tractable size in a vector space.

More recently, the spectral embedding and theory developed in these early years have been applied to video tracking, brain dynamic prediction [Meyer 2007] and satellite image classification [Unsalan 2005].

It should also be noted that, although these works all rely on the spectrum of the graph Laplacian, other properties can be found from the spectrum of the graph affinity matrix. Among the pioneers, [Scott 1990] worked to improved the block structure of the affinity matrix to obtain cleaner clustering. Later, Sarkar and Boyer [Sarkar 1998] on one hand, and Perona and Freeman on the other [Freeman 1998] have used the first and second largest eigenvector of the affinity matrix to bipartition graph nodes.

Of more particular interest to us is the work of [Qiu 2007], in which the authors show that the mapping of the graph nodes associated to the commute time distance is more reliable than the normalised cut of Shi and Malik; they then proceed to present several different applications for image segmentation and video tracking.

2.3.2.5 Connection to multi-dimensional scaling and Isomap

Multidimensional scaling (MDS, [Kruskal 1978]) refers to the embedding of input data described by a similarity matrix in a space of arbitrary dimension. The goal of MDS is to provide an embedding that conserves the similarity matrix. In other words, the distance function between data points must remain stable with the embedding. This problem is usually solved by minimisation of a certain cost function related to the input similarity matrix. Different resolution methods exist for different distance metrics.

Spectral graph theory provides solutions for certain categories of problems: the similarity matrix of the input data allows us to draw an undirected graph of points, where the transition weight matrix W is equal to the similarity matrix. The commute time embedding (for which mathematical details will be given in section 3.3.3) produces an embedding that preserves the commute time distance. Similarly, Isomap embedding (see section 3.3.2.1) preserves the shortest path distance. Thus, both these embeddings achieve the purpose stated by MDS.

2.4 Part-based models

Early successful methods for image understanding aimed at integrating a large amount of information inside visual models. Thus, instead of considering visual data as an arbitrary signal, one of the prominent preoccupations was to relate physical, 3D objects to their 2D projection. The part-based model [Fischler 1973], which later spawned the star-graph [Felzenszwalb 2003, Felzenszwalb 2008] and constellation models [Burl 1996, Weber 2000, Fergus 2003] is an example of that trend that consists in modelling many interactions between image parts. Though we must emphasise that the part-based model considerably differs with our approach, we feel compelled to describe it here because it efficiently models object parts organised in a graphical structure somewhat similar to ours. A large body of the literature deals with part-based models; here we highlight only the most prominent pieces of work and in no case aim for completeness.

The model initially described in [Fischler 1973], and later [Burl 1996], states that an object can be decomposed in N characteristic features, where N is known in advance. Here, the "feature" denomination not only refers to image features, but also to object components. For example, a face is composed of two eyes, one nose and one mouth. The part-based approach models the *appearance* of parts, independently from one another, as well as the layout of the parts, also called the *shape*. In the testing phase, the appearance and shape terms are added to constitute an energy that should be minimised. In general, the shape term itself is a sum of pairwise prior knowledge terms between the different parts: each part has a certain "opinion" on where certain other parts should be located. For instance, in [Fischler 1973],

the location of a face part is given by a spring with a certain length and flexibility. Compressing or extending the spring adds a positive term to the energy to be minimised (see figure 2.4, courtesy of [Fischler 1973]).

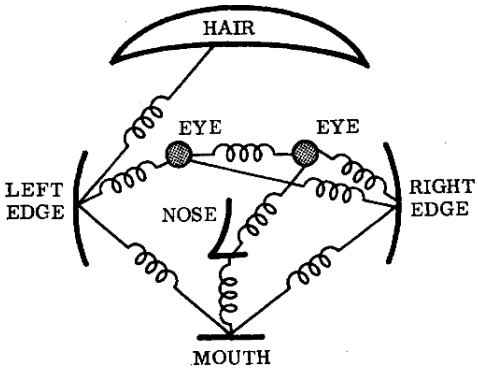

Figure 2.4: The part-based model, as introduced in [Fischler 1973]

In practice, works on part-based models cited above have focused on methods to select visually relevant features, estimate a "good" prior of the feature appearance, model realistic interactions between object parts, learn the model and detect one or multiple object instances in test images.

The general part-based approach proceeds as follows: each one of the object parts is described by a visual feature and appropriate detectors are designed for each feature. In each image, a set of candidate locations for each feature are found and hypotheses are made: each hypothesis assigns a feature to one of its candidate locations, or to none at all (to make up for possible occlusions). The likelihood of each hypothesis is then evaluated by taking into account pairwise interactions between features. Thus, the model takes into account both the appearance of an object (through the appearance of the local parts) and its shape (through the arrangements of parts relatively to one another). [Weber 2000] addresses the problem of automatically finding likely feature locations and formulating more convincing layout hypothesis. The feature selection process is done by quantising the space of visual features and selecting the clusters that are most relevant to the researched features. [Fergus 2003] improves over [Burl 1996, Weber 2000] by modelling the appearance variability of object parts.

Part-based model names are given according to the shape of the graph that describes the interactions between object parts. The star-graph [Felzenszwalb 2003, Felzenszwalb 2008] considers that each object has a central part to which auxiliary, smaller scale parts are connected. On the other hand, the constellation model [Weber 2000, Fergus 2003] only makes assumptions regarding the global shape of the object parts.

The part-based model still enjoys a vigorous popularity today. Felzen-

szwalb et al. [Felzenszwalb 2008] combine the star model with histogram of gradient (HOG, [Dalal 2005]) features and a generalisation of SVM to predict quickly and accurately the location of objects and object parts in the challenging PASCAL VOC 2006 dataset [Everingham 2006]. In [Felzenszwalb 2008], authors have managed to formulate the problems of model training and part detection as a variant of support vector machine (SVM), coined latent SVM (LSVM). Consequently, both their training and testing phases are relatively fast. Moreover, one of the contributions of [Felzenszwalb 2008] over the spring-model of [Fischler 1973] is that the coefficients that quantify the interaction between two different parts are allowed to take negative values, as they are in fact weighting coefficients of an SVM.

In this thesis, we have favoured conceptually weaker models than the part-based models. The — arguably questionable — reason behind this strategic decision is that databases of today labelled images are likely to grow in size and diversity in the future. Thus, we believe that methods that rely on the comparison to many training samples bears good promises. It should not be necessary to explicitly model parts appearance and relationships, because the amount of data should allow us to make up for intra-class variability.

However, several essential comparisons can be drawn between our work and part-based models: first, we rely on local visual features to populate the nodes of our visual graphs. Second, we employ connections between visual features and capture information about these connections for classification. However, this information is not directly stored inside a model, but indirectly, through the image representation.

2.5 Summary

Since our methods rely on fundamental tools coming from different communities, we can cite a great number of works that bear some similarity, or share a common goal with our own work, over a relatively long time span. We briefly sum up here the connections between our work and prior art.

Firstly, our methods strongly rely on the sampling of visual features, as opposed to global features. In this sense we can say that we design bottom-up approaches, as we infer an image label from its local properties.

The visual features we extract are arranged into graphical structures from which we infer the image labels. However, our work should be firmly distinguished from part-based models, as we do not model the pairwise connections themselves.

Another pitfall would be to classify our work among graph matching methods. Indeed we perform graph classification, in a certain way, as in graph matching, but by no means do we perform one-to-one, or even many-to-many node matching. We are dealing with large numbers of graph nodes

and attributes of high dimensionality: two features that strongly distinguish us from the graph matching community. Moreover, we are using tools coming from the field of spectral graph theory, but we are diverting them from their initial purpose — clustering.

Finally, we take a strong stance in favour of unquantised features, when possible, as opposed to a popular tendency of the compute vision community to quantify the feature space. Our work builds on machine learning methods for unquantised features, notably kernels on feature sets, and apply them to the problem of image recognition.

The goal of this thesis is to address two limitations of existing object recognition methods: our first goal is to design an image representation that intrinsically contains information concerning the layout of the visual features, while existing methods usually rely on combinations of classifiers and features. Our second goal is to avoid discarding information regarding the visual features themselves by quantisation of the feature space. The latter objective is all the more challenging that state-of-the-art classifiers all rely on combinations of quantised features.

Chapter 3

Graphical structures for image representation

3.1 Introduction

In the most simple discriminative framework, an image is represented by a vector of fixed, finite dimension and appropriate parametric classifiers are trained over the set of training vectors. This is the context of this chapter, in which we introduce a novel kind of image representation. This image representation is based on properties of an image-inferred graph of visual features. Because this graph captures spatial relationships between image feature points, information related to the layout of the feature points is intrinsically contained in the image representation. We observe a performance gain over orderless feature-based representations.

A valuable property of digital images is that, like many mathematical objects, they can be seen as an assembly of multiple components, or sub-images. Thus, in computer vision, studying the statistical properties of classes of images is a problem that can decomposed and simplified by considering: (1) the properties of the image components and (2) the relationships between the different image components relatively to the image class. In practice, image components can take a wide range of forms. In this chapter, as in many recent image classification approaches, we shall decompose an image in visual features sampled at a sparse set of image locations. Each visual feature is then represented by a finite-dimensional descriptor that captures certain local properties of the image around the feature location. Sampling these feature points is a problem for which a great variety of solutions already exist. However, representing the interactions between these feature points is still an open issue.

The objective of this chapter is to describe a novel method for the representation of the interactions between visual interest points. We will focus on methods that take into account the layout of interest points relatively to their class for efficient image classification. Indeed, we argue that orderless image representation methods are not sufficiently discriminative to meet up actual performance standards. In this sense, our work follows the current trend to introduce some knowledge about the spatial layout of interest points in the image to produce state of the art results [Agarwal 2006, Lazebnik 2006]. However, we do so in principled grounds,

while the cited methods usually improve on the popular visual bag of words, which established its reputation precisely by discarding the information related to the layout of the visual interest points.

Integrating the layout of the interest points in an image representation raises several challenges; one of them is that the representation must preserve the various geometric invariances to which the object classes are subject. For instance, in image labelling, a decision has to be taken whether an image contains an instance of the positive class or not. Usually, the positive class can present quite high levels of intra-class variation. In particular, its 3D location and pose are can vary greatly from one image to another. Therefore, the *absolute* interest point coordinates in the image are not really relevant to the positive class. However, the *relative* positions of interest points, given their appearance, are a useful source of information: "for an instance of the positive class, what is the typical distance between interest points with appearance A and interest points with appearance B?" Providing a quantitative answer to this question for various values of A and B gives useful information about the image content.

The second major difficulty associated to the aggregation of information related to the layout of interest points is that its output must be a mathematical representation that can ultimately be used to train a statistical classifier. We are thus faced with a compromise: how to represent complex local and global interactions between large numbers of points by a mathematical object lying in a (relatively simple) Hilbert space?

We tackle the first issue mentioned above in a natural way, by organising the visual features extracted from the image in an image-inferred *visual feature graph*. This visual feature graph is robust to several rigid geometric transforms: scale, translation, and orientation. In a visual feature graph, nodes correspond to the image interest points and edges reflect a certain image proximity between points. Therefore, a visual feature graph also possesses the same radiometric invariances as the visual features it is based on. Representing the appearance and spatial relationships of interest points is then equivalent to studying the properties of the corresponding visual feature graph; we are then confronted to the second difficulty mentioned above: how to represent an attribute graph of variable size by a finite-dimensional vector?

We observe that a visual feature graph, just like any attribute graph, is completely defined by a set of attributes, which correspond to the graph nodes, and a matrix of distances between the graph nodes (section 3.3). By quantising the image visual features, we can "regroup" (in a sense that will be defined) the graph nodes that possess the same quantised descriptor. By so doing, we obtain a graph with a constant number of nodes. Its distance matrix can now be used as the image representation. We show experimentally that classification performances benefit from the information contained in the graph distance matrix.

The remainder of this chapter is organised as follows: in section 3.2 we introduce two different ways to organise visual features in a visual feature graph. This visual feature graph can be represented by its distance matrix, but graph distances can be measured differently. In section 3.3 we list three possible distance measures. Sections 3.4 and 3.5 shows how to "collapse" a feature graph so as to obtain comparable graph representations. Classification experiments that make use of this representation are described in section 3.6.

3.2 Construction of graphical structures

We present here a variety of methods which produce graphical arrangements of feature points that are robust to the rigid transforms enumerated above. We should here distance ourselves from part-based approaches for object recognition (see section 2.4). We emphasise that our goal is not to build a graph that is optimal with respect to a certain graph model. Instead, we place ourselves in a discriminative context: visual graphs are directly inferred from the image content. A classifier will then be built on the set of training graphs and applied on test graphs to predict the test graph labels. For clarity, the reader should be aware that we are not concerned by the design of the classifier, as we will use an off-the-shelf classifier such as Adaboost or kernel SVM. Instead, the emphasis is laid on the production of repetitive, discriminative image (i.e: graph) representations.

3.2.1 Motivation

In order to motivate our approach, we employ a synthetic dataset of images composed of two classes, the positive and the negative class, as described in appendix A.1. This synthetic example was designed so that the global image colour histograms are undifferentiated for the positive and the negative class. In average, images from the positive and the negative class contain an equal amount of points from all four colours. However, it can be observed that in the positive class blue points are nearer to green points and pink points are nearer to orange points. In the negative class, orange points are closer to green points and pink points are nearer to blue points. The only way to differentiate images from the two classes is to take into account the relative spatial proximity of the different colours: in a given test image, what is the average spatial distance between points of colour A and points of colour B?

This synthetic example is a simplified version of what can be observed in real image datasets, such as satellite images: in urban areas, trees might border roads, but the presence of trees does not necessarily imply the presence of roads, since trees can also be found in parks. Thus, in classification, the relative proximity of "tree" features to "road" features is important.

Which mathematical object could represent the information of relative proximity between different kinds of feature points? We argue that graphs of feature points are well suited to this task. In such graphs, nodes are feature points and node connections reflect spatial proximity.

Naturally, graphical structures based on interest points do not solve the problem of image representation, or image classification by themselves, but they provide an adequate mathematical representation for the image content, and image properties should be observable in the corresponding graph structure. The next problem will then be to quantitatively evaluate these properties. For now, we only show how to infer graphical structures from sets of visual feature points.

3.2.2 Mathematical notations and conventions

In the remaining, we will usually denote $G = (V, E)$ a directed, weighted, attribute graph, where V is the finite set of graph *vertices*, or *nodes*, and E is the set of *edges*, or *links*. Two graph nodes v_i, $v_j \in V$ are *connected* if e_{ij} belongs to E. In this case where nodes i and j are connected, the *weight* $w_{ij} \in \mathbb{R}$ of edge e_{ij} is strictly positive. The case where v_i and v_j are not connected is equivalent to the edge weight w_{ij} being equal to zero. In the case where the graph is oriented, the weight matrix is symmetric: $\forall i, j,\ w_{ij} = w_{ji}$.

We call *neighbourhood* of a vertex v_i the set of vertices to which v_i is connected:

$$\forall i,\ \mathcal{N}(i) = \{j | w_{ij} > 0\}\,. \tag{3.1}$$

The *degree* of a vertex v_i is equal to the sum of the weights of the edges leaving v_i:

$$\forall i,\ \delta_i = \sum_{j \in \mathcal{N}(\rangle)} w_{ij}. \tag{3.2}$$

The *volume* of the graph is equal to the sum of all node degrees:

$$vol = \sum_i \delta_i = \sum_i \sum_{j \in N(i)} w_{ij}. \tag{3.3}$$

Moreover, in the following vertices will be associated to visual features (typically: SIFT features [Lowe 2003]). Thus, a descriptor attribute $d_i \in \mathbb{R}^d$ will be associated to each vertex v_i.

3.2.3 The visual feature graph

In order to evaluate the relative distances between groups of interest points, we propose to build a graph structure in which vertices are interest points and edges represent spatial proximity. We call it: *visual feature graph*. Constructing a representation of an image will then be equivalent to producing a representation of its feature graph.

In the following, we present two different ways to infer different visual feature graphs from image visual features. Both graph construction approaches depend on one or two fixed parameters to make it more general in scope. We shall not provide a method for automated parameter selection. Sensitivity to parameter selection will be discussed in the experiments section 3.6.

3.2.3.1 Hierarchical feature graph

The connectivity of a feature graph should be repeatable across the various instances of the image class: as we are dealing with real images, the objects they contain may exhibit variations in pose and appearance. In particular, because of scale variations, the connection of nearby interest points should be decided relatively to the scale of the interest points. We thus adopt the following graph construction process: given a fixed *scale ratio* α, we connect two points i, j with image coordinates (x_i, y_i) and (x_j, y_j) and scales σ_i and σ_j if and only if: $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} < \alpha \max(\sigma_i, \sigma_j)$. Thus, the higher α, the higher the graph connectivity. Typical values of α are within the $[1, 10]$ range. An example of such a hierarchical feature graph is given in figure 3.1.

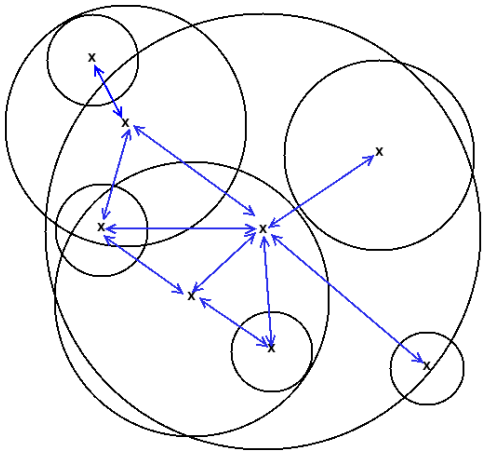

Figure 3.1: Toy example of a hierarchical feature graph. Circles have a radius equal to $\alpha\sigma$, where σ is the detected scale of the interest point and α the feature graph scale ratio. α is a parameter of the method.

Hierarchical feature graphs are unweighted ($\forall i, j,\ w_{i,j} \in \{0, 1\}$) and unoriented.

3.2.3.2 Similarity feature graph

An alternative to the graph construction process described in the above section is to try to label nodes that are likely to belong to the same object, or object part. We consider that object parts are textured regions that display a certain uniformity in appearance over a small spatial extent. Our goal becomes then to link nodes that are located close to one another and that have similar descriptors. We will therefore connect graph nodes for which a certain distance function Δ of the spatial and content proximity will be small. We chose to define this distance Δ between nodes i and j as the weighted product of their normalised spatial distance and their descriptor distance:

$$\Delta(i,j) = (\Delta_{desc}(i,j))^{\beta} (\Delta_{geo}(i,j))^{1-\beta}, \quad (3.4)$$

$$\Delta_{desc}(i,j) = ||d_i - d_j||, \quad (3.5)$$

$$\Delta_{geo}(i,j) = \sqrt{\frac{(x_i - x_j)^2 + (y_i - y_j)^2}{\sigma_i \sigma_j}}, \quad (3.6)$$

where we remind that d_i is the descriptor attribute associated to node i.

Parameter β can be adjusted to construct feature graphs that depend more or less on the spatial layout and the descriptors similarity. Its optimal value will depend on the image classes (see section 3.6). Naturally the definition of distance Δ_{desc} depends on the type of features and could be chosen to be a sum of squared differences or a χ^2 distance for instance (see [Zhang 2007] for a performance review of the different possible distances. Note also that the definition of Δ proposed in this paper can be amended to encode other types of distances between features as well.

The Δ distance[1] will be used to determine the presence of edges between graph nodes as well as their weight: we connect each node to its M closest neighbours and each edge weight is defined as $w(i,j) = e^{\frac{-\Delta(i,j)}{\sigma}}$, where σ is a normalisation factor chosen appropriately (in practice σ depends only on the descriptor distance Δ_{desc}). It should be noted that each node is connected to *at least* M other nodes, as M-nearest relationships are not necessarily symmetrical.

There are thus two parameters for the construction of a similarity feature graph. β indicates whether spatial proximity or appearance similarity should be privileged i the graph construction. M governs the connectivity of the feature graph: when the value of M is above the number of graph nodes, the graph becomes fully connected. When M becomes equal to 0, the set of graph edges is empty. An evaluation of the impact of these two parameters on the performances will be given in section 3.6.3.

[1] Δ does not satisfy the requirements of a metric, but we shall nonetheless use the term "distance" for its convenience.

There are several differences between a similarity feature graph and a hierarchical feature graph. First of all, we made the similarity feature graph a weighted graph while a hierarchical graph is unweighted. Moreover, because the similarity feature graph relies on the M-nearest neighbour distance measure, it is more sensitive to the addition and removal of feature points. In particular, the cropped and discretised nature of an image scale-space can cause the removal and addition of feature points. Thus, while the similarity feature graph is scale-invariant in theory, it is sensitive to scale changes in practice. Experiences will show in which cases establishing stronger connections between regions of similar appearance is an appropriate choice.

3.2.4 Affine invariance

The result of any of the two graph construction processes described above is an unoriented attribute graph that is invariant to scale transform. The graph will be rotation invariant if and only if the descriptors of the visual interest points is rotation invariant. Similarly, in both feature graphs, the graph connectivity can be adapted to the case where affine invariance is required. Intuitively, the graph connection process remains the same except that the circles of figure 3.1 are replaced by ellipses for the hierarchical graph. For the similarity graph, the geographic distance becomes a function of the orientation of the feature points. This is essentially the approach taken by [Ovsjanikov 2009] to represent 3D shapes with a wide range of invariances.

3.3 Distance measures in graphical structures

As mentioned in the introduction to this chapter, our goal is to produce a representation of the feature graphs that integrates the relative distances between the graph nodes. In terms of graphs, the notion of distance between any two nodes is related to the notion of connectivity: nodes that are separated by a small number of strongly weighted edges should be considered close to one another. On the opposite, nodes that belong to disconnected subgraphs should be separated by infinite distance. Based on these principles, different distance measure can be defined inside a feature graph.

The distance functions we present here exhibit the properties of metrics (they are positive definite, symmetric and respect the triangular inequality) in the case of unoriented graphs; however, this is not a major concern in our case. As we will later see, we could just as well employ distances that would not possess the properties of metrics.

3.3.1 Adjacency matrix distance

The binary distance d_τ based on the graph adjacency considers that two graph nodes are located at a finite, fixed distance if they are connected to one another, and located infinitely far from one another otherwise:

$$d_\tau(i,j) = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{if } i \neq j \text{ and } e_{ij} \in E \\ \infty & \text{otherwise.} \end{cases} \tag{3.7}$$

In this case the distance is considerably sensitive to variations in the graph construction process: a slight spatial displacement of an interest point can cause new connections to be added and thus result in large variations of the distance measure. Moreover, this distance measure ignores variations of edge weights: weakly connected nodes will be separated by the same distance as strongly connected nodes.

3.3.2 Shortest path distance

The shortest path distance is the minimum number of edges of a path linking two nodes; in a connected graph it can be computed recursively:

$$d_{SP}(i,j) = \begin{cases} 0 \text{ if } i = j \\ 1 \text{ if } i \neq j \text{ and } e_{ij} \in E \\ \min_{k \in N(i)} d_{SP}(k,j) + 1 \text{ otherwise.} \end{cases} \tag{3.8}$$

The shortest path distance d_{SP} is more robust to changes in the graph structure than the adjacency matrix distance d_τ. Notice that, just as the adjacency matrix distance, the shortest path distance ignores edge weights. As we will see, the shortest path distance not only remains sensitive to node addition and removal, but it it also lacks uniformity, as it can only take integer values. This will become more apparent once we visualise the output of a multidimensional scaling embedding based on the shortest path distance.

3.3.2.1 Isomap embedding

The shortest path distance was applied to spectral clustering in the context of Isomap. Isomap is a non-linear projection technique, mainly used for dimensionality reduction [Tenenbaum 2000]. It consists in constructing a graph of points, either by k nearest neighbours, or r-balls. Then, classic multidimensional scaling (MDS) is applied to the matrix of shortest path distances to project the nodes of the graph in a space of arbitrary dimension. Though the graph construction process takes a large part of the argumentation developed in [Tenenbaum 2000], we argue that the graph structure should not depend on the embedding strategy that will be utilised. We can

thus adapt the embedding technique described in their work to our own hierarchical graph structure. MDS consists in finding the embedding of the graph nodes in a vector space with L^2 norm that best preserves the distance matrix (this distance matrix thus becomes a Gramian matrix).

We show in figure 3.2 (top image) an illustration of an Isomap embedding. We built the hierarchical graph of an image, computed the shortest path distance matrix and the embedding of the graph nodes in a space of dimension 3. The new coordinates of the graph nodes are represented as (R,G,B) triplets and allow us to visualise the typical behaviour of the shortest path distance.

3.3.3 Commute time distance

Given two nodes i and j, the values of $d_\tau(i,j)$ and $d_{SP}(i,j)$ are both unaffected by paths that might connect node i to node j outside of the direct transition or the shortest path. In other words, if many paths of equal shortest length connect i to j are added to the graph, these distance measures will not vary. It might be desirable to make use of more information regarding the various paths that can connect two points; for instance, the average path length could be measured. This is precisely what graph commute times compute.

With the notations introduced in section 2.3.2.1, we are interested in evaluating the commute time matrix C_T of the visual feature graph.

Spectral graph theory has linked the hitting time matrix, and thus the commute time matrix, to the graph Laplacian matrix Δ, which is defined as:

$$\forall i,j \in [1,N]\,,\ \Delta(i,j) = \begin{cases} 1 - \frac{w_{ii}}{d_i} & \text{if } i = j \\ \frac{-w_{ij}}{d_i} & \text{if } i \neq j \end{cases}. \tag{3.9}$$

We briefly outline here the reasoning that leads to the expression of the commute time matrix of equation 3.13. We recalled in equation 2.28 of chapter 2 that it is possible to express the hitting time matrix as a function of the Laplacian operator (see [Aldous 1995]):

$$\Delta H_T = J - volT^{-1}, \tag{3.10}$$

where T is the diagonal degree matrix.

On the other hand, an expression of the normalised Green function can be obtained as a function of the eigenvectors $(\phi_1, \dots \phi_N)$ and eigenvalues $(\lambda_1, \dots \lambda_N)$ of the normalized Laplacian $\mathcal{L} = T^{1/2}\Delta T^{-1/2}$; keeping in mind that, because the graph is finite and connected, $\mathcal{L}$ has exactly one zero eigenvalue and all other eigenvalues are strictly positive, we can write: $0 = \lambda_0 < \lambda_1 \leq \dots \leq \lambda_{N-1}$ (see [Chung 1997]) and:

$$T^{-1/2}GT^{1/2} = \sum_{i>0} \frac{1}{\lambda_i}\phi_i^*\phi_i \tag{3.11}$$

Combining equations 3.10 and 3.11, we can show that the hitting time matrix verifies:

$$Q(x,y) = \frac{vol}{d_x} G(y,y) - \frac{vol}{d_y} G(x,y) \tag{3.12}$$

and the final expression of the commute time matrix follows from equations 3.11 and 3.12 ([Chung 2000]):

$$\forall i,j,\ CT(i,j) = vol \sum_{k=2}^{N} \frac{1}{\lambda_k} \left(\frac{\phi_k(i)}{\sqrt{d_i}} - \frac{\phi_k(j)}{\sqrt{d_j}} \right)^2 \tag{3.13}$$

where $\phi_k(i)$ denotes the i^{th} coordinate of eigenvector k. Thus, the only operation required to compute the commute time matrix is the extraction of the eigenvectors and eigenvalues of $\mathcal{L}$.

As emphasised by [Meyer 2007], it is possible to view the eigenvectors (ϕ_k) of $\mathcal{L}$ as functions on the vertices of the graph. In this light, equation 3.13 can be considered as an L^2 distance function between vectors of coordinates $e_i \sqrt{\frac{vol}{d_i}} \left(\frac{\phi_2(i)}{\sqrt{\lambda_2}}, \ldots \frac{\phi_N(i)}{\sqrt{\lambda_N}} \right) \in \mathbb{R}^{N-1}$. In equation 3.13 we can neglect the terms corresponding to high eigenvalues (low values of $\frac{1}{\lambda_k}$) and obtain an embedding of the graph nodes in a space of arbitrary dimension inferior to $N-1$. The sharper the increase of the sequence $(\lambda_k)_{1<k\leq N}$ the better the approximation.

The *commute time embedding* of the graph nodes is in fact similar to the Isomap embedding reminded in section 3.3.2: the graph nodes are projected in a Euclidean space of finite dimension in which the distance matrix is an approximation of the commute time distance matrix. Figure 3.2 provides an illustration of the embedding of an image graph nodes in a three-dimensional space with the shortest path distance (top image) and the commute time distance (bottom image): each graph node is embedded in $\mathbb{R}^3$, by Isomap or commute time embedding. The three embedding coordinates are then assimilated to an RGB colour. Therefore, points with similar colours share nearby locations in the projection space, and are separated by a small distance in the feature graph. What we observe is that despite the very different manners in which we measure distance in the two experiments, we obtain embeddings that are remarkably similar; the main difference is that the commute time distance seems smoother across the graph nodes. This stronger uniformity reflects the robustness of the commute time distance measure to slight variations in the graph structure, and thus, we may expect, to intra class variations.

It should be noted that because the Isomap and commute time embeddings are only unique *modulo* an arbitrary rotation and translation factors, we had to centre, re-scale and re-orient the distance measures to be able to visually compare the two embeddings.

Figure 3.2: The image feature graph is constructed following section 3.2.3.1. The nodes of the feature graph are in each case embedded in $\mathbb{R}^3$ using either the shortest path distance (section 3.3.2, top image) or the commute time distance (section 3.3.3, bottom image). The 3D coordinates are represented here as (R,G,B) values. (Graph edges are not shown for the sake of readability, image courtesy of [Field 2006])

3.4 Appearance-collapsed graphs

In the previous sections we have described a repeatable image-inferred graph construction process (section 3.2.3) as well as three different metrics to evaluate the distances between graph nodes (section 3.3). For each feature graph, the graph distance matrices $(d_{ij})_{i,j}$ thus produced are arguably good representations of the graph structure. However, they cannot be used as such to compare different graphs since they depend on the number of graph nodes as well as their ordering. In this section we introduce two different ways to deal with the problem of comparing graphs with one another. The first, inspired by the bag of word approach, is to group together graph nodes with similar features and to study the distance matrix of this *appearance-collapsed graph*. This is the method that will be detailed in this section. The second, which will be detailed in section 4.4, is to group together the pairs of features that are located at a comparable distance in *channels* and to

compare the distributions of features inside these channels between images: we coin this process *distance-collapse*.

To produce an appearance-collapsed graph representation, we take advantage of the idea that the space of visual features can be quantised in a finite number of cluster centres, also known as: codebook entries or texture prototypes. Capturing the layout of the feature points relatively to their appearance is the same, *modulo* feature quantisation, as capturing the relative layout of the codebook entries. The idea is thus to group the nodes of the visual feature graph that are assigned to the same codebook entry. This process of grouping graph nodes is called *graph collapse*. The distance matrix of the collapsed graph can then be used as the image representation.

More precisely: prior to the graph construction process, the space of visual descriptors is quantised by k-means in a codebook containing K elements. We denote $c_i \in [1, K]$ the index of the codebook entry nearest to vertex feature d_i. The visual feature graph is then collapsed in a weighted graph $G' = (V', E')$ with K vertices and edge weights $w'_{kl} = \sum_{i|c_=k} \sum_{j|c_j=l} w_{ij}$. A visual representation of such an appearance-collapsed graph is given in figure 3.3. The $K \times K$ distance matrix of this appearance-collapsed graph is used as the image representation.

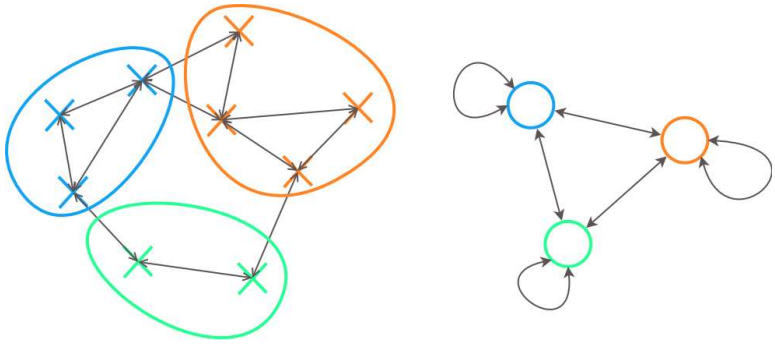

Figure 3.3: Toy appearance-collapsed graph

Obviously, the distance collapse step dismisses a certain amount of information relative to both the visual feature descriptors and the graph transition matrix. We argue that a simplification step is a benefit to the representation process. Indeed, we know, thanks to multi-dimensional scaling, that a graph of N nodes is equivalent to a cloud of N points in a Euclidean space of dimension $N-1$. Intuitively, the amount of information contained in such a structure is very large. In order to obtain a tractable, finite-dimensional representation, a certain quantity of information thus has to be discarded. Moreover, our representation allows us to discard feature noise, similarly to PCA.

The connection between the commute time distance in the collapsed

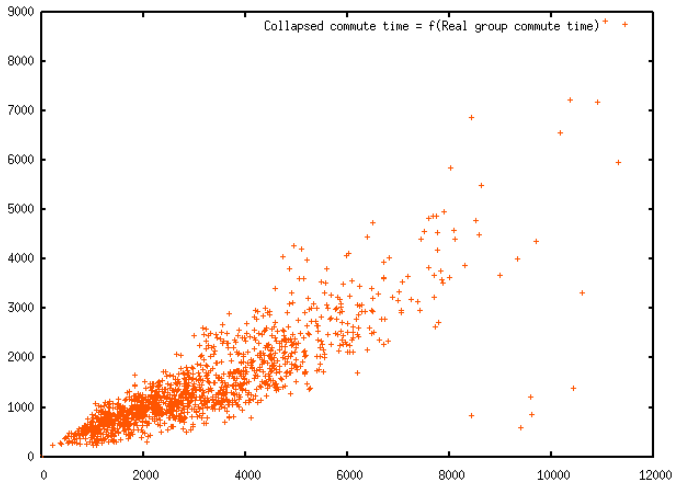

Figure 3.4: The vertical axis represents the commute times in the appearance collapsed graph. On the horizontal axis, we plot the mean commute times between groups of features: the mean commute time between groups A and B is equal to the mean number of steps required to go from A to B, and return to a node inside A. Commute times are measured empirically by randomly walking on the graph.

graph and the average commute time distance in the original feature graph is plotted on figure 3.4. We argue that the relationship between both measurements, though not linear, suffices to justify our approximation.

3.5 Classification of appearance-collapsed graphs

Thanks to the introduction of appearance-collapsed graphs, the definition of an appropriate graph construction process, together with a good graph distance measure allow us to obtain image representations that can be used for classification. In the following, we show how to process these graph representations, prior to the training step, so as to obtain better classification performances.

3.5.1 From distance to proximity

Once the distance matrix of the collapsed graph has been computed, it can be employed to train a machine learning classifier. Prior to this step, we have experimentally realised that a good normalisation of the distance matrix was crucial to obtain good performances. This normalisation is performed in three steps, at the end of which we have transformed the distance matrix into a proximity matrix of the graph nodes.

1. The distance matrix must be transformed into a proximity matrix s by taking the inverse exponential of the graph distances. The smoothing parameter σ involved in this transform is equal to the average distance:

$$\forall i, j,\ s(i,j) = \exp\left(-\frac{d(i,j)}{\sigma}\right) \tag{3.14}$$

$$\sigma = \frac{1}{K^2}\sum_{i,j} d(i,j) \tag{3.15}$$

   In these equations, *both* $s(i,j)$ and $d(i,j)$ are implicitly equal to 0 if codebook entry i or j is not present in the graph. Moreover, $s(i,i)$ is equal to one for all codebook entries i present in the image.

2. SVM works best when dealing with normalised data. Matrix d' is thus normalised by its L^2 norm:

$$\forall i, j, s'(i,j) = \frac{s(i,j)}{\sqrt{\sum_{i',j'} s(i',j')^2}} \tag{3.16}$$

3. Finally, we noticed that the diagonal of s' is proportional to the binary bag of word of the image. In order to improve the performance of the representation, we shall replace the diagonal of s' by the normalised bag of word of the image. In the case of a disconnected graph, the representation will thus be equal to the classical bag of words. Since both the diagonal and the rest of the proximity matrix s' was divided by its L^2 norm, half of the coefficients of s' are distributed on its diagonal (and the other half outside the diagonal).

In the following, we will simply refer to the appearance-collapsed graph normalised proximity matrix as the graph distance matrix.

3.5.2 Dimensionality reduction

The dimensionality of the graph distance matrix is $K(K+1)/2$, where K is the number of codebook entries from our feature dictionary. Typical values for K are of the order of 10^2 or 10^3. Consequently, the dimensionality of the graph distance matrix is of the order of 10^4 to 10^6, which is admittedly very high. The computational overhead might become an issue when we train a classifier with a great number of training samples. Whenever this issue arose, we employed dimensionality reduction by commute-time embedding: the graph distance matrix of each image is connected to its 10 nearest neighbours with a weight that is an inverse exponential of the Frobenius (L^2) distance between the two graph distance matrices. The commute time distances between images is then computed following equation 2.24. Experimentally, we saw that the first 20 coordinates were enough to guarantee

good performance. The drawback of this dimension reduction step is that the nearest neighbours of each graph distance matrix from both the training and the testing sets must be computed. Therefore, a classifier cannot be trained before the testing samples are known. Because of this problem, we did not make use of dimensionality reduction.

3.5.3 Experimental workflow

To sum up, we list here the different steps that lead to the construction of the graph distance matrix. In addition, we specify the values of the few parameters of our approach:

1. Construction of a visual feature codebook by k-means. This step is performed offline, prior to the actual classification process. In general, we deal with codebooks of size $K = 500$. Keypoint detector and descriptor are Speeded-Up Robust Features (SURF, [Bay 2006]).
2. Visual feature points are sampled from each image; each feature point is quantised by its nearest entry from the feature dictionary.
3. A visual feature graph is built in each image. We can generate either a hierarchical graph (section 3.2.3.1) or a similarity graph (section 3.2.3.2). Typical parameter values are $\alpha = 3.5$ for the hierarchical graph and $\beta = 0.5$, $M = 2$ for the similarity graph.
4. The appearance-collapsed graph is inferred from the visual feature graph by grouping all nodes that are assigned to the same codebook entry (section 3.4).
5. A distance measure is chosen (sections 3.3.1, 3.3.2, 3.3.3); the distance matrix of the collapsed graph is evaluated.
6. The distance matrix of the collapsed graph is normalised (section 3.5.1).
7. 1 vs 1 support vector machines (kernel or linear) are trained for each class pair.
8. The estimated class of a test image is the class that maximises the sum of the predictions of the various 1 vs 1 classifiers involving that class.

3.6 Experiments

We tested our approach on several datasets, including the synthetic dataset of section 3.2.1. A parametric evaluation was conducted for the different graph construction methods outlined in section 3.2. We compared our approach with the orderless bag of words representation.

3.6.1 Synthetic dataset

Our approach was tested on the synthetic dataset of section A.1 in order to solve the problem defined in section 3.2.1. Because of the nature of this synthetic dataset, we had to simplify the visual construction graph process: each point was simply connected to its five spatially nearest neighbours. Using the adjacency matrix of the collapsed graph, we obtained a good classification rate of 93.75%. Given the large amount of noise of the dataset, we consider this to be a relatively good result that validates the legitimacy of our approach. We now provide results of our approach on real datasets.

3.6.2 Binary classification

As a first real validation dataset, we employed a simple set of high-resolution (0.6m) optical panchromatic Quickbird satellite images sampled in the area of Beijing (China) (see section A.2 for details). This dataset contains two classes: urban and vegetation areas. The appearance-collapsed graph of each image from the dataset was constructed; from this we computed the commute time distance matrices. The representations were then embedded in a bidimensional space by commute time embedding (see section 3.5.2). The result of this embedding is shown on figure 3.5. As we can see, a linear classifier in just two dimensions should be sufficient to achieve near-perfect classification. As a matter of fact, the only images that were wrongly classified by our linear SVM in an embedding of dimension 20 were found to contain equal portions of both classes.

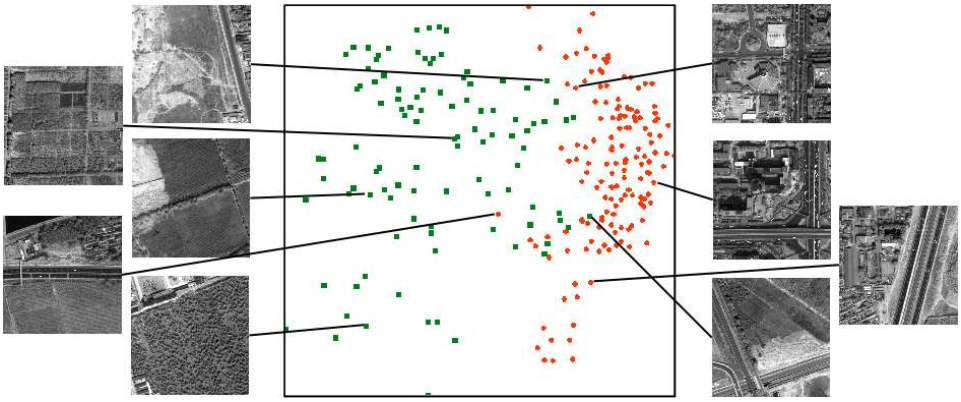

Figure 3.5: Binary classification by appearance-collapsed graph: "Vegetation" versus "Road" image classification. ($\beta = 0.5, M = 2$)

3.6.3 Parametric evaluation

We studied the influence of the graph construction parameters on actual classification performances, for both the hierarchical and similarity feature graphs. In every case, we used the commute time distance.

3.6.3.1 Hierarchical feature graph

We study the influence of the hierarchical graph construction parameter α on classification performances. We made use of the Satellite8 dataset (see appendix A.7) and selected values of α in the $[0, 10]$ range. We plotted the receiver operating characteristic (ROC) and evaluated the area under curve (AUC) for each class. Results are displayed on figure 3.6.

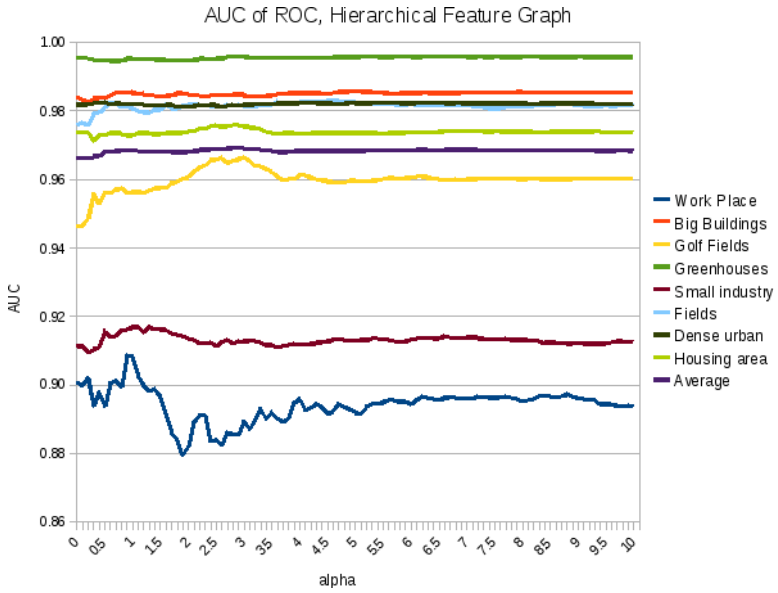

Figure 3.6: Satellite8 dataset, hierarchical feature graph, commute time distance measure. Evolution of the area under curve (AUC) of the receiver operating characteristic (ROC) as a function of graph construction parameter α.

Our first observation is that AUC scores are relatively high, with an average of more than 0.965. The influence of parameter α vary for each class and there is no uniform behaviour pattern. In fact, the optimal parameter value with respect to performance vary for each class, which is problematic. This means that in practice, we should build different graphs to discriminate between different classes. The class on which the graph construction parameter has the most influence is the golf field class; for this class, the gain over the orderless representation ($\alpha = 0$) is 0.02 for a value of $\alpha = 3.0$.

Note that the representation exhibits stable performance for $\alpha >= 4$.

3.6.3.2 Similarity feature graph

We now evaluate the impact of parameters β and M for the construction of similarity graph on the classification performance. The dataset we chose is the indoor dataset (appendix A.6).

The quantitative contribution of our approach can be observed in the classification results of the **indoor scene dataset** as a function of M, the minimum number of connections per node in the feature graph (see table 3.7 and figure 3.8). The value $M = 0$ corresponds to a binary histogram of quantized local descriptors, aka: the bag of features representation. As M is increased the feature graph becomes more connected and the information due to the layout of the different groups of nodes gains greater importance in the image representation (outside the diagonal of the graph distance matrix) relatively to the histogram of quantized features (diagonal of the graph distance matrix). We observe that an increase of M causes variations in the classification performances. These variations can be positive or negative, depending on the classes and the value of M. This reveals two phenomena: first, it shows that taking into account the image layout can raise the ambiguity between image classes that have similar bag of features representations (see bedroom and office classes). For the two others (kitchen and living room) adding spatial information only increases confusion: the content of these images is not sufficiently coherent and our image representation is an overkill compared to the simple bag of features. Second, the extent to which the proximity between image regions should be taken into account varies between classes: a low value of M means that only the interactions between regions that are both spatially close and very similar will be integrated into the image representation.

The influence of parameters β and M in the construction of the feature graph can be seen on table 3.7 and figure 3.8. A value of $\beta = 1$ means that connection between interest points will depend only on the similarity of their descriptors: this leads to feature graphs containing several disconnected subgraphs in which the most similar interest points tend to be grouped. On the contrary, a value of $\beta = 0$ means that only the spatial organisation of the interest points will decide on the connections of the feature graph. Again, this quantitative comparison shows that capturing the information of the layout of the interest points is not evenly important for all image classes. Adjusting the β parameter can lead to substantial performance gain but is not critical. We also see that performance can widely vary with values of parameter M, and no unique value of this parameter provide a positive gain for all classes. For the kitchen class, these measures confirm the previous observation that adding information about the spatial organisation in the image representation is superfluous.

M	Bedroom (108)	Kitchen (105)	Liv.Ro. (145)	Office (108)	Average (466)
0	66.2	56.67	72.66	60.00	64.63
1	74.54	43.33	66.44	49.30	59.14
2	73.61	48.57	70.93	60.93	64.20
3	72.69	48.10	69.20	63.72	63.98
4	71.30	49.05	71.97	64.19	64.85
5	72.69	49.52	70.93	61.86	64.41
6	72.22	48.57	70.93	61.86	64.09
7	72.69	50.00	70.59	62.33	64.52
8	72.69	47.62	70.93	62.79	64.20
9	72.22	49.05	71.28	61.86	64.31
10	67.13	50.00	68.51	57.21	61.40
β	Bedroom (108)	Kitchen (105)	Liv.Ro. (145)	Office (108)	Average (466)
0	72.22	46.19	68.17	62.79	62.91
0.1	71.76	48.57	69.55	62.79	63.77
0.2	72.22	48.57	69.90	61.86	63.77
0.3	70.37	48.10	69.90	61.86	63.23
0.4	71.30	50.00	69.90	61.86	63.88
0.5	70.37	50.48	70.59	60.47	63.66
0.6	71.76	47.62	70.59	62.79	63.88
0.7	72.69	48.57	71.63	63.26	64.74
0.8	71.76	50.00	69.90	60.93	63.77
0.9	70.37	51.90	69.90	63.26	64.41
1.0	69.44	52.86	69.44	60.47	62.69

Figure 3.7: R
estricted SceneClass13 dataset. Good classification (in %) as a function of the minimum number of edges per node in the feature graph M and of parameter β. The number of test images per class is indicated in brackets. β and M are set to 0.5 and 4 in the first and second experiment, respectively. See section 3.6.3 for details.

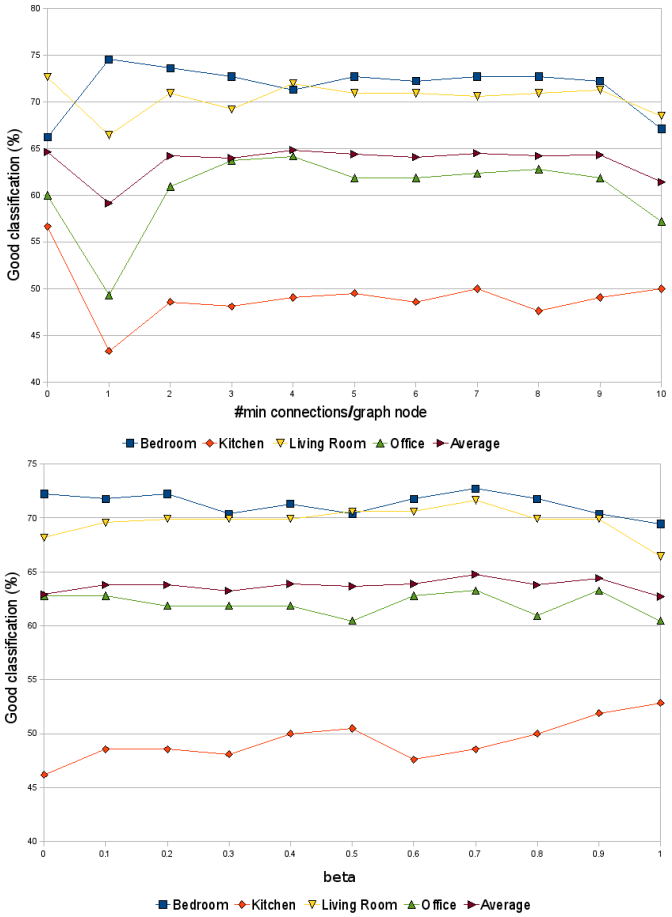

Figure 3.8: Graphic associated to table 3.7

	Hierarchical graph	Similarity graph
d_τ	62.56	63.02
d_{SP}	63.49	63.49
d_{CT}	63.49	63.49
BoW	61.83	

Table 3.1: Classification performances on the SceneClass13 dataset (see sections 3.6.4.1, appendix A.6). Results are expressed in percentage (%) of good classification.

	Hierarchical graph	Similarity graph
d_τ	85.49	85.49
d_{SP}	86.16	86.16
d_{CT}	86.16	86.38
BoW	85.27	

Table 3.2: Classification performances on the Satellite8 dataset (see sections 3.6.5, appendix A.7). Results are expressed in percentage (%) of good classification.

3.6.4 General image classification

3.6.4.1 SceneClass13

We splitted the dataset presented in appendix A.6 in two equal parts: one for training and one for testing. Performances of our approach with a single set of parameter values are reported in table 3.1. We also reported results obtained by a bag of words (BoW) classified by linear SVM. The BoW vector was normalised by its L^2 norm for best performance. Sampled feature points are SURF [Bay 2006] points that were quantised in a codebook of size 500.

Results show that the choice of graph structure or distance measure in the graph is not crucial to obtain best performances. With our approach, results vary in a range of less than ± 0.5 percentage point. In the best case, the graph structure brings an improvement of 1.76 percentage point to the bag of words. This improvement, though significative and consistent over experiments, is a bit disappointing.

3.6.5 Satellite 8 classes

On this dataset we reproduce the experimental setup described in section 3.6.4.1 for the SceneClass13 dataset. The best classification performance that we report is 86.38%, with a combination of a commute time graph distance and a similarity graph. This result should be compared to the 95.63% good classification ratio reported by [Bordes 2008].

Once again, we observe a gain provided by the introduction of the graphical structure over the orderless image representation, but the magnitude of this gain remains disappointing.

3.7 Conclusion

The purpose of this chapter was to propose and design a novel image representation that takes into account the spatial layout of the image visual content. The representation we obtain reflects the general layout of feature points relatively to one another. It consists of a matrix that contains the pairwise distances between prototype regions. The originality of our method is that distances between regions are measured inside a graph of visual features. We have described two different ways to infer this visual graph from the image content. We have also shown how different graph distance measures could equally be used to obtain the graph representation. We have then highlighted the benefits of each of these approaches compared to other, orderless approaches. Experiments clearly show the gain that can be obtained from taking into account the graphical structure of image representations. Moreover, we have shown that choosing the right graph distance is a crucial element for evaluating the graph properties. In particular, we have shown that the commute time distance is more robust to graph variability, and thus better suited to our purpose of image representation.

Our work on appearance-collapsed graphs can be extended in multiple ways. Observations that have been made with bags of words generally remain valid with graph distance matrices. For instance, it is likely a good idea to design pyramid kernels of graph distance matrices, as in [Lazebnik 2006]. Employing different codebooks for different classes is also probably beneficial [Fulkerson 2008]. It is even possible to compute one small codebook per image and to employ earth-mover distance kernels between graph distance matrices, as in [Zhang 2007]. As a possible research lead, we suggest to investigate how the graph commute time matrix changes when a certain edge weight changes. In other words, it would be interesting to study the value of $\phi_{ij}^{kl} = \partial C_T(i,j)/\partial w_{kl}$. Using an SVM classifier of commute time matrices, we could obtain in a test image the visual feature graph that best comply with the discriminative model, thanks to a gradient descent-like method.

In the course of our research, we have realised how much the classification performance can be hurt by the feature quantisation process. By building a graphical structure around visual features, we have proposed a faithful representation of the point layout. However, a large amount of information is lost in the appearance-collapse process. By quantising the sampled visual interest points, we considerably diminish their discriminative power. It seems like it would be very difficult to keep both the shape and appearance information in a single image representation. If the problem concerns the

amount of information stored in the image representation, we would rather dismiss some information that is little relevant to the graph structure to incorporate the full graph attributes. For example, the distance terms between far away graph nodes are of little interest to the construction of the average distance matrix. Thus, we would like to build a graph representation that incorporates only the pairs of (non-quantised) graph nodes located at a distance below a given threshold.

In the following chapter, we will decompose visual feature graphs in sets of unquantised feature pairs. Each set, called a *channel*, contains all pairs of graph nodes located at a distance bounded by an upper and lower value. A graph is then characterised by a certain number of channels. An appropriate classifier must be designed to classify such graphs. We will introduce a generic classifier that works in the space of features sets. We will show that this classifier, based on a nearest neighbour distance, can be trained to combine multiple sources of features in an optimal manner. We will then provide a formulation of the graph classification problem that will allow us to apply this classifier to visual feature graphs.

Chapter 4

Classification and detection of sets of features

4.1 Introduction

As we have seen in the previous chapter, the discrete sampling of visual features is a powerful and flexible tool in image classification and computer vision at large. However, the vast majority of approaches described in the literature, as well as our own work from the previous chapter, suffer from one common issue: feature quantisation considerably reduces the discriminative power of visual features. Consequently, it has the potential to strongly degrade classification performances. Some authors have designed solutions to quantise features over several codebook entries: this kind of quantisation, coined "soft quantisation", as opposed to "greedy quantisation", is designed to reduce the loss of discriminative power of quantised features [van Gemert 2008]. The justification underlying soft feature quantisation is that, for any given feature, there is a small subset of codebook entries that lies in its close neighbourhood, while the vast majority of codebook entries are located much farther away. However, we describe here the results of a simple experiment that point in the direction opposite to soft feature quantisation: we have extracted SIFT features from the Caltech-101 dataset and clustered them by k-means in a codebook containing 500, 1000 or 2000 entries. For each codebook, the distance from each feature to each codebook entry has been computed. Averaging over all features, we obtained the average distance from any feature to the n^{th} nearest entry in the codebook. The corresponding plot is given in figure 4.1. If soft quantisation was well founded, the curves plotted in figure 4.1 would expose an almost flat, short section on the bottom left side followed by a sharp increase. What we observe in every case is exactly the opposite: a relatively rapid increase followed by a long plateau.

The experiment we just described above is an illustration of the well-known curse of dimensionality [Bellman 1961]: because the space of visual features is of large dimension (128 in the case of SIFT features), sampling this space at an acceptable rate requires a great number of samples. Thus, the hypothesis that underlies soft quantisation is verified only when the codebook size is very large. But when the codebook becomes too large, the soft quantisation vector becomes too sparse and features are too discriminative

(distances between quantised features become all very large).

The bottom line is that feature quantisation is a step that is usually taken for the sake of conveniency and simplicity, but it cripples the precision of image representation. The lack of fast, satisfying classifiers based on unquantised features have contributed to the popularity of SVM-like methods based on histograms of quantised features.

In this chapter, we describe two different classifiers of sets of unquantised features: these classifiers are both based on nearest-neighbour distance and they both improve over their quantised counterpart. The first classifier consists in the combination of an SVM and a matching kernel. Because each image is classified by comparing its representation — a set of features, called *handful of features* — to other images, via this matching kernel, this classifier belongs to the category of image-to-image distance-based methods. This handful of features gives us a glimpse of the potential provided by unquantised features. Our second classifier is a reformulation of the naive Bayes nearest neighbour (NBNN) classifier [Boiman 2008]: because each feature from each image is matched to its nearest neighbour among all images from each class, NBNN belongs to the category of image-to-class distance-based methods in practical experiments.

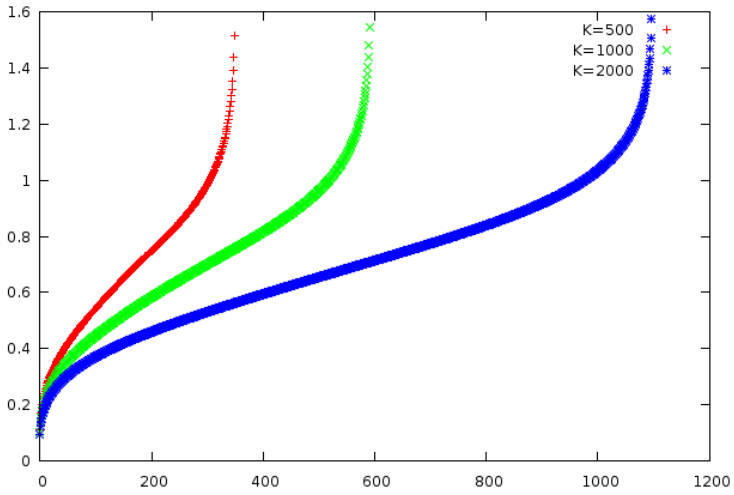

Figure 4.1: Average distance from SIFT visual features to their n^{th} nearest neighbour from the codebook. The vertical axis represents the distance, while the index n is plotted on the horizontal axis.

4.2 Handfuls of features

4.2.1 Introduction

In this section, we briefly depart from the framework based on graphical structures described in the previous chapter to exhibit the level of performances attained by simple representations that consist of unquantised features. We shall first return to the main idea that underlies the bag of words representation and which says that images can be represented by orderless sets of visual features. Our argument is as follows: if we can design a kernel function on sets of unquantised features, then we should be able to employ this kernel in support vector machines. If the obtained classifier produces an improvement over the bag of words representation, then the gap will have been produced solely by the use of unquantised features.

We introduce in the following subsection a kernel on sets of unquantised features by assimilating sets of features to feature distributions. After two simplification steps, the kernel function will be equal to a sum of kernel function values taken between nearest neighbour features. This kernel, known in the literature as the matching kernel [Boughorbel 2005], is not a Mercer kernel (i.e: it is not positive semidefinite). An SVM classifier trained with this kernel is thus not guaranteed to be optimal. Nonetheless, we will show that it consistently outperforms its Mercer counterpart as well as other kernels on bags of quantised features.

4.2.2 A kernel on feature sets

Our goal in this section is to define a distance function between sets of feature points. The distance will then turn into an affinity kernel that measures the similarity between two sets of features. More to the point: we denote $\mathcal{D}$ the space of feature descriptors, and assume it is equipped with a bounded kernel function $k : \mathcal{D} \times \mathcal{D} \to [0, 1]$. In our experiments, the feature space $\mathcal{D}$ is simply a Euclidean space $\mathbb{R}^d$ and k is the Gaussian kernel: $k(x, y) = \exp\left(-\frac{1}{2\sigma^2}||x - y||^2\right)$. We denote $\mathcal{D}_f$ the set of finite subsets of $\mathcal{D}$: $\mathcal{D}_f = \{X \subset \mathcal{D},\ |X| < \infty\}$. Hence we are looking for a bounded kernel function $K : \mathcal{D}_f \times \mathcal{D}_f \to [0, 1]$. Once this kernel will be defined, we will be able to use it with support vector machine for the classification of sets of unquantised features.

To facilitate the practical computation of K, we want to be able to compute values of K using just values of the feature kernel k. To do so, we assimilate a set of features to a feature distribution. Computing a distance between feature sets is then equivalent to computing a distance between feature distributions. We assimilate each point x of $\mathcal{D}$ to a Gaussian probability distribution $\mathcal{G}(x, \sigma, .)$ centred on $x \in \mathcal{D}$ and of variance σ^2, where σ is a parameter of the approach. Intuitively, σ characterises the typical size of a point neighbourhood in which we consider that two points are similar

in appearance. Consequently, a finite set of points $X \in \mathcal{D}_f$ can be considered as a normalised sum of Gaussians distribution: $\frac{1}{|X|}\sum_{x\in X}\mathcal{G}(x,\sigma,.)$. A measure of the proximity of two such distributions is the L^1 norm of their product:

$$\forall X,Y \in \mathcal{D}_f,\ K_1(X,Y) = \frac{1}{|X||Y|}\int_{\mathcal{D}}\left(\sum_{x\in X} f_{x,\sigma}(u)\sum_{y\in Y} f_{y,\sigma}(u)\right)du \quad (4.1)$$

where $f_{x,\sigma}$ denotes the density function of $\mathcal{G}(x,\sigma,.)$. We note that this affinity measure is loosely related to the Bhattacharyya affinity between probability density functions:

$$K_B(f,g) = \int \sqrt{f(x)}\sqrt{g(x)}dx, \quad (4.2)$$

which is itself related to the Hellinger distance:

$$H(f,g) = \left(\int\left(\sqrt{f(x)} - \sqrt{g(x)}\right)^2 dx\right)^{\frac{1}{2}}, \quad (4.3)$$

by $H = \sqrt{2-2K_B}$.

The product of sums from equation 4.1 is equal to a sum of integrals in which each term integrates a product of two Gaussians over $\mathcal{D}$. We will now show how equation 4.1 can be approximated by a sum of kernel products between nearest neighbour features, via two simplification steps.

The first step maps the integral of the density functions product to the real line:

$$\int_{\mathcal{D}} f_{x,\sigma}(u)f_{y,\sigma}(u)du \simeq \int_{\mathbb{R}} f_{0,\sigma}(u)f_{||x-y||,\sigma}(u)du, \quad (4.4)$$

where in the RHS functions $f_{.,.}$ refer to 1D Gaussian density functions of $\mathbb{R}$. This simplification enables us to integrate over $\mathbb{R}$ instead of $\mathcal{D}$; it is justified by considering that the Gaussian distribution $\mathcal{G}(x,\sigma,.)$ is isotropic. Therefore, the integration on $\mathcal{D}$ of the product of $\mathcal{G}(x,\sigma)$ and $\mathcal{G}(y,\sigma)$ can be formulated as the integration on $\{x+\lambda(y-x)|\lambda\in\mathbb{R}\}$. As a matter of fact, equation 4.4 is more a change of integration space than a true approximation, as it cannot be said that both sides of the equation are approximately equal. However, both sides exhibit the same behaviour as $||x-y||$ vary. The change of space serves to avoid the curse of dimensionality that can appear in visual feature spaces of large dimension.

We make a second approximation by considering that the sum on all point pairs (x,y) can be reduced to a sum in which each point is matched

to its nearest neighbour:

$$\sum_{x\in X}\sum_{y\in Y}\int_{\mathbb{R}} f_{0,\sigma}(u)f_{||x-y||,\sigma}(u)du \simeq \frac{1}{2}\sum_{x}\max_{y}\left(\int_{\mathbb{R}} f_{0,\sigma}(u)f_{||x-y||,\sigma}(u)du\right) + \frac{1}{2}\sum_{y}\max_{x}\left(\int_{\mathbb{R}} f_{0,\sigma}(y)f_{||x-y||,\sigma}(u)du\right) \tag{4.5}$$

In effect the purpose of approximation 4.5 is to match each of the $x \in X$ to its nearest neighbour among the $y \in Y$, and vice versa, as each integral becomes maximum when $||x - y||$ is minimum. We argue that this approximation is actually necessary: for instance, if we were to compute the distance between two parallel spatial configurations of points, each one in shape of a straight line, it would make no sense to take into account the distances between furthest point pairs.

The product of two Gaussians is a Gaussian, and the value of each integral of equation 4.5 can be explicitly computed:

$$\int_{\mathbb{R}} f_{0,\sigma}(u)f_{||x-y||,\sigma}(u)du = \frac{1}{2}e^{\frac{-||x-y||^2}{4\sigma^2}} \tag{4.6}$$

Since we want to normalise the proximity (and the distance) measure to the $[0, 1]$ range, the $\frac{1}{2}$ factor should be removed from equation 4.6 and the normalisation by $\frac{1}{|X||Y|}$ should be replaced by $\frac{2}{|X|+|Y|}$ in equation 4.1. Finally, the measure of the proximity between two sets of points can be defined as:

$$\forall X, Y \in \mathcal{D}_f,\ K(X,Y) = \frac{1}{|X|+|Y|}\left(\sum_{x\in X}\max_{y\in Y} e^{\frac{-||x-y||^2}{4\sigma^2}} + \sum_{y\in Y}\max_{x\in X} e^{\frac{-||x-y||^2}{4\sigma^2}}\right) \tag{4.7}$$

Approximation 4.5 is better justified when the number of points is not too high; the distance measure defined here is thus best suited to the comparison of small sets of features, which we call *handfuls of features*.

The role played by parameter σ in equation 4.7 gives us an indication of the value it should take: in our experiments, σ^2 was defined as the variance of the euclidean metric, which can be evaluated experimentally.

4.2.3 Support Vector Machine with non-Mercer kernels

The kernel between sets of features that we have defined above has been employed multiple times in the literature. It was introduced by [Wallraven 2003], but in this paper the authors made the wrong assertion that the kernel is positive semidefinite. In fact, it is possible to find counterexamples for which the kernel matrix has negative eigenvalues, as shown

by [Lyu 2005]. In [Boughorbel 2005], the authors also acknowledge this problem and proceed to solve it by introducing a set of intermediate features $V = \{v_i\}$ to which the features from the kernel arguments are compared. The kernel function becomes then:

$$\forall X, Y \in \mathcal{D}_f,\ K_V(X,Y) = \sum_{v \in V} \exp\left(-\frac{1}{2\sigma^2}||\Phi_v(X) - \Phi_v(Y)||^2\right), \quad (4.8)$$

where $\Phi_v(X)$ is the nearest neighbour of v in X:

$$\forall X \in \mathcal{D}_f,\ \forall v \in V,\ \Phi_v(X) = \arg\min_{x \in X} ||v - x||. \quad (4.9)$$

K_V is indeed a positive semidefinite kernel, but it poses again the problem of the definition of V; the authors obtain V by discrete, finite quantisation of the feature space, but then face the same problem of space quantisation as with the bag of words representation.

The benefit of positive (semi) definite kernels is that they can be employed in a number of classifiers, including support vector machines (SVM). In SVM, the classifier converges towards the optimal solution provided the kernel is positive semidefinite. It is possible to train an SVM classifier with kernels that are not Mercer kernels, but the solution found is not guaranteed to be optimal anymore. In [Boughorbel 2004], a kernel is defined on pairs of feature sets. Like in our work, the kernel consists of a sum of distances between matched points — but the matching is computed so as to maximise the similarity between the feature sets, and not in a nearest neighbour fashion. It is then shown that the probability of the kernel not being positive semidefinite can be bounded by any arbitrary value by setting an appropriate value for the radial basis functions smoothing parameter.

In practice, it is admitted that the kernel we have introduced in the previous subsection is likely *not* to be a Mercer kernel. Thus, an SVM classifier equipped with this kernel is not guaranteed to converge towards its optimal value. Nonetheless, we decided to test the reliability of the handful of features representation with support vector machines.

4.2.4 Kernel performances

The relatively high computational complexity required to compute the distance separating two handfuls of features seems to suggest that this representation is more adequate for small feature sets. Thus, we applied our method to the classification of image subregions.

Images from the Graz-02-bicycles dataset (see section A.3) where divided in regular grids of subimages of variable size. Each subimage is a square with a side length ranging from 50 to 100 pixels. A subimage is considered positive if at least 50% of its surface overlaps with a bicycle instance. Two random, disjoint subsets of 2000 subimages each were constructed from the

sets of training and testing images. Scale-invariant, rotation-variant SURF features were extracted from each subimage. In the case of the bag of word representation, they were quantised by a codebook of size 500. Classification of the subimages was done by SVM, with HoF kernel for HoF representations and linear kernel for bags of words. The smoothing parameter of the HoF kernel was set to the average empirical L^2 distance between SURF features. Classification results are summarised in figure 4.2.

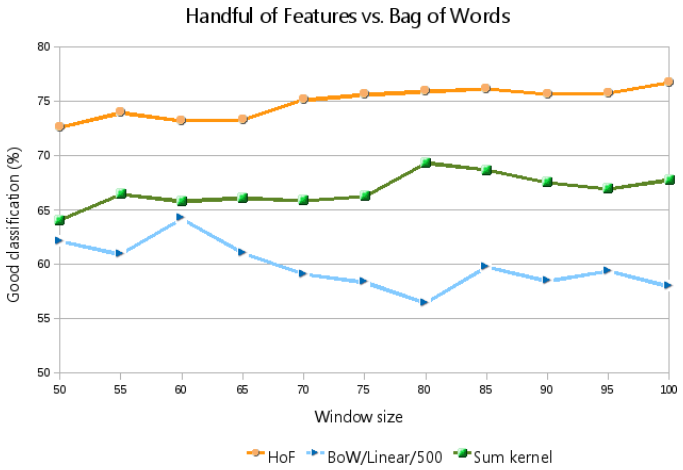

Figure 4.2: Classification of handfuls of features (HoF) versus bags of words (BoW) on the Graz-02-bicycles dataset. The green curve refers to the summation kernel (see text).

We observe an overall gain of the HoF representation over the bag of words that is greater than 9 percentage points. Moreover, this gain remains consistent as the window size (and thus: the average number of feature points contained in each subimage) increases. We are thus confident that image classification, too, can be improved by the use of unquantised features, even though the classifier we employ might be suboptimal.

In order to test the relevance of the main approximation of our approach, we also implemented a "summation kernel" (following the denomination of [Boughorbel 2004]) that sums the distances between all feature pairs:

$$\forall X, Y \in \mathcal{D}_f,\ K_S(X,Y) = \frac{1}{|X||Y|} \sum_{x \in X} \sum_{y \in Y} \exp\left(-\frac{1}{2\sigma^2}||x-y||^2\right). \quad (4.10)$$

It is easy to see that this kernel is positive definite. Indeed, given a sequence

of feature sets $(X_i)_{1\leq i\leq N}$ and real values $(c_i)_{1\leq i\leq N}$, we have:

$$\sum_{i=1}^{N}\sum_{j=1}^{N} c_i c_j K_S(X_i, X_j) = \sum_{i=1}^{N}\sum_{x\in X_i}\sum_{j=1}^{N}\sum_{x'\in X_j} c_i c_j \exp\left(-\frac{1}{2\sigma^2}||x-x'||^2\right) \tag{4.11}$$

$$= \sum_{x\in\bigcup_i X_i}\sum_{x'\in\bigcup_i X_i} c_x c_{x'} \exp\left(-\frac{1}{2\sigma^2}||x-x'||^2\right), \tag{4.12}$$

where: $\forall i \in [1,N]$, $\forall x \in X_i$, $c_x = c_i$. Because $k : (x,x') \rightarrow \exp\left(-\frac{1}{2\sigma^2}||x-x'||^2\right)$ is a positive definite kernel, the sum on the RHS of equation 4.12 is positive. K_S is thus a positive definite kernel.

However, the positive definiteness of this kernel does not imply an improvement of the performances over the handful of features kernel. What the green curve from figure 4.2 tells us is that this kernel still outperforms the linear kernel for a bag of word, but it consistently remains five to ten percentage points below the handful of feature kernel. This emphasises the relevance of our geometrical explanation of the handful of features distance.

4.2.5 Conclusion

Experiments have shown how our representation based on sets of unquantised features can outperform the bag of words representation. Results speak in favour of the unquantised features, despite the sub-optimality of the SVM kernel we use and highlighted in section 4.2.3. Moreover, we have shown that the performance gap obtained does not require any fundamental, theoretical breakthrough. In fact, the same essential bricks already employed for the classification of bags for words can also be employed to classify unquantised sets of features. Consequently, the same improvements brought to SVM and bags of words can probably also be applied to handfuls of features without any major change. For instance, it should be possible to design pyramids of handfuls of features, similarly to [Lazebnik 2006].

These observations encourage us to design more adequate image representations, along with the appropriate classifiers. To do so, we will follow the same line of thought we investigated in chapter 3: we shall include information related to the layout of the image interest points, along with unquantised, multi-dimensional attributes for the image features. This will be done by using a nearest neighbour-based classifier, which is more appropriate than support vector machines when dealing with unquantised features. We will then show how to adapt this classifier to graphical structures of feature points such as the ones introduced in section 3.2.3.

4.3 Optimal Naive Bayes Nearest Neighbour

4.3.1 Introduction

In the previous section, we presented a classifier based on support vector machines (SVM) on the space of feature sets. This classifiers thus relies on image-to-image distances. In this section, we present a classifier that models, in a certain way, the feature distribution relatively to the class label. In other words, this new classifier makes use of image-to-class distances. Once the conditional distribution has been modeled, we will show that the conditional probability of any sample feature can be inferred from the nearest neighbour distance. A naive Bayes hypothesis will then allow us to estimate the conditional probability of an image. We begin by presenting a similar classifier from which we have drawn some inspiration.

Naive Bayes Nearest Neighbour (NBNN) is a non-parametric classifier introduced in [Boiman 2008] that was designed to address the issues raised by feature quantisation, as described in the introductory section 4.1. In [Boiman 2008], the authors give a quantitative assessment of the loss of discriminative power incurred by feature quantisation. To do so, they measure the evolution of the ratio of a feature probability conditioned on its class before and after quantisation: the loss is given by the gap between $P(d|C)/P(d|\overline{C})$ and $P(d_{quant}|C)/P(d_{quant}|\overline{C})$.

One of the conclusions that can be drawn from this experiment is that the popularity enjoyed by the BoW/SVM combination is due to the efficiency of the SVM classifier, not to the representation itself. In simple words, most, but not all, of the information discarded by the feature quantisation step is offset by the efficiency of the classifier.

The second important argument developed in [Boiman 2008] is that nearest-neighbour classifiers perform better when they rely on an image-to-class distance instead of an image-to-image distance. The main observation in favour of this argument is that new observed image samples are more often combinations of several previously observed samples than just one. In other words, the appearance of new objects drawn from a class is closer to the combined appearance of multiple objects than to the appearance of a single object. We will thus need the concept of *feature-to-class* distance, which is the distance of a feature to its nearest neighbour drawn from the considered class.

The following step taken in [Boiman 2008] is to design a classifier that approximates the naive Bayes classifier with respect to this feature-to-class distance. This classifier is coined naive Bayes Nearest Neighbour (NBNN).

In this section, we improve on the work of [Boiman 2008] and show that one of the key simplification steps in the design of NBNN actually hides a strong assumption with regard to the feature distributions of the different classes. Intuitively, the objection that we raise is that if one class con-

tains more features than an other, then the feature-to-class distance measure will be biased towards that class. The initial formulation of NBNN dismisses this issue by implicitly assuming that all classes contain approximately equal numbers of features. In the following, we will show that this assumption greatly damages the generalisation properties of the classifier. We shall see that relaxing this assumption involves the introduction of two distance-correction parameters related to the properties of the feature distribution for each class. Instead of setting these parameters by hand, we show how their optimal values with respect to cross-validation can be obtained as the solution of a linear program. Moreover, the formulation we obtain naturally generalises to multi-channel classification with guaranteed performance increase.

The multi-channel classifier we obtain can be employed for the classification of attribute graphs, which was our primary objective. We introduce the concept of *distance-collapsed graphs* by grouping pairs of nodes separated by the same graph shortest path distance in the same channel. Visual feature graphs introduced in section 3.2 can then be seen as the union of a set of channels; optimal multi-channel NBNN can thus be used to classify sets of visual feature graphs, which was our primary purpose.

The versatility of our optimal NBNN allows us to adapt it to the problems of object detection and classification by detection. Indeed, our classifier shares the property of linearity with support vector machine (SVM). The idea of efficient sub-window search [Lampert 2008] can thus be applied to optimal NBNN.

The remainder of this section is organised as follows: we first summarise in section 4.3.2 the main arguments of [Boiman 2008] on which the initial formulation of NBNN is based. Then we show in section 4.3.3 how a more general formulation involves the introduction of two parameters that bring an affine correction to the density estimation of visual features. This formulation is in turn adapted to the multi-channel case in section 4.3.4. A linear program designed to find the optimal values of these parameters is described in section 4.3.5. The classification framework is adapted to the problem of object detection and image classification by detection in section 4.3.7. The notion of distance-collapsed graph for image classification is then described in section 4.4. A wide range of results on multiple datasets is given in section 4.3.8. In particular, we introduce in section 4.4 the notion of "distance-collapsed graph" that allows us to classify graphical structures based on a quantisation of the entries of their distance matrix.

4.3.2 Initial formulation

For the sake of completeness, we summarise in this section the main arguments of [Boiman 2008] on which the NBNN classifier is based.

In an image I with hidden class label c_I, we extract K_I features $(d_k^I)_k \in$

$\mathbb{R}^D$. Under the naive Bayes assumption, and assuming all image labels are equally probable ($P(c) \sim cte$) the estimated class label $\hat{c}_I$ of image I maximises the product of the feature probabilities relatively to the class label:

$$\hat{c}_I = \arg\max_c \prod_{k=1}^{K_I} P(d_k^I|c). \tag{4.13}$$

The feature probability conditioned on the image class $P(d_k^I|c)$ can be obtained by a non-parametric kernel estimator, such as the Parzen-Rosenblatt estimator: if we note $\chi^c = \left\{d_k^J | c_J = c, 1 \leq k \leq K_J\right\}$ the set of all features from all training images that belong to class c, we can write:

$$P(d_k^I|c) = \frac{1}{Z} \sum_{d \in \chi^c} \exp\left(\frac{- \parallel d_k^I - d \parallel^2}{2\sigma^2}\right), \tag{4.14}$$

where σ is the bandwidth parameter of the density estimator. Boiman et al. argue that this estimator can in turn be approximated by the highest term from the sum on the RHS. This leads to a quite simple expression:

$$\forall d,\ \forall c,\ -\log\left(P(d|c)\right) \simeq \min_{d' \in \chi^c} \parallel d - d' \parallel^2 . \tag{4.15}$$

The decision rule for image I is thus:

$$\hat{c}_I = \arg\max_c P(I|c) \tag{4.16}$$

$$= \arg\min_c \sum_k \min_{d \in \chi^c} \parallel d_k^I - d \parallel^2 . \tag{4.17}$$

This classifier is shown to greatly outperform the usual nearest neighbour classifier. Moreover, it does not require any feature quantisation step, and the descriptive power of image features is thus preserved.

The reasoning above proceeds in three distinct steps: first, the naive Bayes assumption considers that image points are independent identically distributed given the image class c_I (equation 4.13). Then, the estimation of a point probability density is obtained by a non-parametric density estimation process like the Parzen-Rosenblatt estimator (equation 4.14). Finally, NBNN is based on the assumption that this value, which is a sum of distances, can be approximated by its highest term (equation 4.15). In the following section, we will show that the implicit simplification that consists in removing the normalisation parameter from the density estimator is invalid in most practical cases. Furthermore, we will propose a solution to correct the estimator by introduction of a multiplicative and an additive parameter.

We will keep the notation introduced in this section. We will also need the notion of point-to-set distance, which is simply the square Euclidean

distance of a point to its nearest neighbour in the set:

$$\forall \Omega \subset \mathbb{R}^D, |\Omega| < \infty, \forall x \in \mathbb{R}^D, \tau(x, \Omega) = \min_{y \in \Omega} \parallel x - y \parallel^2 . \tag{4.18}$$

Moreover, $\tau(x, \chi^c)$ will be abbreviated as $\tau^c(x)$.[1]

4.3.3 Affine correction of nearest neighbour distance for NBNN

The most important theoretical problem of NBNN is that in order to obtain a simple approximation of the log-likelihood, we need a valid approximation of the probability density. This is achieved by assuming that the normalisation factor $1/Z$ is the same for all classes, an assumption that is wrong in most practical cases. If this factor varies significantly from one class to another, then the approximation leading to the maximum a posteriori class label $\hat{c}_I$ formulation by equation 4.17 becomes unreliable.

It should be noted that the objection that we raise does not concern the core hypothesis of NBNN, namely the naive Bayes hypothesis and the approximation of the sum of exponentials of equation 4.14 by its greatest term. In fact, in the following we will essentially follow and extend the arguments presented in [Boiman 2008] using the same starting hypothesis.

Non-parametric kernel density estimation requires the definition of a smoothing parameter σ, also called bandwidth. We consider the general case of a set of K points $\{x_k | 1 \leq k \leq K\}$ lying in some D-dimensional feature space Ω. The density function evaluated at point x is:

$$\forall x \in \Omega, f(x) = \frac{1}{Z} \sum_{k=1}^{K} \exp\left(-\frac{||x - x_k||^2}{2\sigma^2}\right). \tag{4.19}$$

The value of Z is obtained by normalisation of the density function:

$$\int_\Omega f(x)dx = 1 \Leftrightarrow Z = K(2\pi)^{\frac{D}{2}}\sigma^D. \tag{4.20}$$

We retain the NBNN assumption that the likelihood of a feature is approximately equal to the value of the highest term from the sum on the right hand side of equation 4.19. Here we provide an argument that supports this assumption: it is known that the convergence speed of the Parzen-Rosenblatt (PR) estimator is $K^{-4/(4+D)}$ [Stone 1983]. This means that in the case of a 128-dimensional feature space, such as the SIFT feature space, in order to reach an approximation bounded by $1/2$ we need to sample 2^{33} points:

[1] In the following, the reader will be careful not to mistake the top index x^c with bringing a variable to the power of c. We admit the notations can be confusing when x is a scalar value.

in practice, the PR estimator does not converge and there is little sense in keeping more than just the first term of the sum.

Thus, the negative log-likelihood of a visual feature d relatively to an image label c is:

$$-\log\left(P(d|c)\right) = -\log\left(\frac{1}{Z^c}\exp\left(-\frac{\tau^c(d)}{2(\sigma^c)^2}\right)\right) \tag{4.21}$$

$$= \frac{\tau^c(d)}{2(\sigma^c)^2} + \log(Z^c), \tag{4.22}$$

where $Z^c = |\chi^c|(2\pi)^{\frac{D}{2}}(\sigma^c)^D$. Recall that $\tau^c(d)$ is the square Euclidean distance of d to its nearest neighbour in χ^c (see equation 4.18). In the above equations, we have replaced the class independent notation σ, Z by σ^c, Z^c since, in general, there is no reason to believe that parameters should be equal across classes. For instance, both parameters are functions of the number of training features of class c in the training set.

Returning to the naive Bayes formulation, we obtain:

$$\forall c, -\log\left(P(I|c)\right) = \sum_{k=1}^{K_I}\left(\frac{\tau^c(d_k^I)}{2\sigma^{c^2}} + \log(Z^c)\right) \tag{4.23}$$

$$= \alpha^c \sum_{k=1}^{K_I}\tau^c(d_k^I) + K_I\beta^c, \tag{4.24}$$

where $\alpha^c = 1/(2(\sigma^c)^2)$ and $\beta^c = \log(Z^c)$ is a re-parametrisation of the log-likelihood from equation 4.22 that has the advantage of being linear in the model parameters. The image label is then decided according to a criterion that is slightly different from equation 4.17:

$$\hat{c}_I = \arg\min_c\left(\alpha^c\sum_{k=1}^{K_I}\tau^c(d_k^I) + K_I\beta^c\right). \tag{4.25}$$

We note that this modified decision criterion can be interpreted in two different ways: it can either be interpreted as the consequence of a density estimator to which a multiplicative factor was added, or as an unmodified NBNN in which an affine correction has been added to the square Euclidean distance. In the former, the final formulation we obtain can be considered different from the initial NBNN. In the latter, equation 4.25 can be obtained from equation 4.17 simply by replacing $\tau^c(d)$ by $\alpha^c\tau^c(d) + \beta^c$ (since α^c is positive, the nearest neighbour distance itself does not change). This formulation differs from [Boiman 2008] only in the evaluation of the distance function, leaving us with two parameters per class to be evaluated.

At this point, it is important to recall that the introduction of parameters α^c and β^c does not violate the naive Bayes assumption, nor the assumption

of equiprobability of classes. In fact, the density estimation correction can be seen precisely as an enforcement of these assumptions. If a class is more densely sampled than others (i.e: its feature space contains more training samples), then the NBNN estimator will have a bias towards that class, even though it made the assumption that all classes are equally probable. The purpose of setting appropriate values for α^c and β^c is to correct this bias.

4.3.4 Multi-channel image classification

In the most general case, an image is described by different features coming from different sources or sampling methods. For example, we can sample SIFT features and local colour histogram from an image. We observe that the classification criterion of equation 4.13 copes well with the introduction of multiple feature sources. The only difference should be the parameters for density estimation, since feature types correspond, in general, to different feature spaces.

In order to handle different feature types, we need to introduce a few definitions and adapt our notation. In particular, we define the concept of *channel*: a channel χ is a function that associates a set of finite-dimensional characteristics to an image I: $\forall I, \chi(I) \subset \mathbb{R}^{D_\chi}$. Channels can be defined arbitrarily: a channel can be associated to a particular detector/descriptor pair, but can also represent global image characteristics. For instance, in the most extreme case, an image channel can consist in a single element, such as the global colour histogram. We will later see that channels behave best when containing a large number of very discriminative features.

Let us assume we have defined a certain number of channels $(\chi_n)_{1\leq n\leq N}$, that are expected to be particularly relevant to the problem at hand. We realise that adapting the framework of our modified NBNN to multiple channels is just a matter of changing notation. Similarly to the single-channel case, we aim here at estimating the class label of an image I:

$$\hat{c}_I = \arg\max_c P(I|c) \tag{4.26}$$

$$\forall c, P(I|c) = \prod_n \prod_{d\in\chi_n(I)} P(d|c). \tag{4.27}$$

Since different channels have different features spaces, the density correction parameters should depend on the channel index: α^c, β^c will thus be noted α_n^c, β_n^c. The notation from the previous section are adapted in a similar way: we call $\chi_n^c = \bigcup_{J|c_J=c} \chi_n(J)$ the set of all features from class c and channel n and define the distance function of a feature d to χ_n^c by: $\forall d,\ \tau_n^c(d) = \tau(d, \chi_n^c)$. This leads to the classification criterion:

$$\hat{c}_I = \arg\min_c \sum_n \left(\alpha_n^c \sum_{d\in\chi_n(I)} \tau_n^c(d) + \beta_n^c |\chi_n(I)| \right). \tag{4.28}$$

Naturally, when adding feature channels to our decision criterion, we wish to balance the importance of each channel relatively to its relevance to the problem at hand. Equation 4.28 shows us that the function of relevance weighting can be assigned to the distance correction parameters. The problems of adequate channel balancing and nearest neighbour distance correction should thus be addressed in one single step. In the following section, we present a method to find the optimal values of these parameters by cross-validation.

4.3.5 Parameter estimation

It might be noted that deciding on a suitable value for α^c and β^c simply requires to define an appropriate bandwidth σ^c. Indeed, the dimensionality D of the feature space and the number $|\chi^c|$ of training feature points are known parameters. However, in practice, manually choosing a "good" value for the bandwidth parameter is time-consuming and inefficient. To cope with this issue, we designed an optimisation scheme that finds the optimal values of parameters α^c, β^c with respect to the hinge loss.

We now turn to the problem of estimating values of α_n^c and β_n^c that are optimal for cross-validation in the binary classification case.

Assuming the set of classes is reduced to $\{-,+\}$, an image I will be classified as positive or negative according to the following decision rule:

$$\hat{c}_I = sign\left(\tau^-(I) - \tau^+(I)\right), \tag{4.29}$$

where:

$$\forall c \in \{-,+\},\ \tau^c(I) = \sum_n \sum_{d \in \chi_n(I)} \left(\alpha_n^c \tau_n^c(d) + \beta_n^c\right). \tag{4.30}$$

Developing the term in the RHS of equation 4.29 leads to:

$$\tau^-(I) - \tau^+(I) = \sum_n \sum_{d \in \chi_n(I)} \left(\alpha_n^- \tau_n^-(d) - \alpha_n^+ \tau_n^+(d)\right) + \sum_n |\chi_n(I)|(\beta_n^- - \beta_n^+). \tag{4.31}$$

Given a labelled sample, we can define a constrained linear energy optimisation problem that minimises the hinge loss of a binary multi-channel NBNN classifier:

$$E = \sum_I \max\left\{0, 1 - c_I\left(\tau^-(I) - \tau^+(I)\right)\right\} \tag{4.32}$$

$$= \sum_I \xi_I, \tag{4.33}$$

subject to constraints:

$$\forall I,\ 1 \leq \xi_I + c_I \sum_n \sum_{d \in \chi_n(I)} \left(\alpha_n^- \tau_n^-(d) - \alpha_n^+ \tau_n^+(d)\right) + \sum_n |\chi_n(I)| \beta_n^{-+} \tag{4.34}$$

$$\forall I,\ \xi_I \geq 0 \tag{4.35}$$

$$\forall n,\ \alpha_n^- \geq 0,\ \alpha_n^+ \geq 0,\ \beta_n^{-+} \in \mathbb{R}, \tag{4.36}$$

where labels c_I are in $\{-, +\}$ and where we have replaced $(\beta_n^- - \beta_n^+)$ by β_n^{-+}. This linear program can be solved exactly (and quickly) for a relatively large number of channels and images, with the guarantee that cross-validation performance will be an increasing function of the number of channels[2]. In practice, the only limitation concerns the maximum number of channels: this number should be kept small relatively to the number of training samples to avoid overfitting.

At this point, optimal NBNN exhibits two important qualities:

1. First, optimal NBNN corrects the systematic bias introduced in NBNN by unbalanced datasets. This corrective improves the discriminative power of our classifier.

2. Second, optimal NBNN provides the possibility to optimally combine multiple feature channels. Due to the fact that we estimate the distance correcting weights through a cross-validation strategy, the α_n^c, β_n^c will take higher absolute values for channels that are most relevant to classification.

Overfitting is the main drawback of most parametric classifiers, as opposed to non-parametric classifiers such as the initial NBNN. Nonetheless, in practice, we will see that the benefits of distance correction largely make up for the drawbacks of overfitting.

4.3.6 Multi-class, multi-channel optimal parameter estimation

The first multi-class classifier that we propose derives directly from the binary classifier proposed in the previous subsection. Once the distance-correction parameters have been learned for each pair of classes, we have for each test image a number of predictions in $\mathbb{R}$:

$$\forall c,\ c', \tau^{c,c'}(I) = \sum_n \sum_{d \in \chi_n(I)} \left(\alpha_n^{c'} \tau_n^{c'}(d) - \alpha_n^c \tau_n^c(d) + \beta_n^{c',c}\right), \tag{4.37}$$

[2]Our implementation makes use of the GNU linear programming kit `http://www.gnu.org/software/glpk/` (see appendix D.1.2)

where the reader should be careful to notice that the α_n^c are different in each equation, despite the fact that they are always denoted the same. The final class estimator is then:

$$\hat{c}_I = \arg\min_c \sum_{c' \neq c} H(\tau^{c,c'}(I)) \tag{4.38}$$

where H is the thresholded identity function:

$$\forall x \in \mathbb{R}, H(x) = \begin{cases} -1 & \text{if } x < -1 \\ 1 & \text{if } x > 1 \\ x & \text{otherwise} \end{cases} \tag{4.39}$$

This thresholding is necessary to avoid that the term inside the hinge loss takes values outside of the $[-1, 1]$ bounds.

Our second multi-class classifier generalises the parameter estimation process described in the previous subsection to multiple classes. Let us assume images are labelled with one label from $\{1, \dots K\}$. Let us denote $\mathcal{I}^c$ the set of all images from class c, for all c in $\{1, \dots K\}$. Then the energy corresponding to the binary classifier c versus c' described above can be rewritten:

$$E^{c,c'} = \sum_{I \in \mathcal{I}^c \cup \mathcal{I}^{c'}} \max\left\{0, 1 - \mathbb{1}^{c,c'}(c_I)\left(\tau^c(I) - \tau^{c'}(I)\right)\right\} \tag{4.40}$$

$$= \sum_I \xi_I^{c,c'}, \tag{4.41}$$

subject to constraints:

$$\forall I,\ 1 \leq \xi_I^{c,c'} + \mathbb{1}^{c,c'}(c_I) \sum_n \sum_{d \in \chi_n(I)} \left(\alpha_n^c \tau_n^c(d) - \alpha_n^{c'} \tau_n^{c'}(d)\right) + \sum_n |\chi_n(I)|(\beta_n^c - \beta_n^{c'}) \tag{4.42}$$

$$\forall I,\ \xi_I^{c,c'} \geq 0 \tag{4.43}$$

$$\forall n,\ \alpha_n^c \geq 0,\ \alpha_n^{c'} \geq 0,\ \beta_n^c \in \mathbb{R},\ \beta_n^{c'} \in \mathbb{R}, \tag{4.44}$$

where we have defined:

$$\forall I \in \mathcal{I}^c \cup \mathcal{I}^{c'},\ \mathbb{1}^{c,c'}(c_I) = \begin{cases} -1 & \text{if } c_I = c \\ 1 & \text{if } c_I = c' \end{cases} \tag{4.45}$$

The multi-class energy to be minimised, and the linear program to be solved will thus take the following form:

$$E = \sum_c \sum_{c' < c} E^{c,c'} \tag{4.46}$$

with constraints 4.42, 4.43, 4.44 taken from every binary problems.

Naturally, here, contrary to the sum of binary classifiers from equations 4.37 and 4.38, the coefficients α_n^c, β_n^c represent the same variables inside all $E^{c,c'}$. Their values are then estimated jointly by minimisation of energy 4.46.

In our classification experiments, we always used the binary formulation of equation 4.37 (first multi-class classifier), except in the case of classification by detection (section 4.3.7.2) where we employed the formulation corresponding to equation 4.42.

4.3.7 Naive Bayes Nearest Neighbour for Object Detection

Object detection aims at finding the best location of a given object in an image. We view this problem as finding the sub-image that minimises the decision score of equation 4.29.

In the following, we begin by adapting the problem of fast optimal subwindow search from [Lampert 2008] to the case of detection by optimal multi-channel NBNN. Because optimal NBNN is linear in the image features, the practical difference with linear SVM is small. Then, we adapt the NBNN classifier to the problem of classification by detection. We will show that the classification by detection predictor is a sum of the optimal window predictor with the weighted image-to-background class distance.

4.3.7.1 Detection by efficient subwindow search

Assuming an instance of class c is contained in image I, our goal is to find the rectangular sub-image that minimises $\tau^c(I)$ from equation 4.30. This problem can be solved by a naive sliding window search, but a faster method was introduced by [Lampert 2008]: in their work, Lampert et al. implement a fast and exact branch-and-bound subwindow search to maximise a decision function that is the output of a linear support vector machine (SVM). Though their classifier is not an NBNN classifier, their efficient subwindow search can be adapted to any decision function that is linear in the feature points, such as the multi-channel decision of equation 4.29.

In practice, we are looking for an image rectangular window with pixel coordinates $R = [t, b, l, r]$ inside which the sum of feature-to-class distances is maximum:

$$R_{opt} = \arg\max_R \tau(R) \tag{4.47}$$

$$\forall R,\ \tau(R) = \sum_{d \in R} \left(\tau^-(d) - \tau^+(d)\right) \tag{4.48}$$

Considering a set of image windows $\mathcal{R} = [t_{low}, t_{high}] \times [b_{low}, b_{high}] \times [l_{low}, l_{high}] \times [r_{low}, r_{high}]$ (see figure 4.3) we define the bounding function

of τ over $\mathcal{R}$ as:

$$\hat{\tau}(\mathcal{R}) = \tau_{pos}(R_{max}) + \tau_{neg}(R_{min}) \tag{4.49}$$

where τ_{pos} and τ_{neg} are the positive and negative parts of τ respectively, such that $\tau = \tau_{pos} + \tau_{neg}$, and R_{max} and R_{min} are the maximum and minimum rectangles of $\mathcal{R}$ respectively:

$$\tau_{pos} = \sum_{d \in R} max\left(\tau^{-}(d) - \tau^{+}(d),\ 0\right) \tag{4.50}$$

$$\tau_{neg} = \sum_{d \in R} min\left(\tau^{-}(d) - \tau^{+}(d),\ 0\right) \tag{4.51}$$

$$R_{max} = [t_{low}, b_{high}, r_{low}, l_{high}] \tag{4.52}$$

$$R_{min} = [t_{high}, b_{low}, r_{high}, l_{low}] \tag{4.53}$$

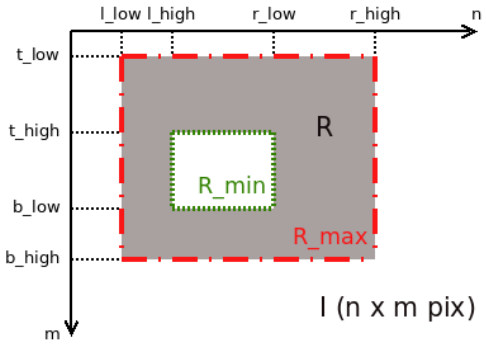

Figure 4.3: In this example of an image $I \in \mathbb{R}^{n \times m}$, $\mathcal{R}$ is delimited by the gray area. The red (long dots) and green (short dots) rectangles are R_{max} and R_{min} respectively. Notice we employed the traditional axis orientations for image coordinates.

It is then possible to find the optimal rectangle R_{opt} with respect to the scoring function from equation 4.48 by algorithm 1.

The loop started at line 4 will stop when the set of candidate windows is reduced to one element. The algorithm is sure to end because the size of $\mathcal{R}$ necessarily decreases on step 5; at this step, the set of candidate windows is split in two along its longest edge. For instance, if $t_{high} - t_{low}$ is greater than $b_{high} - b_{low}$, $l_{high} - l_{low}$ and $r_{high} - r_{low}$, we will have:

$$\mathcal{R}_1 = [t_{low}, t'] \times [b_{low}, b_{high}] \times [l_{low}, l_{high}] \times [r_{low}, r_{high}] \tag{4.54}$$

$$\mathcal{R}_2 =]t', t_{high}] \times [b_{low}, b_{high}] \times [l_{low}, l_{high}] \times [r_{low}, r_{high}] \tag{4.55}$$

The final step of the loop is in line 8 to select the candidate $\mathcal{R}'$ of P with the highest bounding function score $\tau(\mathcal{R}')$. The speed gain produced by this branch and bound subwindow search over naive exhaustive search is

Algorithm 1 Fast branch and bound subwindow search.

given I = image in $\mathbb{R}^{n \times m}$
$\mathcal{R} = [0, m] \times [0, n] \times [0, m] \times [0, m]$ is the set of all rectangles
P = empty candidate queue
while $|\mathcal{R}| > 1$ **do**
  Split $\mathcal{R}$: $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$, $\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset$
  push $(\mathcal{R}_1,\ \hat{\tau}(\mathcal{R}_1))$ in P
  push $(\mathcal{R}_2,\ \hat{\tau}(\mathcal{R}_2))$ in P
  $\mathcal{R} \leftarrow$ best candidate of P
end while
$R_{opt} \leftarrow$ only element of $\mathcal{R}$

relatively large. The complexity observed experimentally is of the order of $O(N^2)$, where N is the number of feature points in the image, instead of N^4 for naive exhaustive search.

4.3.7.2 Classification by detection

We show in this section how the detection scheme proposed in section 4.3.7.1 can be employed to improve classification results. The classification scheme described in sections 4.3.2 to 4.3.5 assumes that the object related to the class label fills the whole content of the image. This is true for scene datasets, such as SceneClass13 (appendix A.6) and simple object datasets such as Caltech-101 (appendix A.4), but the assumption is no more valid for more challenging objects embedded in complex scenes, such as the object classes of the Pascal VOC challenges (appendix A.5) and the Graz-02 dataset (appendix A.3).

In these more challenging datasets, we still assume that each image contains only one instance of just one class, but this object instance can be surrounded by, or overlapped with a relatively high amount of clutter. In this context, finding the best (i.e: most likely) object position can contribute to the improvement of classification performances. Our goal becomes thus to maximise the joint probability of object label c and position π inside the image given the image content. Assuming object positions are independent from object label, the Bayes rule tells us that:

$$(\hat{c}_I, \hat{\pi}_I) = \arg\max_{c,\pi} P(c, \pi | I) \tag{4.56}$$

$$= \arg\max_{c,\pi} P(I | c, \pi). \tag{4.57}$$

Following the same line of thought as in NBNN, we can expand the likelihood term under the naive Bayes assumption:

$$\forall c,\ \pi,\ P(I|c,\pi) = \prod_n \prod_{d \in \chi_n(I)} P(d|c,\pi). \tag{4.58}$$

Remember n loops over all channels and $\chi_n(I)$ is the set of features in channel n of image I.

At this point, we make the additional assumption that a feature probability, knowing the object class and position, only depends on the point belonging or not to the object:

$$\forall n,\ c,\ \pi,\ \forall d \in \chi_n(I),\ -\log\left(P(d|c,\pi)\right) \quad = \quad \begin{cases} \tau_n^c(d) \text{ if } d \in \pi \\ \tau_n^{\overline{c}}(d) \text{ if } d \notin \pi. \end{cases} \tag{4.59}$$

In the above equation, we have written the feature-to-set distance functions τ_n^c and $\tau_n^{\overline{c}}$ without apparent density correction in order to alleviate the notation. We leave to the reader the task of replacing τ_n^c by $\alpha_n^c \tau_n^c + \beta_n^c$ in the equations of this section.

The image likelihood function is now decomposed over all features inside and outside the object:

$$-\log\left(P(I|c,\pi)\right) = \sum_n \left(\sum_{d \in \chi_n(\pi)} \tau_n^c(d) + \sum_{d \in \chi_n(\overline{\pi})} \tau_n^{\overline{c}}(d) \right), \tag{4.60}$$

where $\chi_n(\pi)$ refers to the points from $\chi_n(I)$ located inside π and $\overline{\pi}$ is the complementary of π inside I: $\overline{\pi} = I \setminus \pi$. Since this negative log-likelihood is to be minimised, we rewrite it in the form of an energy: $E(I,c,\pi) = -\log\left(P(I|c,\pi)\right)$. The term on the RHS of equation 4.60 can be rewritten:

$$E(I,c,\pi) = \sum_n \left(\sum_{d \in \chi_n(\pi)} (\tau_n^c(d) - \tau^{\overline{c}}(d)) + \sum_{d \in \chi_n(I)} \tau_n^{\overline{c}}(d) \right). \tag{4.61}$$

We observe that the second sum from the RHS of equation 4.61 does not depend on π. The energy of equation 4.61 can thus be decomposed in two terms:

$$E(I,c,\pi) = E_1(I,c,\pi) + E_2(I,c) \tag{4.62}$$

$$E_1(I,c,\pi) = \sum_n \sum_{d \in \chi_n(\pi)} \left(\tau_n^c(d) - \tau_n^{\overline{c}}(d)\right) \tag{4.63}$$

$$E_2(I,c) = \sum_n \sum_{d \in \chi_n(I)} \tau_n^{\overline{c}}(d). \tag{4.64}$$

Let us define the optimal object position $\hat{\pi}^c$ relatively to class c as the position that minimises the first energy term:

$$\forall c,\ \hat{\pi}^c = \arg\min_{\pi} E_1(I,c,\pi). \tag{4.65}$$

Then, we can express the optimal image class label $\hat{c}_I$ and object position $\hat{\pi}_I$ as:

$$\hat{c}_I \quad = \quad \arg\min_c \left(E_1(I,c,\hat{\pi}^c) + E_2(I,c)\right) \tag{4.66}$$

$$\hat{\pi}_I \quad = \quad \hat{\pi}^{\hat{c}_I}. \tag{4.67}$$

We observe that the first energy term is the one that we have explicitly minimised in section 4.3.7.1. The same subwindow search algorithm can thus be applied to the problem of classification by detection. In short, the most likely class label and object position for a test image I are found by the following algorithm:

Algorithm 2 Classification by detection

```
declare variables ĉ, π̂
Ê = +∞
for each class label c = 1 do
    find π̂^c by efficient branch and bound subwindow search
    π̂^c = arg min_π E_1(I, c, π)
    if E_1(I, c, π̂^c) + E_2(I, c) < Ê then
        Ê = E_1(I, c, π̂^c) + E_2(I, c)
        ĉ = c
        π̂ = π̂^c
    end if
end for
return ĉ, π̂
```

4.3.7.3 Parameter estimation for classification by detection

Classification by detection as described in section 4.3.7.2 assumes the optimal distance-correcting parameters have already been computed. However, the parameter estimation procedure described in section 4.3.5 has to be slightly adapted in the case of classification by detection.

We will adopt a "class vs background" classification framework. The positive class is c and its complementary is $\overline{c}$. The background (negative) class is *back*. Deciding whether an image I contains an instance of class c will depend on the sign of $E_2(I, back) - E(I, c, \hat{\pi}^c)$:

$$\hat{c}_I = sign\left(\sum_n \sum_{d\in\chi_n(I)} \left(\tau_n^{back}(d) - \tau_n^{\overline{c}}(d)\right) + \sum_n \sum_{d\in\chi_n(\hat{\pi}^c)} \left(\tau_n^{\overline{c}}(d) - \tau_n^{c}(d)\right)\right). \tag{4.68}$$

In this equation we have suppressed the distance correcting parameters α_n^c, β_n^c for the sake of clarity. Let us rewrite the full expression of this equation:

$$\begin{aligned}\hat{c}_I = sign\Bigg(&\sum_n \sum_{d\in\chi_n(I)} \left(\alpha_n^{back}\tau_n^{back}(d) - \alpha_n^{\overline{c}}\tau_n^{\overline{c}}(d) + \beta_n^{back} - \beta_n^{\overline{c}}\right) \\ &+ \sum_n \sum_{d\in\chi_n(\hat{\pi}^c)} \left(\alpha_n^{\overline{c}}\tau_n^{\overline{c}}(d) - \alpha_n^{c}\tau_n^{c}(d) + \beta_n^{\overline{c}} - \beta_n^{c}\right)\Bigg).\end{aligned} \tag{4.69}$$

We see that the expressions of $\beta_n^{back} - \beta_n^{\overline{c}}$ and $\beta_n^{\overline{c}} - \beta_n^{c}$ are required. Therefore, values of β_n^{c}, $\beta_n^{\overline{c}}$, and β_n^{back} must be determined jointly. Yet, the linear program described in section 4.3.5 was designed for binary classification and only outputs the difference values of the β_n. We thus have to employ the multi-class formulation of section 4.3.6. $\mathcal{I}^c$, the set of images from class c is composed of all object masks from class c. $\mathcal{I}^{\overline{c}}$ is the set of complementary masks. $\mathcal{I}^{back}$ is the set of images containing no instance of class c. Once these image sets have been composed, we can follow the multi-class, multi-channel parameter estimation procedure of section 4.3.6.

4.3.7.4 Related work

The idea of looking for a window that maximises the score of an NBNN classifier was adopted by the authors of [Yuan 2009] in the context of action detection in video sequences. Like us, they have perceived the need to select appropriate values for the bandwidth of the kernel estimator. However, they solve this problem by picking an appropriate value by hand. Moreover, they do not employ different bandwidth values for different classes, and their classifier thus suffers from the same drawback as the original NBNN: namely, when the positive and negative class are unbalanced, the classifier has a bias towards the most populated class. Finally, their work does not integrate the possibility to make use of multiple channels simultaneously.

4.3.8 Experiments

4.3.8.1 Practical nearest neighbour retrieval

Concerning the nearest neighbour search required by optimal NBNN, the reader might have noticed that in practice, the sets of potential nearest neighbours to explore can be quite large, containing of the order of 10^5 to 10^6 points. We thus need to implement an appropriate search method. However, the dimensionality of the descriptor space is also large and traditional exact search methods, such as kd-trees or vantage point trees [Yianilos 1993] are inefficient. We chose Locality Sensitive Hashing (LSH) and addressed the thorny issue of parameter tuning by multi-probe LSH [Dong 2008] with a recall rate of 0.8 (see appendix D.1.3). We observed that resulting classification performances were not sensitive to small variations in the required recall rate; however, speed performances were. Compared to exhaustive "naive" search, the observed speed increase was ten-fold.

4.3.8.2 Experimental protocol and parameter selection

In every experiment, the datasets are equally split in a testing and a training dataset. The training dataset is itself split in two: one half serves as a feature database while the other is used for parameter selection. Thus, only half the

Datasets	SVM	χ^2-SVM	NBNN	Opt. NBNN
SceneClass13	67.85±0.78	76.70±0.60	48.52±11.35	75.35±0.79
Graz-02	68.18±4.21	77.91±2.43	61.13±5.61	78.98±2.37
Caltech-101 (5 classes)	59.20±11.89	89.13±2.53	73.07±4.02	89.77±2.31

Table 4.1: Performance comparison between the bag of words classified by linear SVM, χ^2-SVM, the NBNN classifier and our optimal NBNN. Scores indicate percentages of good classification.

features coming from the training set for the dataset of training features. We employed 128-dimensional SIFT features sampled by discrete difference of Gaussians (the original SIFT detector, [Lowe 2003]).

4.3.8.3 Single-channel classification

The impact of optimal parameter selection on image classification with our classifier, using just one feature channel, is illustrated and compared to original NBNN in table 4.1.

The first column refers to the classification of bags of words by linear SVM, which was found to be the most consistent classifier. In all experiments involving a feature quantisation step, we selected the codebook size that produced the best results (between 500 and 2000) and feature histograms were normalised by their L^1 norm. The classifier of the experiment from the second column is a kernel SVM with χ^2 kernel, which is recognised as the state of the art in bag of words classification [3] [Zhang 2007].

It should be noted that the NBNN implementation is ours and does not integrate the keypoint coordinates, contrary to [Boiman 2008].

The first observation we make about these experiments comes from comparing the first three columns of table 4.1. Contrary to the claims of [Boiman 2008], our experiments show that the initial formulation of NBNN remains inferior to SVM. The gap is very large for the SceneClass13 and Graz-02 datasets, which are the datasets for which the imbalance in the number of feature points in each class is greatest. On the other hand, NBNN outperforms linear SVM in the Caltech-101 experiment (73.07% versus 59.20%), where the number of points are roughly the same in each class. This observation serves to confirm our initial intuition that the distribution parameters of the training set does have an important influence on the NBNN classifier.

There are two more lessons to be learned from these experiments: first, correcting the NBNN formulation proves to be an absolute necessity if we

[3] Results from classification experiments with χ^2 kernel were drawn with the help of Paul Marcombes

Class	BoW/χ^2-SVM	[Opelt 2004]	NBNN	Opt. NBNN
Airplanes	91.99 ±4.87	97.5	34.17±11.35	95.00 ±3.25
Car-side	96.16±3.84	100.0	97.67 ±2.38	94.00±4.29
Faces	82.67 ±9.10	100.0	85.83±9.02	89.00 ±7.16
Motorbikes	87.80±6.28	94.3	71.33±19.13	91.00 ±5.69
Background	87.50 ±6.22	-	76.33±22.08	78.93±10.67

Table 4.2: Caltech101 (5 classes) class-by-class performance comparison

want to use unquantised features to advantage. Indeed, when we compare the third and fourth columns of table 4.1, the gain produced by parameter selection exceeds 15 percentage points in all experiments, which is considerable. Secondly, the difference between the first and the fourth column shows that NBNN can consistently outperform linear SVM, which is the state-of-the-art linear classifier for the bag of words representation. The gain is greater than 7 percentage points in all three experiments, which is significant given the simplicity of the chosen representation (a set of unquantised features) and classifier. Finally, we observe that, despite its linearity, optimal NBNN is on par with kernel SVM. It should be noted that the use of a χ^2 kernel requires the manual setting of a smoothing parameter, contrary to optimal NBNN. In our experiment, we chose the smoothing parameter that produced the best results.

The best results reported on the SceneClass-13 dataset have a good classification rate of 73.40% [Bosch 2006], though the experimental setting is slightly different, as they were obtained by using half the dataset for training and half the dataset for testing. Given the relatively small training set that we use, our results compare favourably with them.

We listed in table 4.2 class-by-class performances of four different methods, including one by [Opelt 2004] to compare our results with the state of the art. However, it should be noted that [Opelt 2004] train their algorithm with 60 images per class, hence twice as much as we do.

4.3.8.4 Radiometry invariance

In this section we make use of the multiple SIFT radiometry invariants described in [van de Sande 2010] (see appendix B.1).

The results described in [van de Sande 2010] were obtained with bag-of-word representations. In our experiments, we wanted to verify if the best invariant features would correspond to the ones selected by [van de Sande 2010].

We tested the efficiency of the various radiometry-invariant descriptors on the Caltech-101 (5 classes) dataset. Similarly to the experiments from section 4.3.8.3, images were classified by BoW/SVM, by uncorrected NBNN

	linear SVM	NBNN	Optimal NBNN
SIFT	62.05% ±1.87	73.07% ±0.60	89.77% ±2.50
OpponentSIFT	66.35% ±0.11	71.53% ±5.5	88.87% ±2.32
rgSIFT	54.00% ±0.01	74.57% ±5.86	85.20% ±2.34
cSIFT	64.35% ±0.01	73.07% ±5.39	86.20% ±3.40
Transf. color SIFT	64.20% ±0.11	63.5% ±6.41	89.03% ±2.60

Table 4.3: Correct classification rates on Caltech-101 (5 classes): Influence of various radiometry invariant features. Best and worst SIFT invariants are highlighted in green and red, respectively, in each column.

and by our optimal NBNN. Results are indicated in table 4.3. Each descriptor was evaluated separately from the others. Highlighted in red and green are respectively the worst and best results from the radiometry-invariant descriptors. In the following "flat NBNN" will refer to the original formulation of NBNN in which channels are combined with uniform weights, as opposed to out optimal NBNN with variable weights.

We first observe that different descriptors perform much differently, both in the flat and optimal classifiers. In the flat classifier (middle column of table 4.3), we can rank the descriptors by their level of efficiency and obtain approximately the same result as [van de Sande 2010]: rgSIFT is best, followed by cSIFT, opponentSIFT, and transformed color SIFT is last. Results for the density corrected optimal NBNN are always higher than for simple NBNN, as could be expected. However, much surprisingly, we observe that the relative relevance of the different channels is reversed! In particular, transformed color SIFT changes from the worst channel of the flat experiment to the best in optimal NBNN[4]. The gain of 35.53 points between both experiments is considerable and suggests that a very strong bias exists in the density estimation of the space of transformed color SIFT. This could be the case, for instance if the transformed color SIFT samples were distributed along a manifold of much lower dimension.

Further experiments demonstrated that combining all five descriptors did not consistently produce important performance gains, so the results are not shown here. In general, for classification, having just one radiometry-invariant feature channel is sufficient. However, this conclusion will not hold for detection (see experiments in section 4.3.8.6).

Our conclusions on which descriptor is better than others do not necessarily contradict the results given in [van de Sande 2010]. We admit, just like they do, that the relevance of the various invariances to radiometry changes strongly depend on the considered class and dataset. However, what our results show is that the robustness of SIFT features to local average and

[4]Naturally, the impact of these observations should be dampened by the relatively high variance of the classification rates.

variance changes of colour often suffices to obtain good descriptors.

4.3.8.5 Localised features and multiple channels

In this experiment, we show how the notion of feature channel can be employed to make use of the structural information associated to the layout of interest points. We draw our inspiration from the pyramid kernel developed in [Lazebnik 2006]. The main idea behind the pyramid kernel is that an image can be partitioned in rectangular regions of fixed sizes and that each region can be associated to a certain feature channel. In practice, pyramid kernels have been used for bag of words representations in several papers presenting state-of-the-art results, such as [Marszałek 2007b, Bosch 2007, van de Sande 2010]. However, in [Lazebnik 2006, Bosch 2007, van de Sande 2010], each bag of words is uniformly weighted in the final classifier, which means that regions are not weighted according to their relative relevance. [Marszałek 2007b] finds the optimal weights by genetic optimisation, but the obtained gains are disappointingly small.

In optimal NBNN, contrary to [Lazebnik 2006, Bosch 2007, van de Sande 2010], we do not compute distances between different channels. The contribution of each channel to the class prediction is independent from other channels. If one wanted to, it would not be difficult to adapt the formulation of optimal NBNN to truly pyramidal kernels. In such an experiment, we would have cross-channel parameters $\alpha_{n,n'}$, $\beta_{n,n'}$ and the combined predictor would take the following form:

$$\tau^-(I) - \tau^+(I) = \sum_n \sum_{d \in \chi_n(I)} \sum_{n \neq n'} \left(\alpha^-_{n,n'} \tau^-_{n'}(d) - \alpha^+_{n,n'} \tau^+_{n'}(d) + \beta^{-+}_{n,n'} \right) \quad (4.70)$$

We leave to the reader the task of experimenting the benefits of parameter selection in the pyramidal kernel. In the following, cross-channel parameters are set to zero and we optimise over the $\alpha^c_{n,n}, \beta^{c,c'}_{n,n}$, just like described in 4.3.5.

Images are partitioned in horizontal regions that constitute our sets of channels. Each set is denoted by $1 \times n$, where n varies between 1 and 4 (see figure 4.4 for an illustration). We conduct four different experiments on the SceneClass13 dataset with 1 (1×1), 3 ($1 \times 1 + 1 \times 2$), 4 ($1 \times 1 + 1 \times 3$) and 5 ($1 \times 1 + 1 \times 4$) channels. Visual features are SIFT.

Results are shown in table 4.4. The performance gain contributed by the addition of regional channels is positive, both for flat and optimal NBNN. Even though the maximum gain is smaller for optimal NBNN than for flat NBNN (3.24 versus 5.40 percentage points), it remains significant. It should be noted that, in our experiments, combining 1×1 with 1×2, 1×3 and 1×4, for a total of ten feature channels, did not significantly improve the classification results over the $1 \times 1 + 1 \times 4$ experiment.

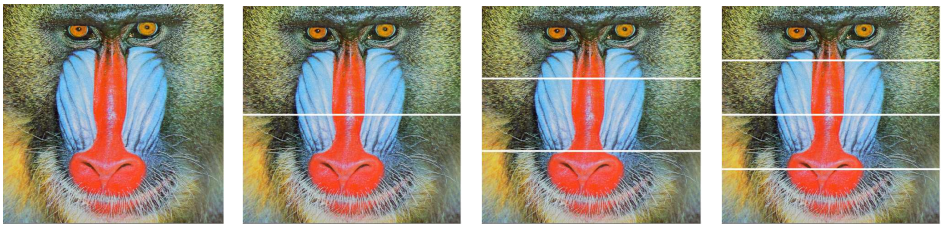

Figure 4.4: Feature channels as image subregions: 1×1, 1×2, 1×3, 1×4

Channels	#channels	NBNN	Optimal NBNN
1×1	1	48.52%	75.35%
$1 \times 1 + 1 \times 2$	3	53.59%	76.10%
$1 \times 1 + 1 \times 3$	4	55.24%	76.54%
$1 \times 1 + 1 \times 4$	5	55.37%	78.26%

Table 4.4: Multi-channel good classification rates, SceneClass13 dataset

4.3.8.6 Classification by detection

The Graz-02 dataset is a good illustration of the necessity of classification by detection in order to diminish the importance of background clutter. In this set of experiments, the dataset is divided in just two classes: the positive class contains images bicycles, while the negative class contains all other images. In this context, the estimated label of a test image I is given by:

$$\hat{c}_I = sign\left(E_2(I, back) - E(I, bike, \hat{\pi}^{bike})\right), \tag{4.71}$$

where we have retained the notations from section 4.3.7.2. The distance correction parameters that have to be computed for this problem are the α_n^c, β_n^c where c is in $\left\{bike, \overline{bike}, back\right\}$. For the sake of parameter selection, the sets of images from classes $bike$ and $\overline{bike}$ are obtained by decomposing each positive image in two complementary parts: the points located on a bicycle instance are in $bike$ while others are in $\overline{bike}$. Labels are obtained by a bounding box present in the training dataset. Density estimation parameters were learned following the procedure described in section 4.3.6.

We combined all five SIFT radiometry invariants already employed in section 4.3.8.4. We obtained a classification rate of 78.70% with our optimal NBNN classifier, while NBNN achieved just 68.35%. Classification by detection raised this rate to 83.60% on the bike dataset, which is a significant improvement. This is close to the results reported in [Opelt 2004], [Mutch 2008], and [Moosmann 2007] which report classification rates of 77.80%, 80.50% and 84.40%, respectively (see table 4.5 for more detailed results). Detection examples are shown in figure 4.5. What can be observed

Class	NBNN	Optimal NBNN	Optimal NBNN (classif. detect.)
bike	68.35±10.66	78.70±4.67	83.60
people	45.10±12.30	76.20±5.85	-
car	42.40±15.41	82.05±4.88	

Class	[Mutch 2008]	[Opelt 2004]	[Moosmann 2007]
bike	80.50	77.80	84.40
people	81.70	81.20	-
car	70.1	70.5	79.9

Table 4.5: Per-class classification, and classification by detection rates for the Graz-02 database.

on these visual results is that non-parametric NBNN usually converges towards an optimal object window that is too small relatively to the object instance. This is due to the fact that the background class is more densely sampled. Consequently, the nearest neighbour distance gives an estimate of the probability density that is too large. It was precisely to address this issue that optimal NBNN was designed.

4.3.9 Optimal NBNN versus Handfuls of Features

In this experiment, we compare results obtained by optimal NBNN and the handful of features by adopting the same experimental framework as in section 4.2.4. Results are plotted in figure 4.6, in addition to the previous results relative to the handful of features, the summation kernel and the bag of words.

We see that optimal NBNN outperforms the handful of features by about 15 percentage points, and this gap remains consistent as the average window size varies. We argue that this is evidence of the superiority of image-to-class distance over image-to-image distance.

4.4 Distance-collapsed graphs

In this section, we take advantage of the optimal multi-channel NBNN classifier developed in section 4.3 to perform classification of visual feature graphs. We do so by using an idea based on the quantisation of graph distances.

In section 2.3.2.4 we argued that the intrinsic dimensionality of graphical structures was, in general, too high and discriminative to be efficiently handled in the context of classification. Moreover, we took in section 3.4 a step toward graph simplification by appearance quantisation that allowed us to produce a finite dimensional graph representation. Considering that a feature graph consists solely of a set of graph nodes (i.e: in the case of the feature graph, visual features) and a transition matrix, or a distance

Figure 4.5: Subwindow detection for NBNN (red) and optimal NBNN (green). For this experiment, all five SIFT radiometry invariants were combined. (see text)

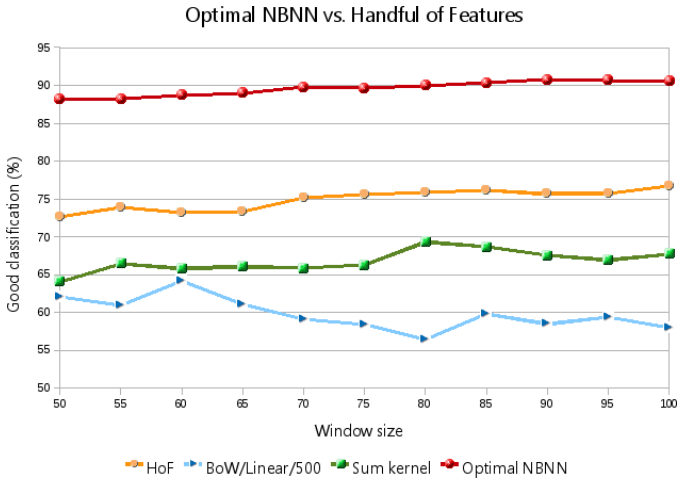

Figure 4.6: Classification by optimal NBNN following the experimental setup of section 4.2.4.

matrix, the next available possibility for graph representation is to quantize the distance matrix, instead of the node set.

Our contribution, in this section, is the classification of attribute graphs by decomposition of the set of all point pairs in disjoint channels. Different channels contain pairs of points that are separated by different graph distances.

More to the point, we begin by choosing a certain type of visual feature graph (see section 3.2) and a distance measure inside the graph (section 3.3). For a given visual feature graph, we then consider all point pairs and quantize the value of the graph distance that separates them in K bins $[t_0, t_1[, \dots [t_{K-1}, \dots t_K[$. The result of this distance quantisation, or *distance-collapse*, is K sets of point pairs $(\chi_k)_{0 \leq k < K}$:

$$\forall k \in [1, K],\ \chi_k = \{(v_i, v_j) \mid d(i,j) \in [t_{k-1}, t_k[\} \text{ with } t_{k-1} < t_k, \qquad (4.72)$$

where d is the graph distance that we have chosen (adjacency matrix distance, shortest path distance or commute time distance).

Note that we adopted the same notations as the feature channels of section 4.3.4; indeed, the different sets of point pairs can be considered as channels which collectively contribute to the faithful description of the original feature graph. We can thus use the multichannel NBNN classifier for graph classification.

In effect, what distance-collapse does is to regroup pairs of points that are separated by similar distances. The first channels two t_0, t_1 will collect the point pairs separated by a short distance, while the channels with higher indices will capture the information related to long distance interactions.

4.4.1 Experiments

In our experiments, we used the shortest path distance d_{SP} so as to obtain graph distances invariant to the graph size. We are thus dealing with discrete, integer graph distances between graph nodes, and the feature channels of a graph G can be written:

$$\forall k \in \mathbb{N},\ \chi_k(G) = \{(v_i, v_j) | d_{SP}(i,j) = k\}\,. \tag{4.73}$$

This means that channel 0 consists of all individual feature points (each feature point is separated from itself by a distance equal to 0); channel 1 consists of feature pairs that are directly connected; channel 2 is the set of point pairs that are neighbours of neighbours, but not direct neighbours; etc. Because any two nodes of a graph are separated by a shortest-path distance in $\mathbb{N}$, we can say that the disjoint union $\bigcup_{k\in\mathbb{N}} \chi_k(G)$ forms an adequate representation of feature graph G. Moreover, we are entitled to consider that close range interactions between graph nodes are most informative of the graph structure. In the representation by feature channels of the graph, we can thus choose to retain a certain finite number K of channels, such that point pairs separated by a a distance greater than K will not be taken into account. This is similar to the approximation that allows us to dismiss most of the large eigenvalues and the corresponding coordinates in an commute time embedding of the graph nodes.

In practice, we limited the number of channels to a number below 5. This choice made sense, not only from the point of view of computational practicability, but also from the point of view of global performances.

We conducted our experiments on the SceneClass13 dataset (see appendix A.6, with 20 training and 20 validation images per class.

Exp. 1 In the first experiment, we employed the unoriented, unweighted hierarchical feature graphs described in section 3.2.3.1.

Exp. 2 In the second experiment, we tested a novel kind of oriented graph in which each node was connected to its three nearest neighbours relatively to the normalised spatial distance Δ_{geo} from equation 3.6.

Experiment results are reported in table 4.6, both for NBNN and optimal NBNN.

Again, we observe in table 4.6 a strong gain of optimal NBNN over NBNN, but that is hardly a surprise anymore. We also observe that the performances of NBNN systematically degrade as the number of channels increase, which is another indication of its sub-optimality with regard to multiple channels. On the other hand, there is a performance increase of optimal NBNN as we increase the number of channels from one to two, but it is disappointing to see that this gain does not persist as we keep increasing the number of channels. Naturally, the main benefit of additional channels

	Exp.1		Exp.2	
#channels	NBNN	Opt. NBNN	NBNN	Opt. NBNN
1	32.02%	46.08%	32.02%	46.08%
2	28.96%	47.35%	29.00%	47.81%
3	27.46%	45.73%	27.38%	45.69%
4	26.58%	42.81%	26.85%	42.88%

Table 4.6: Classification of distance-collapsed visual feature graphs. See text of section 4.4.1.

is a guaranteed performance increase on the cross-validation set. In practice, we observe this gain on the cross-validation set, but it does not extend to the validation dataset. This means that channels beyond χ_1 do not contain any relevant information with regard to the image class. Therefore, the most we can get out of graphical structures comes from direct connections between graph nodes. As we realised in this experiment as well as in other datasets, the gain brought by the information related to the direct connectivity of visual feature graphs is of the order of one to two percentage points.

4.5 Conclusion

The goal of this chapter was to introduce two feature-based image classifiers for which feature quantisation was not required. The handful of features (HoF) represents an image by an unordered set of features and a specific SVM kernel is employed for classification. The SVM kernel that we have introduced is in essence a sum of nearest-neighbour distances that closely resembles the Gaussian kernel with L^2 distance. Optimal NBNN, the second classifier, differs with the handful of features mainly because NBNN relies on an image-to-class distance that improves its generalisation capacity. However, the linear program employed for parameter selection remains very similar to SVM training. At the end of the day, the main difference of our classifiers with the BoW/SVM approach is that they avoid feature quantisation. It is the fact that we make use of feature points to their full capacity that is at the source of the performance gain that we obtain. For questions of computational practicability, the HoF is better suited to smaller sets of features while optimal NBNN works best when the number of features extracted from the image is large. These two classifiers are thus complementary. In both cases, in all attempted experiments, the performance gain is above five percentage points, which is considerable given previously observed SVM performances and the similarity of our classifiers with SVM.

The initial motivation for building our optimal NBNN was the need of a multi-channel classifier that was able to deal with continuous attributes. Thus, our contribution does not only consist in an improvement over the

orderless image representation: in our eyes, the most useful properties of optimal NBNN is that it naturally generalises to multiple channels. This property allows us to design a classification method for attributed graphs that we use to address the problem raised at the beginning of this manuscript: the distance-collapsed graph approach allows us to represent images by taking into account the general layout of points and without quantising visual features. Even though the quantitative benefit we observe for the classification of images is small (of the order of one to two percentage points), we believe the distance-collapsed graph representation can be employed in different contexts that best exploit its characteristics.

Finally, the simplicity and the linearity of the classifier allows us to foresee further improvements akin to the improvements developed for linear SVM. We believe that concerns of the computer vision community will shift from adequately quantising visual features, a step that is in our view no longer required, to the construction of better feature indexes.

Chapter 5

Conclusions

As we have emphasised in the introduction, the practice of computer vision requires the joint use of techniques stemming from multiple areas of scientific expertise. In this thesis, we have built novel connections between research topics that are seldom studied together. In particular, we have combined the effectiveness of visual descriptors to powerful results coming from spectral graph theory. This has allowed us to produce a novel image representation that improves on other representations that do not incorporate the layout of interest points. Moreover, we have introduced a classifier of sets of features that is sufficiently flexible to incorporate various image representation types, such as graphical structures.

5.1 Main contributions

5.1.1 Image representations

The new image representation we propose is based on spectral properties of a graph of visual features extracted from the image at sparse locations. Interesting properties of this representation are its robustness to rigid geometric transforms on one hand, and the expression of properties related to the global geometric layout of visual features on the other hand. Its main drawback is its reliance on a lossy quantisation of the feature space that is detrimental to the discriminative power of visual features. Among other results, we investigated the impact on the performance of the different strategies for graph construction and of the choice of the graph distance matrix.

5.1.2 A linear, multi-channel classifier of point sets

The issue of point visual feature quantisation is addressed by an innovative nearest neighbour-based classifier of feature sets. This classifier possesses two interesting properties: first, its linearity allows us to use for fast object detection. Second, because of its ability to optimally combine multiple feature channels, we can formulate an entorely new solution to the problem of graph classification.

5.1.3 Quantitative results

We observed that the benefits of adding a graph layer to the image representation varied between 1 and 2 percentage points. On the other hand, we have shown that an appropriate classifier that avoids a feature quantisation step can bring an improvement of more than 10 percentage points, and up to 20 percentage points, over linear support vector machine (SVM), its quantised counterpart, on a wide range of experiments. Moreover, performance are better or comparable to the state-of-the-art kernel SVM with χ^2 kernel. However, as opposed to kernel SVM, the linearity of our classifier allows us to perform object detection by sliding windows. The benefit of the graph layer is conserved in the unquantised experiments and is added to the gain provided by the classifier, on the condition that only very short range interactions be taken into account in the representation.

5.2 Limitations and suggestions for improvement

Seeing these quantitative gains, we can make two observations: first, incorporating the spatial layout information in the image representation by means of a graphical structure does improve the result, but not sufficiently to justify the supplementary requirements in memory storage and processing power due to the augmented representation. Though it is hard to predict with certainty what kind of method will definitely solve the problem of image classification, we believe that properties of the visual graph for image representation will not be the decisive factor of this method. However, the performance gap produced by the removal of the feature quantisation step points us in a direction that is likely to bear some very fruitful progress. Keeping these observations in mind, it is possible to point at several bottlenecks that might hinder our whole approach, and think about working ideas that should be investigated to address these bottlenecks.

5.2.1 Model overfitting

While using optimal NBNN for the classification of distance-collapsed feature graphs, we have experimentally observed that model overfitting quickly becomes an issue as we add more channels to take into account far-range interactions of the feature graph. The reason for this is that far-range interactions are much less significative than short-range interactions for image representation. A solution would consist of learning a parametric model for feature graph generation, but then the potentially large number of feature points in the image becomes an issue of computational power.

5.2.2 Nearest neighbor search in large databases

Another limiation of optimal NBNN appears when we increase the number of training images and the number of classes: an index of visual features must be built for each class and the nearest neighbour of each visual feature from the testing set must be found in each index. This search step is hungry both in terms of memory requirements and processing time. Though we have only partially addressed this issue in this thesis, by making use of locality sensitive hashing, we suspect large improvements can be obtained from distributed computing techniques.

5.3 Future work

5.3.1 Relaxing the naive Bayes assumption

First, we need to admit that the naive Bayes assumption that we make in optimal NBNN is too strong and that some level of spatial coherence between the visual features should be conserved by adding a correlation term between the various features. The resulting energy would take the form of a conditional random field (CRF) in which conditional probabilities could be computed by parametric estimation, similarly to optimal NBNN. Although the exact nature of the interaction between features due to their respective scale, location and orientation should be investigated more precisely, some inspiration could be drawn from the shape factor of the constellation model (see section 2.4).

5.3.2 Unsupervised sub-classification

Secondly, we need to take into account the fact that a single object class can in reality be composed of several sub-classes. While the source class presents a large appearance variability over its various instances, each sub-class should have a more consistent appearance. For instance, a class can be decomposed over the various viewpoints: cars seen from the side and from the front should constitute different sub-classes, as it would diminish the intra-class variations. However, available training labels usually do not incorporate such precise information and "sub-labelling" should thus be done in an unsupervised manner. Using the framework provided by optimal NBNN, we can design a strategy that subdivides the features of a class in different channels so as to obtain multiple distributions of features. The purpose of this separation would be to improve the class specificity of the various channels. It could be achieved by taking into account global visual features, or at least features sampled at a larger scale than the original features in order to take into account the visual context of the individual points.

5.3.3 Training data pruning

Thirdly, in the constitution of a training set of feature points for optimal NBNN, computational limitations and some practical considerations prompt us to limit the size of the feature distributions. Indeed, by blindly adding all class features to the database we unwillingly incorporate many features that are not relevant to the investigated class; the nearest neighbour searches that follow are also slowed down. To tackle this issue, we propose to suppress training points for which the probability distribution of the opposite class is higher than for the point class by a given threshold. This can be done by leave-one-out cross-validation.

5.4 Last word

If we want to take the methods of image classification out of the research world and build practical industrial applications, several barriers must be lowered. Algorithms must be made more efficient, in terms of quality, and they must be scaled to a much greater scale: the web scale. We believe that methods based on visual features and nearest neighbour classifiers have the potential to take great steps in both these directions. We have shown that they can outperform sparsely quantised features while retaining the important linearity property. Moreover, their simplicity and the fact that they do not require the definition of a global model make them suitable for large-scale applications.

The field of computer vision has been progressing at such a speed for the past decade that it is safe to predict the advent of large scale, challenging image classification applications in the few years to come. The scientific and technical advances that will make this achievement possible will define a new milestone for the world wide web, redefining the transmission of information in ways that can be barely foreseen.

Appendix A

Datasets

The experiments performed during this thesis were conducted on a wide variety of image datasets, most publicly available. We describe them here briefly.

A.1 Synthetic graph dataset

We introduce an artificial dataset that was specifically designed to exemplify the need to take into account the layout of features in the image to perform image classification. We consider a set of images belonging to one of two classes, the positive and the negative class. Each image contains a variable number (~ 1000) of points of four different colours. In each image, the positions of these points are generated by two bivariate Gaussian distributions with random mean. In an image belonging to the positive (respectively: negative) class, the first distribution evenly generates orange and pink points (resp: orange and green), and the second distribution evenly generates green and blue points (resp: pink and blue).

More precisely, for each image the locations $\mu_1 = (x_1, y_1)$, $\mu_2 = (x_2, y_2) \in [-1, 1] \times [-1, 1]$ of the center of two gaussian distributions of equal covariance $\sigma = 0.5$ are randomly generated following a uniform distribution. In a positive image, the first distribution generates orange and pink points, and the second distribution generates green and blue points. In a negative image, the first distribution generates orange and green points, and the second distribution generates pink and blue points. Image examples of the two classes are shown on figure A.1. For the purpose of image classification, ten images were randomly generated for each of the two classes.

A.2 Urban/Vegetation satellite dataset

This dataset was been generated from a sample of panchromatic Quickbird satellite images from the Beijing (China) area. The resolution of these images is 0.5 cm, which places them in the category of high resolution images. Since satellite swaths produce relatively large image files (typically of the order of 10^2 Gb, the input image was split into smaller, non-overlapping 512×512 pixels subimages. A subset of 231 subimages was manually assigned one of two labels: 128 subimages were placed in the "urban" category

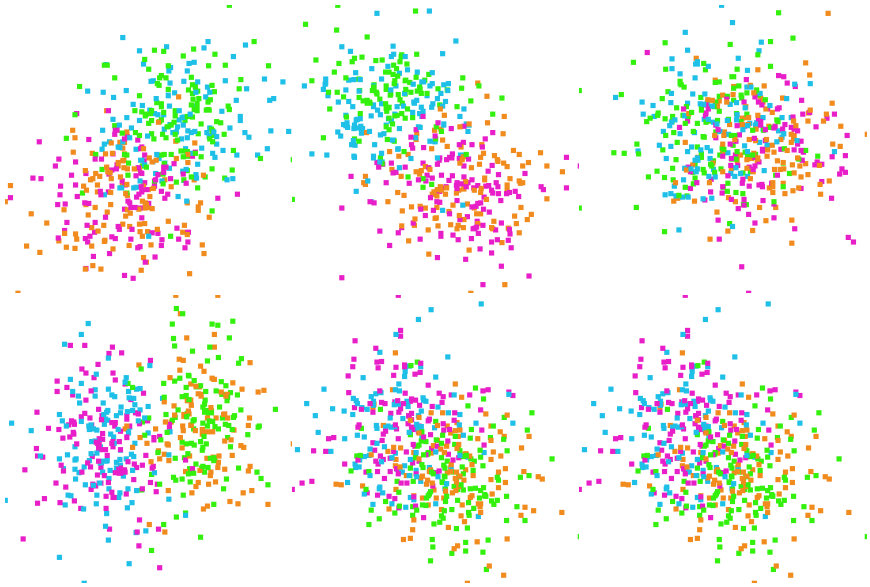

Figure A.1: Synthetic dataset. Top row: positive dataset samples. Bottom row: negative dataset samples. (Best viewed in colour)

and 103 in the "vegetation" category. This dataset was partitioned in two equal halves, thus producing a training and a testing dataset.

Examples of each category (urban and vegetation) is given in figure A.2.

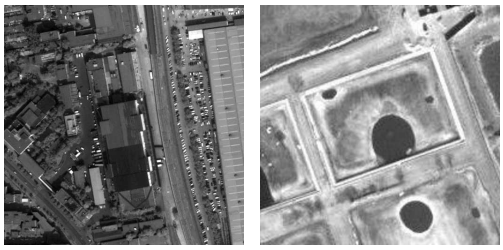

Figure A.2: A sample of high resolution subimages: urban (left) and vegetation (right).

A.3 Graz-02 and Graz-02-bicycles

The Graz-02 dataset [Marszałek 2007a] is a publicly available set of 1096 colour photographs of size 640×480 taken in the area of Graz, Austria. Each image contains at least one instance of exactly one among three classes: bike (365 images), car (420 images) or person (311 images). The particularity of

this dataset is that each image has only one label. Moreover, the training data includes the exact segmentation of every instance from the three classes. For these reasons, the Graz-02 dataset is an ideal candidate both for image classification and object detection.

Because we do not always need labels from multiple classes to perform image recognition tasks, in particular in the context of object detection, we realised certain experiments using just the "bike" label from the Graz-02 dataset. In such cases, images containing cars or persons are labelled as "background". This dataset is referenced under the name of Graz-02-bicycle.

Figure A.3: Samples from the Graz-02 dataset: bikes (top), cars (middle), persons (bottom).

A.4 Caltech-101

The Caltech-101 dataset [Fei-Fei 2006] is one of the most popular benchmarking datasets in the context of image classification. It consists of images of variable sizes labelled with one among 101 class labels. Each class contains a variable number of images. This dataset presents a relatively low intra-class variability, as many objects are shown in a similar "canonical" pose.

Because the large number of classes makes this dataset impractical for applications requiring a large amount of memory, we sometimes limited ourselves to the five most populated classes of this dataset: faces, airplanes, cars-side, motorbikes and background. In these cases, this dataset is referenced under the name Caltech-101 (5 classes).

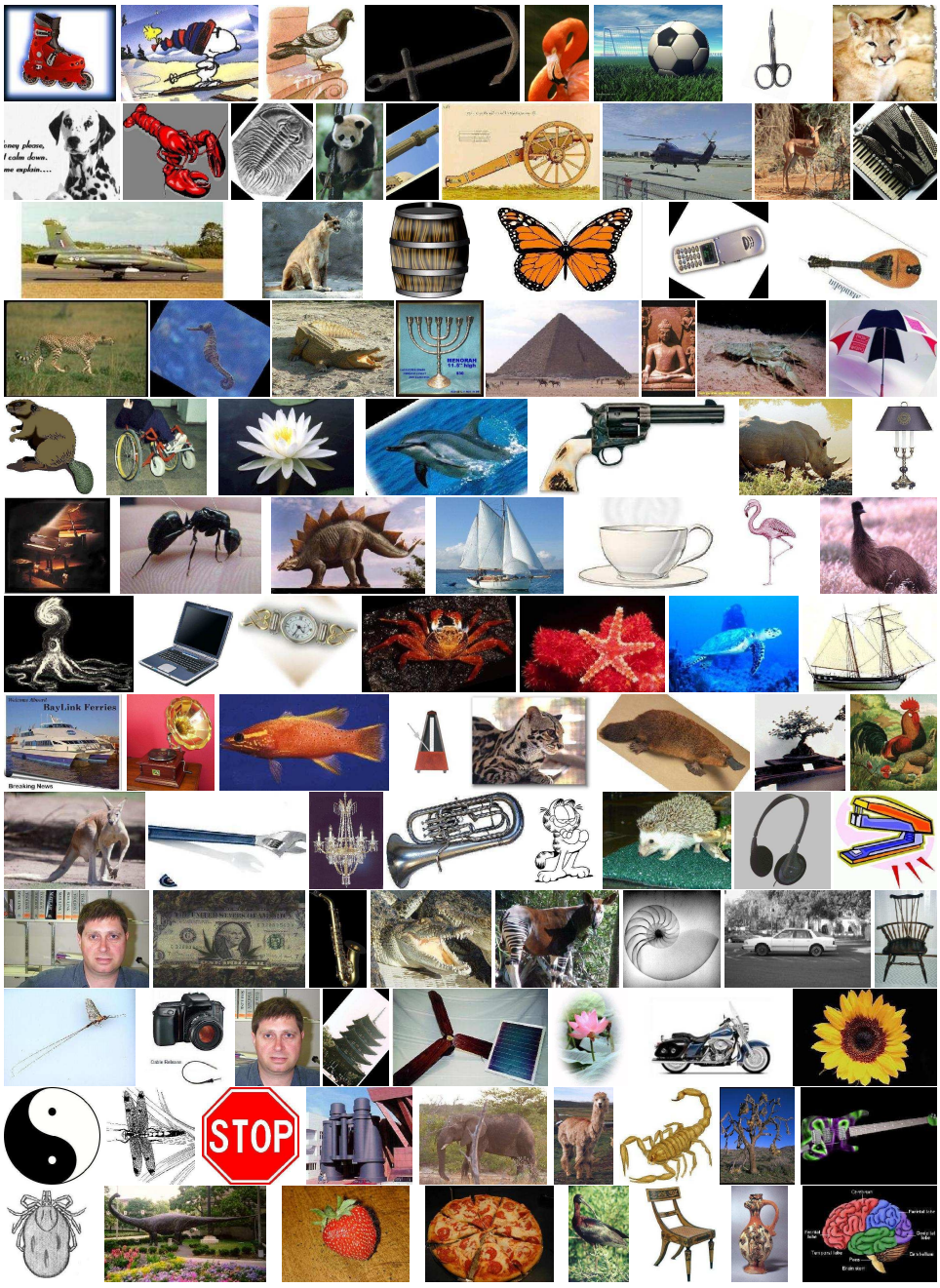

Figure A.4: The Caltech101 dataset

A.5 PASCAL VOC 2006-2009

The yearly PASCAL visual object classes (VOC) challenges [Everingham 2006, Everingham 2007, Everingham 2008, Everingham 2009] provide datasets that rank among the most challenging for image classification, object detection and segmentation. The challenge contains twenty object classes with a variable number of images that increases year after year. Images are sampled from the online Flickr database and manually labelled with rectangular bounding boxes for each object instances. Because of the realism of this dataset, intra-class variation is large and object instances are often degraded by occlusions and extreme lighting conditions.

In the classification task, the goal of the PASCAL VOC challenge is to assign one or multiple labels to each image with a certain confidence score. These scores are used to draw precision/recall curves (or receiver operating curves (ROC) for challenges prior to 2008), from which an average precision is computed.

Figure A.5: Samples from the PASCAL 2007 classification dataset

A.6 SceneClass13 and indoor dataset

The SceneClass13 dataset, also called "13 natural scene categories" [Fei-Fei 2005], contains images of variable sizes representing "scenes", rather than objects. The thirteen classes are: bedroom, suburb, kitchen, living room, coast, forest, highway, inside city, mountain, open country, street, tall building and office. Because a scene occupies the full extent of an image, this dataset is ideal for image classification.

The indoor dataset restricts the SceneClass13 dataset to the four indoor classes: bedroom, kitchen, living room and office.

A.7 Satellite8

This dataset of satellite images, illustrated in figure A.7, is composed of 878 images of size 200×200 coming from eight classes: (1) work place, (2) big buildings, (3) golf fields, (4) greenhouses, (5) small industry, (6) fields, (7) dense urban, (8) housing area. It was introduced by [Bordes 2008].

(a) Bedroom (216) (b) Coast (360) (c) Forest (328) (d) Highway (260) (e) Inside City (308)

(f) Kitchen (210) (g) Living room (289) (h) Mountain (374) (i) Office (215) (j) Open country (410)

(k) Street (292) (l) Suburb (241) (m) Tall building (356)

Figure A.6: SceneClass13 dataset. Number of images for each class is indicated in brackets.

(a) Work place (21) (b) Big buildings (68) (c) Golf field (104) (d) Greenhouses (111)

(e) Small industry (122) (f) Fields (52) (g) Dense urban (238) (h) Housing area (183)

Figure A.7: "Satellite8 dataset". The number of images for each class is indicated in brackets.

Appendix B

Visual features

B.1 SIFT radiometry invariants

It has been shown in [van de Sande 2010] that the use of color for visual features could be a decisive parameter for image classification. However, there are many different ways of taking colour into account in visual features, as an object can be subject to a wide array of photometric transforms. As an illustration of the amount of variation that a given object can sustain under different lighting conditions, the authors of [van de Sande 2010] include an example based on the Pascal VOC2007 challenge [Everingham 2007] (see figure B.1). As can be observed, the range of values that the colour of a given object class can take is very large, depending on the lighting conditions and the visual sensor employed. It is thus necessary to design descriptors that are robust to certain photometric changes.

Figure B.1: Instances of a given object class (potted plant) in world scenes. (image courtesy of [van de Sande 2010], Pascal VOC 2007)

In order to select the best features, the authors of [van de Sande 2010] perform image classification with various descriptors and combine the four or five best performing channels to produce results that outperform the state of the art on the Pascal VOC2007 dataset. Apart from the original SIFT, each descriptor is a concatenation of three SIFT, where each one has been computed in a channel corresponding to a particular colour space:

SIFT is the original 128-dimensional descriptor introduced in [Lowe 2003].

Opponent SIFT is computed in the opponent space. The opponent space

is composed of linear combinations of RGB channels:

$$\begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R+G-2B}{\sqrt{6}} \\ \frac{R+G+B}{\sqrt{3}} \end{pmatrix} \tag{B.1}$$

Opponent SIFT is invariant to linear combinations of light changes and shifts.

cSIFT is also computed in the opponent space, but channels O_1 and O_2 are divided by O_3 to add invariance to intensity change.

rgSIFT corresponds to intensity invariant channels $r = R/(R+G+B)$, $g = G/(R+G+B)$ and $r+g$.

Transformed color SIFT consists of the SIFT features computed in a space where the colour information has been centered and normalised by the local mean and variance. However, because SIFT is a histogram of normalised gradient, this representation is strictly equivalent to a SIFT computed in each of the RGB channels.

B.2 Speeded up robust features (SURF)

SIFT are highly reliable and discriminative visual features, but they have relatively large requirements in terms of memory and extraction speed. It was to address this issue that Bay et al introduced SURF [Bay 2006]. SURF features build on the same fundamental principles as SIFT, including the difference of Gaussian detector and the image scale space, but does so by using a certain number of approximations that increase the speed of both the detection and description steps.

First, the authors of SURF noticed it was possible to approximate the Gaussian filters required for computation of the image first and second order derivatives by box filters. Box filters make it possible to compute image derivatives using integral images, which are extremely fast. In particular, the computation of box filters using integral images is done in constant time, independently from the filter size; therefore, the image scale space can be computed without image resizing. The drawback of this approximation is that it makes the detector more less robust to rotations of angle an odd factor of $\pi/4$.

Secondly, the dimensionality of the SURF descriptor is only half the dimensionality of a SIFT. With 64 dimensions, distances between visual features are computed in half the time and require only half as much storage.

These benefits do not have a large impact on the performance of the detector/descriptor combination. For this reason, we frequently made use of SURF in our experiments.

Appendix C

PASCAL 2008 VOC Challenge: Propagation of class assignment belief in feature graphs

---

In October 2008, we took part in the classification task of the PASCAL VOC challenge. We summarise here our method, based on the inference of feature labels thanks to non-loopy belief propagation inside feature graphs.

In each image of the training and testing sets, we build a hierarchical feature graph (see section 3.2.3.1) with oriented edges: edges are drawn from high-scale features to low-scale features. The nodes of these feature graphs consist of scale invariant interest points (Laplacian of Gaussians and SURF [Bay 2006]) and edges represent spatial proximity of the points. Each interest point descriptor is quantised according to a codebook of size 3000. The attribute y of each node is the index of the codebook entry to which it is assigned. x is a probabilistic estimate of the class to which it belongs.

Learning of graph model The training set allows us to learn a simple statistical model of the graph nodes and edges by estimating the values of $P(y|x)$ and $P(x|x')$ for all x, y and x', where x' refers to the label of a node that is an ancestor, in terms of graph connections, of the node labelled by (x, y). In practice we learn one model $\mathcal{M}_c$ per class c and the labels for model $\mathcal{M}_c$ are x_c and x_0, where x_0 is the background class. During the construction of model $\mathcal{M}_c$ we consider that nodes that do not belong to an object of class c belong to the background class.

Graph model propagation and image classification For each image and for each model $\mathcal{M}_c$ the labels of the graph nodes (x_c or x_0) are computed by non-loopy belief propagation [Yedidia 2003]. A representation of the image for class c is then built in the form of a double bag of features. The first histogram contains the features for wich the belief of belonging to class c is below 0.5, and the second contains the features of high belief. We thus obtain one representation per image and per object class. Similarly, representations are built for each training image. Finally, images are

separated by a set of Adaboost classifiers; each classifier is trained by an equal number of positive and negative images in order to compensate for the unevenness of the dataset.

Results Lack of space prevents us from listing the detailed class-by-class results to the competition; these results are available on the PASCAL 2008 VOC webpage[1] ("ECPLIAMA" entry). With a median average precision score of 28.4, our contribution ranked 11 out of 19, though we admit that the median average is not really an appropriate measure of performance for this particular challenge.

[1] http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2008/results/index.shtml

Appendix D

Algorithm implementation

The experiments described in this manuscript were realised by computer programs designed, mostly, in C/C++. All the programs were entirely written by the author, with the exception of a great number of libraries that performed functions the author felt unnecessary to reimplement.

Coding specifically for research is a process that has a certain number of specificities over coding for productivity. The main difference is that is can seldom be foreseen, which methods will be implemented, and therefore which data structures will be necessary. At the same time, while the amount of code grows, there must be no doubt regarding the correctness of the programs (i.e: they must effectively be doing what they are expected to be). Programming requirements thus include strong agility and strong reliability. In our particular case, reliability was replaced by confidence, as each program part was considered reliable after it had been employed successfully multiple times. Agility was obtained by intensive code reuse: by breaking down long pieces of code into multiple pieces, and by heavily relying on templated function arguments and object classes, we were able to frequently modify large parts of our programs to suit our needs. In particular, we had to redesign certain open-source libraries so that they could accept custom data structures as arguments

D.1 Libraries

D.1.1 OpenCV

The image loading steps, as well as most various image processing function were carried out by the OpenCV library (v1.0.0). This library is among the most competitive image processing libraries in terms of speed. However, it is regrettable that development of extensions of the the OpenCV library itself are so difficult to implement: because OpenCV has to deal with so many types of incompatible objects (`CvMat*`, `IplImage*`, each with different pixel types, storage and pixel access conventions), the development of supplementary functions by outside groups remains very time consuming. Nonetheless, OpenCV remains the image processing library of reference for many.

D.1.2 GNU Linear Programming kit (GLPK)

The GLPK is a C library for linear programming and mixed integer programming. Despite the fact that it is not object-oriented, the GLPK is extremely easy to use, in addition to being greatly reliable. We used the GLPK for estimation of distance correction parameters in optimal naive Bayes nearest neighbours.

D.1.3 Multi-probe Locality Sensitive Hashing (MP-LSH)

Locality sensitive hashing is a method for approximate nearest neighbour search that remains efficient in high dimension, contrary to k-d trees. However, LSH requires a time-consuming step of parameter tuning; multi-probe LSH (MP-LSH) alleviates this difficulty by finding optimised parameters that match a certain correct retrieval precision criterion. An open source implementation exists[1] for indexing of `float*` objects. We had to adapt it to data points of different types, namely `std::vector<float>`. Our solution was to produce a templated version of the library.

D.1.4 Support Vector Machine (SVM)

Similarly to our implementation of MP-LSH, we had to adapt existing libraries for SVM to templated objects and kernels. This was a requirement of the handful of features implementation (see section 4.2).

D.2 Parallel computing

We were able to get over some major computational limitations by multithreading of the greatest part of our software. In other words, multithreading made possible experiments that would have otherwise taken weeks to achieve. The simplicity and efficiency of OpenMP[2] allowed us to run our software to machines equipped with multiple cores and processors with minimal effort. For instance, testing often requires to apply a certain model to every testing image, independently from one another: with OpenMP, such a loop can be made parallel with simply one line of code.

The huge benefits granted by the use of multiple cores and processors prompt us to imagine what it would be like to run image recognition software on multiple *machines*. The cost of use of a machine has dramatically decreased with the emergence with cloud computing such as Amazon EC2[3] and the Google App Engine[4]. The combination of such solutions with effi-

[1] http://lshkit.sourceforge.net/
[2] http://www.openmp.org
[3] http://aws.amazon.com/ec2/
[4] http://code.google.com/appengine/

cient and free distributed computing software, such as Hadoop[5], and simple distributed programming models, such as Map/Reduce [Dean 2008], make distributed computing more open and affordable than ever. Although we did not make use of these solutions in our work, we believe that the computer vision community at large could greatly benefit from them.

[5] http://hadoop.apache.org/

Appendix E

Publications of the author

---

1. R. Behmo, P. Marcombes, A. Dalalyan, V. Prinet. Towards Optimal Naive Bayes Nearest Neighbor, In *Proc. European Conference on Computer Vision (ECCV)*, Hersonissos, Greece, September 5-11 2010.

2. R. Behmo, V. Prinet, N. Paragios. An Application of Graph Commute Times to Image Indexing, In *Proc. IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 6-11 2008.

3. R. Behmo, V. Prinet, N. Paragios. Graph Commute Times and Image Classification, In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Anchorage, USA, June 24-26 2008

Bibliography

[Agarwal 2006] Ankur Agarwal and Bill Triggs. *Hyperfeatures – Multilevel Local Coding for Visual Recognition*. In European Conference on Computer Vision (ECCV), 2006.

[Aha 1997] David W. Aha. Lazy learning. Kluwer Academic Publishers, Norwell, MA, USA, 1997.

[Aldous 1995] David Aldous and James Allen Fill. *Reversible Markov Chains and Random Walks on Graphs - Chapter 9: A Second Look at General Markov Chains*, 1995.

[Alpert 2008] Jesse Alpert and Nissan Hajaj. *We knew the web was big...* http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html, 2008.

[Barla 2002] Annalisa Barla, Emanuele Franceschi, Francesca Odone and Alessandro Verri. *Image Kernels*. In SVM '02: Proceedings of the First International Workshop on Pattern Recognition with Support Vector Machines, pages 83–96, London, UK, 2002. Springer-Verlag.

[Bay 2006] Herbert Bay, Tinne Tuytelaars and Luc Van Gool. *SURF: Speeded Up Robust Features*. In European Conference on Computer Vision (ECCV), 2006.

[Behmo 2008a] Régis Behmo, Nikos Paragios and Véronique Prinet. *An Application of Graph Commute Times to Image Indexing*. In IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2008.

[Behmo 2008b] Régis Behmo, Nikos Paragios and Véronique Prinet. *Graph Commute Times for Image Representation*. In Conference on Computer Vision and Pattern Recognition (CVPR), 2008.

[Behmo 2010] Régis Behmo, Nikos Paragios and Véronique Prinet. *Towards Optimal Naive Bayes Nearest Neighbors*. In European Conference on Computer Vision (ECCV), 2010.

[Belkin 2001] Mikhail Belkin and Partha Niyogi. *Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering*. In Advances in Neural Information Processing Systems (NIPS), volume 14, pages 585–591, 2001.

[Bellman 1961] R. E. Bellman. Princeton University Press, 1961.

[Belongie 2002] S. Belongie, J. Malik and J. Puzicha. *Shape matching and object recognition using shape contexts*. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 24, no. 4, pages 509–522, April 2002.

[Bengoetxea 2002] E. Bengoetxea. *Inexact Graph Matching Using Estimation of Distribution Algorithms*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, Paris, France, Dec 2002.

[Biederman 1987] I. Biederman. *Recognition-by-components: a theory of human image understanding*. Psychol Review, vol. 94, no. 2, pages 115–147, April 1987.

[Boiman 2008] Oren Boiman, Eli Shechtman and Michal Irani. *In Defense of Nearest-Neighbor Based Image Classification*. In Conference on Computer Vision and Pattern Recognition (CVPR), 2008.

[Bordes 2008] Jean-Baptiste Bordes and Véronique Prinet. *Mixture Distributions for Weakly Supervised Classification in Remote Sensing Images*. In British Machine Vision Conference (BMVC), 2008.

[Bosch 2006] A. Bosch, A. Zisserman and X. Munoz. *Scene Classification via pLSA*. In Proceedings of the European Conference on Computer Vision, 2006.

[Bosch 2007] Anna Bosch, Andrew Zisserman and Xavier Munoz. *Representing shape with a spatial pyramid kernel*. In International Conference on Image and Video Retrieval (ICIVR), 2007.

[Boser 1992] Bernhard E. Boser, Isabelle M. Guyon and Vladimir N. Vapnik. *A training algorithm for optimal margin classifiers*. In Proceedings of the fifth annual workshop on Computational learning theory, pages 144–152. ACM, 1992.

[Boughorbel 2004] S. Boughorbel, J.-P. Tarel and F. Fleuret. *Non-Mercer Kernels for SVM Object Recognition*. In British Machine Vision Conference (BMVC), 2004.

[Boughorbel 2005] S. Boughorbel, J. P. Tarel and N. Boujemaa. *The intermediate matching kernel for image local features*. In IEEE International Joint Conference on Neural Networks (IJCNN), volume 2, 2005.

[Boyer 1988] K.L. Boyer and A.C. Kak. *Structural Stereopsis for 3-D Vision*. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 10, pages 144–166, 1988.

[Burl 1996] M. C. Burl and P. Perona. *Recognition of planar object classes*. In Conference on Computer Vision and Pattern Recognition (CVPR), 1996.

[Caputo 2002] B. Caputo and G. Dorko. *How to combine color and shape information for 3d object recognition: kernels do the trick*. Advances in Neural Information Processing Systems (NIPS), 2002.

[Champ 2009] Heather Champ. *Flickr Blog - 4000000000*. http://blog.flickr.net/en/2009/10/12/4000000000/, 2009.

[Chung 1997] Fan R. K. Chung. Spectral graph theory. American Mathematical Society, 1997.

[Chung 2000] F. Chung and S. Yau. *Discrete Green's function*. Journal of Combinatorial Theory, vol. A, pages 141–214, 2000.

[Conte 2004] D. Conte, P. Foggia, C. Sansone and M. Vento. *Thirty Years Of Graph Matching In Pattern Recognition*. International Journal of Pattern Recognition and Artificial Intelligence, 2004.

[Cover 1967] T. Cover and P. Hart. *Nearest neighbor pattern classification*. IEEE Transactions onInformation Theory, vol. 13, no. 1, pages 21–27, 1967.

[Cula 2001a] Oana G. Cula and Kristin J. Dana. *Compact Representation of Bidirectional Texture Functions*. Conference on Computer Vision and Pattern Recognition (CVPR), 2001.

[Cula 2001b] Oana G. Cula and Kristin J. Dana. *Recognition methods for 3d textured surfaces*. In Proceedings of SPIE Conference on Human Vision and Electronic Imaging VI, pages 209–220, 2001.

[Dalal 2005] N. Dalal and B. Triggs. *Histograms of Oriented Gradients for Human Detection*. In Conference on Computer Vision and Pattern Recognition (CVPR), 2005.

[Dance 2004] Chris Dance, Jutta Willamowski, Lixin Fan, Cedric Bray and Gabriela Csurka. *Visual categorization with bags of keypoints*. In International Workshop on Statistical Learning in Computer Vision, European Conference on Computer Vision (ECCV), 2004.

[Dean 2008] Jeffrey Dean and Sanjay Ghemawat. *MapReduce: simplified data processing on large clusters*. Communications of the ACM, vol. 51, no. 1, pages 107–113, 2008.

[Deng 2009] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei. *ImageNet: A Large-Scale Hierarchical Image Database*. In Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[Domeniconi 2005] Carlotta Domeniconi, Dimitrios Gunopulos and Jing Peng. *Large margin nearest neighbor classifiers*. IEEE transactions on neural networks, vol. 16, no. 4, pages 899–909, July 2005.

[Dong 2008] Wei Dong, Zhe Wang, William Josephson, Moses Charikar and Kai Li. *Modeling LSH for performance tuning*. In ACM Conference on Information and Knowledge Management (CIKM), pages 669–678. ACM, 2008.

[Dubuisson 1994] M. P. Dubuisson and A. K. Jain. *A modified Hausdorff distance for object matching*. 1994.

[Everingham 2006] M. Everingham, A. Zisserman, C. K. I. Williams and L. Van Gool. *The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results*. http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf, 2006.

[Everingham 2007] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman. *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results*. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html, 2007.

[Everingham 2008] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman. *The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results*. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html, 2008.

[Everingham 2009] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman. *The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results*. http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html, 2009.

[Fac 2009] *Facebook statistics*. http://www.facebook.com/press/info.php?statistics, 2009.

[Farmer 1999] S.J. Farmer. *Probabilistic Graph Matching*. Unpublished manuscript, University of York (United Kingdom), 1999.

[Fei-Fei 2005] L. Fei-Fei and P. Perona. *A Bayesian Hierarchical Model for Learning Natural Scene Categories*. Conference on Computer Vision and Pattern Recognition (CVPR), 2005.

[Fei-Fei 2006] Li Fei-Fei, R. Fergus and P. Perona. *One-Shot Learning of Object Categories*. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 28, no. 4, pages 594–611, 2006.

[Felzenszwalb 2003] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. *Pictorial Structures for Object Recognition*. International Journal of Computer Vision (IJCV), vol. 61, page 2005, 2003.

[Felzenszwalb 2008] P. Felzenszwalb, D. McAllester and D. Ramanan. *A Discriminatively Trained, Multiscale, Deformable Part Model*. In Conference on Computer Vision and Pattern Recognition (CVPR), 2008.

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

[Field 2006] Matthew Field. *http://www.mattfield.com*, 2006.

[Fischler 1973] M. A. Fischler and R. A. Elschlager. *The Representation and Matching of Pictorial Structures*. IEEE Transactions on Computers, vol. C-22, no. 1, pages 67–92, 1973.

[Freeman 1991] W. T. Freeman and E. H. Adelson. *The Design and Use of Steerable Filters*. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 13, no. 9, pages 891–906, 1991.

[Freeman 1998] W. T. Freeman and P. Perona. *A factorization approach to grouping*. In European Conference on Computer Vision (ECCV), pages 655–670, 1998.

[Fulkerson 2008] B. Fulkerson, A. Vedaldi and S. Soatto. *Localizing Objects With Smart Dictionaries*. In European Conference on Computer Vision (ECCV), 2008.

[Gorkani 1994] Monika M. Gorkani and Rosalind W. Picard. *Texture Orientation for Sorting Photos at a Glance*. In International Conference on Pattern Recognition (ICPR), 1994.

[Grauman 2005] K. Grauman and T. Darrell. *The pyramid match kernel: discriminative classification with sets of image features*. In Conference on Computer Vision and Pattern Recognition (CVPR), 2005.

[Harris 1988] Chris Harris and Mike Stephens. *A Combined Corner and Edge Detector*. In Proceedings of the 4th Alvey Vision Conference, pages 147–151, 1988.

[Hastie 1994] Trevor Hastie and Robert Tibshirani. *Discriminant Adaptive Nearest Neighbor Classification*, 1994.

[Huang 1997] Jing Huang, S. Ravi Kumar, Mandar Mitra, Wei-jing Zhu and Ramin Zabih. *Image Indexing Using Color Correlograms*. IJCV, 1997.

[Huttenlocher 1993] Daniel P. Huttenlocher, Gregory A. Klanderman, Gregory A. Kl and William J. Rucklidge. *Comparing Images Using the*

Hausdorff Distance. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 15, pages 850–863, 1993.

[Joachims 1998] Thorsten Joachims. *Text categorization with support vector machines: learning with many relevant features*. In European Conference on Machine Learning, 1998.

[Khoo 2002] K. G. Khoo and P. N. Suganthan. *Evaluation of genetic operators and solution representations for shape recognition by genetic algorithms*. Pattern Recognition Letters, vol. 23, no. 13, pages 1589–1597, November 2002.

[Koenderink 1984] Jan J. Koenderink. *The structure of images*. Biological Cybernetics, vol. 50, pages 363–370, 1984.

[Kondor 2003] Risi Imre Kondor and Tony Jebara. *A Kernel Between Sets of Vectors*. In ICML, pages 361–368, 2003.

[Kruskal 1978] J. Kruskal and M. Wish. Multidimensional scaling. Sage Publications, 1978.

[Lampert 2008] C. H. Lampert, M. B. Blaschko and T. Hofmann. *Beyond sliding windows: Object localization by efficient subwindow search*. In Conference on Computer Vision and Pattern Recognition (CVPR), 2008.

[Lazebnik 2005a] S. Lazebnik, C. Schmid and J. Ponce. *A Maximum Entropy Framework for Part-Based Texture and Object Recognition*. International Conference on Computer Vision (ICCV), 2005.

[Lazebnik 2005b] Svetlana Lazebnik, Cordelia Schmid and Jean Ponce. *A sparse texture representation using local affine regions*. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2005.

[Lazebnik 2006] S. Lazebnik, C. Schmid and J. Ponce. *Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories*. In Conference on Computer Vision and Pattern Recognition (CVPR), 2006.

[Lazebnik 2007] Svetlana Lazebnik and Maxim Raginsky. *Learning Nearest-Neighbor Quantizers from Labeled Data by Information Loss Minimization*. In International Conference on Artificial Intelligence and Statistics, 2007.

[Leibe 2004] B. Leibe, A. Leonardis and B. Schiele. *Combined object categorization and segmentation with an implicit shape model*. In Workshop on Statistical Learning in Computer Vision, European Conference on Computer Vision (ECCV), 2004.

[Leung 2001] Thomas Leung and Jitendra Malik. *Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons*. International Journal of Computer Vision (IJCV), vol. 43, no. 1, pages 29–44, June 2001.

[Li 2008] J Li and N Allinson. *A comprehensive review of current local features for computer vision*. Neurocomputing, vol. 71, no. 10-12, pages 1771–1787, 2008.

[Lindeberg 1998] Tony Lindeberg. *Feature detection with automatic scale selection*. International Journal of Computer Vision (IJCV), vol. 30, pages 79–116, 1998.

[Ling 2007] Haibin Ling. *Proximity Distribution Kernels for Geometric Context in Category Recognition*. In International Conference on Computer Vision (ICCV), 2007.

[Lovász 1993] L Lovász. *Random walks on graphs: a survey*. In Combinatorics, Paul Erdös is eighty, pages 353–397, 1993.

[Lowe 2003] D. Lowe. *Distinctive image features from scale-invariant keypoints*. In International Journal of Computer Vision (IJCV), 2003.

[Luo 2001] B. Luo and E.R. Hancock. *Structural Graph Matching Using the EM Algorithm and Singular Value Decomposition*. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 23, pages 1120–1136, 2001.

[Luxburg 2007] Ulrike Luxburg. *A tutorial on spectral clustering*. Technical Report, 2007.

[Lyu 2005] Siwei Lyu. *Mercer Kernels for Object Recognition with Local Features*. In Conference on Computer Vision and Pattern Recognition (CVPR), 2005.

[Marszalek 2006] Marcin Marszalek and Cordelia Schmid. *Spatial Weighting for Bag-of-Features*. In Conference on Computer Vision and Pattern Recognition (CVPR), 2006.

[Marszałek 2007a] Marcin Marszałek and Cordelia Schmid. *Accurate Object Localization with Shape Masks*. In Conference on Computer Vision and Pattern Recognition (CVPR), 2007.

[Marszałek 2007b] Marcin Marszałek, Cordelia Schmid, Hedi Harzallah and Joost van de Weijer. *Learning Object Representations for Visual Object Class Recognition*, oct 2007. Visual Recognition Challange workshop, in conjunction with ICCV.

[Marčelja 1980] S. Marčelja. *Mathematical description of the responses of simple cortical cells.* Journal of the Optical Society of America, vol. 70, no. 11, pages 1297–1300, 1980.

[Meila 2000] Marina Meila and Jianbo Shi. *A random walks view on spectral segmentation.* In Advances in Neural Information Processing Systems (NIPS), pages 873–879, 2000.

[Meyer 2007] F. Meyer. *Learning and predicting brain dynamics from fMRI: a spectral approach.* In SPIE, 2007.

[Mikolajczyk 2005a] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir and L. Van Gool. *A comparison of affine region detectors.* International Journal of Computer Vision (IJCV), 2005.

[Mikolajczyk 2005b] Krystian Mikolajczyk and Cordelia Schmid. *A performance evaluation of local descriptors.* IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 27, no. 10, pages 1615–1630, 2005.

[Moosmann 2007] Frank Moosmann, Bill Triggs and Frederic Jurie. *Fast discriminative visual codebooks using randomized clustering forests.* In Advances in Neural Information Processing Systems (NIPS), 2007.

[Moravec 1980] Hans Moravec. *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover.* In tech. report CMU-RI-TR-80-03, Robotics Institute, Carnegie Mellon University & doctoral dissertation, Stanford University. September 1980.

[Mutch 2008] Jim Mutch and David G. Lowe. *Object class recognition and localization using sparse features with limited receptive fields.* International Journal of Computer Vision (IJCV), vol. 80, no. 1, pages 45–57, October 2008.

[Myers 2000] Richard Myers, Richard C. Wilson and Edwin R. Hancock. *Bayesian graph edit distance.* IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 22, pages 628–635, 2000.

[Ng 2001] Andrew Y. Ng, Michael I. Jordan and Yair Weiss. *On Spectral Clustering: Analysis and an algorithm.* In Advances in Neural Information Processing Systems (NIPS), pages 849–856. MIT Press, 2001.

[Odone 2001] Francesca Odone, Emanuele Trucco, Alessandro Verri and Ro Verri. *General Purpose Matching of Grey Level Arbitrary Images.* In 4th International Workshop on Visual Forms, Lecture Notes on Computer Science LNCS 2059, pages 573–582. Springer, 2001.

[Oliva 2001] A. Oliva and A. Torralba. *Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope*. International Journal of Computer Vision (IJCV), vol. 42, no. 3, pages 145–175, 2001.

[Oliva 2006] Aude Oliva and Antonio Torralba. *Building the Gist of a Scene: The Role of Global Image Features in Recognition*. In Visual Perception, Progress in Brain Research, volume 155, pages 23–36, 2006.

[Opelt 2004] Andreas Opelt, Axel Pinz, Michael Fussenegger and Peter Auer. *Generic object recognition with boosting*. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 28, 2004.

[Ovsjanikov 2009] M. Ovsjanikov, A. M. Bronstein, M. M. Bronstein and L. J. Guibas. *ShapeGoogle: a computer vision approach for invariant shape retrieval*. In Proc. Workshop on Nonrigid Shape Analysis and Deformable Image Alignment (NORDIA), 2009.

[Puzicha 1998] Jan Puzicha, Thomas Hofmann and Joachim M. Buhmann. *Histogram Clustering for Unsupervised Segmentation and Image Retrieval*. Pattern Recognition Letters, 1998.

[Qiu 2007] H. Qiu and E. R. Hancock. *Clustering and Embedding Using Commute Times*. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2007.

[Rubner 2000] Yossi Rubner, Carlo Tomasi and Leonidas J. Guibas. *The Earth Mover's Distance as a Metric for Image Retrieval*. International Journal of Computer Vision (IJCV), vol. 40, no. 2, pages 99–121, November 2000.

[Sanfeliu 1983] Alberto Sanfeliu and King-Sun Fu. *A Distance measure between attributed relational graphs for pattern recognition*. IEEE transactions on systems, man, and cybernetics, vol. 13, no. 3, pages 353–362, 1983.

[Sarkar 1998] Sudeep Sarkar and Kim L. Boyer. *Quantitative measures of change based on feature organization: eigenvalues and eigenvectors*. Computer Vision and Image Understanding, vol. 71, no. 1, pages 110–136, 1998.

[Savarese 2006] S Savarese, J Winn and A Criminisi. *Discriminative Object Class Models of Appearance and Shape by Correlatons*. In Conference on Computer Vision and Pattern Recognition (CVPR), 2006.

[Schmid 2001] Cordelia Schmid. *Constructing Models for Content-Based Image Retrieval*. In Conference on Computer Vision and Pattern Recognition (CVPR), 2001.

[Schmid 2004] Cordelia Schmid. *Weakly Supervised Learning of Visual Models and its Application to content-based retrieval.* IJCV, 2004.

[Schneiderman 2000] Henry Schneiderman and Takeo Kanade. *A Statistical Method for 3D Object Detection Applied to Faces and Cars.* In Conference on Computer Vision and Pattern Recognition (CVPR), 2000.

[Scott 1990] G. Scott and H. Longuet-Higgins. *Feature grouping by relocalisation of eigenvectors of the proximity matrix.* In British Machine Vision Conference (BMVC), 1990.

[Shapiro] L.G. Shapiro and R.M. Haralick. *A Metric for Comparing Relational Descriptions.* IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI).

[Shi 2000] Jianbo Shi and Jitendra Malik. *Normalized Cuts and Image Segmentation.* IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2000.

[Sivic 2005] Josef Sivic, Bryan C. Russell, Alexei A. Efros, Andrew Zisserman and William T. Freeman. *Discovering objects and their location in images.* In International Conference on Computer Vision (ICCV), 2005.

[Stone 1983] Charles Stone. *Optimal uniform rate of convergence for nonparametric estimators of a density function or its derivatives.* Recent advances in statistics, 1983.

[Szummer 1998] Martin Szummer and Rosalind W. Picard. *Indoor-Outdoor Image Classification.* In IEEE International Workshop on Content-based Access of Image and Video Databases, pages 42–51, 1998.

[Tenenbaum 2000] Joshua B. Tenenbaum, Vin Silva and John C. Langford. *A Global Geometric Framework for Nonlinear Dimensionality Reduction.* Science, vol. 290, no. 5500, pages 2319–2323, December 2000.

[Tsai 1979] W. H. Tsai and K. S. Fu. *Error-correcting isomorphisms of attributed relational graphs for pattern analysis.* vol. 9, page 757–768, 1979.

[Unsalan 2005] C. Unsalan and K. L. Boyer. *A Theoretical and Experimental Investigation of Graph Theoretical Measures for Land Development in Satellite Imagery.* IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2005.

[van de Sande 2010] K. E. A. van de Sande, T. Gevers and C. G. M. Snoek. *Evaluating Color Descriptors for Object and Scene Recognition*. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2010.

[van Gemert 2008] Jan van Gemert, Jan M. Geusebroek, Cor J. Veenman and Arnold W. M. Smeulders. *Kernel Codebooks for Scene Categorization*. In European Conference on Computer Vision (ECCV), 2008.

[Varma 2002] M. Varma and A. Zisserman. *Classifying materials from images: to cluster or not to cluster?* In Proceedings of the 2nd International Workshop on Texture Analysis and Synthesis, Copenhagen, Denmark, pages 139–144, June 2002.

[Wallraven 2003] Christian Wallraven, Barbara Caputo and Arnulf B. A. Graf. *Recognition with Local Features: the Kernel Recipe*. In International Conference on Computer Vision (ICCV), pages 257–264. IEEE Computer Society, 2003.

[Weber 2000] Markus Weber, Max Welling and Pietro Perona. *Unsupervised Learning of Models for Recognition*. In European Conference on Computer Vision (ECCV), pages 18–32, 2000.

[Witkin 1983] Andrew P. Witkin. *Scale Space Filtering*. In Proceedings of the 8th International Conference on Artificial Intelligence, pages 1019–1022, 1983.

[Wolf 2003] Lior Wolf and Amnon Shashua. *Kernel Principal Angles for Classification Machines with Applications to Image Sequence Interpretation*. In Conference on Computer Vision and Pattern Recognition (CVPR), pages 635–642, 2003.

[Yang 2007] Lin Yang and David J Foran. *Multiple Class Segmentation Using A Unified Framework over Mean-Shift Patches*. In Conference on Computer Vision and Pattern Recognition (CVPR), 2007.

[Yedidia 2003] J. S. Yedidia, W. T. Freeman and Y. Weiss. *Understanding belief propagation and its generalizations*. Technical Report, 2003.

[Yianilos 1993] Peter N. Yianilos. *Data Structures and Algorithms for Nearest Neighbor Search in General Metric Spaces*. In SODA: ACM-SIAM Symposium on Discrete Algorithms (A Conference on Theoretical and Experimental Analysis of Discrete Algorithms), 1993.

[You 2009] *YouTube fact sheet*. http://www.youtube.com/t/fact_sheet, 2009.

[Yuan 2009] Junsong Yuan, Zicheng Liu and Ying Wu. *Discriminative subvolume search for efficient action detection.* In Conference on Computer Vision and Pattern Recognition (CVPR), June 2009.

[Zhang 2006] H Zhang, A Berg, M Maire and J Malik. *Svm-KNN: Discriminative nearest neighbor classification for visual category recognition.* Conference on Computer Vision and Pattern Recognition (CVPR), 2006.

[Zhang 2007] J. Zhang, M. Marszalek, S. Lazebnik and C. Schmid. *Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study.* In International Journal of Computer Vision (IJCV), 2007.