Modélisation et recherche de graphes visuels : une approche par modèles de langue pour la reconnaissance de scènes

Trong-Ton Pham

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

N^o attribué par la bibliothèque
|_|_|_|_|_|_|_|_|_|_|

École Doctorale
Mathématiques, Sciences et Technologies de l'Information, Informatique

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE DE GRENOBLE

Spécialité : **Informatique**

préparée au LABORATOIRE INFORMATIQUE DE GRENOBLE

présentée et soutenue publiquement par

PHAM TRONG-TON

02 December 2010

Visual Graph Modeling and Retrieval: A Language Model Approach for Scene Recognition

---

Co-directeurs de thèse :

Dr. Philippe MULHEM et Dr. Joo-Hwee LIM

Composition du jury :

M.	Augustin LUX	Président
M.	Mohand BOUGHANEM	Rapporteur
M.	Salvatore-Atoine TABBONE	Rapporteur
M.	Florent PERRONNIN	Examinateur
M.	Philippe MULHEM	Directeur de thèse
M.	Joo-Hwee LIM	Co-directeur de thèse

This thesis is dedicated to my beloved parents, Pham Trong-Kiem and Nguyen Thi-Hien.

Special thanks to my co-supervisors Philippe Mulhem and Lim Joo-Hwee, Prof. Eric Gaussier, Loic Maisonnasse, Nicolas Maillot, Andy Tseng, Rami Albatal, my family and friends for their helps and encouragements.

Grenoble, December 2010.

Abstract

Content-based image indexing and retrieval (CBIR) system needs to consider several types of visual features and spatial information among them (i.e., different *point of views*) for better image representation. This thesis presents a novel approach that exploits an extension of the language modeling approach from information retrieval to the problem of graph-based image retrieval. Such versatile graph model is needed to represent the multiple points of views of images. This graph-based framework is composed of three main stages:

Image processing stage aims at extracting image regions from the image. It also consists of computing the numerical feature vectors associated with image regions.

Graph modeling stage consists of two main steps. First, extracted image regions that are visually similar will be grouped into clusters using an unsupervised learning algorithm. Each cluster is then associated with a visual concept. The second step generates the spatial relations between the visual concepts. Each image is represented by a visual graph captured from a set of visual concepts and a set of spatial relations among them.

Graph retrieval stage is to retrieve images relevant to a new image query. Query graphs are generated following the graph modeling stage. Inspired by the language model for text retrieval, we extend this framework for matching the query graph with the document graphs from the database. Images are then ranked based on the relevance values of the corresponding image graphs.

Two instances of the visual graph model have been applied to the problem of *scene recognition* and *robot localization*. We performed the experiments on two image collections: one contained 3,849 touristic images and another composed of 3,633 images captured by a mobile robot. The achieved results show that using visual graph model outperforms the standard language model and the Support Vector Machine method by more than 10% in accuracy.

Keywords: Graph Theory, Image Representation, Information Retrieval, Language Modeling, Scene Recognition, Robot Localization.

Table of Contents

Chapter 1

Introduction

Napoleon once declared that he preferred a drawing to a long report. Today, I am certain he would say that he would prefer a photograph.
Brassaï

As an old saying goes, *"A picture is worth a thousand words"*, pictorial information is a crucial information complementary to the textual information. Brassaï, a photo-journalist, had captured the same importance of the visual information for his interview for *Camera* magazine in 1974. Indeed, human tends to prefer using visual information to express their ideas and their communication needs.

In recent years, the number of image acquired is growing rapidly, thanks to the invention of digital cameras and the creation of photo sharing sites such as Flickr[1], Picasa[2], Photobucket[3], etc. Digital cameras are becoming cheaper and more friendly to the amateurs. This fact has encouraged the users to explore the image world and generate more and more visual contents. Reported by Media Culpa[4] that Flickr, one of the best social photo sharing sites, has reached the milestone of *5 billions* photos uploaded to their website in September 2010. The increase in terms of the number of photos uploaded is very steep over the years. Other social networking sites, such as Facebook[5], has also claimed to have *2.5 billion* photos uploaded per month in February 2010.

As consequence, a user will need an effective system for organizing their photos, searching for a particular photo or automatically tagging their photos with

[1] http://www.flickr.com
[2] http://www.picasa.com
[3] http://www.photobucket.com
[4] http://www.kullin.net/2010/09/flickr-5-billion-photos/
[5] http://www.facebook.com

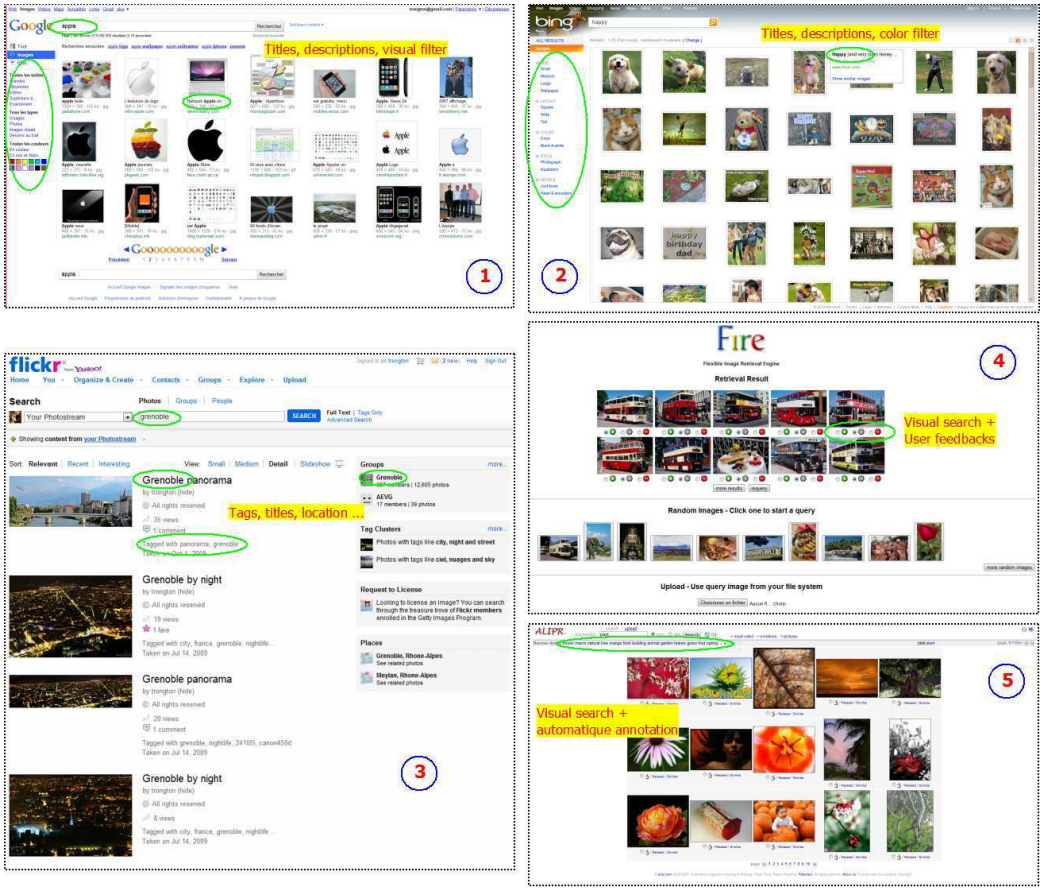

Figure 1.1: Example of the current state-of-the-art systems in image search. (1) Google Images, (2) Bing Image by Microsoft, (3) Flikr photo by Yahoo, (4) FIRE visual search engine and (5) ALIPR photo tagging and image search

some keywords. This raises an important challenge for research and industry. Eventually, Annotation-Based Image Retrieval (ABIR) is widely used in the real-world image search thanks to the success of the web search engine such as Google or Bing. Figure 1.1 shows some current state-of-the-art engines used for image search. Some of the current ABIR systems are:

- **Google Images Search**[6]: As of today, Google has indexed more than 10 billion[7] of images on the Web. The success of the web search engine led Google to create an image search engine. However their search engine is still heavily based on textual metadata related to the image such as image

[6]http://www.google.com
[7]http://googleblog.blogspot.com/2010/07/ooh-ahh-google-images-presents-nicer.html

title, description, or links. Recently, Google has added some new search features with the image option panel. They implemented some simple image filters based on the *color information* (full color vs. black & white) and *picture type* (photos, drawing, etc.) and *face detection* engine.

- **Bing Images Search**[8]: Similar to Google's engine, the Microsoft search engine mainly uses the textual information to index their photos. Images results can be narrowed down by some options such as *image size* (big, medium, small), *image layout* (square, rectangle), and the integrating of *face detection technology*.

- **Flickr photo**: In order to deal with a large amount of photos uploaded to their website, Flickr allows users to add *tags* to their photos or to organize them into *groups and sets* of photos or to localize using the *geographical information* (i.e., GPS coordination). However, the provided textual information is subjective. Hence, the search results rarely satisfied user's needs.

Another type of image search is based principally on the analysis of the visual image content. These systems are known as Content Based Image Retrieval (CBIR) engines. However, we observe that there are only few CBIR systems that have been implemented in the real-world context. Most of the systems are for experimental research purposes. Some of these systems are:

- **Flexible Image Retrieval Engine (FIRE)**[9]: This is one of the visual search engines that used several image features such as color, texture and shape information for similar image searching. Moreover, the system allows user to fine-tune their queries by using a relevant feedback mechanism (i.e., scoring the search result with positive or negative indication). This system produces encouraging results, although it is far from perfect.

- **Automatic Photo Tagging and Visual Image Search (ALIPR)**[10]: This is the first automatic image tagging engine developed by researchers at the Penn State University. This engine will automatically analyze and associate with some keywords to the photos (such as a "person" or "car" or a more general "outdoors" or "manmade") according to their visual content. In return, these keywords are used to index the photos for searching later. The researchers claimed that the system achieved a high accuracy (approximately 98% of all photo analyzed). However, ALIPR system tends to assign more general and higher frequency terms.

[8] http://www.bing.com/images
[9] http://www-i6.informatik.rwth-aachen.de/deselaers/cgi_bin/fire.cgi
[10] http://www.alipr.com

Even though the text retrieval field has received an enormously success, image indexing and retrieval is still a very challenging problem and requires a lot of research efforts. The need for a reliable image retrieval system is the research trends for the decade. In the scope of this dissertation, we intend to add some new perspectives to this challenging puzzle.

1.1 Motivations

CBIR is an active research domain for more than 20 years. CBIR systems are complex retrieval platforms which combine multiple areas of expertises from computer vision and machine learning to information retrieval (Figure 1.2). Achievements have been made to contribute to the advancement of the domain. However, a good CBIR system is still far from a reality.

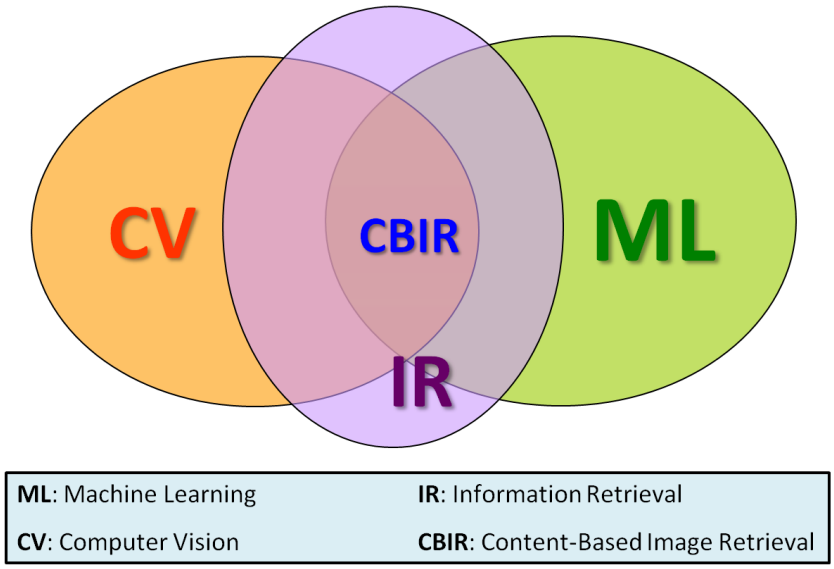

Figure 1.2: Content Based Image Retrieval (CBIR) in the intersection of different research fields.

On the other hand, still image representations for computer are about combining multiple points of views. A broader perspective for multimedia document indexing and retrieval is given by R. Datta, D. Joshi, J. Li, and J. Z. Wang in [Datta *et al.* 2008]: *"The future lies in harnessing as many channels of information as possible, and fusing them in smart, practical ways to solve real problems. Principled approaches to fusion, particularly probabilistic ones, can also help provide performance guarantees which in turn convert to quality standards for public-domain systems"*

This reflexion also holds in the specific context of image indexing and retrieval. The points of views of images rely on different regions extracted, different features generated and different ways to integrate these aspects in order to annotate or retrieve images based on their visual similarity.

Let us present a short overview of the diversity of approaches encountered in the image indexing and retrieval domain. Image indexing and retrieval may use predefined segmentation in blocks [Chua *et al.* 1997], or try to consider segmentation techniques based on color/texture [Felzenszwalb & Huttenlocher 2004] or point of interest like the well-known work of D. Lowe [Lowe 2004]. The feature considered are mostly represented using histograms of features (colors, textures or shapes) or of *bag-of-word* (BoW) [Sivic & Zisserman 2003] or of latent semantic analysis (LSA) [Pham *et al.* 2007]. Other approach may consider spatial relationships between regions [Smith & Chang 1996]. When considering more complex representations, other approach may use *conceptual graph* representations [Ounis & Pasca 1998].

A short survey on the state-of-the-art leads us to several thinking:

- **Integration of spatial relation**. Most of current image representation is based on the flat and numerical vector presentation of BoW model. The information on the spatial relations between visual elements is not well considered. Therefore, we believe that a complete image representation of image contents should include in the right way this important information together with the visual features.

- **Graph-based image representation**. While studying the graph theory, we think that it should be appropriate to use this type of representation to combine the visual contents and the relations among them. Graph has been used as a general framework for structural infomation representation [Sowa 1984, Ballard & Brown 1982]. Considering image content as a special source of information (i.e., visual features, spatial relations), graph is a well-suited representation for image contents.

- **Bridging the semantic gap**. An important underlying issue that we would like to address is to reduce the "*semantic gap*" between high-level of knowledge representation (e.g., text description, conceptual information) and the middle-level of image representation (e.g., BoW model, visual concept detection). Indeed, the graph-based image representation will add an intermediate layer to fill this gap.

- **Graph matching with probabilistic framework**. Classical graph matching algorithm is a main bottleneck for the graph-based knowledge representation. However, probabilistic approaches (such as Bayesian methods,

Probabilistic Latent Semantic Analysis (pLSA), Language Modeling, etc.) have been developed widely in the information retrieval field for the decades. We think that it should be interesting and important to express the graph matching process with the probabilistic matching framework.

Therefore, the objective of this thesis aims to answer a part (if not all) of the above mentioned questions.

1.2 Problem statements

In this dissertation, we address two specific problems, namely a *graph-based image representation* and a *generative graph matching method*.

1. First, we focus on a representation of image content, more precisely graph-based representation, which is able to represent different points of views (namely several visual representations and spatial relationships between regions). Despite the fact that selecting relevant regions and extracting good features are very difficult tasks, we believe that the way we represent different points of views of the image (like several segmentations and/or several features for instance) will also have a great impact on image annotation and image retrieval.

   Considering a graph that represents the visual features which are intended to preserve the diversity of content when needed. In fact, such graphs are versatile, because they can handle early fusion-like approaches when considering several representations in an integrated matching process as well as late fusion-like approaches when considering matching on specific sub-graphs before fusion.

2. Second, we define a language model on such graphs that tackles the problem of retrieval and classification of images. The interest of considering language models for such graphs lies in the fact that it benefits from this successful research field of information retrieval since the end of the 90s and in particular the seminal work of Ponte and Croft in [Ponte & Croft 1998]. Such language models are well-defined theoretically, and also have shown interesting experimental results, as synthesized in [Manning *et al.* 2009]. Therefore, our main focus is to propose an extension of language models in the context of graph-based representation for image content.

On the practical side, we will apply the above graph model in two applications (Figure 1.3): *scene recognition system* and *robot self-localizing system*:

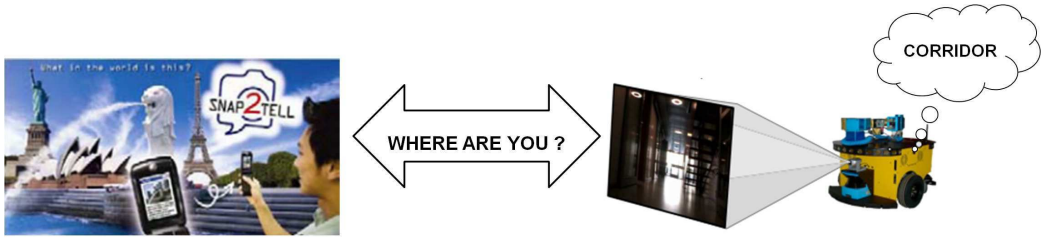

Figure 1.3: Two applications of this thesis: (a) a scene identification system for mobile phone and (b) a self-localization system for mobile robot.

1. The first application is a *scene recognition system* for mobile phone service, for instance the Snap2Tell system developed by the IPAL lab[11]. This system enables user to take a picture of a monument with their camera phone, send it to Snap2Tell's server and to receive in return touristic information about the monument. To do so, a set of images taken from 101 Singapore landscapes has been collected and used for experimental purposes. The main task of the recognition system is to match a query image to one of the 101 different scenes (or 101 classes).

2. The second application is a *robot self-localizing system* using only visual information, known as the RobotVision[12] task in ImageCLEF international benchmark and competition. The robot has to determine in real-time its topological location based on the images acquired. The image acquisition was performed within an indoor laboratory environment consisting of five rooms of different functionality under various illumination conditions. The main task of the localization system is to identify the correct rooms of the robot in an *unknown* condition and with different time spans.

1.3 Main contributions

Coping with the specific problems as stated above, the contributions of this thesis are as follows:

- First, we present a **unified graph-based framework** for image representation which allows us to integrate different types of visual concepts and different spatial relations among them. This graph can be used for different image points of views in the very flexible way. Actually, this visual graph

[11] http://www.ipal.i2r.a-star.edu.sg
[12] http://www.imageclef.org/2009/robot

model is a higher layer of image representation that approaches the image semantics.

- Second, we extensively study the **extension of language model for graph matching** which allows a more reliable matching based on a well studied theory of information retrieval. The matching method allows matching a complex graph composed of multiple concept sets and multiple relations set. We also propose a smoothing method that adapts to the specific graph model.

- Finally, the experimental results, performed on STOIC-101 and RobotVision '09 image collections, confirm the **performance and the effectiveness of the proposed visual graph modeling**. The proposed method outperforms the standard language modeling and the state-of-the-art SVM methods in both cases.

The results of this work have been published in the Journal on Multimedia Tools and Applications (2011), the proceeding of IEEE International Workshop on Content Based Multimedia Indexing (CBMI 2010), the proceeding of ACM Conference on Research and Development in Information Retrieval (poster session of SIGIR 2010), the proceeding of Singaporean-French IPAL Symposium (SinFra 2009) and the proceeding of ACM Conference on Information and Knowledge Management (CIKM 2007).

Our participation in RobotVision track, part of ImageCLEF 2009 international evaluation, also led to good results. The technical methods have been reported in a working note for the ImageCLEF 2009 workshop and a book chapter in Lecture Notes for Computer Science (LNCS) published by Springer. A complete list of publications can be found in the Appendix A.

1.4 Thesis outline

We describe here the structure of this thesis. This thesis has six chapters:

Chapter 2 introduces the early works on image indexing and retrieval. We will give an overview of the image processing such as image decomposition (grid partition, region segmentation or local keypoints), visual feature extraction (color, edge histogram and local invariant features). A preliminary indexing models based on the Bag-of-Word (BoW) model is also introduced. We describe how the visual concepts are constructed from the low-level visual features and quantized with the vector model. How latent semantic technique was used successfully with the BoW model is also discussed. Our goal is to present in this chapter the basic

steps in representing image contents. Based on these elementary steps, we present in chapter 3 the different learning methods of visual concepts in the literature.

Chapter 3 concentrates on different machine learning techniques based on the numerical representation of an image. We review two main approaches in information retrieval: generative-based model and discriminative-based model. The generative models include two main methods: Naive Bayes and Probabilistic Latent Semantic Analysis (pLSA). The discriminative models include two main methods: k-NN classification and the famous Support Vector Machine (SVM). We also mention in this chapter how the structure been captured to represent image content with the graph-based model. One important model that our method relied on is Language Modeling (LM) method will be detailed in this chapter.

Chapter 4 gives an overview of our proposed approach. The proposed model includes 3 main stages:

- **Image processing stage** aims at extracting image regions and keypoints from the image. It also consists of computing the numerical feature vectors associated with image regions or keypoints.

- **Graph modeling stage** consists of grouping similar visual features into clusters using the unsupervised learning algorithm. The visual concepts are generated for each type of visual feature. Then, the spatial relations between the visual concepts are extracted. Finally, an image is represented by a visual graph composed of a set of visual concepts and a set of spatial relations.

- **Graph retrieval stage** is to retrieve the relevant graphs to a new image query. Inspired by the language model, we extend this framework for matching the query graph with the trained graph from the database. Images are then ranked based on their probability likelihoods.

Chapter 5 details the proposed visual graph model. We formalize the definition of visual graph model and give examples of two graph instance. The graph matching model takes the query graph model and the document graph model as input to rank the image based on their probability likelihood. The matching model is an extended version of the language modeling to graphs. We also explain how we transform the normal probability into the log-probability domain to compute the relevance status value of image.

Chapter 6 presents the first application using the proposed approach: *outdoor scene recognition system*. We will present the proposed visual graph models adapted for the STOIC collection. The experimental result will be studied with different impacts of the relation and of multiple image queries on the classification performance. We will describe different techniques for optimizing the smoothing

parameter with cross validation technique and optimization based on the test set. The implementation of the scene recognition system will also be detailed in this chapter.

Chapter 7 demonstrates the second application of the visual graph model, namely *mobile robot localization*. The proposed visual graph models adapted to this image collection will be presented. We will provide the experimental results with different impacts of the relation and of the room classification accuracies. We also give a comparison of the proposed model with the SVM method. Then, we will discuss on how validation set has been used to choose the appropriate features for representing the image contents. The post-processing step and the official results of the run submitted to the ImageCLEF will also be discussed.

Chapter 8 concludes this dissertation with the discussion on the contribution and also on the perspective of the future works.

Part I

State of The Art

Chapter 2

Image Indexing

To take photographs means to recognize - simultaneously and within a fraction of a second - both the fact itself and the rigorous organization of visually perceived forms that give it meaning.
Henri Cartier-Bresson.

2.1 Introduction

In [Marr 1982], Marr described the three layers of a classical paradigm in machine vision: the *processing layer* (1), the *mapping layer* (2), the *high-level interpretation layer* (3) (detailed in Figure 2.1). These three layers can be aligned to the three levels of image representation in CBIR, namely *feature layer* (low level), *conceptual layer* (middle level) and *semantics layer* (high level). The feature layer concerns how to extract good visual feature from the pictorial data of an image. This layer is close to the actual computer representation of image. The conceptual layer maps the low-level signal information to a higher visual perception form, called visual concept. A visual concept is represented for a set of homogenous group of visual features. The semantics layer represents image with the highest form of knowledge representation which is close to the human understanding, i.e., textual description or textual concept.

For this reason, the "*semantic gap*" is often referred to "*the lack of co-incidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation*" [Smeulders *et al.* 2000]. More precisely, it is the lack of knowledge representation between the low-level feature layer and the high-level semantics layer. Since this problem is still unsolved, our objective is to inject a new *intermediate-level* of image representation in between conceptual layer and semantics layer. We believe that will help to reduce this *gap*.

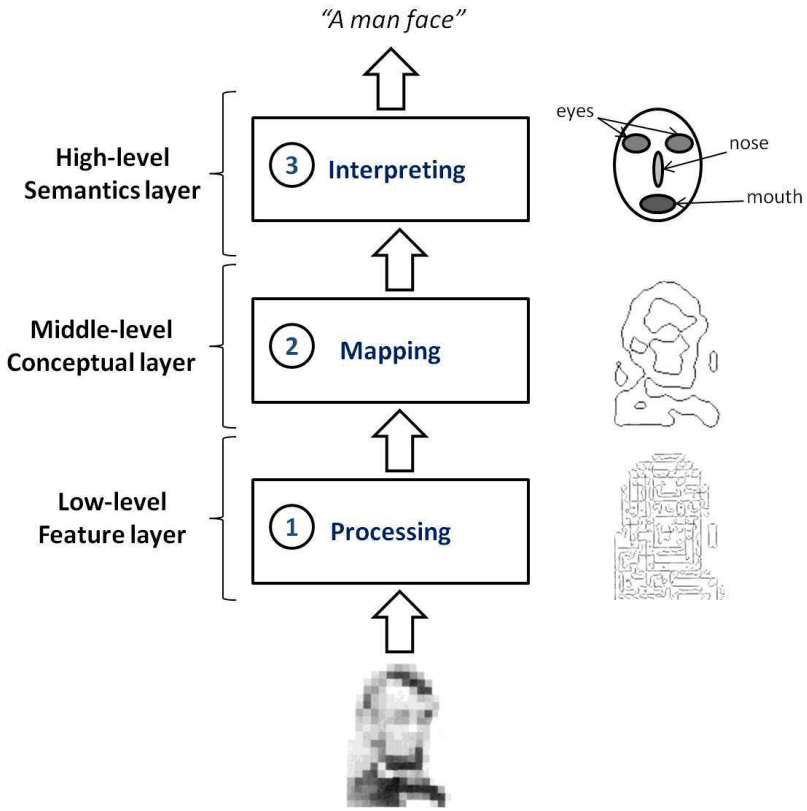

Figure 2.1: Illustration of Marr's paradigm [Marr 1982] for a vision system.

In this chapter, we will describe the works concerning mostly the first two layers (visual feature layer and conceptual layer) in a CBIR system. In the next section, we will present three different methods for region extraction: grid partitioning, region segmentation and interest point detection. Section 2.3 provides the information on the visual features extraction step. Section 2.4 gives more details on the indexing models, such as vector model, bag-of-words model and latent semantics indexing model, from the CBIR fields. Finally, section 2.5 will summarize this chapter.

2.2 Image representation

In CBIR, images are often divided into smaller parts to extract visual features from each part. The objective of image partitioning aims at obtaining more informative features by selecting a smaller subset of pixel to represent a whole image. Several image representations have been proposed. In this section, we

summarize some frequently used methods in CBIR such as uniform partitioning into regular grid, region segmentation or local region extraction.

2.2.1 Grid partitioning

This is a simple method for segmenting an image. A rectangular grid with fixed-size [Feng *et al.* 2004] slides over (can be overlap) the image (see Figure 2.2). For each rectangular grid, a feature vector is extracted. The rectangular size can be variable to make a multi-scale version [Lim & Jin 2005] of grid partitioning. Combining overlapping and multi-scale partitioning enables to cope with changes in object positions and image scale changes.

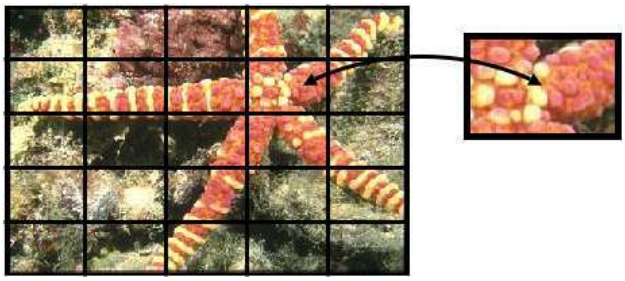

Figure 2.2: An image decomposed into 5x5 sub-images using regular grid

Using grid provides a number of advantages. The performance of rectangular grid as pointed out in [Feng *et al.* 2004] is better than the method based on region segmentation in annotation tasks. In addition, there is a significant reduction in the computational time required for segmenting the image. Grid partitioning (with more regions than produced by the segmentation algorithm) allows the model to learn how to associate visual features with images using a much larger set of training samples.

2.2.2 Region segmentation

Segmenting an image into regions may help to find out the relations between visual features and objects contained in the image. Image segmentation frees us from considering every pixel of the image but rather only groups of pixels that condense more information during subsequent processing. As defined in [Smeulders *et al.* 2000], there are two types of image segmentation:

- **Strong segmentation** is a division of the image data into regions in such a way that region T contains the pixels of the object O ($T = O$).

- **Weak segmentation** is a grouping of the image data in conspicuous regions T internally homogeneous according to some criterion, hopefully with T a subset of O ($T \subset O$).

These segmentation algorithms are based on some homogeneity criterion in each region such as color and texture. It is also difficult to obtain a strong segmentation so that each region contains an object. The weak segmentation helps to eliminate this problem and sometimes helps to identify better objects in image [Carson *et al.* 1999].

Many algorithms have been proposed for region segmentation. A graph-based algorithm has been used to find minimum normalized-cut (or N-cut) [Shi *et al.* 1998] in a pixel graph of image. A *Normalized-cut* algorithm gives bad results with cluttered background as they use only color as homogeneous criterion. The computational time of N-cut algorithm is also excessive due to the operation based on complex graph. The Blobworld system [Carson *et al.* 1999] used this algorithm to build image tokens (often called blobs).

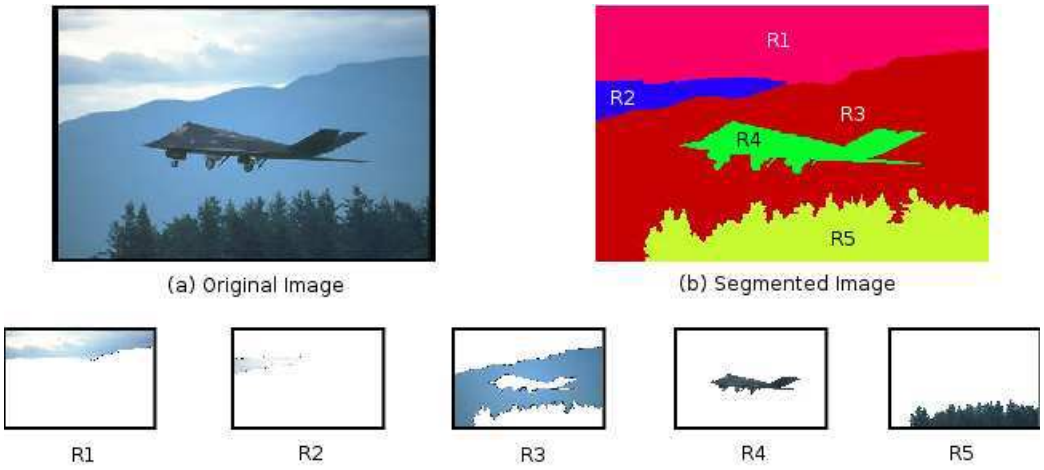

Figure 2.3: Example of image segmentation using the Mean-shift algorithm

Likewise, the *mean-shift segmentation* [Comaniciu & Meer 2002] algorithm searchs for a higher density of data distribution in images. The mean-shift segmentation algorithm is recognized as a very flexible algorithm (user can choose different parameters: window size, filter kernel, region threshold, etc...) and perhaps the best segmentation technique to date.

2.2.3 Interest point detection

Saliency-based models have been studied for image indexing and retrieval by [Schmid & Mohr 1997, Hare & Lewis 2005] for several years and later have been

experimented for the object recognition by [Lowe 1999]. The saliency regions are extracted from the interest points using local detector, such as Harris corner [Harris & Stephens 1988], Different of Gaussian (DOG) detector [Lowe 1999] and affine invariant point [Mikolajczyk & Schmid 2002] (see Figure 2.4). These points are localized in the zones of the image which contain rich information. They held also some invariant properties to image transformations (e.g., affine, scale, rotation) and illumination conditions.

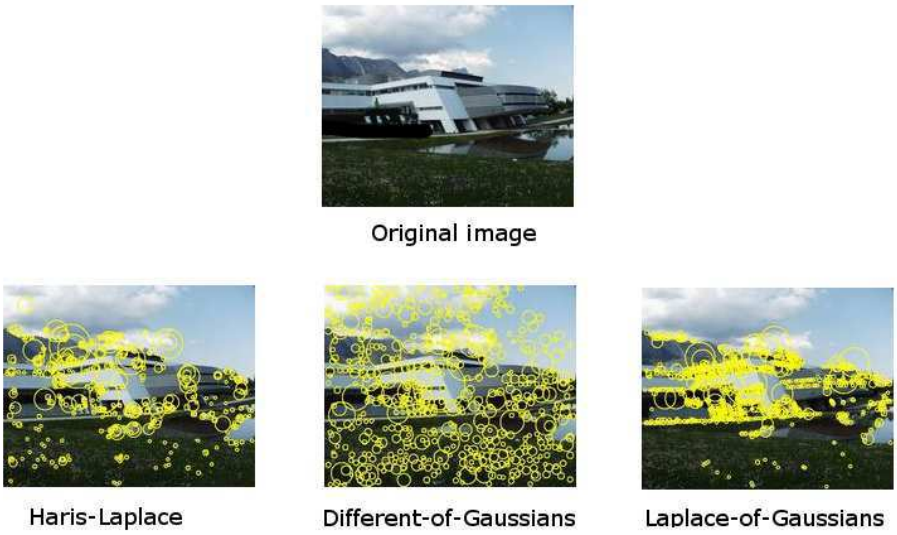

Figure 2.4: Saliency regions detected using different methods: Haris-Laplace, DOG and LOG [Mikolajczyk & Schmid 2002].

The saliency region-based model has shown good performance in object recognition problems with very high accuracy on some limited object databases and with certain kind of objects (building, car, bicycle ...) [Lowe 2004]. However, dealing with more general objects and with a large dataset, the performance of saliency-based decreases substantially.

2.3 Visual features

2.3.1 Color histogram

RGB space. This is the fundamental representation of color in computer. RGB uses an additive model in which red (R), green (G) and blue (B) are combined in various ways to reproduce the color space. This color model is simple. But it is sensitive to illumination changes. Nevertheless, this color model

in widely used in object recognition [Duffy & Crowley 2000] and in region-based color retrieval systems [Carson *et al.* 1999].

HSV[1] **space**. Artists sometimes prefer to use the HSV color model over alternative models such as RGB or CMYK[2] space, because of its similarities to the human color perception. HSV encapsulates more information about a color. Using this color model in object representation has shown its efficiency and its invariance to illumination changes.

L*a*b space. The CIE 1976 L*a*b color model, defined by the International Commission on Illumination (Commission Internationale d'Eclairage, hence its CIE initialism), is the most complete color model used conventionally to describe all the colors visible to the human eye. The three parameters in the model represent the lightness of the color L, its position between magenta and green $a*$ and its position between yellow and blue $b*$. This color description is very interesting in the sense that computer perceives the color close to the human vision.

According to a color sapce, a *color histogram* is then extracted for each image. Considering a three-dimensional color space (x, y, z), quantized on each component to a finite set of colors which correspond to the number of bins N_x, N_y, N_z, the color of the image I is the joint probability of the intensities of the three color channels. Let $i \in [1, N_x],\ j \in [1, N_y]\ and\ k \in [1, N_z]$. Then, $h(i, j, k) = Card\{p \in I \mid color(p) = (i, j, k)\}$. The color histogram H of image I is then defined as the vector $H(I) = (..., h(i, j, k), ...)$.

In [Swain & Ballard 1991], an image is represented by its color histogram. Similar images are identified by matching theirs color histograms with the color histogram of the sample image. The matching is performed by histogram intersection. Similar approach has been installed in the QBIC[3] system [Flickner *et al.* 1995]. This is also the first commercial image retrieval system developed by IBM. This method is robust to changes in the orientation, scale, partial occlusion and changes of the viewing position. However, the main drawback of the method is its sensitivity to illumination conditions as it relies only on color information.

2.3.2 Edge histogram

Edge or shape in images constitutes an important feature to represent the image content. Also, human eyes are sensitive to edge features for object recognition. Several algorithms have been applied for edge detection using different

[1] Hue Saturation Value
[2] Cyan Magneta Yellow blacK
[3] http://wwwqbic.almaden.ibm.com/

methods [Harris & Stephens 1988, Ziou & Tabbone 1998], such as, Prewitt and Sobel mask, Canny filter, or Laplacians of Gaussian filters, etc. As shown in Figure 2.5, edge detection process preserves only the important information on the contours of the object. These contours are then described by the shape descriptors (or edge histogram) and stored for further matching step.

(a) Original image (b) Edge detection

Figure 2.5: Edge detection using Sobel mask.

In the literature, various shape descriptors have been proposed, for example, chain code boundary [Freeman 1974], Shape Context [Belongie & Malik 2000], and Radon transform descriptor [Tabbone *et al.* 2006], etc. The edge histogram is invariant to image *translation* and *rotation*, and normalizing the histogram leads to *scale invariance*. Exploiting the above properties, these methods are useful for object recognition [Belongie *et al.* 2002, Ferrari *et al.* 2010] and image retrieval [Zhang & Lu 2001, Prasad *et al.* 2001].

As proposed in [Won *et al.* 2002], the local edge histogram has been used for shape descriptor in MPEG-7 video standard. Basically, the local edge histogram represents the distribution of 5 types of edges in each local area called a sub-image. As shown in Figure 2.6, the sub-image is defined by dividing the image space into 4×4 non-overlapping blocks. Hence, the image partition always yields 16 equal sized sub-images regardless of the size of the original image. To characterize the sub-image, a histogram of edge distribution is generated for each sub-image. Edges in the sub-images are categorized into 5 types: vertical, horizontal, 45-degree diagonal, 135-degree diagonal and non-directional edges. Thus, the histogram for each sub-image represents the relative frequency of occurrence of the 5 types of edges in the corresponding sub-image. As a result, each local histogram contains 5 bins. Each bin corresponds to one of 5 edge types. Since there are 16 sub-images in the image, a total of $5 \times 16 = 80$ histogram bins is required.

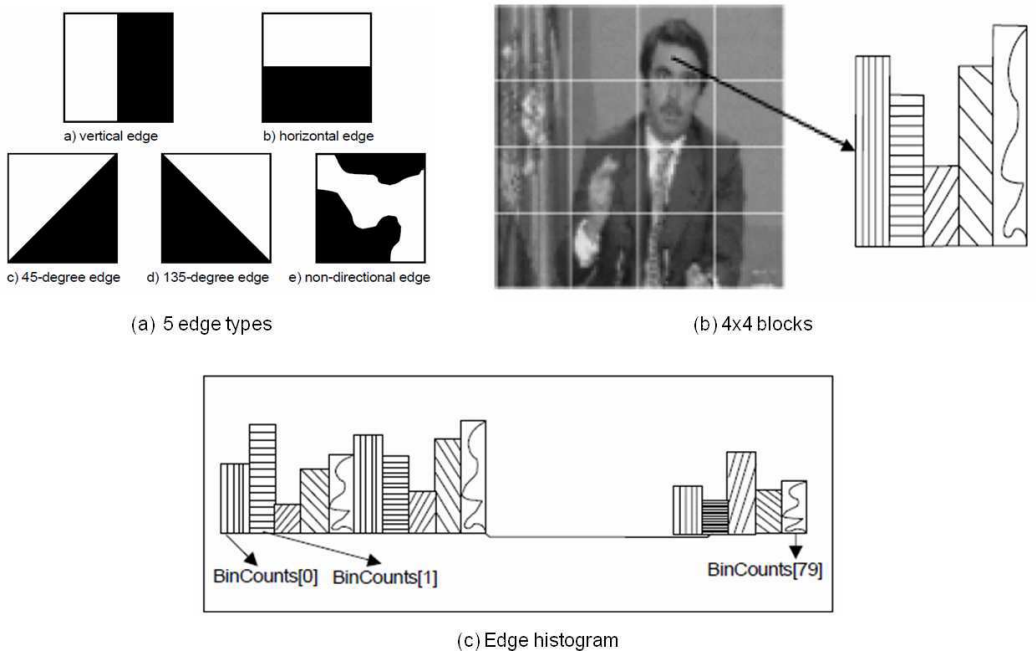

Figure 2.6: Local edge histogram extraction for an image with MPEG-7 standard [Won *et al.* 2002].

2.3.3 Scale Invariant Feature Transform (SIFT)

SIFT extractor has been first introduced in [Lowe 1999]. These features belong to the class of local image features. They are well adapted for characterizing small details. Moreover, they are invariant to image *scaling*, image *translation*, and partially invariant to *illumination changes* and *affine* for 3D projection. Thanks to these invariant properties, SIFTs are become more and more popular visual features for image and video retrieval [Lazebnik *et al.* 2006, Lowe 2004].

First, features are detected through a staged filtering approach that identifies stable points in scale space. The result of this detection is a set of key local regions. Then, given a stable location, scale, and orientation for each key point, it is possible to describe the local image regions in a manner invariant to these transformations. Key locations are selected at maxima and minima of a difference of Gaussians (DOG) applied in scale space. The input image I is first convolved with the Gaussians function to give an image A. This is then repeated a second time with a further incremental smoothing to give a new image B. The difference of Gaussians function is obtained by subtracting image B from A. This difference of Gaussians is formally expressed as:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y)$$

with k corresponding to the strength of smoothing and

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

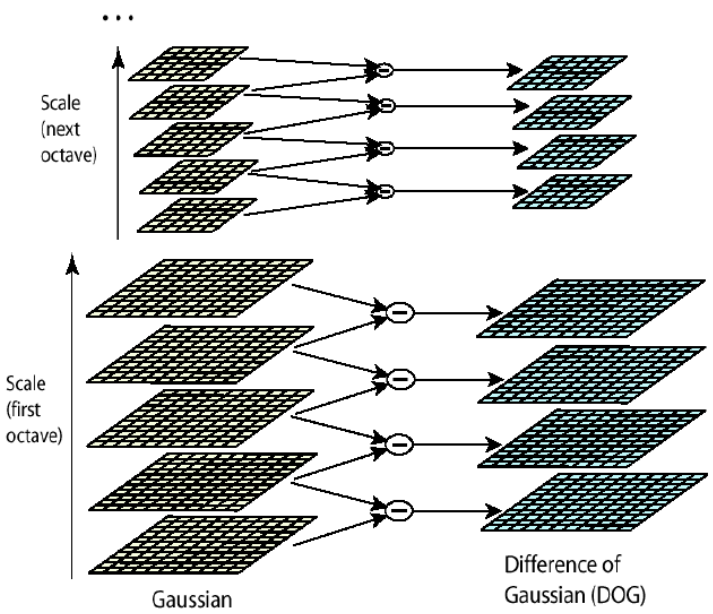

Figure 2.7: Construction of the scale space pyramid.

This differentiation process is repeated with different values of k. A change of scale consists of sampling the smoothed images by using a bilinear interpolation. The combination of scaling and smoothing produces a scale space pyramid. An overview of the scale/space construction is shown in Figure 2.7.

Minima and extrema detection of $D(x, y, \sigma)$ uses this scale space pyramid and is achieved by comparing each sample point to its neighbors in the current image and 9 neighbors in the scale above and below. It is selected only if it is larger than all its neighbors or smaller than all its neighbors. The result of this selection is a set of key-points which are assigned a location, a scale and an orientation (i.e. obtained by gradient orientation computation).

The last step consists of assigning a numerical vector to each keypoint. The 16×16 neighborhood around the key location is divided into 16 sub-regions. Each sub-region is used to compute an orientation histogram. Each bin of a given histogram corresponds to the sum of the gradient magnitude of the pixels in the sub-region. The final numerical vector is of dimension 128.

2.4 Indexing Models

For the past two decades, several indexing models have been proposed in the literature. The objective of image indexing is to store images effectively in the database and to retrieve similar images from a database for a given query image. Image can be indexed using directly the extracted visual features (such as, color, texture and shape) with the vector representation. Recently, the bag-of-visual-features (or bag-of-words) inspired from textual indexing draw more attention for its simplicity and effectiveness on storing visual content. This section is dedicated to the presentation some of these indexing methods.

2.4.1 Vector space model

This is a the simplest model in CBIR system. Images are represented by their feature vectors. These vectors have the same dimension and normalized with the same scale (usually between 0 and 1). The $tf.idf$[4] normalization is often used in information retrieval and text mining. This technique has also adopted widely in CBIR systems. This weighting scheme comes from a statistical measure to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.

Given 2 feature vectors V^q and V^d extracted from image query q and image document d, the visual similarity is computed using two different measurement functions: *Euclidian distance* or *cosines similarity*.

Euclidean distance

The Euclidean distance is probably the most common approach to compare directly two images. Given V^q and V^d are two vectors in Euclidean n-space, then the metric distance of two images p and q is given by:

$$d(V^q, V^d) = ||V^q - V^d|| = \sqrt{||V^q||^2 + ||V^d||^2 - 2V^q \bullet V^d}$$

The smaller distance indicates the closer of two images are. This value reflects the visual similarity of the two images.

Cosine similarity

In contrast to the distance measure, two vectors V^q and V^d can be considered to be similar if the angle between their vectors is small. To compute the cosine similarity, the normalized scalar product is used to measure the angle between two vectors :

[4] term frequency, inverse document frequency

$$cos(\theta) = \frac{V^q \bullet V^d}{||V^q||||V^d||}$$

In information retrieval, the cosine similarity of two documents will range from 0 to 1. A similarity of 0 implies that documents are identical, and a similarity of 1 implies they are unrelated.

2.4.2 Bag-of-words model

A simple approach to indexing images is to treat them as a collection of regions, describing only their statistical distribution of typical regions and ignoring their spatial structure. Similar models have been successfully used in the text community for analyzing documents and are known as "*bag-of-words*" (BoW) models, since each document is represented by a distribution over fixed vocabulary.

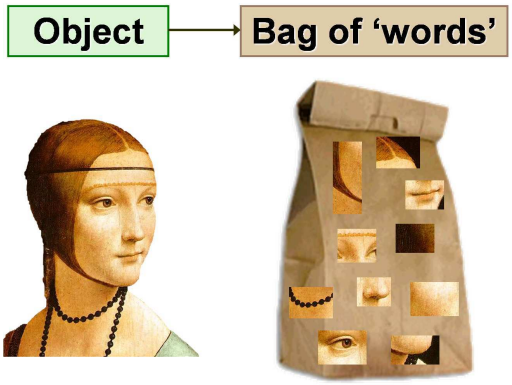

Figure 2.8: Image is represented by a collection of visual words [Fei-Fei & Perona 2005].

The construction of this model is based on four main steps:

1. Image segmentation consists of dividing image into smaller parts. As introduced in previous section 2.2, we can consider different types of image segmentation such as pixels, regions or interested points.

2. Feature extraction step consists of representing each image region by a set of visual features as detailed in section 2.3. Each feature is quantized and normalized by a vector with fixed size.

3. Visual vocabulary construction step converts feature vector represented image regions to "*visual words*" or "*visual concepts*" (analogy to words in text documents), which also produces a "*visual dictionary*" (analogy to a word dictionary). A visual word can be considered as a representative of several similar image regions. One simple method is performing k-means clustering over all the vectors. Visual words are then defined as the centers of the clusters. The number of the clusters k is the vocabulary size.

4. Each image region is mapped to a certain visual word through a clustering process and the image can be represented by the quantized vector of the visual vocabulary.

In step 3, *k-means clustering* is performed on a set of visual features to construct the visual words. We present in the following a brief description of this algorithm.

K-means clustering is a popular technique for automatic data partitioning in machine learning. The goal is to find *k* centroid vectors $\mu_1, ..., \mu_k$ for representing each cluster. The basic idea of this interactive algorithm is to assign each feature vector x to the cluster such that the sum of squared error Err is minimum

$$Err = \sum_{i=1}^{k} \sum_{j=1}^{N_j} ||x_{ij} - \mu_i||^2$$

where x_{ij} is the j^{th} point in the i^{th} cluster, μ_i is the mean vector of i^{th} cluster and N_j is the number of pattern in the j^{th} cluster. In general, the k-means clustering algorithm works as follows:

1. Select an initial mean vector for each of *k* clusters.

2. Partition data into *k* clusters by assigning each pattern x_n to its closest cluster centroid μ_i.

3. Compute new mean clusters $\mu_1, ..., \mu_k$ as the centroids of *k* clusters.

4. Repeat step 2 and 3 until the cluster criterion is reached.

The initial mean vectors can be chosen randomly from *k* seed points in the data in the first step. The partitioning is then performed from these initial points. In the second step, to measure the distance between two patterns, different metric distances (e.g., Hamming distance, Euclidean distance, etc.) can be applied. Usually, the Euclidean distance is good enough to measure the distance between two vectors in the same feature space. In step 3, the centroid μ_i for each cluster is re-estimated by computing the mean of cluster members. The number of iterations

can be used in the last step as a convergence criterion. The k-means algorithm has a time complexity of $\mathcal{O}(nk)$ for each iteration. Only one parameter which needs to be fixed is the number of clusters k.

As demonstrated in [Fei-Fei & Perona 2005], this model is simple but yet effective for image indexing. However, the lack of spatial relation and location information of visual words are the mains drawbacks of this model. Using this representation, methods based on latent semantics extraction, such as latent semantic analysis [Monay & Gatica-Perez 2003, Pham *et al.* 2007] and probabilistic latent semantic analysis [Monay & Gatica-Perez 2004] and latent Dirichlet allocation [Blei *et al.* 2003], are able to extract coherent topics within document collections in an unsupervised manner. Other approaches are based on discriminative methods with annotated or slightly annotated examples, such as support vector machine [Vapnik 1995] and nearest neighbors [Shakhnarovich *et al.* 2005]. In the next chapter, we will review of some of these learning methods.

2.4.3 Latent Semantic Indexing

Latent Semantic Analysis (LSA) was first introduced as a text retrieval technique [Deerwester *et al.* 1990] and motivated by problems in textual domain. A fundamental problem was that users wanted to retrieve documents on the basis of their conceptual meanings, and individual terms provide little reliability about the conceptual meanings of a document. This issue has two aspects: *synonymy* and *polysemy*. *Synonymy* describes the fact that different terms can be used to refer to the same concept. *Polysemy* describes the fact that the same term can refer to different concepts depending on the context of appearance of the term. LSA is said to overcome these deficiencies because of the way it associates meaning to words and groups of words according to the mutual constraints embedded in the context which they appear. In addition, this technique is similar with the popular technique for dimension reduction, i.e., principal component analysis [Gorban *et al.* 2007], in data mining. It helps to analyze the document-by-term matrix by mapping the original matrix into lower dimensional space. Hence, the computational cost is also contracted.

Considering each image as a document, a coocurrence matrix of document-by-term M, a concatenation of vectors extracted from all document with model BoW is built. Following the analogy between textual document and image document, given a coocurrencce document-by-term matrix M rank r, M is decomposed into 3 matrices using Singular Value Decomposition (SVD) as follows:

$$M = U\Sigma V^t$$

where

$$\begin{cases} U \text{ : is the matrix of eigenvectors derived from } MM^t \\ V^t \text{ : is the matrix of eigenvectors derived from } M^tM \\ \Sigma \text{ : is an } r \times r \text{ diagonal matrix of singular values } \sigma. \\ \sigma \text{ : are the positive square roots of the eigen-values of } MM^t \text{ or } M^tM \end{cases}$$

This transformation divides matrix M into two parts. One is related to the documents and the second related to the terms. By selecting only k largest values from matrix Σ and keep the corresponding column in U and V, the reduced matrix M_k is given by:

$$M_k = U_k \Sigma_k V_k^t$$

where $k < r$ is the dimensionality of the concept space. Indeed, the choice of parameter k is not obvious and depends on each data collection. It should be large enough to allow fitting the characteristics of the data. On the other hand, it must be small enough to filter out the non-relevant representation details. To rank a given document, the query vector q is then projected into the latent space to obtain a *pseudo-vector*, $q_k = q * U_k$, with dimension reduced.

Recently, LSA has been applied for scene modeling [Quelhas *et al.* 2007], image annotation [Monay & Gatica-Perez 2003], improving multimedia documents retrieval [Pham *et al.* 2007, Monay & Gatica-Perez 2007] and indexing of video shots [Souvannavong *et al.* 2004]. In [Monay & Gatica-Perez 2003], Monay and Gatica-Perez have demonstrated that the LSA outperformed the pLSA of more than 10% on annotation and retrieval task based on COREL collection. Unfortunately, LSA lacks a clear probabilistic interpretation comparing to other generative models such as probabilistic latent semantic analysis.

2.5 Conclusion

In this chapter, we have introduced the basic steps in constructing an image indexing system. Images are decomposed into image regions and then visual features are extracted for indexing. Each type of image representation and visual features described in this chapter represents a *point of view* of an image. It can be combined in different ways for effective use of the retrieval process. Most of the current approach are based on the early fusion method which relies on the vector combination for the image indexing. Next chapter will discuss on how the machine learning methods will be used for image modeling and retrieval.

Chapter 3

Image Modeling and Learning

3.1 Introduction

In the previous chapter, we presented the popular techniques that have been used for image indexing. An image is decomposed in several ways (from pixels to image regions) for faciliting visual feature extraction. From the extracted image regions, several visual features have been considered, such as color histogram, edge histogram and SIFT. The early image indexing model with vector representation of the bag-of-word model were also described.

In this chapter, we study some machine learning methods used for image modeling in the literature. Following the paradigm of Marr [Marr 1982], these steps correspond to the *mapping layer* and the *interpretation layer*.

First, we will give an overview on the state-of-the-art of the two major branches of learning models: generative approaches and discriminative approaches. The important theory of language modeling for text retrieval will also be presented. Structured image representation has been introduced early in the computer vision [Ballard & Brown 1982] and then applied for image modeling [Boutell *et al.* 2007, Aksoy 2006, Ounis & Pasca 1998]. The main issue of structured image representation is the matching methods based on graph. Classical approaches on sub-graph isomorphism [Ullmann 1976] are costly and ineffective, with its computational complexity cast as NP-complete problem. Modern approaches, such as kernel based and 2D HMMs, express the graph matching by classifying of *paths* and *walks* with SVM kernel or as the stochastic process of Markov's model.

Currently, the generative model, such as language modeling [Wu *et al.* 2007, Maisonnasse *et al.* 2009] are extensively studied for the generative matching process. We will also give a discussion on this active topic. From these pivots, we propose an approach that takes the advantage of both graph-based image

representation and the generative matching process to construct the visual graph modeling. With this approach, we hope to add a new layer to reduce the semantic gap discussed in the literature.

Section 3.2 presents two methods of generative approaches: *Naive Bayes* and *Probabilistic Latent Semantic Analysis* (pLSA). The language modeling approach from information retrieval will be detailed in section 3.3. Two others methods of discriminative approaches, namely *Nearest Neighbors* and *Support Vector Machine* (SVM), will be described in section 3.4. Then, section 3.5 concentrates on the structured representation of the image with the graph model, such as *Conceptual Graph* (CG) and *Attributed Relation Graph* (ARG). We will also introduce some graph matching techniques developed in the literature, for exmple, *(sub)graph isomorphism*, *kernel based* methods and *two dimensional multiresolution hidden Markov models* (2D MHMMs). Finally, based on the review of the state-of the art, we propose our graph-based image representation approach and the matching method inspired from the language modeling in section 3.6.

3.2 Generative approaches

3.2.1 Naive Bayes

Naive Bayes is a simple probabilistic classifier based on Bayes's theorem. It has a strong condition on the class where each feature is estimated independently. In general, the probability model for a classifier is a conditional model over a dependent class variable C with a small number of classes, conditional on several feature variables F_1 through F_n. Using Bayes' theorem, we write:

$$p(C|F_1,\ldots,F_n) = \frac{p(C)\,p(F_1,\ldots,F_n|C)}{p(F_1,\ldots,F_n)}$$

Asume that each feature F_i is conditionally independent of every other feature F_j for $j \neq i$. This leads to

$$p(C|F_1,\ldots,F_n) = \frac{1}{Z}p(C)\prod_{i=1}^{n} p(F_i|C)$$

where Z is a scaling factor dependent only on $F_1,\ldots,F_n$. Finally, the corresponding classifier is defined as follows:

$$\text{classify}(f_1,\ldots,f_n) = \underset{c}{\text{argmax}}\; p(C=c)\prod_{i=1}^{n} p(F_i=f_i|C=c)$$

This is known as the *maximum a posteriori* (MAP) decision rule. This model is popular in text analysis and retrieval, for example: SPAM email detection and document classification. Despite the strong independence assumption, the naive Bayes classifier has successfuly been used for text classification [Iwayama & Tokunaga 1995] and scene categorization [Fei-Fei & Perona 2005]. A hierarchical version of this classifier has been developed by David Blei [Blei 2004] and been applied to both text and image data.

3.2.2 Probabilistic Latent Semantic Analysis (pLSA)

pLSA is a statistical technique for the analysis of co-occurrence data which evolved from Latent Semantic Analysis (LSA) [Deerwester *et al.* 1990], proposed initally by Jan Puzicha and Thomas Hofmann [Hofmann & Puzicha 1998]. In contrast to standard latent semantic analysis which stems from linear algebra and downsizes the occurrence tables (usually via a singular value decomposition), probabilistic latent semantic analysis is based on a mixture decomposition derived from a latent class model. This results in a more principled approach which has a solid foundation in statistics.

Considering observations in the form of co-occurrences (w,d) of words and documents, pLSA models the probability of each co-occurrence as a mixture of conditionally independent multinomial distributions:

$$P(d, w) = P(d)P(w|d)$$

and

$$P(w|d) = \sum_{z \in Z} P(w|z)P(z|d)$$

where z is the latent variable or hidden topic extracted from a set of topics Z of image documents.

The standard procedure for maximum likelihood estimation in latent variable models is the Expectation Maximization (EM) algorithm. EM alternates two steps: (i) an expectation (E) step where posterior probabilities are computed for the latent variables z, based on the current estimates of the parameters, (ii) an maximization (M) step, where parameters are updated for given posterior probabilities computed in the previous (E) step. However it is reported that the pLSA has severe over fitting problems. The number of parameters grows linearly with the number of documents.

pLSA methods are very popular for text indexing and retrieval [Hofmann 1999] thanks to its solid probabilistic foundation. This technique was also adopted by the CBIR community [Lienhart *et al.* 2009, Lu *et al.* 2010] and for image annotation

[Monay & Gatica-Perez 2004, Monay & Gatica-Perez 2007]. However, estimating parameter using E-M step is a very costly process which is a main limitation of this method.

The following section present the principal theory of the language modeling which is a key model of this thesis. We also give a short survey of the application of the language modeling for image classification.

3.3 Language modeling approach

Language modeling (LM) was first introduced in linguistic technologies, such as speech recognition, machine translation and handwriting recognition [Rosenfeld 2000]. Ponte and Croft [Ponte & Croft 1998] applied the probabilistic language modeling in text retrieval and obtained good retrieval accuracies on TREC collections. Similar to the previous generative models, the documents are ranked by the probability that the query could be generated by the document models. The query likelihood $P(D|Q)$ is computed by using Bayes' Rule:

$$P(D|Q) = \frac{P(Q|D)P(D)}{P(Q)}$$

We can ignore the normalizing constant $P(Q)$, the former fomular leads to

$$P(D|Q) \propto P(Q|D)P(D)$$

where $P(D)$ is the prior probability of a document, which is assumed to be uniform in most cases. Therefore, the documents are ranked equivalent to the joint probability of $P(Q|D)$. This is known as *maximum a posteriori* (MAP) technique which selects the most probable document D to maximize the posterior distribution of $P(D|Q)$.

3.3.1 Unigram model

The simplest form of language modeling is the unigram model where each word is estimated independently of each other. To estimate the probability of a word in the documents, one has to make an assumption about the distribution of the data. In the literature, a number of different assumptions have been made about the distribution of words in document. The *multiple-Bernoulli distribution* captures a set of binary events that some word appears in the document or not. Therefore, the document can be represented by a binary vector of 0 and 1 to indicate the occurrence of a corresponding word. The multiple-Bernoulli distribution is well suited for representing the presence of

query word and insisting on the explicit negation of words (e.g. $apple$ but not $orange$). In [Ponte & Croft 1998], the original language modeling for IR was based on multiple-Bernoulli distribution assumption. Given multiple-Bernoulli assumption, the query likelihood gives:

$$P(Q|D) = \prod_{w \in q_1,\ldots,q_k} P(w|D) \prod_{w \notin q_1,\ldots,q_k} (1 - P(w|D))$$

where w is a word in document D. This assumption is simple and straightforward. However, one limitation of this distribution is that the latter does not deal with the importance (i.e. the frequency of occurrence) of word in the document. For this reason, most of the current modeling assumptions in IR are now centered on multinomial distributions.

The *multinomial distribution* takes into account the number of occurrences of words (e.g. $apple$ appears 3 times and $orange$ appears 2 times in the document). This suggests that the document can be encoded by a vector with the number of times each word appears in the document. Assuming a multinomial distribution over words, we can compute the query likelihood using unigram model. The query likelihood is then calculated using unigram model for the document as follows

$$P(Q|D) = \prod_{i=1}^{m} P(q_i|D)$$

where q_i is a query word and m is the number of word in the query. To calculate this score, probability of query word q_i is estimated from the document

$$P(q_i|D) = \frac{\#(q_i, D)}{\#(*, D)}$$

where $\#(q_i, D)$ is the number of times word q_i occurs in document D, and $\#(*, D)$ is the total number of words in D. For a multinomial distribution, maximum likelihood refers to the estimate that makes the observed value of (q_i, D) most likely.

One problem with this estimate is that if any of the query words is missing from the document, the score of query likelihood will be zero. This is not appropriate for long query which may have frequently "missing words". In this case, it should not yield a zero score. To overcome this problem, one solution is to give a small probability for missing words which will enable the document to receive a non-zero score. In fact, this small probability is taken from the prior information of the document collection. This solution is known as *smoothing* techniques or *discounting* techniques. We will address this problem in the following section.

3.3.2 Smoothing techniques

Smoothing is a popular technique used in information retrieval to avoid the probability estimation problem and to overcome the data sparity of collection. Typically, we do not have large amount of data to use for the model estimation. The general idea is to lower (or *discount*) the probability estimates for words that are observed in the collection and apply that probability to the unseen words in the document.

A simple method is known as the *Jelinek-Mercer smoothing* [Jelinek *et al.* 1991] involving the linear interpolation of the LM from the whole collection C. Given $P(q_i|C)$ is the probability of query word q_i estimated from the collection C and λ is the smoothing coefficient assigned to the unseen word, the estimate probability of query from document model becomes:

$$P(q_i|D) = (1-\lambda)P(q_i|D) + \lambda P(q_i|C)$$

The collection model for estimating the query word q_i is $P(q_i|C) = \frac{\#(q_i,C)}{\#(*,C)}$, where $\#(q_i, C)$ is the number of time query word q_i appears in collection C and $\#(*, C)$ is the total number of words in the whole collection. Substituting this probability in the query likelihood gives:

$$P(Q|D) = \prod_{i=1}^{m}((1-\lambda)\frac{\#(q_i, D)}{\#(*, D)} + \lambda\frac{\#(q_i, C)}{\#(*, C)})$$

The smoothed probabilities of document model still verify $\sum_{i=1}^{n} P(q_i|D) = 1$. This smoothing method is simple and straightforward. However, it is more sensitive to λ for the long queries than the short queries. The reason is long queries need more smoothing and less emphasis on the weighting of words.

Another smoothing technique called *Dirichlet smoothing* takes into account the document length. The parameter λ becomes

$$\lambda = \frac{\mu}{\#(*, D) + \mu}$$

where μ is a parameter whose value is set empirically. The probability estimation of query word q_i leads to:

$$P(q_i|D) = \frac{\#(q_i, D) + \mu\frac{\#(q_i,C)}{\#(*,C)}}{\#(*, D) + \mu}$$

Similar to Jelinek-Mercer smoothing, parameter μ gives more importance to the relative weighting of words for small values. On the other hand, this parameter also takes into account the prior knowledge of long documents. Therefore, Dirichlet smoothing is generally more effective than Jelinek-Mercer, especially for short queries that are common in the current retrieval engines.

3.3.3 n-gram model

The extension of unigram to the higher order language model is known as n-gram language model. In n-gram model, the probability of estimate for word q_i depends on the $n-1$ preceding words. Hence, it is able to model not only occurrences of independent words like unigram model, but also the fact that several words often occur together. This effect is interesting in text retrieval because the combination of words can have different meaning comparing to the same words used independently (e.g. "swimming pool" or "Wall Street Journal"). The n-gram models will help to capture efficiently this cooccurrence information.

The query likelihood probability $P(Q|D)$ of observing the query $Q = (q_1, \ldots, q_m)$ is approximated as:

$$P(Q|D) = \prod_{i=1}^{m} P(q_i|q_1, \ldots, q_{i-1}, D)$$

Following the assumption that the probability of observing the word q_i in the context history of the preceding $i-1$ words can be approximated by the probability of observing it in the preceding $n-1$ words (n^{th} order Markov property):

$$P(Q|D) \approx \prod_{i=1}^{m} P(q_i|q_{i-(n-1)}, \ldots, q_{i-1}, D)$$

The conditional probability can be calculated from n-gram frequency counts:

$$P(q_i|q_{i-(n-1)}, \ldots, q_{i-1}, D) = \frac{\#(q_{i-(n-1)}, \ldots, q_{i-1}, q_i, D)}{\#(q_{i-(n-1)}, \ldots, q_{i-1}, D)}$$

The *bigram* and *trigram* language models correspond to language models with $n = 2$ and $n = 3$, respectively. Similar to the unigram model, n-gram models also suffer from the problem of probability estimation. Hence, smoothing technique is also required to overcome this problem. The occurrence of bigrams or trigrams in the document to some extent are rather rare comparing to the unigram. More details on the smoothing techniques with n-gram models (such as Good-Turing discounting, Witten-Bell discounting, etc.) can be found in [Jelinek 1998].

Although the standard language models have yielded good performance in text retrieval, several works have investigated further the use of more advanced representations of words within this framework. Gao [Gao *et al.* 2004] and Lee [Lee *et al.* 2006] proposed to incorporate syntactic dependencies structure in the language model. These models defined a *linkage* over query terms which is related automatically through a parse in document. However, there is a certain ambiguity in the way the linkage is used in this model. As pointed out in

[Maisonnasse *et al.* 2008], this model is theoretically inconsistent to represent graphical structure in the language modeling approach to IR.

In contrast, Maisonnasse [Maisonnasse *et al.* 2008] relied on the notion of graph model to integrate the relation between concepts in the language modeling. Concepts and semantic relations are extracted from knowledge source such as UMLS[1] for medical concepts. The authors also proved that the use of concept and the semantic relation on graph achieved a substantial improvement over purely term-based language models (such as unigram and n-gram model). Based on this work, we will extend the graph-based language modeling in [Maisonnasse *et al.* 2009] to take into account of the visual elements and their spatial relations in a unified framework for image retrieval.

3.3.4 Language modeling for image classification

Language modeling has also been applied for capturing the spatial information of the BOW models for image classification. Tirilly et al [Tirilly *et al.* 2008] proposed to use the principal component analysis (PCA) to find the main axis of visual words to be extracted from object. Keypoints are then orthogonally projected back to main axis to construct a visual sentence. The authors also applied the pLSA method in order to eliminate the noisy visual words. The *n-grams* model is estimated for each object. The retrieval process is similar to the one of textual document in standard language model. This method has been experimented on CALTECH-101 image dataset and obtained a promising result for image classification. However, this method is limited to one object per image because of its sensibility in selecting the main axis with PCA. Moreover, the spatial relation of visual word in this case needs more explanation.

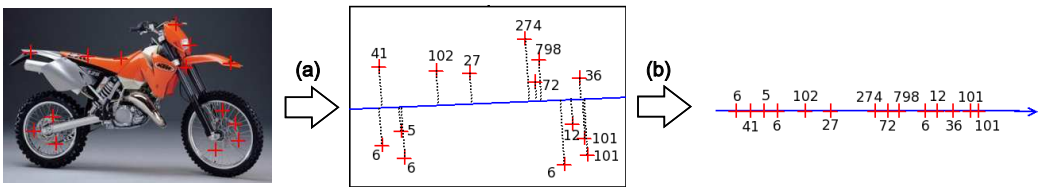

Figure 3.1: Visual words are considered as words in a visual sentence. The construction is followed by two steps: (a) main axis is defined by PCA and (b) keypoints is then orthogonally projected back to main axis to construct a sentence.

Similar work has been exploited by Wu et al. [Wu *et al.* 2007]. Image is divided into regular patches. Each patch is conditionally dependent on the neighbors from top and from left of the current patch. This relation is interesting

[1] Unified Medical Language System

in the sense that it captures the most basic relation in image which is analogous to the relation of words in a document. Three language models (*unigram*, *bigrams* and *trigrams*) constructed follow strictly the theoretical language model. For this reason, the model is hard to extend for more complicated relation between visual words.

3.4 Discriminative approaches

Unlike the generative approach, which is based on the probabilistic principle, the discriminative approach treats each document as a point in some geometric space. There is no explicit assumption on the data itself. The principle is "let data speaks", which means the model will find the decision boundary to separate automatically the annotated samples for the training set and the generalizes to the test sets.

3.4.1 Nearest neighbors approach

Nearest neighbors (or k-NN) is a well-known method for object classification in pattern recognition [Shakhnarovich *et al.* 2005]. The main principle is to match a test sample to the given training samples. An object is classified by a *majority vote* of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (k is usually small). If $k = 1$, then the object is simply assigned to the class of its nearest neighbors.

A drawback to the basic *majority voting* classification is that classes with the more frequent examples tend to dominate the prediction of the new sample, as they tend to come up in the k nearest neighbors when the neighbors are computed due to their large number. One way to overcome this problem is to weight the classification taking into account the distance from the test point to each of its k nearest neighbors.

3.4.2 Support Vector Machines (SVM)

SVM is the most popular discriminative algorithm for classification. Introduced by Vapnik in 1995 [Vapnik 1995], SVM has since become one of the most developed classification algorithms, especially for pattern recognition. The strength of SVM is twofold: in terms of maximizing the margins around the separator hyperplane it provides good capacity of generalization and the application of kernel allow it to solve the problem of non linear separable space.

Figure 3.2 illustrates the operation of SVM for classification in a linear space of two dimensions. H denotes the hyperplane which separated white dots and

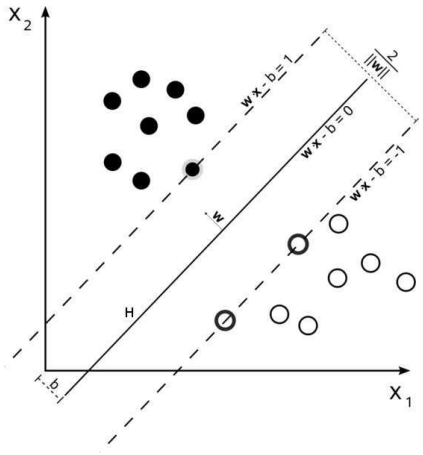

Figure 3.2: SVM is to search for the maximal margin that separates the training set in a linear space of two dimensions. In this case, the training set is separable.

black dots.

Let $\mathcal{L}$ be the set of training points, where each point x_i has m attributes (i.e. vector of dimensionality m) and belongs to one of two classes $y_i \in \{-1, +1\}$. Here we assume the data are linearly separable, meaning that we can draw a hyperplane on the space $\mathcal{L}$. This hyperplane can be described by $w \cdot x_i - b = 0$ where:

- w is normal to the hyperplane.
- $\frac{b}{||w||}$ is the perpendicular distance from the hyperplane to the origin.

Then the goal is to minimize the value $||w||$ of the margin such that the objective function is maximum. Minimizing $||w||$ is equivalent to minimizing $\frac{1}{2}||w||^2$ and the use of this term makes it possible to perform Quadratic Programming (QP) optimization . Therefore, we need to find:

$$min \frac{1}{2}||w||^2$$

subject to

$$y_i(w \cdot x_i - b) - 1 \geq 0, \forall i$$

In order to cater for the constraints in this minimization, we need to allocate them Lagrange multipliers α_i. It can be shown that this is equivalent to the minimization of:

$$\min_{\mathbf{w},b,\boldsymbol{\alpha}} \{\frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i[y_i(\mathbf{w} \cdot \mathbf{x_i} - b) - 1]\}$$

with $\alpha_i \geq 0$ and under constraint $\sum_{i=1}^{n} \alpha_i y_i = 0$. This can be achieved by the use of standard QP methods. Once we obtained the solution vector α^0 of the minimization problem, the optimal hyperplane (w_0, b_0) will be defined by:

$$w_0 = \sum_{n}^{i=1} \alpha_i^0 y_i x_i$$

Points corresponding to solution α^0 are called *support vectors*. The decision rule for new point x is then defined by function $f(x)$:

$$f(x) = \sum_{n}^{i=1} \alpha_i^0 y_i x_i \cdot x - b^0$$

The *sign* of *f(x)* is usually used as binary decision. If it is positive (respectively negative), the test point *x* belongs to the class of training set with label +1 (respectively -1).

This approach can also be applied to non linear separable data with some mapping functions $\Phi(x)$ of the input feature vectors into a high-dimensional feature space (see Figure 3.3). This technique is called *kernel trick*. The kernel trick is useful because there are many classification/regression problems that are not linearly separable/repressible in the space of the input features.

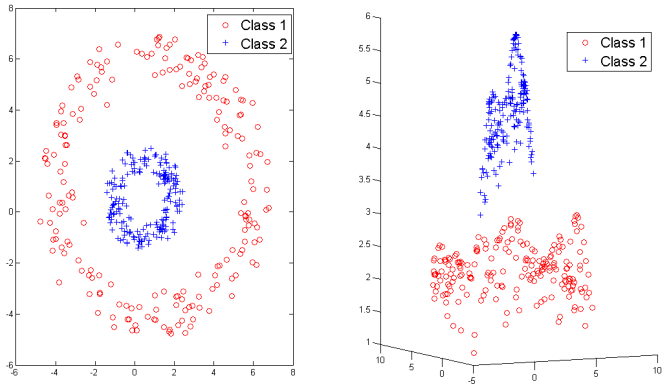

Figure 3.3: Example of Radial Basis Function (RBF) kernel mapping data from non linear separable space to high-dimensional separable space.

The kernel is expressed by the dot product of mapping function which gives $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$. Once kernel K satisfying the Mercer condition[2], the output function then becomes:

[2]symmetric and positive matrix

$$f(x) = \sum_{n}^{i=1} \alpha_i^0 y_i K(x_i, x_j) - b^0$$

There are different types of kernel, such as polynomial kernel, sigmoid kernel and radial basis kernel. But the most commonly used in pattern recognition is radial basis function (RBF) kernel, defined as follows:

$$K(x, y) = e^{-\frac{||x-y||^2}{2\sigma^2}}$$

where $||.||$ is L_2-norm of two vectors x and y and σ is the smoothing parameter of Gaussian function. In general, this parameter is estimated by cross validation on the data from training set.

3.5 Structured representation approaches

Most of the learning methods described previously are based on the *bag-of-words* models, thus taking into account only the visual information in the form of vector representation. The spatial information among visual features is disregarded. Although this representation is simplistic, the result proved surprisingly promising. Some efforts have been attempted to go beyond this primitive and flat representation of *bag-of-words* model by adding visual feature correlation information [Lazebnik *et al.* 2006]. However, none of them really considered the spatial relationship between image regions. This section aims at introducing some state-of-the-art in graph representation, which is popular in interpreting structural information, and learning methods for image matching.

3.5.1 Graph for image modeling

Graph is one of the most formal representations of structural information in computer vision [Marr 1982]. It is a natural way to encode the relation between objects. A famous example of graph is presented in a book of computer vision by Ballard and Brown [Ballard & Brown 1982]. The idea is to represent structural information of face by a set of *templates* connected by *springs*. Nodes represent the instances of face, for example: eyes, nose, mouth, hair, etc. Edges indicate the structural relations between these instances. This is one of the most primitive forms for representing a graph where nodes and edges indicate the object instances and links between them respectively. However, suffering from the combinatorial explosion issue in graph matching, the classical graph representation has not been used widely for image modeling in image retrieval.

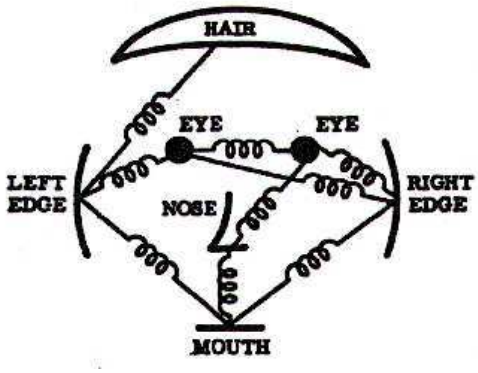

Figure 3.4: An example of graph representation for human face [Ballard & Brown 1982].

For real image representation, nodes and edges are associated with a set of labels. Harchaoui and Bach [Harchaoui & Bach 2007] presented their images by a planar graph where each node corresponds to a segmented region and edge corresponds to a link between two connected regions. One interesting constraint is all the regions are non-overlapping, thus creating a planar graph. This representation is simple and intuitive, although the inferring process on these graph representations is challenging problem. A method to reduce the complexity of this graph, combining with kernel methods, will be introduced in the next section.

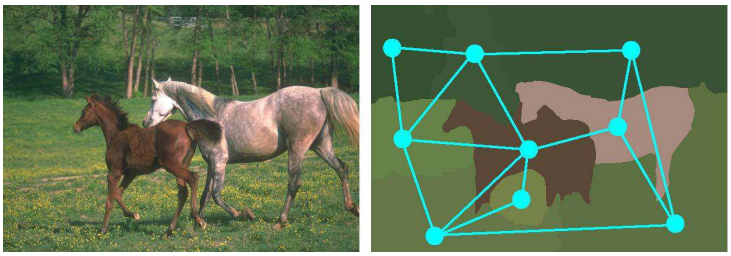

Figure 3.5: An example of planar graph extracted from an image. Segmented regions are obtained by using the median RGB color. [Harchaoui & Bach 2007]

Conceptual graphs have been first introduced in the early 80's to model knowledge representation. In [Sowa 1984], Sowa presented the theoretical formalism for conceptual graph which was consistent and flexible for knowledge representation. This framework can capture semantic representation of data and it offers some useful extension which is likely applicable for other knowledge-based representation such as semantic web or data mining.

Conceptual graphs are widely considered as a chanel to express the representation of image content. Figure 3.6 illustrates an example of conceptual graph extracted from a landscape scene. Nodes represent the visual entities composed of image and directed arcs indicate semantic relations between these nodes. Conceptual graphs have also been used for scene recognition [Boutell *et al.* 2007] and for image retrieval [Ounis & Pasca 1998]. With the integrating of semantic relations, conceptual graph allows to describe better the nature of image contents [Mulhem *et al.* 2001, Boutell *et al.* 2007].

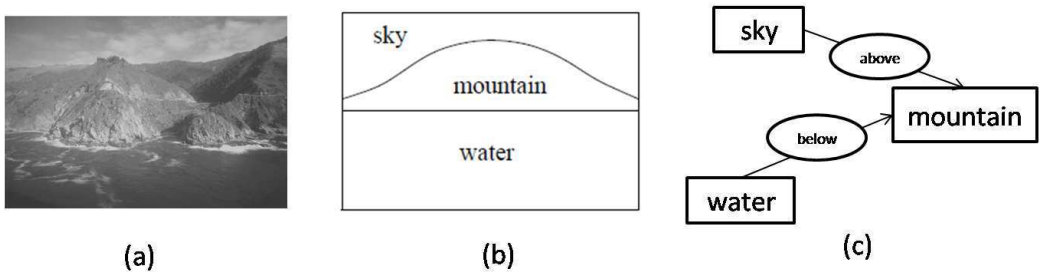

Figure 3.6: Example of a conceptual graph extracted from natural scene: (a) original image, (b) manual segmented image, (c) conceptual graph representation.

An extension of conceptual graph is attributed relational graph (ARG) where nodes and relations are associated with some attributes represented by their weights or their probability of contribution [Mulhem *et al.* 2001]. Attributed relational graphs have been widely used for image modeling [Aksoy 2006] and near duplicated image detection [Zhang & Chang 2004]. One advantage of the ARG is that it can be used to represent complex visual content in the very flexible way. Node and link can be easily embedded with some properties, such as weight, numeric or symbolic value or even with the estimated probabilities [Boutell *et al.* 2007].

Figure 3.7 presents some of the attributes used for representing the spatial relations between two image regions. With this representation, the spatial relationships among regions are expressed with more details, for instance, symbolic relations *near* and *far*, or relative distance $d = 0.35$ and relative angle $a = \pi/3$, etc. However, matching the attributes relation graphs requires a special technique to adapt to some specific problems. We will present in the next section some matching algorithms for graph-based image presentation in the literature.

3.5.2 Matching methods on graphs

After defining a graph based on their representation, several matching process on graph have been summarized in [Ballard & Brown 1982]:

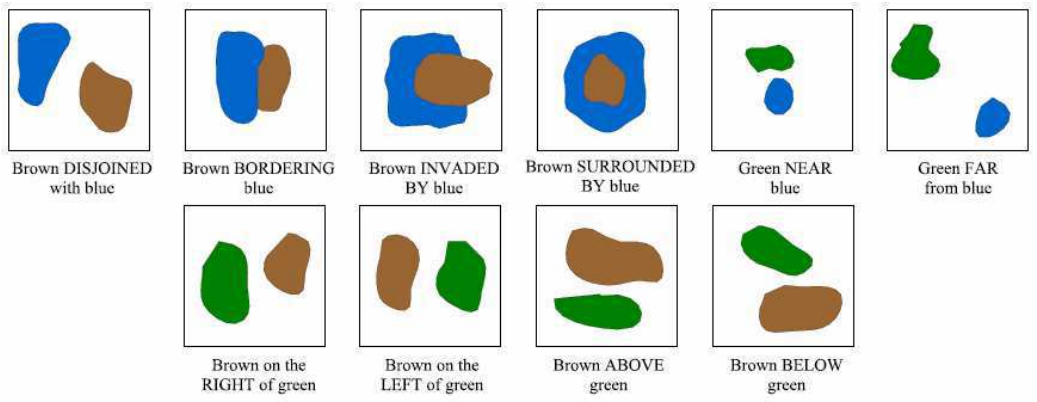

Figure 3.7: Spatial relationships of region pairs: *disjoined*, *bordering*, *invaded by*, *surrounded by*, *near*, *far*, *right*, *left*, *above* and *below* [Aksoy 2006].

- Exact matching: graph isomorphism, subgraph isomorphism (see Figure 3.8).
- Inexact matching: partial graph matching, attributed graph matching.

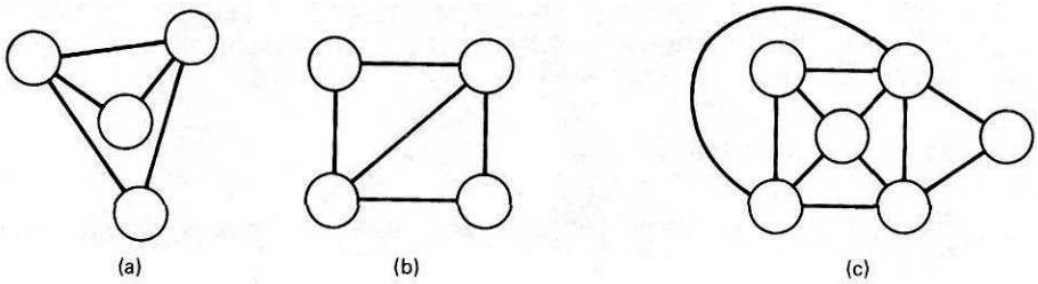

Figure 3.8: Exact graph matching: graph (a) has an isomorphism with graph (b) and has various subgraph isomorphism with graph(c) [Ballard & Brown 1982].

Graph matching is related to string theory [Gusfield 1997] (such as substring matching and edit distances) which is complex and costly operation. Therefore, exact graph matching is a combinatorial problem and subgraph matching is a NP-complete problem [Ullmann 1976]. The deterministic algorithms run (in the worst case) in time exponential with the size of the constructed graphs. Several works [Shokoufandeh *et al.* 2002, Cordella *et al.* 1998] have been involved to solve the graph matching problem in polynomial time. However, these algorithms are very complicated in implementation. Soft graph matching have to adapt to the nature of the application [Shokoufandeh *et al.* 2002]. Hence, most of works so far have focused on finding the approximate solutions to this problem.

To compute the similarity of conceptual graph, several matching methods have been proposed, such as: partial subgraph isomorphism [Ullmann 1976], error correction graph matching [Mulhem *et al.* 2001], median graphs [Jiang *et al.* 2001]. As demonstrated in [Ballard & Brown 1982] the matching algorithms of conceptual graph may also suffer from the computational problems as they used basically the morphological matching algorithm. In [Ounis & Pasca 1998], authors proposed to use inverted file for indexing and retrieval of conceptual graph extracted manually from images to accelerate the performance of graph matching. Recently, Kostin et al. [Kostin *et al.* 2005] have applied probabilistic relaxation matching technique for object recognition. Despite the fact that graph matching is still a challenging problem, graphs are very promising for structural image representation.

Recently, applications of kernel-based methods are widely used for semi-supervised learning [Shawe-Taylor & Cristianini 2004, Bach *et al.* 2004] and in computer vision [Suard *et al.* 2005]. In [Harchaoui & Bach 2007], authors proposed a method to combine graph planarity with a kernel of a SVM classifier for image classification. Image is represented as a planar graph in which labeled nodes correspond to segmented regions and edges are the neighboring regions. Image graphs are fixed with the same number of segments. Figure 3.9 shows a simple planar graph constructed for an image. Each color represents a different label and each edge encodes a spatial relation.

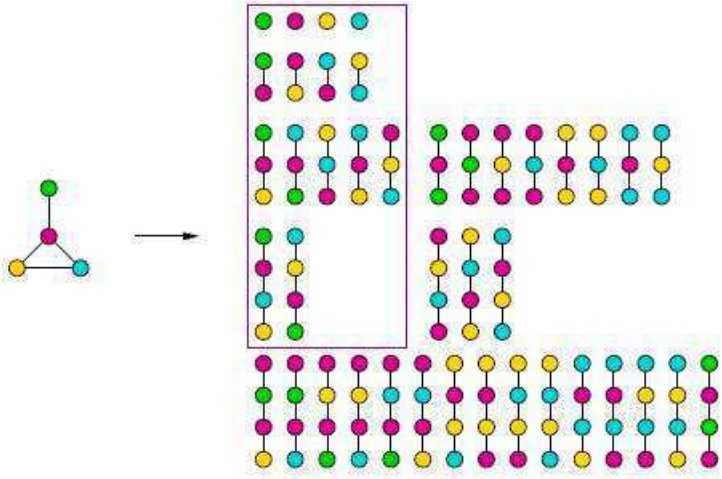

Figure 3.9: Enumeration of paths and walks from a simple graph. *Walk* is a finite sequence of neighboring vertices, while a *path* is a walk such that all its vertices are distinct (inside the rectangle) [Harchaoui & Bach 2007].

Kernel graph is an efficient way to solve the graph matching problem by soft-matching tree-walks in order to *obtain kernels computable in polynomial*

time [Harchaoui & Bach 2007]. The kernels keep the underlying topological structures of graph through *walks* and *paths* (see Figure 3.9). Moreover, the kernels also embed the local information of the segments (such as color histogram of local features). Due to the computational operation on *paths*, the authors choose to implement only the kernels with *walks*. After defining a corresponding walks kernel (for example Dirac kernel for exact graph matching) between two graphs, these parameters are then fed to SVM classifiers for training. This work showed very promising results on image classification.

Likewise, Li and Wang [Li & Wang 2003] introduced a statistical modeling approach to the problem of automatic linguistic indexing of pictures. A *two dimensional multiresolution hidden Markov models* (2D MHMMs) is used to model the stochastic process of associating an image with the textual description of a concept. First of all, each image is summarized by a collection of feature vectors extracted and spatially arranged on a pyramid grid (see Figure 3.10). The 2D MHMM aims at describing statical dependence of the feature vectors at multiresolution and their spatial relations in the same resolution. The number of block is reduced by half at each lower resolution. Blocks at lower resolution cover spatially more abstract information of the image.

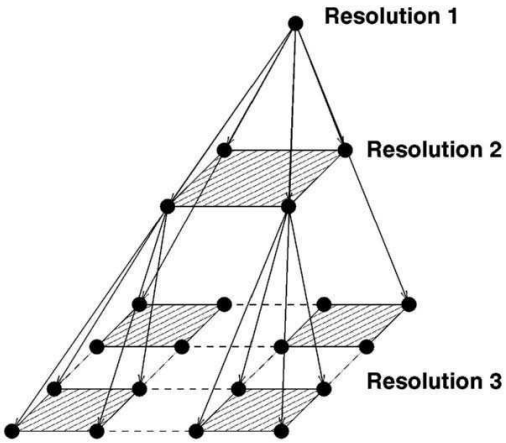

Figure 3.10: The hierarchical statistical dependence across different resolutions [Li & Wang 2003].

Images are trained based on given concepts. For example, concept $Paris/France$ includes images with category description such as "*Paris, european, historical building, beach, landscape, water*". This helps to improve statically the 2D MHMM profile for each visual concept. For a test image, feature vectors are extracted from the pyramid grid. The likelihood of the feature vectors being generated by each profiling 2D MHMM is computed. Images are ranked based

on their likelihood with each image concept. This approach is interesting in the sense that it considers a statistical model for each group of images. Moreover, it takes into account the spatial information throughout the 2D HMM framework. This method has shown good accuracy and also has a high potential for automatic image annotation. However, it is limited for image retrieval as the matching process based on the Markov's model is time consuming.

3.6 Our proposition within graph-based framework

Firstly, we aim to provide an alternative method for image modeling which can consider different type of image representations and different visual features. The need of a model that could take several image points of views is one of our objectives. We are also motivated by the fact that there is still a gap between the low-level features model and that of the high-level semantic ones. We create an intermediate-level image representation layer between image semantics and the middle-level of concepts included various visual features along with the spatial relations among them. Such image representation layer can easily describe the image contents, for example, "*building is in the left of the tree*", "*cloud is in the top of the building*", etc.

Secondly, generative models have been around for decades and been applied successfully to textual retrieval. These methods are both practical in terms of implementation and effective in term of computational cost. Moreover, the extension of the generative matching process does exist for the complex knowledge representation, such as for conceptual graph [Maisonnasse *et al.* 2008]. To the best of our knowledge, no one has tried to use generative methods for graph matching process. In this regard, our second objective is to study the effect and benefit of using a probabilistic framework for matching of the graph-based image representation.

Therefore, our proposition graph-based framework will include the following original contributions to the current state-of-the art:

- **A unified graph-based representation for image modeling**. Our goal is to automatically deduce for each image a visual graph representing the image contents. For this, image regions are automatically associated with the visual concepts, and spatial relations are used for creating links between these regions and keypoints. The frequency of visual concepts and their relations are also captured as the weights in our visual graphs.

  The advantage of this model is that it offers an intuitive representation of image content. Moreover, by allowing the user to select the image

representations (such as visual concepts) and the spatial relations to be considered it can be more easily matched to a particular image category.

- **A generative matching method using language modeling**. To reduce the computational cost, we propose to use the language modeling for generative graph matching process. Unfortunately, the current conceptual language modeling framework is limited to only a set of concept and a set of relation [Maisonnasse *et al*. 2008]. Therefore, we will extend the theory of this framework in order to take in to account of multiple concept sets and multiple relation sets. To do that, we have to make several independence assumptions based on the concept sets and relation set. We also propose a simple smoothing method for the probability estimation of concept and relation in this framework.

3.7 Conclusion

To summarize, in this chapter we surveyed the current learning models, such as generative approaches and discriminative approaches. The important theoretical aspect of the language modeling inspired from information retrieval is also provided in section 3.3. Furthermore, we have investigated different structured image representations on image modeling, for instance conceptual graph and attributed relational graph. We have also studied some graph matching methods based on discriminative approach (such as embedding of paths and walks in kernel based classification) or generative approaches (such as Markov's model and language modeling). Motivated by the limitation of the current state-of-the-art methods, we have proposed a new approach based on the graph-based image representation and a generative process for graph matching.

The next part contributes on designing the proposal method. As said, chapter 4 explains how the framework works with three principal steps: image processing, graph modeling and graph retrieval. Chapter 5 details the graph formulation and the graph matching based on the language modeling. We will give some examples to illustrate the constructed graph and how we compute the likelihood probability for a pair of graphs.

Part II

Our Approach

Chapter 4

Proposed Approach

Design is not just what it looks like and feels like. Design is how it works.
Steve Jobs

4.1 Framework overview

Inspired by the bag-of-word model, images are modeled as a set of visual words (concepts) described and supported by different visual features and representations.. As we explained previously, our goal is to automatically deduce, from a given image, a graph that represents the image content. Such a graph will contain concepts directly associated with the elements present in the image, as well as spatial relations which express how concepts are related in the image.

The reason that we have choosen graph as the image representation is due to its capacity of embedding complex symbolic relations and attributes of concepts (such as numerical value or probability estimation). Alternatively, with this presentation we can apply an extension of language modeling, which is a generative probabilistic model, for the graph retrieval process.

To do so, we present in this section the system architecture that consists of three main stages (see Figure 4.1).

1. **Image processing** aims at extracting image regions (i.e., segmentation, grid partition or saliency point detection) from the image. It also consists of computing the numerical feature vectors (e.g., color, edge histogram, and local feature information) associated with regions or saliency points.

2. **Graph modeling** consists of two main steps. First, extracted image regions that are visually similar will be grouped into clusters using an unsupervised learning algorithm (e.g., k-means clustering). Each cluster is then associated with a visual concept. The second process consists of

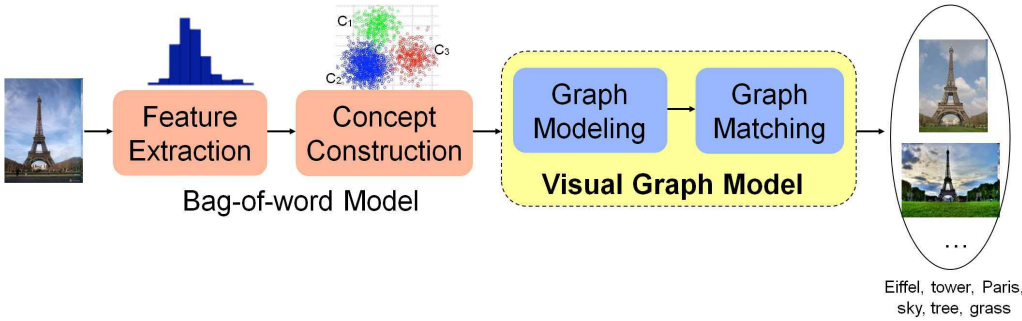

Figure 4.1: System architecture of the graph-based model for image retrieval.

generating the spatial relations between the visual concepts. After these two steps, each image is represented by a visual graph generated from a set of visual concepts and a set of spatial relations among the visual concepts.

3. **Graph retrieval** is to retrieve images relevant to a new image query. Query graphs are generated following the graph modeling step described above. Inspired by the language model for text retrieval, we extend this framework for matching the query graph with the trained graph from the database. Images are then ranked based on their probabilities of the corresponding graphs.

Indeed, these three phrases are clearly distinct from each other. They can be associated with the three layers of a classical paradigm in machine vision of Marr as introduced in chapter 2: the *processing layer* (1), the *mapping layer* (2), the *high-level interpretation layer* (3). Our contributions are mainly related to the graph modeling and graph retrieval problem. In the graph modeling step, we propose a unified graph-based framework for image representation. After that, we propose a graph matching algorithm based on the extension of the language model that was initially proposed in the information retrieval community. We will describe these steps in the following sections.

4.2 Image processing

Given an image I, finding a good representation of image content is a difficult task. In the literature, we can find various techniques for image segmentation. In this section, we present three segmentation techniques of image content that have been applied in our experimentations. As we concentrate more on the graph modeling process, we choose simple and popular techniques for image segmenting, such as pixel sampling, grid partitioning and keypoint detection.

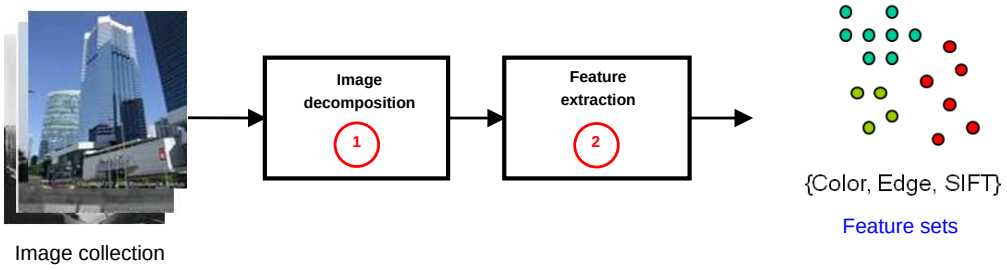

Figure 4.2: Image processing pipeline: (1) Image decomposition, (2) Feature extraction.

From these primitive regions, we extract different visual features (such as color, edge orientation and scale invariant features). These features are associated with one type of image region to represent an image representation (or a point of view). Our objective is to provide a common framework to effectively represent the different viewpoints of image contents using a graph model.

4.2.1 Image decomposition

Image region is the primitive part of image representation. Classical approaches for image representation consider image as a whole in order to take into account of the global visual information of image content. Recently, local region approaches try to represent an image as a composition of different objects (or different parts of object). Several segmentation techniques have been proposed (e.g., N-cut segmentation [Shi *et al.* 1998] and mean-shift segmentation [Comaniciu & Meer 2002]). However, these techniques are more computationally consuming compared to the simple technique such as grid partitioning [Lim & Jin 2005]. Therefore, we present here three types of image regions used in our work as depicted in figure 4.3.

- **Pixel sampling** is the basic form of image representation. The idea is to down-sample the image into smaller set of pixels (i.e., image *thumbnail*). Considering equal size rectangles, each image is decomposed into a set of $n \times m$ regular rectangles. For each of these rectangles, only the center pixel is considered to represent the contents of this image region. This method requires less computational effort as the features are already computed. However, large amount of visual information is reduced after the sampling.

- **Grid partitioning** divides the image into $n \times m$ regular rectangles with the same size. For example, we apply a regular grid partitioning resulting in 5×5 sub-windows. This value yields 25 rectangular patches for each

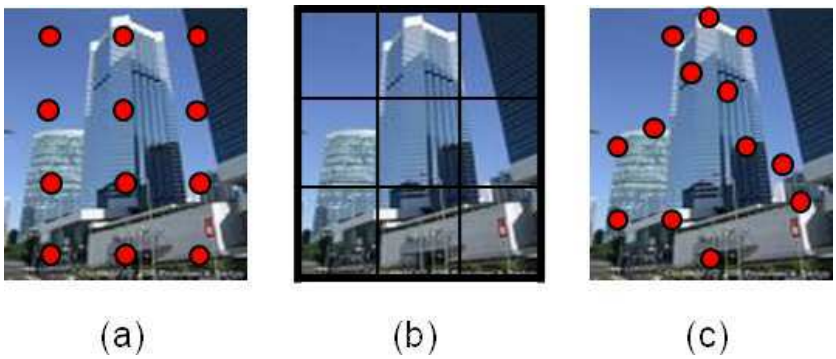

Figure 4.3: Image decomposing into pixel or region: (a) pixel sampling, (b) grid partioning and (c) keypoint detection.

image which is a good tradeoff between the visual content and the patch size. In contrast to the pixel sampling, this method preserves all the visual information contained in the image patches. However, it requires much more effort to extract good visual features from these patches.

- **Keypoint detection** identifies the locations in the image which contain rich information according to some invariant features. These points possess some invariant properties to image transformations (e.g., affine, scale and rotation). The salient points have shown good performances in object recognition with very high accuracy on certain kind of rigid objects (building, car, bicycle, etc.) [Lowe 2004]. Salient points are detected based on the multi-scale filters (e.g., Different of Gaussian (DOG)) in the scale space. The result is a set of keypoints associated with their locations and scales.

  Indeed, invariant keypoints give a good compromise between pixel sampling and grid segmenting for image representation. It not only keeps the important regions of an image but also reduces the computational cost to generate a set of visual feartures.

4.2.2 Feature extraction

The feature extraction step aims at representing each region as a set of feature vectors for clustering purposes. We consider here several visual features (i.e., several points of views) extracted from one pixel or for an image region. We denoted the set of visual features as $\mathcal{F}$, which f is a specific visual feature from $\mathcal{F}$ extracted from an image region.

For the pixel sampling method, each region is represented by its central pixel.

The HSV color value of this pixel can be used as visual feature to represent images. Each pixel is represented by a 3 dimensional vector . We choose to focus on the HSV color space because of its robustness against illumination changes.

For image regions, several visual features can be extracted. Color histograms and edge descriptors [Won *et al.* 2002] are frequently used as visual features for image patches as mentioned in chapter 2. For the keypoint extraction, SIFT descriptors [Lowe 1999] are extracted within a given radius around the keypoint. Note that, we can extract the same visual features (e.g., color and edge histogram) for the keypoint knowing the region covering around this keypoint. The dimensionality for each type of visual feature is summarized in Table 4.1.

Table 4.1: Summary of visual features used for each type of representation.

Feature type f	**Quantization**	**Dimensions**
(H,S,V) value	3 bins	3
HSV histogram	4 x 4 x 4 bins	64
Edge histogram	16 patches x 5 edge types	80
SIFT descriptor	16 patches x 8 orientations	128

4.3 Visual graph modeling

After the image processing step, we obtain a set of visual features extracted from image regions. These features are used for visual concept learning using the unsupervised learning method. These visual concepts, together with the spatial relations, allow us to form the visual graph which better represents the image content. Figure 4.4 shows the pipeline of our graph modeling process.

4.3.1 Visual concept learning

Given a set of features $\mathcal{F}$ extracted from regions or keypoints, the goal of the training stage is to classify these feature vectors into the homogenous groups that can be represented by a set of visual concepts. For this purpose, we apply the k-means algorithm to the pool of feature sets $\mathcal{F}$ and cluster them to the k clusters. The clustering algorithm is applied to the set of feature vectors. The result is a set of numerical label c_i associated with each image region or keypoint. For each visual feature $f \in \mathcal{F}$, a corresponding visual vocabulary $\mathcal{C}_f$ is created. The number of clusters is the number of visual concepts contained in the corresponding visual vocabulary.

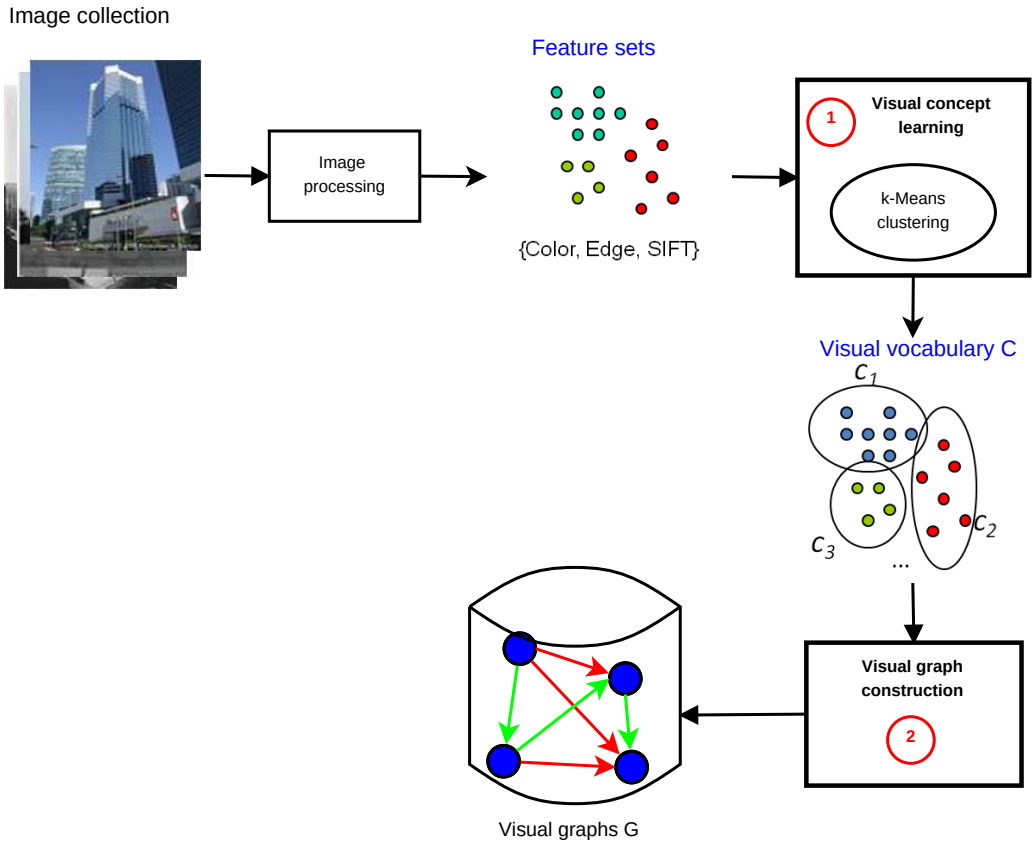

Figure 4.4: Visual graph modeling pipeline for image collection.

The reasons why we choose the $k - means$ clustering method, as presented in chapter 2, for visual concept learning over the other methods, such as *EM clustering* [Moore 1998], are twofold:

- It is a popular technique used in image retrieval thanks to its simple implementation and it requires minimum number of paremeter to operate.

- For language modeling, an important assumption over the visual concepts is that the probability follows the *multinomial distribution* where the random variables have discrete values. Therefore, other segmentation techniques are not valid under this framework.

The visual concept learning step will be discussed further in the future works in chapter 8. Meanwhile, we rely on the common bag-of-words representation.

4.3.2 Visual graph construction

Concept set generation

From the constructed visual vocabulary $\mathcal{C}_f$ and for each image, we will build a set of visual concepts that represents its visual content. First, a set of visual features will be extracted for the corresponding image region, such as pixel, region or keypoint. Then, the next process assigns each image region to the nearest cluster based on the distance of the vector quantizing for this visual feature and the centroid vector of each cluster. Finally, image region is denoted by a corresponding *visual concept* (or *concept* in short) c_i that it has been associated to.

Taking the bridge scene in figure 4.5 as an example, we can see that the visual concepts are assigned to the corresponding patches after the concept set quantization. More precisely, the concept c_1 corresponds to the "tree", while concept c_2 and c_3 are more likely associated with the "bridge". In this way, this image can be denoted by a set of symbolic concepts, for instance $\{c1, c1, c1, c2, c2, c2, c3, c3, c4\}$.

Relation set extraction

Once these visual concepts are defined and characterized independently, the last step is to define the relationships among them. Existing works have suggested the use of topological relations between points [Egenhofer & Herring 1991] or between regions [Boutell *et al.* 2007, Aksoy 2006]. Inspired by these works, we will define the similar relationships between the regions and keypoints. Although, different from the latter approaches, the relation in our context is stricly symbolic in the sense that it does not take into account any relation attributes. The attributed relation will be addressed as part of the future work.

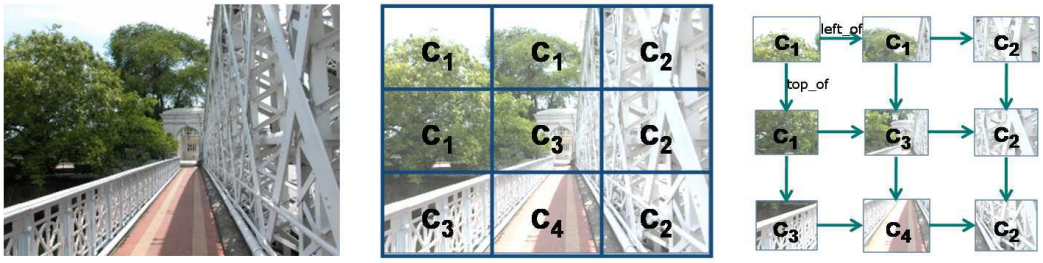

Figure 4.5: Example of spatial relations extracted from image. (a) scene of a bridge, (b) visual concept generation, (c) relations *left_of* and *top_of* extracted from concepts

We denote a labeled relation as l and a set of labeled relations as $\mathcal{L}$. Figure 4.5 gives an example of spatial relations between visual concepts with STOIC collection used in chapter 6. Relation sets $\mathcal{L} = \{left\_of, top\_of\}$ are extracted from the two connected concepts. These relations help to capture the spatial co-occurrence information of two visual concepts. For example, instances of the concept "sky" are usually in the top_of instances of the concept "tree", while instances of concept "tree" appears more frequently in the $left\_of$ instances of concept "bridge". If the number of training images is large enough, the graph framework will capture the statistical consistency for this type of relation.

Similar to the above, we can denote these relation sets using symbolic representation, for example {*(c1,c1,left_of), (c1,c2,left_of), (c1,c3,left_of), (c3,c2,left_of)*}, *(c3,c4,left_of), (c4,c2,left_of)* and {*(c1,c1,top_of), (c1,c3,top_of), (c1,c3,top_of), (c3,c4,top_of)*}, *(c2,c2,top_of), (c2,c2,top_of).*

Graph formulation

At the end of the graph construction procedure, we obtain a set of visual concepts $\mathcal{C}_f$ and a set of predefined relations E_l for each type of concept f and relation l. Each concept is associated with a weight that represents its number of occurrences in the image. Similarly, each relation is also given a weight corresponding to the number of times this relation has occurred in the image. We will denote the weighted concepts set by WC_f and the weighted relations set by WE_l. As we may have several image representations (or point of views) and different kind of spatial relationships between them, we denote a set of weighted concept sets as $S_{WC_\mathcal{F}} = \bigcup_{f \in \mathcal{F}} WC_f$ and a set of weighted relation sets as $S_{WE_\mathcal{L}} = \bigcup_{l \in \mathcal{L}} WE_l$ for an image I.

Given a graph which is represented theoreticaly by a set of nodes and a set of arcs. We map the set of concept sets $S_{WC_\mathcal{F}}$ and the set of relation sets $S_{WE_\mathcal{L}}$ to the set of nodes and to the set of arcs respectively. In our case, we denote this graph as a visual graph $G =< S_{WC_\mathcal{F}}, S_{WE_\mathcal{L}} >$. The weight of concepts and relations are also mapped with the corresponding nodes and arcs. These visual graphs are then stored in the graph database.

By using the graph-based representation, we can include several image representations (i.e., different point of views) into this generic framework. Note that we tend to choose different representations for image regions (i.e., patch, keypoint) and visual features (i.e., color, edge, SIFT) which are considered visually independent of each other to represent image content. Therefore, concept sets WC_f are disjoint. From this stand point, we will make an independent assumption based on the set of weighted concept sets $S_{WC_\mathcal{F}}$. The similar assumption is also applied to weighted relation sets $S_{WE_\mathcal{L}}$. The details of graph formulation will be given in section 5.2 of the next chapter.

4.4 Visual graph retrieval

Once the visual graphs are established based on the concept sets and the relation sets, the next question is how we can compare a new image against the document graphs stored in our database. Figure 4.6 shows the pipeline for our graph retrieval scheme. The retrieval scheme includes three main stages:

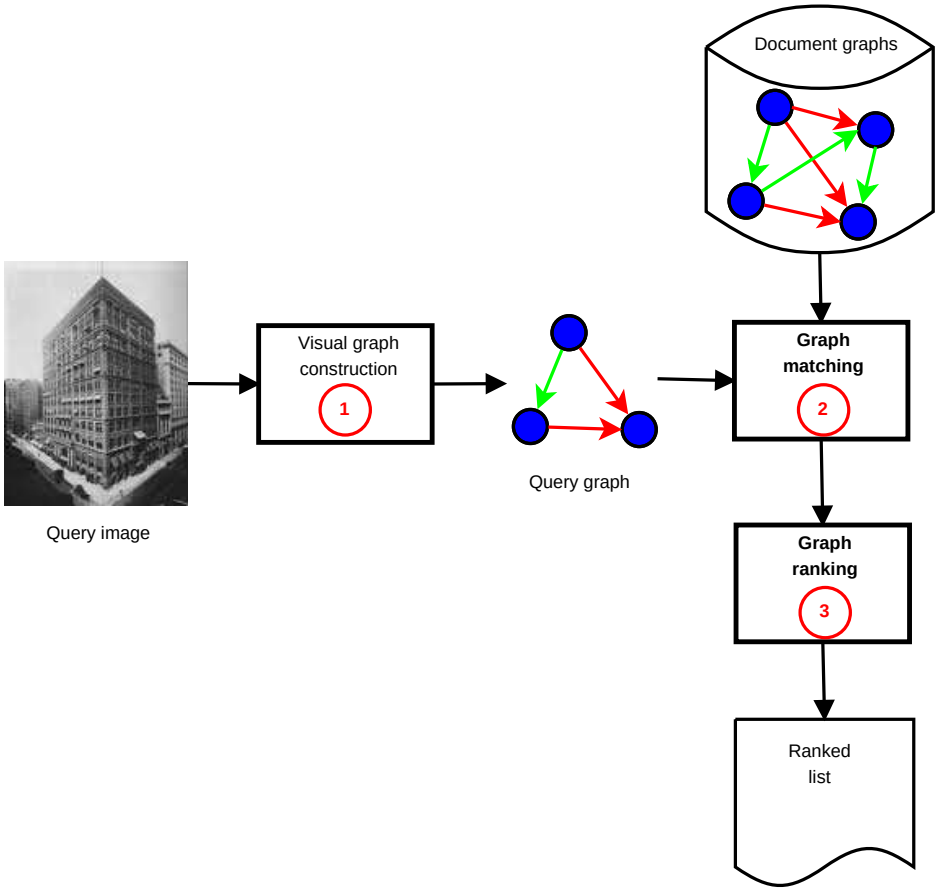

Figure 4.6: Graph retrieval pipeline for a query image

1. Given a query image Iq, we follow the same process of **graph modeling** as above to extract the visual feature and cluster them into visual concepts. The same type of spatial relations are also extracted from the concepts. From these concept sets and relation sets, a visual graph for user image G^{Iq} is automatically generated.

2. **Graph matching** consists of comparing the user query graph G^{Iq} with the trained graphs G^{Id} stored in our database. Inspired by the language model

from the IR domain, which is a generative model, we extend this framework to take into the account of the multiple concepts and multiple relations embedded in our graph-based model. We compute the probabilities of generating of the concept sets and the relation sets assuming concept sets and relation set are independent.

3. **Graph ranking** consists of the computing of relevance status values of the document graph and the query graph in the log-probability space. Finally, document images are ranked based on their relevance values associated with the documents graphs.

4.5 Discussion

In this chapter, we have given an overview of the system implemented in our work. It includes three main layers: the *image processing* layer, the *graph modeling* layer and the *graph matching* layer. The main contributions of this thesis are related to the graph modeling layer and the graph matching layer. These contributions rely on the graph-based presentation of image content and the extension of language modeling for graph matching.

The graph modeling consists steps of visual concepts construction and spatial relation extraction. The visual concepts are learned from the low-level features (e.g., color, edge, and local features) which are computed directly from different type of image representations such as pixels, patches or local keypoints, etc. The visual concepts learning phrase which consists of mapping the low-level image features into a discrete space of the visual vocabulary is done by the *k-means clustering algorithm*. In the other words, the visual concepts represent the middle layer of image representation. To complete the graph-based representation, spatial relations are extracted from the visual concepts. It should be considered that our visual graph model, which adds another layer above the *conceptual layer*, represents the *intermediate layer* of image representation approaching the *semantics layer*.

The graph matching stage consists of generating the probabilities of new graph from the trained graphs in our database. The classical approaches of graph matching are usually complicated and time consuming. Therefore, we would like to address another perspective of graph matching based on the *generative probabilistic framework* of *language modeling*. This approach is simple in term of computational perfomance, as well as a well-founded theory from IR fields. The language model has been successful in the text retrieval domain. Moreover, the extension of this model is straightforward from our graph-based model. The next chapter will detail our proposed approach for image retrieval.

Chapter 5

Visual Graph Modeling and Retrieval

5.1 Introduction

In the previous chapter, we have presented an overview of our graph-based system. This system composes of three main stages: the *image processing step*, the *visual graph modeling step* and the *visual graph retrieval step*. The image processing step provides the tool for extracting the low-level visual features (such as color, texture or edge). The visual graph modeling step includes two processes. First, it automatically induces a set of visual concept from a set of visual features based on the unsupervised learning algorithm, e.g., k-means clustering. Second, the relation extractor generates a set of spatial relations from the constructed visual concepts. Finally, a visual graph is formulated from these set of concepts and set of relations. The visual graph retrieval process consists of matching the query image graph with the graphs stored in the database and ranks the results using their probability values.

The goal of this chapter is to define formally the visual graph model and to describe the matching process based on the formalism of language modeling. We will show some examples of graph instance derived from the general graph model. As we have shown in the chapter 3, the main bottleneck of using the graph-based image representation is the matching step. In the literature, graph matching with classical algorithm is a costly process. To avoid that problem and to provide a more reliable matching algorithm, we rely on the idea of language modeling for generating the query graph from the document graphs in the database. Graphs are then ranked with their corresponding probability likelihood values. As a consequence, images are ranked in the lists with the same order of their relevance values. As we will show in chapter 6, the proposed graph model may also be used

for image categorization. The category of a query image is decided based on the class of the image which maximizes this probability likelihood.

This chapter is structured into 5 sections. Section 5.2 introduces a formal definition of our visual graph. Also, we give some examples of graph instances used in our application. Section 5.3 will show how visual graphs are matched using our extended language modeling framework. We also give an example on how we compute the probability likelihood in section 5.3.4. Then, section 5.4 discusses how we actually rank our graph retrieval results using the relevance status value. Finally, we conclude the chapter and give some insight discussions in section 5.5.

5.2 Visual graph formulation

5.2.1 Definition

In this section, we introduce a set of formal definitions associated with the visual graph. To facilitate reading, the following notations will be used:

$$\begin{cases} I \text{ : an image} \\ G^I \text{ : visual graph for image } I \\ \mathcal{F} \text{ : set of visual features associated with an image region} \\ f \text{ : a low-level visual feature, } f \in \mathcal{F} \\ \mathcal{L} \text{ : set of possible labeled relations} \\ l \text{ : label of relation, } l \in \mathcal{L} \\ \mathcal{C}_f \text{ : set of concepts (or visual vocabulary) extracted for a feature } f \\ c \text{ : a visual concept, } c \in \mathcal{C}_f \\ E_l \text{ : set of concept pairs extracted for a relation labeled } l \\ c, c', l \text{ : a labeled relation, } c \in \mathcal{C}_f, c' \in \mathcal{C}_{f'}, l \in \mathcal{L} \\ WC_f \text{ : weighted concept set} \\ WE_l \text{ : weighted relation set} \\ S_{WC_\mathcal{F}} \text{ : set of weighted concept set } WC_f \\ S_{WE_\mathcal{L}} \text{ : set of weighted relation set } WE_l \end{cases}$$

Our visual graph may contain different sets of visual concepts sets and different sets of relation sets, which reflects multiple points of views (i.e., image decompositions and low-level visual features). Each visual concept is constructed for each type of image region and its low-level feature from the collection as in section 4.2. Figure 5.1 shows how we formulate the set of concept set and the set of relation set from a collection C and then transfer them to the formulation of image I.

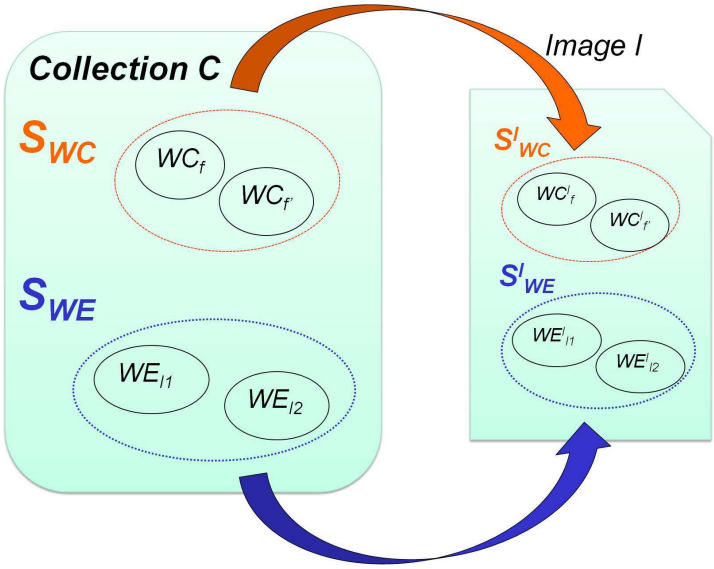

Figure 5.1: Formulation of the set of concept set and set of relation set from the image collection C and for the image I.

Definition 1 Let $\mathcal{F}$ be the set of low-level features. For each feature $f \in \mathcal{F}$, C_f^I is the concept set extracted for feature f of an image I:

$$C_f^I = \{c | c \in \mathcal{C}_f, I\}$$

where c is a visual concept that characterizes image I. Assuming the concept set independent hypothesis, we have:

$$\bigcap_{f \in \mathcal{F}} C_f^I = \emptyset$$

Definition 2 Given the concept set C_f^I, WC_f^I denotes a weighted concept set which contains a set of pairs $(c, \#(c, I))$:

$$WC_f^I = \{(c, \#(c, I)) | c \in C_f^I\}$$

where $\#(c, I)$ is the number of times a visual concept c occurs in the image I. The weight of concept captures the importance of this concept in the image. By default, WC_f^I captures only the visual concept c that appears in the image, which means $\#(c, I) > 0$.

Definition 3 The set of weighted concept sets $S^I_{WC_\mathcal{F}}$ is a union of weighted concept sets WC_f^I:

$$S^I_{WC_\mathcal{F}} = \bigcup_{f \in \mathcal{F}} WC_f^I$$

Assuming the concept set independent hypothesis, the weighted concept sets are disjoint. We get:

$$\bigcap_{f \in \mathcal{F}} WC_f^I = \emptyset$$

Similarly, we define the relation sets extracted from the visual concepts for an image I as follows.

Definition 5 Let $\mathcal{L}$ be the set of the possible labels. For each labeled relation $l \in \mathcal{L}$, the relation set E_l^I is defined by:

$$E_l^I = \{((c, c'), l) | (c, c') \in C_f^I \times C_{f'}^I, l \in \mathcal{L}\}$$

where (c, c') is a pair of concept extracted from two concept sets C_f^I and $C_{f'}^I$, and l is a relation that occurs in the image I.

If a pair of concepts (c, c') comes from the same concept set (i.e., $C_f^I = C_{f'}^I$), we refer this relation set as *intra-relation* set. Otherwise, if it comes from two concept sets extracted from different visual features (i.e., $C_f^I \neq C_{f'}^I$), we refer this relation set as *inter-relation* set.

Assuming the relation set independent hypothesis, we have:

$$\bigcap_{l \in \mathcal{L}} E_l^I = \emptyset$$

Definition 6 Given relation set E_l^I, WE_l^I denotes a weighted relation set which represented by a set of triplet $((c, c'), l, \#(c, c', l, I))$:

$$WE_l^I = \{((c, c'), l, \#(c, c', l, I)) | (c, c') \in C_f^I \times C_{f'}^I, l \in \mathcal{L}\}$$

where $\#(c, c', l, I)$ is the number of times c and c' are related with label l in image I. The weight of relation signifies the relation importance as the frequency appeared in the image. By default, we capture only the relation that appears in the image I, therefore, $\#(c, c', l, I) > 0$.

Definition 7 The set of weighted relation sets $S_{WE_\mathcal{L}}^I$ is a union of weighted relation sets WE_l^I:

$$S_{WE_\mathcal{L}}^I = \bigcup_{l \in \mathcal{L}} WE_l^I$$

Assuming the relation set independent hypothesis, the relation sets are disjoint. We have:

$$\bigcap_{l \in \mathcal{L}} WE_l^I = \emptyset$$

Definition *3* and *7* are important as they provide the generality of our visual graph model. As pointed out earlier, the visual graph can integrate smoothly different image points of views, as well as the relations among them. Moreover, extracting visual concepts and spatial relation from image content is a difficult task (like in text retrieval domain). Visual concepts and relations are sometimes defined in a subjective way. Therefore, the independence hypotheses based on the concept sets and relation sets have been stated clearly to facilitate our proposition of graph retrieval in the next section.

Finally, the set of concept sets $S^I_{WC_\mathcal{F}}$ and the set of relation sets $S^I_{WE_\mathcal{L}}$ are mapped to the set of nodes and the set of arcs respectively, in our graph-based framework. The following is the definition of visual graph for image I.

Definition 9 Given a set of weighted concept sets $S^I_{WC_\mathcal{F}}$ and a set of weighted relation sets $S^I_{WE_\mathcal{L}}$ for an image I, the visual graph G^I is defined by:

$$G^I =< S^I_{WC_\mathcal{F}}, S^I_{WE_\mathcal{L}} >$$

The definition of our visual graph model provides a general framework which allows us to derive to different graph instances. Depending on the visual contents of image, for example: outdoor scenary, building or indoor photos, we can create different visual graph instances to fit the image content in this graph framework. In the next section, we will present two graph examples used in our experiments.

5.2.2 Graph instance 1

In this section, we illustrate how the visual graph model is constructed from one concept set and two relation sets. This graph instance is used in our experiment with the STOIC-101 image collection. Most of the photos are captured famous scenes of Singapore landmarks and are mostly outdoors. We build the concept set based on the patch-based division and extract the color information from these patches. We denote this concept set as $\mathcal{C}_{color} = \{c1, c2, c3, ..., cN\}$, where N are the number of concept defined for this visual vocabulary.

Nevertheless, spatial information embedded in these photos is an important factor. Firstly, these photos were taken for touristic purposes. Most of the photos are centered with the main object. Second, images are mostly in portrait or landscape mode. Therefore, the vertical and horizontal information are very useful clues for recognizing of the image content. For example, the sky is above the buildings, trees are next to the statue, river is below the bridge etc. For these reasons, we decide to choose two type of spatial relations, denoted $\mathcal{L} = \{left\_of, top\_of\}$, extracted from the concept set $\mathcal{C}_{color}$. The *left_of* indicates

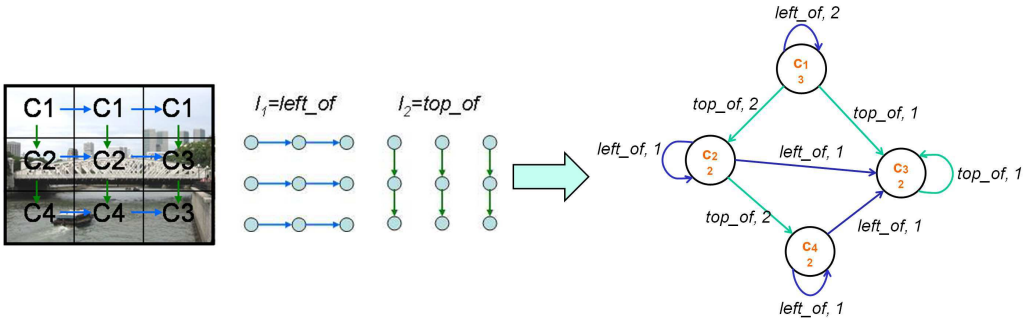

Figure 5.2: Example of a visual graph extracted from STOIC image collection. Concepts are represented by nodes and spatial relations are expressed by directed arcs. Nodes and arcs are weighted by the number of times they appear in the image.

the horizontal relation between a pair of concepts. In the same way, the *top_of* indicates the vertical relation between a pair of concepts.

Figure 5.2 shows an example of the visual graph constructed from an image of a bridge scene. This example corresponds to a visual graph containing one visual concept set $\mathcal{C}_{color}$ and two intra-relation sets E_{left_of} and E_{top_of}. The visual graph for an image $G^I =< S^I_{WC_\mathcal{F}}, S^I_{WE_\mathcal{L}} >$ is composed of:

- The set of concept set contains one weighted concept set WC_{color} extracted from color feature, denoted $S^I_{WC_\mathcal{L}} = WC_{color}$. In the figure, each node corresponds to a concept and the number of time it occurs in the image. For example, concept $c1$ appeared 3 times in the image and is denoted by $(c1, 3)$, concept $c2$ appeared 2 times in the image and is denoted by $(c2, 2)$ etc.

- The set of relation set $S^I_{WE_\mathcal{L}}$ contains two intra-relation sets E_{left_of} and E_{top_of} extracted from two spatial relations $l1 = left\_of$ and $l2 = top\_of$, denoted by $S^I_{WE_\mathcal{L}} = WE_{left\_of} \bigcup WE_{top\_of}$. The relation between a couple of concepts is captured by the directed arcs in this graph. Precisely, the blue arcs express the relation $left\_of$ and the green arcs express the relation top_of between two concepts. For example, concept $c1$ is related to concept $c2$ with the relation top_of 2 times and is related to itself by the relation $left\_of$ 2 times. It is denoted by $(c1, c2, top\_of, 2)$ and $(c1, c1, left\_of, 2)$.

5.2.3 Graph instance 2

The second example is a graph instance extracted from two concept sets and one relation set. The idea is to integrate different image representations

(such as patch division and keypoints as described in chapter 2) together with their relations to represent a unique view of image contents. Image patches are characterized by a set of visual features, such as color features and edge features. These features capture the global information of the objects. On the other hand, the visual feature extracted from keypoints represent the details of the objects. We denote two concept sets as $\mathcal{C}_{patch} = \{p1, p2, p3, ..., pN\}$ and $\mathcal{C}_{sift} = \{s1, s2, s3, ..., sM\}$, where N, M are the number of visual concepts defined for each visual vocabulary. The co-occurrence information between two concept set is denoted by the relation $\mathcal{L} = \{inside\}$ if one keypoint is localized inside the area of an image patch.

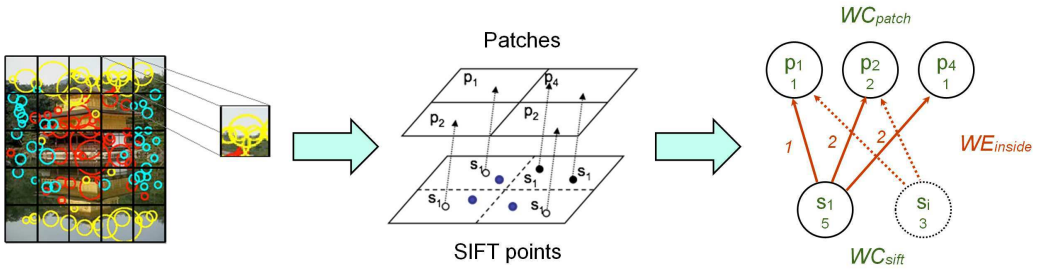

Figure 5.3: Example of a visual graph extracted from two different visual concept sets and related by a set of relation $inside$.

Figure 5.3 shows an example of an image and the corresponding visual graph constructed. The above graph example corresponds to a visual graph containing two visual concept sets $\mathcal{C}_{patch}$, $\mathcal{C}_{sift}$ and one inter-relation set E_{inside}. The visual graph for an image $G^I =< S^I_{WC_\mathcal{F}}, S^I_{WE_\mathcal{L}} >$ is then composed of:

- The set of concept set contains two weighted concept sets WC_{patch} and WC_{sift} extracted from color and edge features and SIFT features, denoted $S^I_{WC_\mathcal{F}} = WC_{patch} \bigcup WC_{sift}$. In the figure, the set of node above corresponds to a concept set $\mathcal{C}_{patch}$ and the set of node below corresponds to a concept set $\mathcal{C}_{sift}$. For each node, the weight is calculated by the number of time its occurrence in the image. For example, concept $s1$ of concept set $\mathcal{C}_{sift}$ appeared 5 times in the image and is denoted by $(s1, 5)$. Concept $p2$ of concept set $\mathcal{C}_{patch}$ appeared 2 times in the image and is denoted by $(p2, 2)$

- The set of relation set contains a weighted inter-relation sets WE_{inside} extracted from two concept sets $\mathcal{C}_{patch}$ and $\mathcal{C}_{sift}$, denoted $S^I_{WE_\mathcal{L}} = WE_{inside}$. Similar to above, the relation between a couple of concepts is also captured by the directed arcs in this graph. For example, concept $s1$ is linked to concept $p1$ with the relation $inside$ for 1 times and is related to concept

$p2$ for 2 times. These relations are denoted by $(s1, p1, inside, 1)$ and $(s1, p2, inside, 2)$.

After defining the representation for graphs, we turn to the problem of matching a query graph with the document graphs. The language model defined over the conceptual graph proposed in [Maisonnasse *et al.* 2009] is considered only one concept set and one relation set in the same time. In the next section we will explain how we extend this framework to take into account of multiple concept sets and multiple relation sets in our visual graph matching method.

5.3 Graph matching for image retrieval

For a new image query, our objective is to provide the relevant photos from the collection that match the user needs. Inspired by the information retrieval theory, we define here three models:

- **Document graph model** G^{Id} is extracted from the document image Id in the collection.
- **Query graph model** G^{Iq} is constructed for a new query image Iq with the same configuration as the document graph model.
- **Matching model** $\mathcal{M}(G^{Iq}, G^{Id})$ includes a ranking function that computes the probability for generating query graph model G^{Iq} from the document graph model G^{Id}.

Figure 5.4 provides the common diagram for image indexing and retrieval process in our graph-based model. First, we generate for the set of image documents in the collection a set of corresponding graphs. These models are then stored in our database for matching purpose. A graph model is also constructed for the image query. Then, the matching model takes into account of both query graph model and document graph model to compute the similarity of these graph models. Finally, images are ranked based on their similarity values of the document graph models.

5.3.1 Query likelihood ranking

Inspired by the language modeling approach proposed in Section 3.3, the matching model $\mathcal{M}(G^{Iq}, G^{Id})$ ranks the image documents based on the probability of generating the query graph model from the document graph models, which is defined by:

$$\mathcal{M}(G^{Iq}, G^{Id}) = rank_{Id \in C}\{P(G^{Iq}|G^{Id}), Id\} \tag{5.1}$$

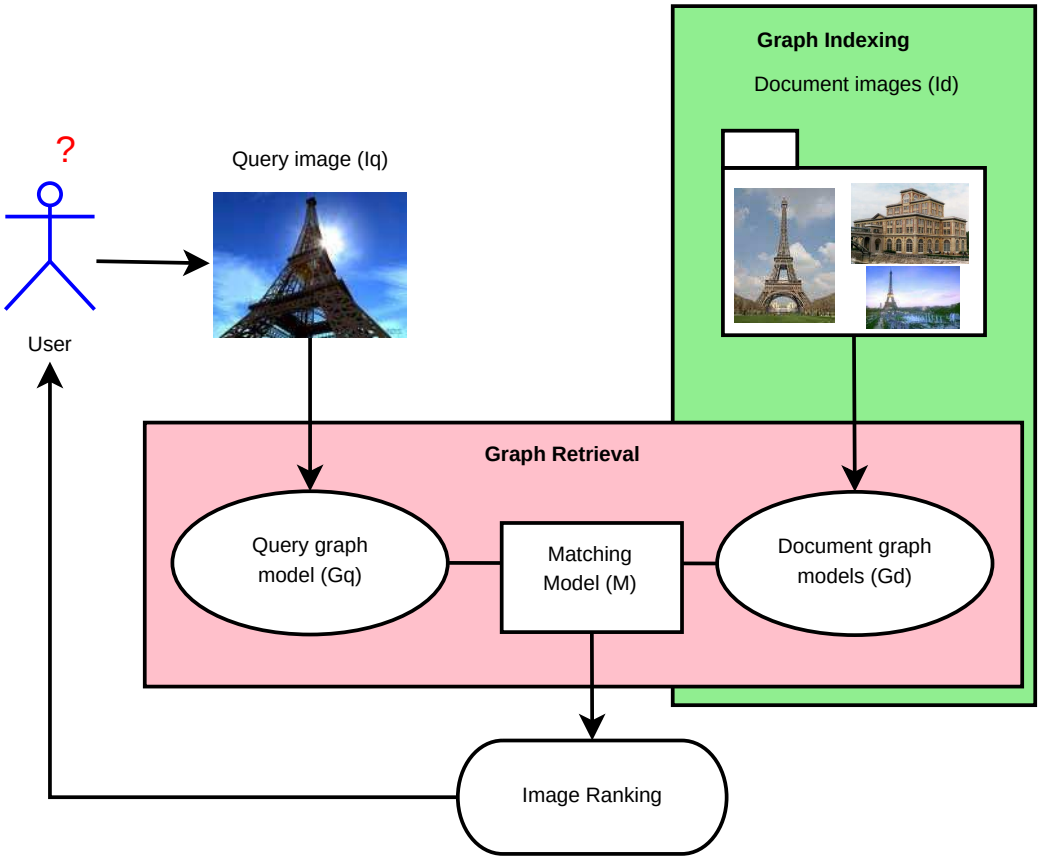

Figure 5.4: Flowchart for graph indexing and graph retrieval in our system.

This is a generative model in the sense that the probability of query graph generation is measured based on the probability taken from the document graph model. The probability $P(G^{Iq}|G^{Id})$ indicates how likely the document graph G^{Id} is close to the query graph G^{Iq}. In [Maisonnasse *et al.* 2009], the probability P has been calculated for a conceptual unigram model. However, our graph model is composed of multiple concept sets, as well as, multiple relation sets. To expand this framework, we present here an extension of the matching model $\mathcal{M}$ that handles both set of concept sets and set of relation sets.

In other words, the probability for a query graph model $G^{Iq} =< S^{Iq}_{WC_\mathcal{F}}, S^{Iq}_{WE_\mathcal{L}} >$ to be generated from a document graph G^{Id} is composed of the probability of generating independently the set of concept set and the set of relation set. Using the conditional probability rule, this can be written as:

$$\begin{aligned} P(G^{Iq}|G^{Id}) &= P(S^{Iq}_{WC_\mathcal{F}}, S^{Iq}_{WE_\mathcal{L}}|G^{Id}) \\ &= P(S^{Iq}_{WC_\mathcal{F}}|G^{Id}) \times P(S^{Iq}_{WE_\mathcal{L}}|S^{Iq}_{WC_\mathcal{F}}, G^{Id}) \end{aligned} \tag{5.2}$$

where $P(S^{Iq}_{WC_f}|G^{Id})$ is the probability of generating set of concept sets from the document graph, and $P(S^{Iq}_{WE_l}|S^{Iq}_{WC_f}, G^{Id})$ is the probability of generating set of relation set given the set of concept set $S^{Iq}_{WC_f}$ from the document graph G^{Id}. To calculate this score, we will estimate separately the two probabilities: the probability of generating concept set and probability of generating relation set.

5.3.2 Matching of weighted concept set

For generating the probability of query concept sets from the document model $P(S^{Iq}_{WC_f}|G^{Id})$, we assume a concept set independence hypothesis (related explanation in section 5.2). The probability can thus be estimated as:

$$P(S^{Iq}_{WC_{\mathcal{F}}}|G^{Id}) = \prod_{WC_f{}^{Iq} \in S^{Iq}_{WC_{\mathcal{F}}}} P(WC_f{}^{Iq}|G^{Id}) \tag{5.3}$$

Assuming *concept independence* which is standard in information retrieval, the number of occurrences of the concepts (i.e., the weights considered previously) are integrated through the use of a *multinomial* distribution model. We compute $P(WC_f{}^{Iq}|G^{Id})$ as follows:

$$P(WC_f{}^{Iq}|G^{Id}) \quad = \quad \frac{(\sum_c \#(c, Iq))!}{\prod_c \#(c, Iq)!} \prod_{c \in \mathcal{C}_f} P(c|G^{Id})^{\#(c,Iq)}$$

where $\#(c, Iq)$ denotes the number of times concept c occurs in the query graph G^{Iq}. This contribution corresponds to the unigram conceptual probability as proposed in [Maisonnasse *et al.* 2009].

The proportion in the above equation will not affect the ranking of the document images for the given query image Iq. By omitting the constant value α, the equation leads to:

$$P(WC_f{}^{Iq}|G^{Id}) \quad \propto \quad \prod_{c \in \mathcal{C}_f} P(c|G^{Id})^{\#(c,Iq)} \tag{5.4}$$

To calculate this score, we need to estimate the probabilities $P(c|G^{Id)}$ from the document graph model. The maximum likelihood estimate would be:

$$P(c|G^{Id}) = \frac{\#(c, Id)}{\#(|D|, Id)}$$

where the quantity $\#(c, Id)$ represents the number of times c occurs in the document image Id. The quantity $\#(|D|, Id)$ is the total number of concept in the document image and equal to $\sum_c \#(c, Id)$.

The major problem with this probability estimation is that if any of the concepts in the query image is missing from the document, the probability of $P(WC_f{}^{Iq}|G^{Id})$ will be *zero* (as refered in section 3.3). Consequently, the score given by the query likelihood $P(G^{Iq}|G^{Id})$ becomes *zero* probability. This is clearly not appropriate for ranking this type of image while it should have a small value instead. To avoid this bottleneck, the quantity $P(c|G^{Id})$ is estimated through maximum likelihood using *Jelinek-Mercer smoothing*:

$$\begin{aligned} P(c|G^{Id}) &= (1-\lambda_f)P(c|G^{Id}) + \lambda_f P(c|C) \\ &= (1-\lambda_f)\frac{\#(c,Id)}{\#(|D|,Id)} + \lambda_f \frac{\#(c,C)}{\#(|C|,C)} \end{aligned} \tag{5.5}$$

where λ_f is the smoothing parameter for each concept set $\mathcal{C}_f$. Similarly, the quantities $\#(c,C)$ and $\#(|C|,C)$ are defined over the whole collection C (i.e., over the union of all images in the collection).

We choose to use the *Jelinek-Mercer smoothing* for its simplicity and its proved effectiveness in the text retrieval domain. In general, the parameter λ_f can depend on the nature of the document such as the used visual feature, image category, etc. In our case, this parameter can be optimized using a validation set or a cross-validation technique.

In order to respect the consistency of multinomial distribution, the probabilities of the concepts must sum to one, i.e. $\sum_c P(c|G^{Id}) = 1$. It's been proved that the *Jelinek-Mercer smoothing* respects this prior condition. To illustrate, we consider a simple example with only 3 concepts, $c1, c2$ and $c3$, in the concept set. Supposing that the probabilities for these concepts in the document graph are 0.6, 0.4 and 0.0. The probabilities estimated for these concepts in the collection are 0.3, 0.5 and 0.2. Given a smoothing value, for example $\lambda_f = 0.2$, the smoothed probability for the document graph are:

$$\begin{aligned} P(c1|G^{Id}) &= 0.6 \times (1-\lambda_f) + 0.3 \times \lambda_f = 0.54 \\ P(c2|G^{Id}) &= 0.4 \times (1-\lambda_f) + 0.5 \times \lambda_f = 0.42 \\ P(c3|G^{Id}) &= 0.0 \times (1-\lambda_f) + 0.2 \times \lambda_f = 0.04 \end{aligned}$$

Note that concept $c3$ has zero probability. Even though, the smoothed probability of concept $c3$ has non-zero score thanks to the background probability estimated from the collection. In the end, we get:

$$\begin{aligned} \sum_c P(c|G^{Id}) &= P(c1|G^{Id}) + P(c2|G^{Id}) + P(c3|G^{Id}) \\ &= 0.54 + 0.42 + 0.04 \\ &= 1.0 \end{aligned}$$

which confirms that the probabilities using Jelinek-Mercer smoothing method are consistent.

5.3.3 Matching of weighted relation set

As shown in the previous section, we follow a similar process for generating the probability of the relation sets from document graph model. Assuming the relation set independence (cf. section 5.2), this leads to:

$$P(S^{Iq}_{WE_{\mathcal{L}}}|S^{Iq}_{WC_{\mathcal{F}}}, G^{Id}) = \prod_{WE^{Iq}_l \in S^{Iq}_{WE_{\mathcal{L}}}} P(WE^{Iq}_l|S^{Iq}_{WC_{\mathcal{F}}}, G^{Id}) \tag{5.6}$$

For the probability of generating query relation set from the document graph, we assume that a relation depends only on the two linked concept sets. Assuming that the relation sets are conditionally independent according to the set of concept set $S^{Iq}_{WC_{\mathcal{F}}}$ and the graph document G^{Id}, and following a *multinomial distribution* model, we can compute:

$$P(WE^{Iq}_l|S^{Iq}_{WC_{\mathcal{F}}}, G^{Id}) = \frac{(\sum_{(c,c',l)} \#(c,c',l,Iq))!}{\prod_{(c,c',l)} \#(c,c',l,Iq)!} \times \prod_{(c,c',l)\in \mathcal{C}_f \times \mathcal{C}_{f'} \times \mathcal{L}} P(L(c,c') = l|WC^{Iq}_f, WC^{Iq}_{f'}, G^{Id})^{\#(c,c',l,Iq)}$$

where the quantity $\#(c,c',l,Iq)$ is the number of time the relation l of concept pair (c,c') appears in the query graph G^{Iq}.

Similar to the concept set, the first proportion in the above equation will not affect the final ranking. By eliminating the constant value, this equation leads to:

$$P(WE^{Iq}_l|S^{Iq}_{WC_f}, G^{Id}) \propto \prod_{(c,c',l)\in \mathcal{C}_f \times \mathcal{C}_{f'} \times \mathcal{L}} P(L(c,c') = l|WC^{Iq}_f, WC^{Iq}_{f'}, G^{Id})^{\#(c,c',l,Iq)} \tag{5.7}$$

where $c \in \mathcal{C}_f$, $c' \in \mathcal{C}_{f'}$ and $L(c,c')$ are variables which values in $\mathcal{L}$ reflects the possible relation labels between c and c', in this relation set.

Similar to the concept set, the relation set suffers the same problem of zero probability when a relation is missing from the document graph. Hence, the smoothing technique has been applied for the relation set. The probabilities $P(L(c,c') = l|WC^{Iq}_f, WC^{Iq}_{f'}, G^{Id})$ are estimated by maximum likelihood with *Jelinek-Mercer smoothing* method, giving:

$$P(L(c,c') = l|WC^{Iq}_f, WC^{Iq}_{f'}, G^{Id}) = (1-\lambda_l)\frac{\#(c,c',l,Id)}{\#(c,c',|D|,Id)} + \lambda_l \frac{\#(c,c',l,C)}{\#(c,c',|C|,C)} \tag{5.8}$$

where λ_l is the smoothing parameter for each relation set E_l. The quantity $\#(c, c', l, Id)$ represents the number of times concepts c and c' are linked with label l in the document image Id, and the quantity $\#(c, c', |D|, Id)$ is equal to $\sum_{l \in \mathcal{L}} \#(c, c', l, Id)$. In the same way, this parameter can be optimized by the cross-validation technique for each relation. Note that if one of the two concepts does not appear in the image d, it yields:

$$\frac{\#(c, c', l, Id)}{\#(c, c', |D|, Id)} = 0$$

Again, the quantities $\#(c, c', l, C)$ and $\#(c, c', |C|, C)$ are counted in a similar way but computed on the whole collection C (i.e., over the union of all the graphs from all the documents in the collection).

This graph model is a generalization of the model defined in [Pham *et al.* 2010] which corresponds to the case where only one concept set is used. In some special cases, our model corresponds to the standard language model used in [Pham *et al.* 2009] where relations are not considered (i.e., documents and queries correspond to multiple bag-of-words model). In the next section, we will give an example of graph matching with our graph models.

5.3.4 Graph matching example

For a better understanding of the graph matching function, we provide here a simple example of matching with three graph models. These graphs are generated with the graph instance 1 introduced in the previous section. In the figure 5.5, $G1$ and $G2$ are the document graphs and G is the query graph. Intuitively, graph $G1$ is closer to the query graph G than the latter. $G1$ contains the same set of concept and have similar relation set as query graph G. While, $G2$ is missing a concept $c4$ and have less relation than the query graph G. The collection C is then defined on two graphs $G1$ and $G2$.

Supposing that the visual graphs are constructed from 4 visual concepts $(c1, c2, c3, c4)$ and two relations $(l1 = left\_of, l2 = top\_of)$. Graph models

are described as follows:

$$
\begin{aligned}
G1 = < & \{(c1,3),(c2,3),(c3,2),(c4,1)\}; \\
& \{(c1,c1,l1,2),(c2,c3,l1,2),(c4,c3,l1,2), \\
& (c1,c2,l2,1),(c2,c4,l2,1),(c1,c3,l2,2),(c3,c3,l2,2)\} > \\
G2 = < & \{(c1,4),(c2,3),(c3,2)\}; \\
& \{(c2,c3,l1,2),(c2,c2,l1,4),(c1,c2,l2,2),(c1,c3,l2,2),(c3,c3,l2,2)\} > \\
G = < & \{(c1,3),(c2,2),(c3,2),(c4,2)\}; \\
& \{(c1,c1,l1,2),(c2,c2,l1,1),(c2,c3,l1,1),(c4,c3,l1,2), \\
& (c1,c2,l2,2),(c2,c4,l2,2),(c1,c3,l2,1),(c3,c3,l2,1)\} >
\end{aligned}
$$

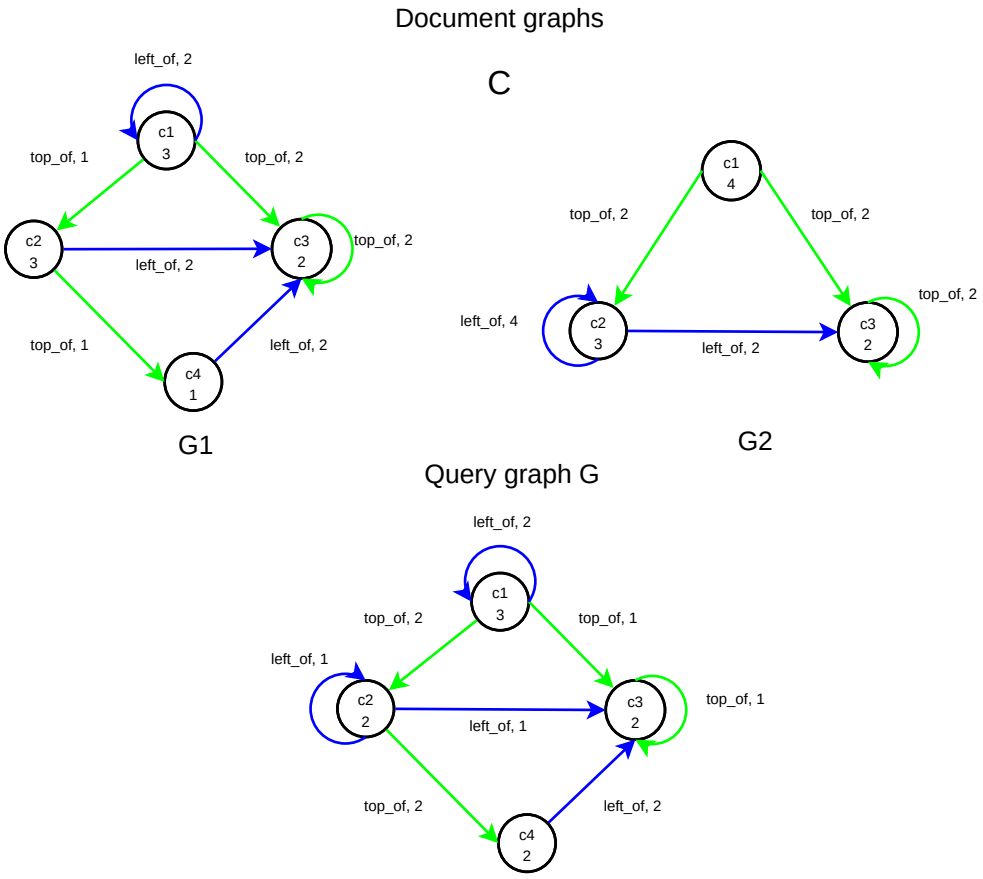

Figure 5.5: Example of matching graph with our model. $G1$ and $G2$ are the document graphs from collection C and G is the query graph.

Assuming the multinomial distribution, we calculate the probability likelihood

for query graph to be generated by the document graph $G1$:

$$\begin{aligned}P(G|G1) =& P(c1|G1)^3 P(c2|G1)^2 P(c3|G1)^2 P(c4|G1)^2 \times \\ & P(L(c1,c1) = l1|G1)^2 P(L(c2,c2) = l1|G1) P(L(c2,c3) = l1|G1) \\ & P(L(c4,c3) = l1|G1)^2 P(L(c1,c2) = l2|G1)^2 P(L(c2,c4) = l2|G1)^2 \\ & P(L(c3,c3) = l2|G1) P(L(c1,c3) = l2|G1)\end{aligned}$$

Assuming a small value of the smoothing parameters (i.e., $\lambda_f = \lambda_l = 0.2$), the probabilities are estimated with the Jenlinek-Mercer smoothing method. We calculate the probabilities for each concept, for example $P(c1|G1)$, as follows:

$$\begin{aligned}P(c1|G1) &= (1 - 0.2) \times P(c1|G1) + 0.2 \times P(c1|C) \\ &= 0.8 \times 3/9 + 0.2 \times 7/18 \\ &= 0.344\end{aligned}$$

For estimating the probabilities of the relations, for example $P(c1, c1, l1|G1)$ and $P(c1, c2, l2|G1)$, we have:

$$\begin{aligned}P(L(c1,c1) = l1|G1) &= (1 - 0.2) \times P(L(c1,c1) = l1|G1) + 0.2 \times P(L(c1,c1) = l1|C) \\ &= 0.8 \times 2/6 + 0.2 \times 2/12 \\ &= 0.3 \\ P(L(c1,c2) = l2|G1) &= (1 - 0.2) \times P(L(c1,c2) = l2|G1) + 0.2 \times P(L(c1,c2) = l2|C) \\ &= 0.8 \times 2/6 + 0.2 \times 2/12 \\ &= 0.183\end{aligned}$$

Note that the relation $(c2, c2, l1|G1)$ does not appear in the document graph G1. Although, it still obtain a non-zero probability of 0.067 which has been leveraged from the collection. The final score for graph G1 is:

$$\begin{aligned}P(G|G1) =& (0.344)^3 (1/3)^2 (2/9)^2 (0.1)^2 \times (0.3)^2 (0.067)(1/3)(0.3)^2 \times \\ & (0.183)^2 (0.15)^2 (1/3)(1/3) \\ P(G|G1) \approx& 3.377 \times 10^{-14}\end{aligned}$$

Similar to the graph $G1$, we calculate the probability likelihood for query graph to be generated by the document graph $G2$:

$$\begin{aligned}P(G|G2) =& P(c1|G2)^3 P(c2|G2)^2 P(c3|G2)^2 P(c4|G2)^2 \times \\ & P(L(c1,c1) = l1|G2)^2 P(L(c2,c2) = l1|G2) P(L(c2,c3) = l1|G2) \\ & P(L(c4,c3) = l1|G2)^2 P(L(c1,c2) = l2|G2)^2 P(L(c2,c4) = l2|G2)^2 \\ & P(L(c3,c3) = l2|G2) P(L(c1,c3) = l2|G2)\end{aligned}$$

Also note that the concept $c4$ does not appear in the document graph $G1$. However it still get a small value of 0.0111 from the collection. The final score for graph $G2$ is:

$$\begin{aligned} P(G|G2) =& (0.433)^3(1/3)^2(2/9)^2(0.011)^2 \times (0.033)^2(0.6)(0.6)(0.033)^2 \times \\ & (0.316)^2(0.017)^2(1/3)(1/3) \\ P(G|G2) \approx& 6.464 \times 10^{-20} \end{aligned}$$

As we can observe from the above scores, graph $G1$ obtains a larger score value compared to graph $G2$, which is consistent with our initial intuition.

5.4 Ranking with relevance status value

In practice, multiplying many small numbers may lead to numerical precision problems. Moreover, the multiply operation is considered more costly when compared to the addition operation. As usual in IR, we can use logarithm function to turn the likelihood score into rank-preserving sum. As done in [Pham *et al.* 2009], the relevance status value (RSV) of a document image Id for query image Iq is computed in the log-probability domain. In the context of multinomial distributions, we have:

$$\begin{aligned} RSV_{log}(G^{Iq}|G^{Id}) &= \log P(G^{Iq}|G^{Id}) \\ &= \log(P(S^{Iq}_{WC_{\mathcal{F}}}|G^{Id}) \times P(S^{Iq}_{WE_{\mathcal{L}}}|S^{Iq}_{WC_{\mathcal{F}}}, G^{Id})) \\ &= \log P(S^{Iq}_{WC_{\mathcal{F}}}|G^{Id}) + \log P(S^{Iq}_{WE_{\mathcal{L}}}|S^{Iq}_{WC_{\mathcal{F}}}, G^{Id}) \end{aligned}$$

By submitting the probabilities of the set of concept set and the set of relation set in equation with the equations 5.3 and 5.6, it leads to:

$$\begin{aligned} RSV_{log}(G^{Iq}|G^{Id}) = & \sum_{WC_{\mathcal{F}}{}^{Iq} \in S^{Iq}_{WC_{\mathcal{F}}}} \log P(WC_f{}^{Iq}|G^{Id}) + \\ & \sum_{WE_l^{Iq} \in S^{Iq}_{WE_{\mathcal{L}}}} \log P(WE_l^{Iq}|S^{Iq}_{WC_{\mathcal{F}}}, G^{Id}) \end{aligned}$$

Again, by approximating the probabilities of the concept set and the relation set in the equations 5.4 and 5.8, we get:

$$RSV_{log}(G^{Iq}|G^{Id}) \propto \sum_{WC_f{}^{Iq} \in S^{Iq}_{WC_{\mathcal{F}}}} \sum_{c \in \mathcal{C}_f} \log P(c|G^{Id})^{\#(c,Iq)} +$$
$$\sum_{WE_l^{Iq} \in S^{Iq}_{WE_{\mathcal{L}}}} \sum_{(c,c',l) \in \mathcal{C}_f \times \mathcal{C}_{f'} \times \mathcal{L}} \log P(L(c,c') = l | WC_f^{Iq}, WC_{f'}^{Iq}, G^{Id})^{\#(c,c',l,Iq)}$$
$$\propto \sum_{WC_f{}^{Iq} \in S^{Iq}_{WC_{\mathcal{F}}}} \sum_{c \in \mathcal{C}_f} \#(c, Iq) \times \log P(c|G^{Id}) +$$
$$\sum_{WE_l^{Iq} \in S^{Iq}_{WE_{\mathcal{L}}}} \sum_{(c,c',l) \in \mathcal{C}_f \times \mathcal{C}_{f'} \times \mathcal{L}} \#(c,c',l,Iq) \times \log P(L(c,c') = l | WC_f^{Iq}, WC_{f'}^{Iq}, G^{Id})$$

Finally, the probabilities of the concept and relation of the graph document are estimated using the Jelinek-Mercer smoothing as seen in the previous sections.

For image categorization, document images are categorized into a specific *classes*, for example: “person”, “animal”, “nature”, “sport”, etc. Query image Iq is matched against the document images and then classified into the class of the closest document image Id given by the following estimate:

$$class(Iq) = class(\arg\max_{Id \in C} RSV_{log}(G^{Iq}|G^{Id})) \tag{5.9}$$

More details on the classification task will be given in the experiment part of the next chapter.

5.5 Conclusion

We have presented in this chapter the formulation of the visual graph model and the graph matching algorithm inspired by the language modeling from information retrieval. The visual graph is defined from a set of concept sets and a set of relation sets. The visual concept set is achieved by image segmentation, feature extraction and by visual concept learning. The relation set is generated based on the predefined extraction rules. The proposed graph model reflects the modern approach in the contend-based image retrieval, which try to combine multiple viewpoints of the visual content. This can be done in several ways, such as vector combination for BoW model or kernel fusion for SVM method. However, these approaches lack the capacities of integrating efficiently the spatial information, which is crucial to represent the image content, among the visual concepts. On the contrary, with our general visual graph framework, one can

integrate smoothly different types of visual concept and also the spatial relation among them.

One limitation of this approach is that the relation between concepts is defined manually and varies depending on the nature of the image collection or application used. For this reason, two graph instances have been shown with different configuration that adapted to the visual content. Once again, this also proves the flexibility and the expendability of our visual graph model.

Section 5.3 showed how the document graphs are matched against the query graph using the extension of the language modeling framework. Indeed, the matching model not only takes into account a set of concepts but also a multiple concept sets and a multiple relation sets. We have made several hypotheses in order to adapt to the specific context of the image. The smoothing technique is also modified to fit into our visual graph framework. Finally, we have demonstrated an example of graph matching to illustrate the idea.

The next chapter is dedicated to the application of our proposed approach. We will present two applications of the image categorization problem. The first application is a system for scene recognition of the Singapore's famous landmarks. Different graph models will be created to take into account of different visual features. We will also show how the spatial relations are improving the accuracy of the recognition process. We also discuss on the aspect of optimizing the smoothing parameters using the cross-validation technique. The second application is a self-localizing system of a mobile robot in an indoor environment. We will show how the graph model has been created to adapt to different environment conditions (such as light changing, object moving, etc.). Finally, in both applications we will show that visual graph models are actually performing better than the state-of-the-art SVM methods.

Part III

Applications

Chapter 6

Scene Recognition

Imagination is more important than knowledge ...
Albert Einstein

6.1 Introduction

The first application of the theoretical approach is an *outdoor scene recognition system*. This work has been done in the context of French-Singaporean collaboration. This has been partly realized at the Image Perception and Access lab (IPAL) in Singapore and at the Multimedia Modeling Information Retrieval (MRIM) team in Grenoble. Part of the project was funded by the Merlion programme, supported by the French embassy in Singapore.

(a) prototype (b) phone interface

Figure 6.1: Snap2Tell application is a prototype of an image-based mobile tour guide [Lim *et al.* 2007].

One application of this scene recognition systems is for mobile touristic scene identification, called Snap2Tell [Lim *et al.* 2007], developed by IPAL lab. A user uploads an image taken with a hand-phone and sends it as a query to the Snap2Tell system. The Snap2Tell system will be able to identify the particular scene and sends back the tourist information. The Snap2Tell prototype is implemented as 2-tier architecture: client-server protocol. The client provides a mobile user interface and has functionalities to capture images and interact with the server. The client-server protocol is developed for communication over wireless connection (such as WiFi and GPRS) as depicted in figure 6.1.

On the server-side, the recognition system identifies the captured scene, retrieves and sends back the scene descriptions. The images from collection are matched against the user query. Finally, information related to the matched scenes (i.e. text or audio) will be sent back to the user mobile through the wireless connections.

In this regard, our works concentrate on developing *a good scene recognition engine* which needs to address the following challeging issues:

- No prior knowledge of the image content and of the categorization of objects/scenes;
- Difficulty to separate foreground and background;
- Occlusion and moving objects, for example: people, vehicles, trees, etc.;
- Variation of viewpoint, scale, lighting condition;
- Fast and reliable response to the user query given a limited computing resources.

6.1.1 Objectives

Our first objective is to build for this specific image collection an adequate visual graph model that compromises both visual features and spatial relations. We will show that with the integration of spatial relation, our visual graph model obtains a better performance versus the standard conceptual model. Moreover, we will compare the proposed model with the state-of-the-art SVM method on the image classification.

Second, as user can take one or several images of the same scene and query the system, we have considered several usage scenarios for training and query: with single image (I) or with multiple images (S). Therefore, our second objective is to demonstrate that with multiple image queries, which accumulated viewpoints of the scene, will help to improve significantly the recognition accuracy. Table 6.1 summarizes the different scenarios implemented in our experiments.

Figure 6.2: Different viewpoints of the Merlion statue. User can use single image or multiple images as a query.

Table 6.1: Summary of experiments on STOIC-101 collection. A scene (S) corresponds to a group of images and (I) corresponds to a single query image.

	Training by (I)	Training by (S)
Query by (I)	√	√
Query by (S)	√	√

Last but not least, we will discuss on how we optimize the smoothing parameters with 3-fold cross validation on the training set. Comparing to the *a posteriori* optimized methods, we will show that the effect of the cross validation parameters is not significant. Issues related to technical implementation will also be discussed.

6.1.2 Outline

In the next section, we will describe the STOIC-101 image collection used in our experiments. Then, we present the proposed visual graph models adapted for the image collection in section 6.3. The experimental result will be shown in section 6.4 with different impacts of the relation and multiple image queries on the classification performance. In section 6.5, we will discuss on how we used the cross validation technique on training set to optimize the smoothing parameters. Finally, we summarize the chapter in section 6.6

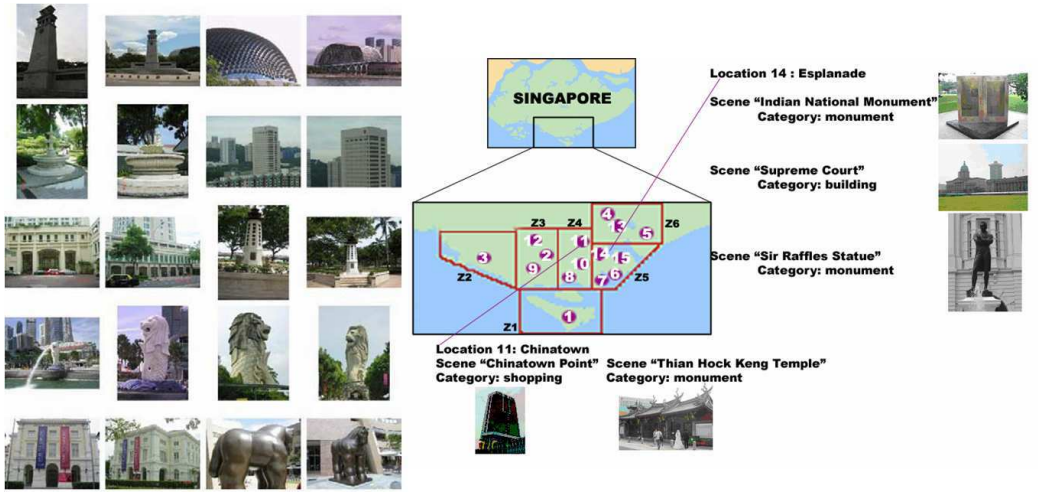

Figure 6.3: Images of STOIC-101 collection are taken from different locations across Singapore [Lim *et al.* 2007].

6.2 The STOIC-101 collection

The Singapore Tourist Object Identification Collection (STOIC) has been collected for the experimental purpose of the Snap2Tell application. STOIC collection contains of 3,849 images taken from 101 popular tourist landmarks in Singapore (mostly outdoor). These images were taken, mainly with consumer digital cameras in a manner typical of a casual tourist, from 3 distances and 4 angles in natural light, with a mix of occlusions and cluttered background to ensure a minimum of 16 images per scene. Images in the collection are affected by different weather patterns and different image capturing styles. Figure 6.3 shows some example images taken from the STOIC collection. Note that some images in the collection have been rotated into the correct orientation (for portrait and landscape layouts).

For experimental purposes, the STOIC-101 collection has been divided into a training set containing 3,189 images (82.8% of the collection) and a test set containing 660 images (17.2% of the collection). The average number of images per class for training is 31.7, and 6.5 for testing respectively. In the test set, the minimum number of images per class is 1, and the maximum is 21. Table 6.2 summarizes some key statistics on the STOIC-101 collection.

Table 6.2: Statistics of the STOIC-101 collection

	Training	Test	Overall
Number of scenes	101	101	101
Number of images	3189	660	3849
Percentage	82.85%	17.15%	100%
Mean (per scene)	31.57	6.53	38.11
Maximum (per scene)	160	21	181
Minimum (per scene)	5	1	8

6.3 Proposed models

6.3.1 Image modeling

Several studies on the STOIC collection have shown that color plays a dominant role, and should be preferred to other visual features such as edge or texture [Lim *et al.* 2007]. Furthermore, color histogram can be easily and efficiently extracted. For these reasons, we rely only on HSV color features in our experiments. In order to assess the validity of our methodology, we followed different ways to divide image into regions as proposed in chapter 4 and we retained:

1. A division of medium grain, where blocks of 10x10 pixels are used, the center pixel being considered as a representative for the region. We refer to this division as *mg*.

2. A patch division where the image is divided into 5x5 regions of equal size. We refer to this division as *gg*.

For *mg* divisions, we used the (H, S, V) values as a feature vector for each pixel. Similarly, each patch in *gg* division is quantized by a HSV histogram (4 bins/channel) that yields a 64 dimensional vector for each region. We then clustered the HSV feature vectors of all regions into $k = 500$ classes with *k-means* clustering algorithm. This results in a hard assignment of each region to one concept. The set of weighted concepts, WC, is then obtained by counting how many times a given concept occurs in the image. The choice of $k = 500$ is motivated by the fact that we want a certain granularity in the number of concepts used to represent an image. With too few concepts, one is likely to miss important differences between images, whereas too many concepts will tend to make similar images look different.

6.3.2 Visual graph models

We refer to the indexing obtained in this way as *mg-LM* and *gg-LM*, respectively for "division *mg* with automatically induced concepts" and "division *gg* with automatically induced concepts". For the methods *mg-LM* and *gg-LM*, we extracted the spatial relations between concepts as mentioned previously: *left_of* and *top_of*, and counted how many times two given concepts are related through a particular relation in order to obtain the weights for our relations. This last step provides a complete graph representation for images. We will refer to these two complete methods as *mg-VGM* and *gg-VGM*.

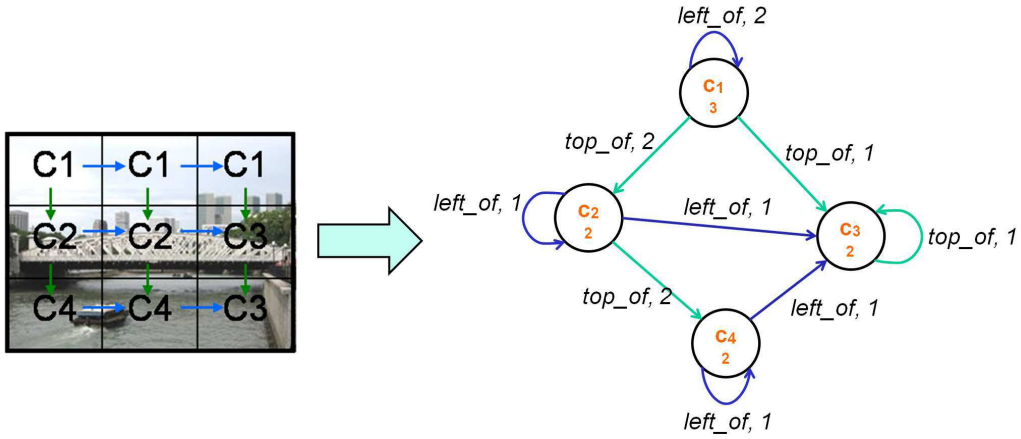

Figure 6.4: A visual graph extracted with gg division and two relations $left\_of, top\_of$ from a riverside scene.

Figure 6.4 shows an example of visual graph extracted with gg concepts and the $left\_of, top\_of$ relations. To summarize, we have constructed four models based on the visual concept sets and the relation sets:

1. *mg-LM*=$< WC_{mg}, \emptyset >$, that used only *mg* division concepts.

2. *mg-VGM*=$< \ WC_{mg}, W_{Eleft\_of} \bigcup WE_{top\_of} \ >$, that used *mg* division concepts and two intra-relation sets *left_of* and *top_of*.

3. *gg-LM*=$< WC_{gg}, \emptyset >$, that used only *gg* concepts.

4. *gg-VGM*=$< \ WC_{gg}, WE_{left\_of} \bigcup WE_{top\_of} \ >$, that used *gg* concepts and two intra-relation sets *left_of* and *top_of*.

6.4 Experimental results

To classify query images in the 101 scenes, we used the language model for visual graphs as mentioned in equation 5.9. When there is no relation, as in the cases of *mg-LM* and *gg-LM*, the term $P(S^q_{WE}|S^q_{WC}, G^d) = 1$ so that only concepts are taken into account to compare images.

6.4.1 Classification accuracy

The performance of the different methods was evaluated using the accuracy, *per image* and *per scene*. They are defined as the ratio of correctly classified images or scenes. More precisely:

$$\text{Image accuracy} = \frac{TP_i}{N_i}, \quad \text{Scene accuracy} = \frac{TP_s}{N_s}$$

where TP_i and TP_s represent the number of images and the number of scenes (respectively) correctly classified. N_i is the total number of test images (i.e., 660 images), and N_s the total number of scenes (i.e., 101 locations).

6.4.2 The impact of multiple training/query images

Table 6.3 shows the results we obtained when using automatically induced (through clustering) concepts. As one can see, automatically induced concepts with a medium grain division of the image yields the best results (the difference with the patch division for the S-I scenario being marginal). Overall, the *mg* division outperforms the *gg* division in most of the cases. Especially in the S-S scenario, the *mg* models obtained the best performance. One possible reason is that in *mg* division the number of concepts is far more than the one in the *gg* division.

Table 6.3: Impact of spatial relations and multiple training/query images on the performance (best result for each scenario is in bold, relative improvement over the method without relations is in parentheses)

Training	Query	*mg-LM*	***mg-VGM***	*gg-LM*	***gg-VGM***
I	I	0.789	**0.794** (+0.6%)	0.484	0.551 (+13.8%)
I	S	0.822	**1.00** (+21.6%)	0.465	0.762 (+63.8%)
S	I	0.529	0.594 (+12.3%)	0.478	**0.603** (+26.1%)
S	S	**1.00**	**1.00**	0.891	0.920 (+3.2%)

This being said, there is a difference between the I-S and S-I scenarios: The system is queried with more information in the I-S scenario than in the S-I scenario. This difference results in a performance which is, for all methods, worse for the S-I scenario than for the other ones. We conjecture that this is why the results obtained for the *mg-VGM* method on S-I are not as good as the ones for I-I. There seems to be a plateau for this scenario around 0.6, a hypothesis we want to explore in future work.

6.4.3 The impact of the relations

We also assessed the usefulness of spatial relationships by comparing the results obtained with the different methods that include or not such relations. These results are displayed in Table 6.3. As one can note, except for the S-S scenario with the *mg* division, the use of spatial relations always improves the accuracy of the classifier. This justifies the framework we developed in section 5.3 of language model for visual graphs including automatically induced concepts and spatial relations among them.

6.4.4 Comparing to the SVM method

In order to confirm the validity of our methods, we have compared the results with the state-of-the-art method in image categorization such as SVM classification method (implemented thanks to the *libsvm*[1]). We applied the same visual features used for graph model in our experiment. The input vector in SVM classifier is the early fusion of the multiple bag-of-word models. Then, each image class was trained with a corresponding SVM classifier using radial basis function (RBF) kernel. To optimize the kernel parameters, we trained SVM classifiers with 3-fold cross validation on the training set. Finally, these classifiers are used to classify the new query image.

Similar to above, we refer to the model with only the contribution of concept as LM and model with the spatial relation as VGM. We choose the *mg* concepts as a comparison model.

Table 6.4: Results on categorizing STOIC-101 collections comparing to SVM method using I-I scenario.

	$\#class$	*SVM*	*LM*	***VGM***
STOIC	101	0.744	0.789 (+ 6.0%)	**0.794** (+ 6.3%)

[1] http://www.csie.ntu.edu.tw/cjlin/libsvm/

Table 6.4 summarizes the results obtained from collection STOIC-101 with three methods: SVM, LM, and VGM. We can see that in all cases the VGM outperforms other methods. More precisely, with the integration of spatial relation into VGM helps improving the accuracy of classical approaches of LM by 6.0%. Moreover, VGM increases roughly the accuracies of 6.3% comparing to SVM method. This proves that the proposed methods (LM and VGM) are consistently performing better than the state-of-the-art SVM.

6.5 Discussion

In this section, we will give some discussion on how we optimize *a posteriori* the smoothing parameter based on the test set. We also employed the cross validation technique on the training set to optimize *a priori* these parameters. We will show that the difference in term of parameter values and classification accuracy is not significant among the two methods.

6.5.1 Smoothing parameter optimization

The results presented above are optimized *a posteriori*, i.e., we exhaustively tested the parameters on the test set to get the best configuration. We vary the value of each parameter in between $[0, 1.0]$ with the increment of 0.1 for each step. Then, we test these values against the test set and choose the configuration which gives the best results. However, this approach overestimates the proposed algorithms, by giving an upper bound of the evaluation results and not a correct estimation.

In a way to estimate more precisely the results, we optimized the smoothing parameters on a validation set for the *mg* division models because this approach gives the best results. To achieve this optimization, a 3-fold cross validation was performed. Once the parameters were optimized for each of the three training/validation sets, we processed the test set using the whole training set. This technique is called optimizing *a priori* of the smoothing parameters.

Table 6.5 compare the two techniques mentioned above for parameter optimization. For the model *mg-LM*, only the concept smoothing parameter λ_c has been tested. We can see that there are only two cases that the optimized parameters are different, e.g., for the I-I and I-S scenarios. However, the gap between the *a posteriori* and the *a priori* is not significant (with the difference of 0.1).

Similar with the model *mg-VGM*, two parameters λ_c and λ_l have been tested. Note that we applied specifically the same parameter value λ_l for both relation *left_of* and *top_of* as they contribute an equal role in the graph model. We observe that it obtained almost the same values in all cases, except in the case of I-S

Table 6.5: Comparison of the smoothing parameters obtained by cross validation set (*a priori*) and by test set (*a posteriori*) with model ***mg-LM*** and model ***mg-VGM***. Bold values signify the different cases of the two methods.

Training	Query	λ_c		λ_l	
		a posteriori	*a priori*	*a posteriori*	*a priori*
mg-LM					
I	I	**0.1**	**0.2**	-	-
I	S	**0.8**	**0.7**	-	-
S	I	0.1	0.1	-	-
S	S	0.2	0.2	-	-
mg-VGM					
I	I	0.2	0.2	0.3	0.3
I	S	0.7	0.7	**0.9**	**1.0**
S	I	0.1	0.1	0.7	0.7
S	S	0.2	0.2	1.0	1.0

scenario with the relation smoothing parameter λ_l. This proves that the smoothing parameters obtained with the *a posteriori* optimization technique are consistent compared to the *a priori* optimization technique.

Table 6.6 shows the average (Avg) and standard deviation (Std-dev) of the 3 results obtained. The last column of the table exhibits the difference (Diff) in percentage for the evaluation measurement between the 3-fold results and the *a posteriori* optimization. As shown in the table, the results obtained by the cross validation and by a posteriori optimization are very similar. If we focus on the results of the I-I, S-I and S-S configurations, the differences are smaller than 1%, and for the configuration I-S the 3-fold results are 4.46% lower. So, the optimization used on the validation sets provides satisfying results for a medium grain and for automatically defined visual concepts.

Table 6.6: Comparison of the results *mg-LM-val* on 3-fold cross validation, and percentage of difference in accuracy compared to the *a posteriori* optimization model *mg-LM*

Training	Query	*mg-LM*	***mg-LM-val***		Diff
			Avg	Std-dev	
I	I	0.789	0.784	5.8×10^{-3}	-0.68%
I	S	0.822	0.785	5.8×10^{-3}	-4.46%
S	I	0.529	0.529	0.0	0%
S	S	1.00	0.990	1.7×10^{-2}	-0.01%

We also tested 3-fold cross validation with relationships, as presented in Table 6.7. Here again the results with the cross validations are very close to the *a posteriori* optimized results: the S-I and S-S results are almost equal. A small difference is observed as in the case of I-I and I-S.

Table 6.7: Comparison of the results *mg-VGM-val* on 3-fold cross validation, and percentage of difference in accuracy compared to the *a posteriori* optimization model *mg-VGM*

Training	Query	*mg-VGM*	***mg-VGM-val***		Diff
			Avg	Std-dev	
I	I	0.794	0.788	6.4×10^{-3}	-2.64%
I	S	1.00	0.939	5.3×10^{-2}	-6.07%
S	I	0.594	0.594	0.0	0%
S	S	1.00	0.990	1.7×10^{-2}	-0.01%

Another conclusion drawn from Tables 6.6 and 6.7 is that, with a cross validation procedure, the usage of relationships still outperforms the results without relationships: +0.5% for the case I-I, +19.6% for I-S, and +12.3% for S-I. For the case S-S no improvement is achieved, which is also consistent with the *a posteriori* optimized results.

6.5.2 Implementation

The system is implemented in C/C++ with the LTI-Lib[2] and compiled on a Linux platform. LTI-lib is a well designed and well documented for image processing library, developed by the Aachen University of Technology. Image indexing and querying are performed on a computer with 3.0 GHz quad-core CPU and 8.0 Gb of memory. Training step takes about 2 hours for the whole training images set from extracting visual features, clustering the concepts and modeling trained graphs. For the query step, it takes about 0.22 second on average (or 5 images/second) for computing the likelihood of graph query with all the graphs stored in database. However, the computation is highly parallelizable given graph models are stored and are processed independently. It shows that the graph matching step is very reliable for image matching comparing to classical graph matching algorithm.

[2]http://ltilib.sourceforge.net/

6.6 Summary

We have shown in this chapter the first application of the visual graph model, namely the *outdoor scene recognition*. The context in which this work has been realized is to develop a scene recognition engine for the Snap2Tell prototype, an image-based mobile tourist guide. For this purpose, we have tested our proposed graph models on the STOIC collection containing of 101 famous scenes of Singapore.

We have proposed different visual graph models in order to adapt to the specific visual contents of the image collection. The proposed graphs were constructed based on the *color concepts* and two spatial relations *left_of* and *top_of*. The results obtained shown that the integration of spatial relations into the visual graph model outperformed the standard language model and the SVM classification which based only on the visual concept. A key strength of the proposed approach is the possibility of combining several images for training and for querying. The results have shown clearly an improvement in term of accuracy of the multiple image queries comparing to that of the single image query. This also confirmed the *flexibility* and *extenbility* of this new graph-theoretic framework.

Finally, we have discussed the process of optimizing the smoothing parameter with the cross validation technique. Parallel to the *a posteriori* optimizing method based on the test set, it has shown a very small difference in result with the parameter optimized with *cross validation technique*. This fact confirmed the consistency of the proposed *Jelinek-Mercer smoothing* method. In fact, we also wish to study the cross validating with other smoothing method (such as *Dirichlet smoothing*) as referred in the state-of-the-art. This should be considered in our future works. Some details on the implementing of the system have also provided to prove the *reliability* of the graph-based framework.

In the next chapter, we will present the second application of our method to the self-localizing of a mobile robot in an indoor environment. Coping with the specific condition of the *indoor and laboratory environment*, we experiment another instance of the proposed graph model. We will show how it can be adapted to the indoor changes (such as lighting condition, object moving, human involving and the *unknown* room).

Chapter 7

Robot Localization

7.1 Introduction

RobotVision[1] track is organized by the ImageCLEF[2] evaluation campagin. The main task is to exploit the location information within a known environment of a mobile robot based only on the visual information [Luo *et al.* 2006]. This chapter focuses on applying the proposed visual graph modeling for the RobotVision track of the ImageCLEF 2009. This works was partly funded by the AVEIR[3] (Automatic annotation and Visual concept Extraction for Image Retrieval) project, supported by l'Agence Nationale de la Recherche (ANR).

The challenge was to build a system able to answer the question "Where are you?" for a mobile robot. The visual system has to determine the topological localization of a mobile robot based on a sequence of training image. One difficulty of this task is that the robot has to recognize a room in different illumination conditions and adapt as the environment changes (such as moving people or objects, new furniture added over the time, etc.). This might pose a problem for a visual recognition system as the trained data usually obtained at a fixed time. Meanwhile, the system has to provide the location of the robot in real-time and in different time spans (6 months to 20 months) (see figure 7.1).

Several classical approaches in computer vision have been proposed for this problem. In [Pronobis *et al.* 2008], the authors suggested an appearance-based method using Support Vector Machine (SVM) to cope with illumination and pose changes. This method achieved a satisfactory performance when considering a short time interval between training and testing phrases. Other possible approach is to detect the interest point (such as SIFT, Harris-Laplace, etc.) and do a

[1] http://www.imageclef.org/2009/robot
[2] http://www.imageclef.org/
[3] http://aveir.lip6.fr/

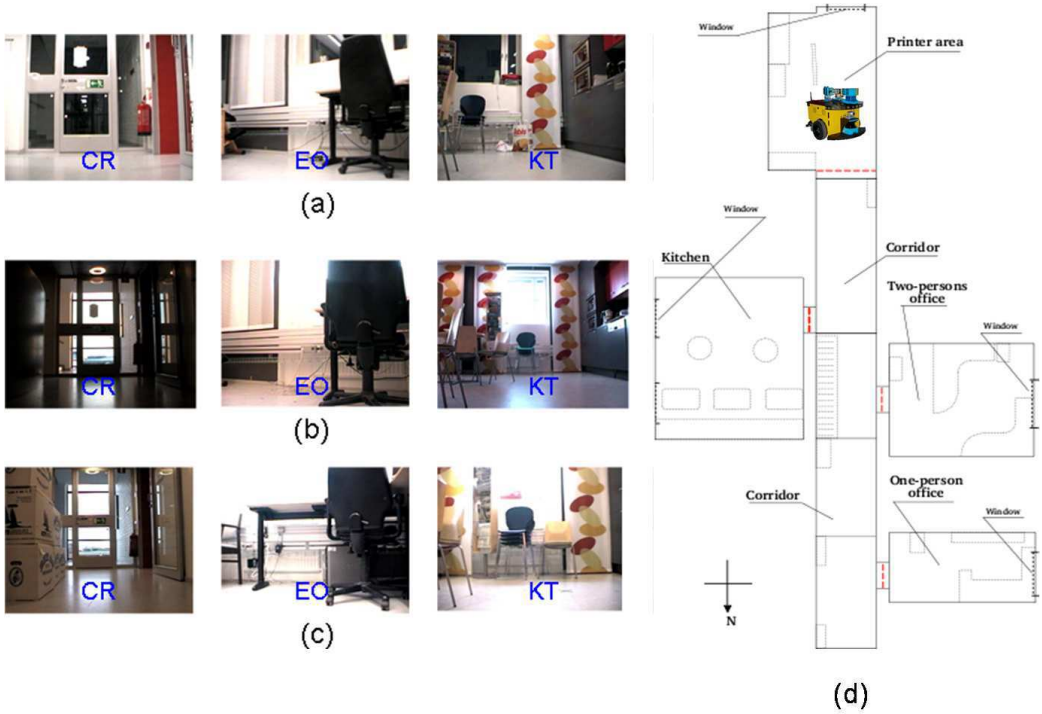

Figure 7.1: Example images from RobotVision'09 collection: (a) training set in night condition, (b) validation set in sunny condition, (c) test set in unknown condition, (d) the local area map.

topological matching of these points [Lowe 1999]. This is a simple approach but quite effective for recognizing some types of non rigid objects (e.g., building, car, motorbike, etc.). However, this method is heavily based on the quality of the interest points detected.

In the context of the RobotVision, we have developed an indoor recognition system that deals with some specific conditions:

- Small number of images in training set;
- Occlusion and moving objects, for example: people, furniture, object, etc;
- Lighting conditions changes, such as sunny, night and cloudy day;
- Different time span of image collections;
- *Unknown* environments, *unknown* objects/classes.

7.1.1 Objectives

To participate in this competition, we apply our visual language model (VLM) with the enhancement to cope with specific conditions of this task. We will show the *robustness* and the *adaptability* of the proposed models with different kind of image representations as well as different type of visual features. The validating process helps us to choose the appropriate features for our VLM. The relevance status value (RSV) proposed in section 5.4 will be also employed for ranking the results. In order to enhance the classification quality, we will perform some post-processing of the ranked results based on their relevance values. We will also provide the official results of our runs submitted to the ImageCLEF 2009 campaign.

The visual graph model (VGM) with the addition of spatial relation to the VLM was done after the competition. However, VGMs have shown a clear improvement comparing to the VLMs. We will show that the impact on different room accuracies proved *stability* of the VGM. Finally, we compare both approaches (VLM and VGM) with the SVM method for image classification.

7.1.2 Outline

Next section describes the IDOL2 image collection used in for the RobotVision experiments. Then, we present the proposed visual graph models adapted for this image collection in section 7.3. The experimental results will be shown in section 7.4 with different impacts of the relation and of the room classification accuracies. We also give a comparison of the proposed model with the SVM method. Section 7.5 discusses how we used the validation set to choose the appropriate features for representing the image contents. The post-processing step and the official results of the run submitted will also be detailed. Finally, we conclude this chapter in section 7.6.

7.2 The IDOL2 collection

The RobotVision collection consists of a subset of the IDOL2 database[4]. The image sequences in the IDOL2 database were captured with a Canon VC-C4 perspective camera with the resolution of 320x240 pixels, mounted on a mobile robot platform. The robot was manually driven through rooms while continuously acquiring images (see figure 7.2). The acquisition was performed in a five room of a laboratory environment and one *unknown* room for test set. These rooms was captured under three different illumination conditions: in *cloudy* weather, in

[4]http://cogvis.nada.kth.se/IDOL2/

sunny weather, and at *night*. Each of the rooms represented a different functional area, annotated as follows:

- CR - corridor
- PA - printer area
- KT - kitchen
- BO - one person office
- EO - two persons office
- UK - *unknown* room from test set

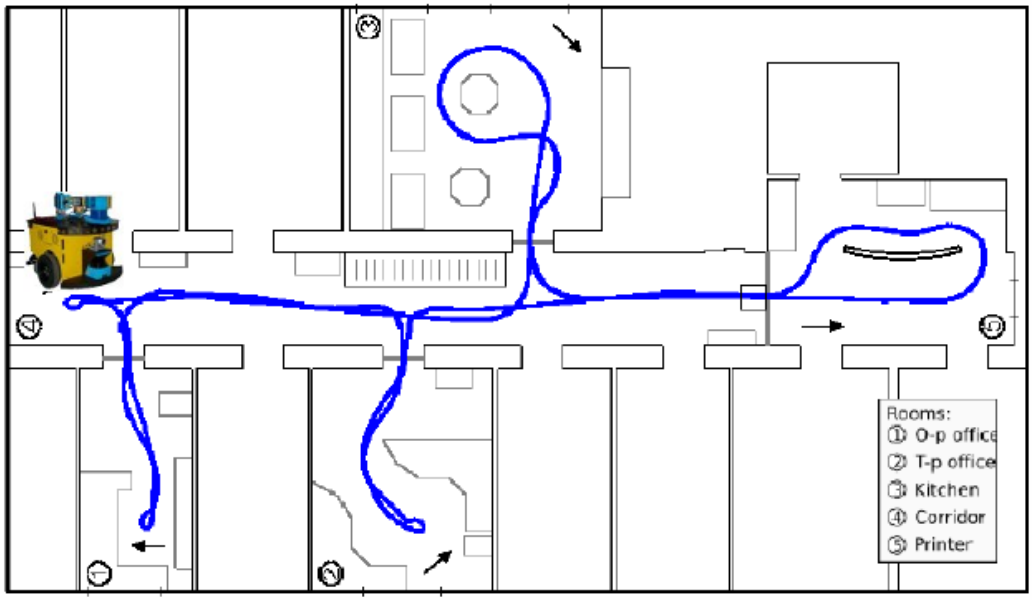

Figure 7.2: Path of the robot doing acquisition of one of the image sequences

The acquisition process was conducted in two phases. Training and validation sequences were acquired for each type of illumination conditions over the time span of 6 months. Therefore, the sequences captured variability introduced not only by illumination but also natural activities in the environment (e.g., moving people, furniture relocated etc.). The test sequences were acquired in the same environment but performed 20 months after the acquisition of the training set. Test sequences contain an additional room that was not captured in the training and validation sets. Examples of images showing the interiors of the rooms, variations of activities and changing of illumination condition are presented in Figure 7.3.

Figure 7.3: Example images from IDOL2 collection: (a) variation of illumination changing, (b) variation over time span.

For experimental purpose, the RobotVision collection consists of three image sets: training set, validation set and test set. Training set contains a sequence of 1,034 images and validation set contains a sequence of 909 images. Training and validation sets consist of five rooms across a span of 6 months. The official test was released as a sequence of 1,690 images with an additional room and recorded 20 months later.

7.3 Proposed models

The system we used for the RobotVision competition was composed of two processes: a recognition step and a post-processing step. However, we describe

and evaluate here only the recognition step, in such a way to assess the impact of the proposed model. The post-processing step of the results will be discussed in the section 7.5.2. The robot was trained with a sequence of images taken in the night condition. Then, we used a validation set captured in sunny condition to estimate the system parameters.

7.3.1 Image modeling

As described in chapter 4, the concept sets and relation sets were extracted from the image collection as follows:

1. Each image was divided into 5x5 patches. We extracted for each patch a HSV color histogram and an edge histogram as in section 4.2. Then, the visual vocabulary of 500 visual concepts was constructed by using k-means clustering algorithm. From this vocabulary, we built the weighted concept set WC_{patch}.
2. Similar to the previous step except that the visual features were extracted from the local keypoints. To be more precise, we detected scale invariant keypoints using SIFT detector [Lowe 2004] for each images. Local features were then used to create the weighted concept set WC_{sift}.
3. Using the two previous features we defined an inter-relation set $\{inside\}$ between patch concepts and SIFT concepts, denoted as WE_{inside}, if one key-point is located **inside** the area of a corresponding patch.

7.3.2 Visual graph models

Similar to above, we referred to the model without relation as LM (simply the production of probability generated by different concept sets) and the graph model with the spatial relation as VGM (with the contributing of relation probability to graph model). Based on this definition, we have implemented several graphs to measure the performance of our proposed model:

1. $LM^{P} =< WC_{patch}, \emptyset >$, that used only patch concepts.
2. $LM^{S} =< WC_{sift}, \emptyset >$, that used only SIFT feature concepts.
3. $LM^{S.P} =< WC_{sift} \bigcup WC_{patch}, \emptyset >$, that used both patch and SIFT feature concepts.
4. $VGM^{S \rightarrow P} =< WC_{sift} \bigcup WC_{patch}, WE_{inside} >$, that used patch concepts, SIFT feature concepts and the $inside$ relations between them.

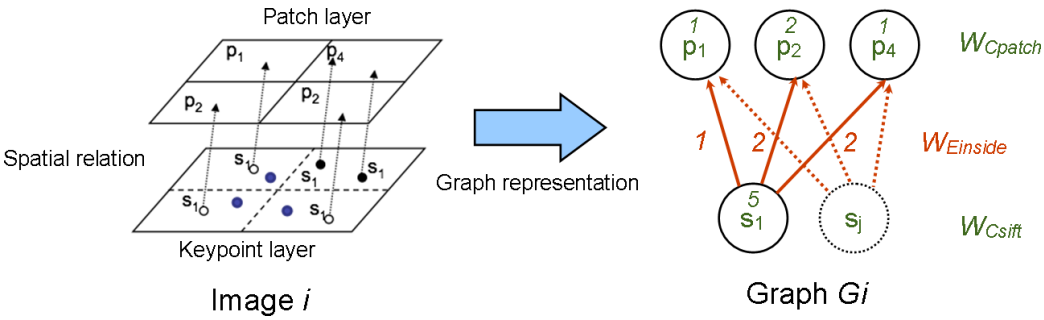

Figure 7.4: Graph model constructed for RobotVision includes two type of image representation and one type of relation.

Figure 7.4 gives an example of the graph extracted from the concept sets and relation sets defined above. In fact, the first three models were estimated following the equation presented in section 5.3.2. The fourth model is the fusion graph combined with spatial relation. Its probability was computed according to the equation defined in section 5.3.3.

7.4 Experimental results

7.4.1 Evaluation methods

The image sequences used in the competition were annotated with ground truth. The annotations of the training and validation sequences were available to the participants, while the ground truth for the test sequence was released after the results were announced. Each image in the sequences was labeled according to the position of the robot during acquisition as belonging to one of the rooms used for training or as an unknown room. The ground truth was then used to calculate a score indicating the performance of an algorithm on the test sequence. We have applied two methods for the evaluating of the system performance.

Room accuracy In order to compare our method with other classical approaches and for the validation purpose, we also adopt the evaluation based on the classification accuracy as proposed for STOIC collection (section 6.4.1). This measurement is computed as follows:

$$\text{Room accuracy} = \frac{TP_{room}}{N_{room}}$$

where TP_{room} represents the number of images correctly classified for the specific room. N_{room} is the total number of images annotated for this room (i.e., 1034

images for corridor CR) in the test collection.

Recognition score (official measurement)

The recognition score measured the differences between the actual room id and the one classified by the systems. The following rules were used when calculating the official score for a test sequence:

- **+1.0 points** for each correctly classified image
- **-0.5 points** for each misclassified image
- **0 points** for unclassified image (the algorithm refrained from the decision)

This score is more strict in the sense that the robot will get penalty point (-0.5 point) for an uncorrected guest. Each participant has to decide their strategy to adapt to this specific context. Similar to the room accuracy, higher score means higher accuracy.

7.4.2 Impact of the spatial relation

Table 7.1 describes the results in terms of score value for each model. As expected, the two basic models LM^P and LM^S gave a good score for the validation set. However, the model LM^P did not perform well on the test set due to the introduction of new room and new arrangement of interior furniture. The simple fusion model $LM^{S.P}$ underperformed the best results of LM^P and LM^S. However, this result was more robust in the sense that it leveraged on the spurious effects of each visual feature (i.e., $LM^{S.P}$ outperformed the averaged result of LM^P and LM^S in both cases). Moreover, the introduction of $inside$ relations to the completed graph $VGM^{S\rightarrow P}$ boosted its results respectively by 39.5% and 40.1% comparing to the fusion graph $LM^{S.P}$ for both validation set and test set. This fact confirmed that the integration of relations played a significant role to improve the results. In addition, it showed that the link between object details and its global presentation provides a better abstraction for image content.

Table 7.1: Recognition scores of different graph models

Graph model	LM^P	LM^S	$LM^{S.P}$	$\boldsymbol{VGM}^{S\rightarrow P}$
Validation	345	285	334.5	**466.5** (+39.5%)
Test	80.5	263	209.5	**293.5** (+40.1%)

7.4.3 Impact on room classification

We present in detail the classification accuracies for each class (represented by its room id) as categorized by our algorithms in Table 7.2. For each class, the accuracy is computed by the number of correctly labeled images divided by the total number of images belonging to this class. Note that we only consider the classification accuracies of 5 rooms as we did not treat the *unknown* room in the test sequence at this step. The post-processing step of the results will be discussed in the section 7.5.2.

Generally, the graph model for SIFT concepts LM^S performs better than the graph model for patch concepts LM^P. This leads us to conclude that the details of object are important clues for scene recognition. In addition, the simple fusion model $LM^{S.P}$ tried to leverage the effect on both model LM^S and LM^P and improved the results only in the case of two-person office (EO). All four models gave good accuracies for the corridor (CR) regardless of brutal changes in light conditions. We also noted that the number of training images for corridor (CR) was the highest (483/1034 images) comparing to other classes. It suggests that the higher the number of image samples, the more robust the performance is.

Table 7.2: Classification accuracies of graph models for each room. Bold values indicate the best results obtained for each class.

	BO	CR	EO	KT	PA	Mean
Validation set						
LM^P	0.257	0.779	0.524	0.450	0.434	0.489
LM^S	0.354	0.658	0.581	0.426	0.402	0.484
$LM^{S.P}$	0.398	0.679	0.613	0.519	0.426	0.527
$VGM^{S\to P}$	**0.416**	**0.829**	**0.702**	**0.550**	**0.492**	**0.598**
Test set						
LM^P	0.163	0.701	0.385	0.236	0.279	0.353
LM^S	0.331	0.721	0.478	0.509	**0.348**	0.477
$LM^{S.P}$	0.206	**0.756**	0.484	0.410	0.286	0.428
$VGM^{S\to P}$	**0.369**	0.736	**0.540**	**0.516**	0.344	**0.501**

As a whole, the visual graph with spatial relations $VGM^{S\to P}$ led to higher accuracies in all cases except in the cases of corridor (CR) and printer area (PA) in test set. However, the difference was not significant comparing to other models (only 2% less than the $LM^{S.P}$ graph model). Furthermore, the mean accuracy of model $VGM^{S\to P}$ achieved on the test set and the validation set were the best of four models, with more than 7% better than the simple fusion model $VGM^{S.P}$. This result confirms again the strength of spatial relationships that contributed to

our graph model.

7.4.4 Comparing to SVM method

Similar to above, we refer to the model with only the contribution of concept as LM and model with the spatial relation as VGM. For RobotVision collection, we choose the model $LM^{S.P}$ as LM and $VGM^{S \to P}$ as VGM.

Table 7.3: Results on categorizing with different methods

	$\#class$	*SVM*	*LM*	***VGM***
Validation	5	0.535	0.579 (+ 8.2%)	**0.675** (+ 26.2%)
Test	6	0.439	0.416 (– 5.2%)	**0.449** (+ 22.8%)

Table 7.3 summarizes the results obtained from collection RobotVision'09. We can see that in all cases our VGMs outperform other methods. More precisely, with the integration of spatial relation into VGM helps to improve the accuracy of classical LM approaches by more than 8%. The LMs perform roughly similar to the SVMs. Likewise, VGMs increase sharply the accuracies from 22.8% to 26.2% comparing to those of SVMs for both the test and validation sets respectively. Once again, this fact confirms that if we can integrate the relation in a smart way, it could increase the overall performance of the recognition systems.

7.5 Discussion

In this section, we will discuss on how we choose the visual features for generating the language model based on the validations sets with different weather conditions. Then we will describe the post processing step for enhancing the quality of the results. Finally, we report the result of our submissions to the official evaluation ImageCLEF campaign.

7.5.1 Validation process

The validation aims at evaluating robustness of the algorithms to visual variations that occur over time due to the changing conditions and human activity. We trained our system with the night condition set and tested against all the other conditions from validation set. Our objective is to understand the behavior of our system with the changing conditions and with different types of features. Moreover, the validation process can help us to fine-tune the model parameters that the latter will be used for the official test.

We built 3 different language models corresponding with 3 types of visual features. The training set used is *night* set. Model Mc and Me correspond to color histogram and edge histogram extracted from image with the division of 5x5 patches. Model Ms corresponds to SIFT color feature extracted from interest points. We measure the precision of system using the accuracy rate. Summary of the results is reported in Table 7.4.

Table 7.4: Results obtained on different conditions with 3 visual language models

Training	Validation	HSV(Mc)	Edge(Me)	SIFT color(Ms)
Night	Night	**84.24%**	59.45%	79.20%
Night	Cloudy	39.33%	58.62%	**60.60%**
Night	Sunny	29.04%	52.37%	**54.78%**

We noticed that, under the same condition (e.g. night-night), the HSV color histogram Mc outperformed all the other models. However, under different conditions, the result of this model dropped significantly (from 84% to 29%). It showed that the color information is very sensitive with the changing of illumination condition. On the other hand, the edge model (Me) and the SIFT color model (Ms) are practically robust with the changing of the illumination condition. In the worst condition (night-sunny), we still obtained a quite good recognition rate of 52% for Me and 55% for Ms. As the result, edge histogram and SIFT feature are chosen as the appropriate features for our recognition system.

7.5.2 Post-processing of the results

For the official evaluation, the algorithm must be able to provide information about the location of the robot separately for each test image (**obligatory task**) (e.g. when only some of the images from the test sequences are available). This corresponds to the problem of global topological localization. However, results can also be reported for the case when the algorithm is allowed to exploit continuity of the sequences and rely on the test images acquired before the classified image (**optional task**). The reported results will be compared separately for each task.

We have performed some fine-tuning steps of these results in order to enhance the accuracy of our system. Figure 7.5 shows the flowchart of the post-processing of the results come from different models. This flowchart includes four main functions:

1. **Linear fusion**: we take the advantage of the different features extracted from the images. We represent an image by a set of concept sets $\mathcal{C}_i$

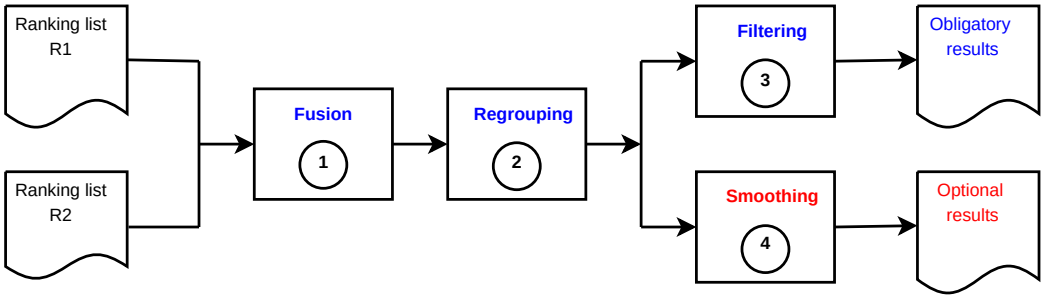

Figure 7.5: Post-processing steps of the results. Blue scheme is for the obligatory track and red scheme is the for optional track

and each $\mathcal{C}_i$ corresponds to a visual feature. Assuming the concepts sets independence, we fuse the relevance status values (RSV) from the ranked list of the individual concepts sets using sum operator:

$$RSV(Q, D) = \sum_i RSV(q_i, d_i) \tag{7.1}$$

where $Q = \{q_i\}$ and $D = \{d_i\}$ are the set of concept sets corresponding to the query image and to the document image respectively. This step corresponds to the visual graph fusion with the absent of the relation as developed in the above section.

2. **Regrouping by room id**: On the basis that using only the closest image to determine the room id of a query image is not enough, we proposed to group the results of the n-best images for each room. We compute a ranked list of room RL instead of an image list:

$$RL = \{R, RSV(Q, R)\} \tag{7.2}$$

with

$$RSV(Q, R) = \sum_{f_{n-best}(Q_j, R)} RSV(Q, D) \tag{7.3}$$

where R correspond to a room and f_{n-best} is a function that select the n images with the best RSV belonging to the room R.

3. **Filtering the "unknown" room**: we measured a difference from the score of the 4th room to the 1st room in the room list RL. If the difference is big enough ($>$ threshold β) we keep this image. Otherwise we remove it from the list (or consider as an "unknown" room). In our experiment, we choose the value $\beta = 0.003$ as a threshold.

4. **Smoothing window**: we exploited the continuity in a sequence of images by smoothing the result in the temporal dimension. To do that, we use a smoothing window sliding on the classified image sequences. Here, we choose the width of window $w = 40$ (i.e. 20 images before and after the classified image). So, the score of the smoothed image is the mean value of their neighborhood images.

$$RSV_{window}(Q_i, R) = \frac{\sum_{j \in [j-w/2; j+w/2]} RSV(Q_j, R)}{w} \tag{7.4}$$

where w is the width of the smoothing window. In the real case, we could only use a semi smoothing window which considers the images before the current classified image. This leads to:

$$RSV_{semi-window}(Q_i, R) = \frac{\sum_{j \in [j-w; j]} RSV(Q_j, R)}{w} \tag{7.5}$$

where w is the width of the semi-window.

Below is the result for the post-processing step (see Table 7.5) based on the ranked lists provide by two models Me and Ms from the validation process. The training and validation conditions used for post-processing are in *night* and *sunny* respectively.

Table 7.5: Result of the post-processing step based on 2 models Me and Ms

Me	**Ms**	Fusion	Regrouping $n - best = 15$	Filtering $\beta = 0.003$	Smoothing $w = 20$
52.37 %	54.78 %	62%	67%	**72%**	**92%**

As we can see, the linear fusion of these 2 models gives overall of 8% of improvement. The regrouping step helped to pop-up some prominent rooms from the score list by averaging from *n-best* room's scores. The filtering takes part in eliminating some of the uncertain decisions base on the difference of their score after the regrouping step. Finally, the smoothing step (which is an optional step) helps to increase significantly the performance of a sequence of images by 20%.

7.5.3 Submitted runs to the ImageCLEF 2009

Participating in this competition, we have built 3 graph models based on the previous validating process. We eliminated the HSV histogram model because of its poor performance on different lighting conditions. We used the same visual vocabulary of 500 visual concepts generated for night condition set. Each model provided a ranked result corresponding with the test sequence released. The post-processing steps were performed similar to the validating process employing the same configuration. The visual language models built for the competition are listed as follows:

- **Me1**: visual language model based on edge histogram extracted from 10x10 patches division
- **Me2**: visual language model based on edge histogram extracted from 5x5 patches division
- **Ms**: visual language model based on color SIFT local features

Based on the 3 visual models constructed, we have submitted 5 runs to the ImageCLEF evaluation:

- **01-LIG-Me1Me2Ms**: linear fusion of the results coming from 3 models (Score = 328)
- **02-LIG-Me1Me2Ms-Rk15**: re-ranking the result of 01-LIG-Me1Me2Ms with the regrouping of top 15 scores for each room (Score = 415)
- **03-LIG-Me1Me2Ms-Rk15-Fil003**: if the result of the 1st and the 4th in the ranked list is too small (i.e. β <0.003), we remove image that from the list. We refrain the decision from some cases other than to mark them as unknown room (Score = 456.5)
- **04-LIG-Me1Me2Ms-Rk2-Diff20**: re-ranking the result of 01-LIG-Me1Me2Ms with the regrouping of top 2 scores for each room and using smoothing window (20images/frame) to update the room-id from image sequences (Score = 706)
- **05-LIG-Me1Ms-Rk2-Diff20**: same as 04-LIG-Me1Me2Ms-Rk2-Diff20 but with the fusion of 2 model Me1 and Ms (Score = 697)

Our best run **03-LIG-Me1Me2Ms-Rk15-Fil003** for the obligatory track is ranked at 10^{th} place among all the 21 runs submitted. The best run in the competition (score = 793 points) was obtained with an approach based on local

feature matching. Run **04-LIG-Me1Me2Ms-Rk2-Diff20** had not met the criteria of the optional task which only used the sequence before the classified image. Nevertheless, this run has increased by roughly 250 points from the best obligatory run. It means that we still have room to improve the performance of our systems with the valid smoothing window.

7.6 Summary

To summarize, we have shown in this chapter the second application of the visual graph model, namely *mobile robot self-localization*. Coping with the specific condition of an *indoor laboratory environment*, we have implemented another instance of the proposed graph model. The proposed visual graph models have to adapt to the specific visual contents of the image collection, as well as adapt to the environment changes (such as lighting condition, object moving, human involving and the *unknown* room).

We have constructed different graph models based on *patch concepts* and *SIFT concepts* which represented the abstract form and the object details respectively. A particular relation between the two concepts is also included to capture the co-occurrence information among the concepts. The results obtained shown that the integration of spatial relations into the visual graph model outperformed the standard language model and the SVM classification which based only on the visual concept.

We have also performed a validation process based on the validation sets to choose the best visual features adapting to the environment changes. Post-processing step of the ranked list was also studied. Finally, we provided the official results of our submitted run to the ImageCLEF 2009 forum.

In the next chapter, we will conclude our thesis and give some perspectives into the future works.

Part IV

Conclusion

Chapter 8

Conclusions and Perspectives

We are not interested in the unusual, but in the usual seen unusually.
Beaumont Newhall

Content-Based Image Retrieval (CBIR) has been an open problem for the past two decades. Several attempts have been made to overcome the information gap between low-level visual features and the semantics layer of image. In [Marr 1982], Marr proposed a common paradigm for designing a visual recognition system which includes three sub-modules: *image processing*, *mapping* and *high-level interpretation*. Our works aimed at solving the two latter problems.

In this thesis, we have introduced a graph-based model for representing image content which added an intermediate layer to image representation. This graph captured the spatial relations among visual concepts associated with extracted regions of images. The graph matching process is based on the extension of unigram conceptual modeling, proposed initially in [Maisonnasse *et al.* 2008]. Theoretically, our model fits within the language modeling approach for information retrieval, and expands previous proposals for graph-based representation.

Even though we have chosen to illustrate the proposed approach with the scene recognition problems, this method is not fundamentally tied to a specific type of images. The designed framework can be extended for several types of image representations, as well as several applications in different fields, such as, *image retrieval/annotation*, *object recognition*, *video classification/categorization*, or *medical imaging classification*. This list, by all means, is not exhaustive. As suggested by Nicolas Maillot, the combination with a reasoning layer or an ontology network [Maillot 2005] will equippe the graph model with the capacity of understanding the scenic contents. The system is then able to detect multiple object instances embeded in a particular scene, e.g, car, people, building, street ...

8.1 Summary

We summarize here some main points mentioned in this dissertation:

Part I introduced the current state-of-the-art in the Content-Based Image Retrieval field.

In chapter 2, we gave a survey on different methods of image processing such as: image decomposition and visual features extraction. This is a basic step in representing of the image contents. Based in the extracted visual fetures, the *bag-of-words* model has been introduced. The bag-of-words model often represents image content by a sparse vector of visual concepts. Images are matched based on the Euclidean distances or the cosine similarity of the quantized vectors. The bag-of-words model is simple but lacks the information on the spatial relations between visual concepts.

In chapter 3, we reviewed two principal branches of learning methods based on the conceptual representation: *generative approaches* and *discriminative approaches*. Important approaches, such as, Naive Bayes, Language Modeling, Support Vector Machines, have also been introduced. Then, we discussed on the need of embedding the structural information of visual concepts into a graph-based image representation. We also investigated some current graph matching algorithms and their limitations. Finally, an initial proposal of the graph-based image retrieval framework was sketched.

Part II described the proposed approach based on the *graph-based image representation* and a *generative matching algorithm*.

In chapter 4, we presented the system architecture for the graph-based image modeling. This framework included three main stages: *image processing*, *graph modeling* and *graph retrieval*. The *image processing step* aims at extracting the different visual features from image regions to build a set of visual vocabularies. The *graph modeling step* consists of visual concepts construction and spatial relation extraction. Each image is then represented by a corresponding visual graph. Finally, the *graph retrieval stage* generates the probabilities likelihood for the query image from the trained graphs in the database. Images are ranked based on their relevance values.

Chapter 5 defined formally the visual graph model based on a set of concept sets and a set of relation sets. Two instances of the visual graph models were used to illustrate the adaptability of the latter to the real applications. Then, we showed how the document graphs are matched against the query graph using the extension of the language modeling framework. For better understanding, we have demonstrated with an intuitive example of graph matching. Finally, we showed how visual graphs were actually ranked in the log-probability space.

Part III demonstrated the proposed approach in **Part II** with two applications: *scene recognition* and *robot localization*. These experimentations aimed at assessing the validity of our approach in certain aspects. We have conducted the test on two image collections: STOIC-101 and RobotVision'09.

In chapter 6, the consideration of regions and associated concepts allows us to gain generality in the description of images, a generality which may be beneficial when the usage of the system slightly differs from its training environment. This is likely to happen with image collections that, for example, use one or several images to represent a scene. The proposed model able to adequately match images and sets of images represented by graphs. As we conjectured, being able to abstract from a low level description enables robustness with respect to the usage scenarios. On the other hand, querying a specific location with a group of images is very promising for future applications (such as mobile localization services) that allows higher accuracy score with less computational effort comparing to video sequence. In addition, the way of combining different image representations/features in the graph framework is more versatile comparing to other fusion approaches. On the experimental side, we have proved a positive impact of the relations, as well as of multiple image queries. We also discussed on the smoothing parameter optimization with a cross validation technique based on the training image set.

In chapter 7, we showed that integrating inter-relations between two different concept sets to represent images led to a significant improvement in the results. We hoped that the combination of the two different image representations (such as patch and keypoints) can take advantage of the different visual features of both the abstract-level of the scene as well as the details of the objects. The strength of our approach is that this fusion-like model can be expressed naturally through the links of graph-based model. The experimental results confirmed the superiority of the visual graph model comparing to the conceptual modeling approach. We also showed that the graph models performed better than the state-of-the-art SVM method for image classification. Finally, the proposed approach has been validated and submitted to the ImageCLEF for the evaluation.

8.2 Contributions

From the point of view of a *graph-based framework*, the major contributions of our approach are:

- **A well-founded graph representation for image retrieval**. We have presented a unified graph-based framework for image representation which is able to integrate different types of visual concepts and spatial relations

among them. Such graph can represent different image point of views in a very flexible way. Indeed, the visual graph model represents an intermediate layer of image representation that could fill the *semantic gap* between high-level knowledge and visual concepts.

- **A simple and effective graph matching process**. We have extended the language model in infomation retrieval for graph matching. The language modeling has been studied extensively for text retrieval and proved its effectiveness. Standing from this well-foundeness framework, our proposed method allowed matching of complex graph composed of *multiple concept sets* and *multiple relations sets*. This can be done under certain independence hypotheses of the concept sets and relation sets. Furthermore, we used the *Jelinek-Mercer smoothing method*, which is a popular approximation technique for re-estimating of the probability distribution.

- **Application to the problem of image categorization**. We have shown how the proposed approach can be applied to the problem of *scene recognition* and *robot localization*. Different graph instances have been developed for each application to adapt to the image contents. The experimental results performed on two image collections (STOIC-101 and RobotVision) have confirmed the good performance and the effectiveness of the visual graph modeling. Moreover, the proposed method also outperformed both the standard language modeling and the state-of-the-art SVM methods. The results obtained show a promising direction for the image categorization.

8.3 Future works

Our objectives aim at designing a graph-based framework which gains the capable of *generality*, *re-usability* and *expendability* in different contexts. In the future, several aspects can be considered to extend our visual graph model.

8.3.1 Short-term perspectives

Integration of textual information for multimedia retrieval/annotation

First of all, as the language model is coming from textual domain, we could combine the graph representation of image with the graph representation of the annotated text as done in ImageCLEF photographic retrieval track. Hence, multi-modalities image indexing and retrieval should be a promising direction for the future model extension.

In our case, the integration should be done smoothly as they shared the same probabilistic framework. The conceptual language modeling has been investigated successfully in [Maisonnasse *et al.* 2009] with the use of UMLS (Unified Medical Language System) for conceptual relation extraction in medical document retrieval. Therefore, we wish to fuse these two approaches in the same graph-based framework in order to enhance the system performance.

Moreover, the common framework between textual/visual graph makes it possible to learn the mutual information for each text category / image topic. Similar to [Pham *et al.* 2007], the proposed graph can be used for image annotation with the COREL image collection. For each image topic, we can train a *representative graph* for a set of visual graphs with a specific classifier (for example SVM classifier). Then, this representative graph can be used for classifying of the new images with the associating annotations.

The need of further study on visual concepts and spatial relations

In chapter 4, we have shown that the choice of visual concepts and relations are subjective. Hence, the future work should include more types of visual concepts and their relations and study the effect of these concepts and relations on the accuracy. Then the selection of good visual features (using LSA techniques to eliminate the *synonymy/polysemy* effects on the visual concepts) and spatial relations can be processed. This should be adapted following a specific image context or towards a typical scenario of the application. We also wish to investigate different possible couplings of the low-level and high-level representations, with the hope to come up with a single representation that could be used in a general case.

Study the impact on the number of visual concepts

Another technical issue that we would like to address is the choice of number of clusters for visual concept learning. This number might affect the quality of the constructed visual concepts. Actually, this parameter has been chosen empirically and fixed for each application. As done in [Pham 2006], a practical study on this aspect is needed for subsequent step of building visual graph models.

Evaluation of the proposed approach for object/video retrieval

Last but not least, experiment on a large collection of images is necessary to test the *scalability* and the *stability* of the proposed method. In the near future, the graph-based model can be used to tackle the video retrieval applications (e.g., TRECVID collection). Moreover, the graph instance proposed in chapter 5 (with the combination of patches and keypoints) seems appropriate for object classification. Hence, the proposed graph could be applied for VOC challenge task which comprises of more than 10K images in training and test sets.

8.3.2 Long-term perspectives

The long-term perspectives of this work cover a broad-range of theoretical extensions, as well as the practical aspects, for example: integration of an interactive relevant feedback and a prototype user-interface for retrieval platform.

Relevance feedback modeling using information divergence

Although the query likelihood model has a number of advantages, it is limited in terms of how it models the user needs. It is difficult to incorporate information about the relevant documents to a specific query using the current ranking algorithm. However, it is possible to extend the current query-based model to the *pseudo-relevance feedback* model with the measure of divergence between query models and document models. A well-known measure from information theory is the *Kullback-Liebler (KL) divergence* which measures the difference between two probability distributions. Hence, the first theoretical aspect we want to address is to incorporate this measurement in our graph-based framework. Similarly, the KL divergences should be computed independently for the concept set distributions and the relation set distribution respectively.

Extension of the current probabilistic framework

As explained in chapter 5, the proposed approach relies mainly on the theoretical assumption that the concept sets and relation sets are following the *multinomial probability distribution*. This assumption is widely used for text retrieval domain where the random variable takes only the discrete values (i.e., $0, 1, 2...N$). The k-means clustering defines a *hard version* of visual concept for visual vocabulary construction. Thus, the multinomial distribution assumption still holds true.

However, the visual concepts can be defined differently from the textual concept, for example using *fuzzy c-means* or *EM clustering* [Moore 1998], which can create a *soft version* of visual concept that is likely to be closer to the reality. It means that a visual concept might belong to several clusters with different weights/probabilities. For example, a visual concept could belong to cluster $c1$ with a probability of 0.8 and to cluster $c2$ with a probability of 0.2. In this case, the multinomial probability distribution is no longer valid and *Dirichlet distribution* seems more suitable for modeling of the random variable. As a consequence, the concept independence hypothesis is not correct anymore. Therefore, one theoretical direction is to extend the current framework with the Dirichlet probability distribution.

Graph clustering for visual concept navigation

One of the current trends in CBIR system is automatically regrouping image

into homogenous visual clusters (or sub-topics) for each image topic. In the ImageCLEF 2009 photo retrieval track, relevant images were asked to be clustered into sub-topics based on their visual similarities. For example, if a topic asks for photos of animals, clusters will be formed based on animal type. The objective is to promote the diversity of image search system. Another example is *Image Swirl* system developed by Google labs (see Figure 8.1) that automatically estimates the image clusters based on their visual appearances (apparently using color and shape information) of photos for the textual queries.

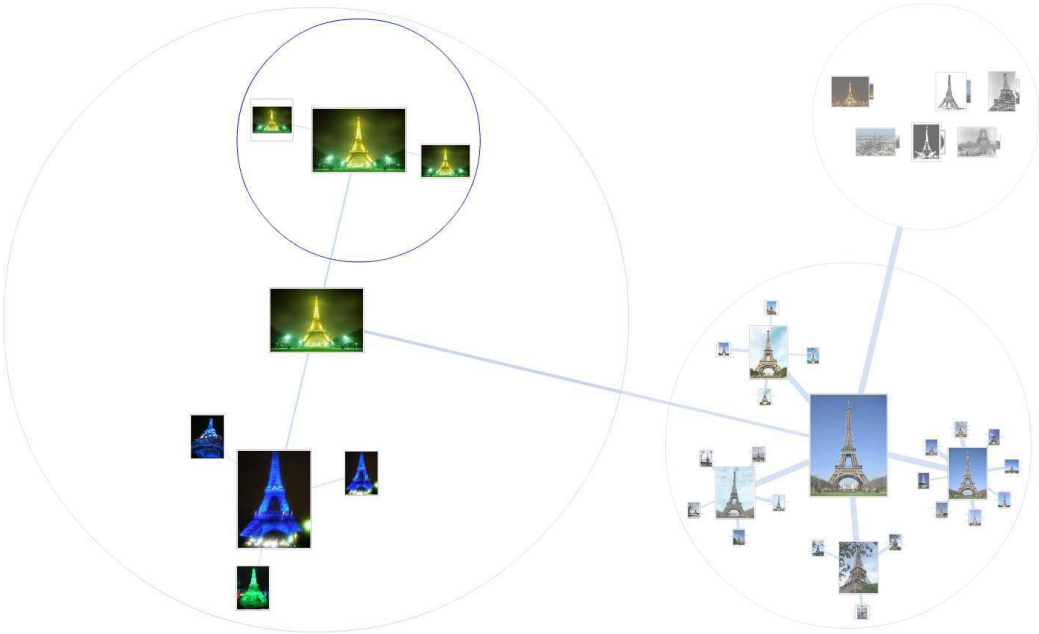

Figure 8.1: Photo clusters for the query "Eiffel tower" based on their visual appearances (Google Image Swirl).

This function enables users a quicker way to browse and visualize the result of searched images. Therefore, we would like to address this problem within our graph-based framework by performing a *"graph clustering"* algorithm on a set of visual graph. This can be done thanks to the pre-computed similarity values between pairs of image graphs.

Towards a sketchable user interface (UI) for graph retrieval

With the current proposal, the image search system allows users to query the image collection using an image (or a group of images). There is a system that allows user to express their ideas by drawing any shape and picking colors

from a limited selection, such as Retrievr[1] system (see Figure 8.2). One possible direction for the graph-based framework application is that we can design an UI that allows user to generate their own visual graph providing a set of annotated visual concepts from the visual vocabulary and a set of relations. Concept and relation can be assigned with a weight/probability based on their importance. Finally, users can arrange these inputs to form a visual graph/subgraph and query it against the visual graph database.

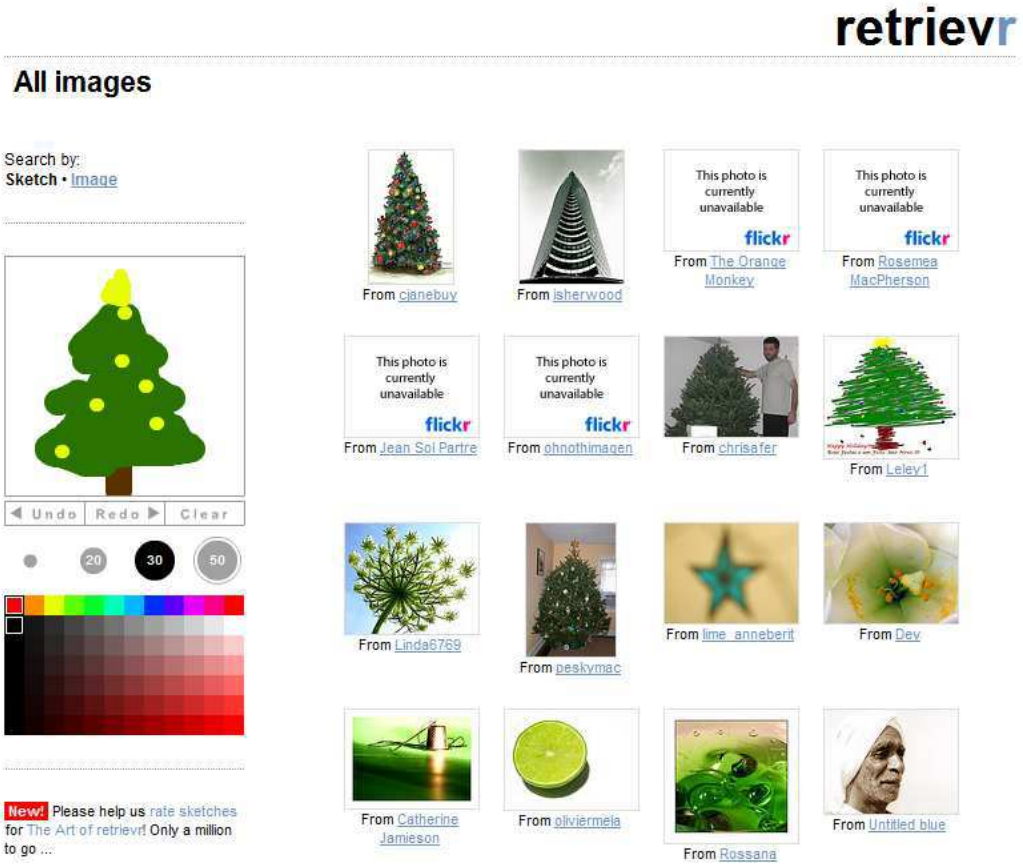

Figure 8.2: Retrievr's user interface allows user retrieving photos by drawing a simple sketch.

Finally, we believe that the work achieved in this thesis as well as the future works will allow to create image retrieval systems with *better quality*, *easier to extend* and *more interactive*.

[1] http://labs.systemone.at/retrievr/

Appendix A: Publication of the Author

Journal Peer-reviewed Articles

1. Trong-Ton Pham, Philippe Mulhem, Loic Maisonnasse, Eric Gaussier, Joo-Hwee Lim. Visual Graph Modeling for Scene Recognition and Robot Localization. Journal on Multimedia Tools and Applications, pages 20, Springer, January 2011.

2. Trong-Ton Pham, Loic Maisonnasse, Philippe Mulhem, Eric Gaussier. Modèle de graphe et modèle de langue pour la reconnaissance de scènes visuelles. Numéro spécial du revu Document Numérique, Vol 13 (211-228), Lavoisier, Juin 2010.

International Peer-reviewed Conference Articles

1. Trong-Ton Pham, Philippe Mulhem, Loic Maisonnasse. Spatial Relationships in Visual Graph Modeling for Image Categorization. Proceedings of the 33rd ACM SIGIR'10, pages 729-730, Geneva, Switzerland, 2010.

2. Trong-Ton Pham, Philippe Mulhem, Loic Maisonnasse, Eric Gaussier, Ali Ait-Bachir. Integration of Spatial Relationship in Visual Language Model for Scene Retrieval. IEEE 8th International Workshop on Content-Based Multimedia Indexing (CBMI), pages 6, Grenoble, France, 2010.

3. Trong-Ton Pham, Loic Maisonnasse, Philippe Mulhem, Eric Gaussier. Visual Language Model for Scene Recognition. Singaporean-French IPAL Symposium (SinFra'09), Singapore, 2009.

4. Trong-Ton Pham, Nicolas Maillot, Joo-Hwee Lim, Jean-Pierre Chevallet. Latent Semantic Fusion Model for Image Retrieval and Annotation. ACM 16th Conference on Information and Knowledge Management (CIKM), pages 439-444, Lisboa, Portugal, 2007.

National Peer-reviewed Conference Articles

1. Trong-Ton Pham, Philippe Mulhem et Loic Maisonnasse. Relations explicites entre différentes représentations d'image dans un modèle de graphe visuel. Actes de la conférence CORIA, pages 211-222, Sousse, Tunisie, 2010.

2. Trong-Ton Pham, Loic Maisonnasse, Philippe Mulhem, Eric Gaussier. Modèle de langue visuel pour la reconnaissance de scènes. Actes de la conférence CORIA, pages 99-112, Giens, France, 2009.

3. Trong-Ton Pham, Jean-Pierre Chevallet, Joo-Hwee Lim. Fusion de multi-modalités et réduction par sémantique latente: Application à la recherche de documents multimédia et à l'annotation automatique d'images. Actes de la conférence CORIA, pages 39-53, Tregastel, France, 2008.

Working Notes, Book chapter

1. Trong-Ton Pham, Philippe Mulhem, Loic Maisonnasse, Jean-Pierre Chevallet, Georges Quenot and Rami Albatal. MRIM-LIG at ImageCLEF 2009: Robot Vision, Image annotation and retrieval tasks. Lecture Notes for Computer Science, pages 324-331, Springer, 2010.

2. Trong-Ton Pham, Loic Maisonnasse and Philippe Mulhem. Visual Language Modeling for Mobile Localization: LIG Participation in RobotVision'09. Working Notes for ImageCLEF '09, Corfu, Greece, 2009.

Bibliography

[Aksoy 2006] Selim Aksoy. *Modeling of Remote Sensing Image Content using Attributed Relational Graphs*. In IAPR International Workshop on Structural and Syntactic Pattern Recognition, volume 4109, pages 75–483, 2006.

[Bach *et al.* 2004] F. Bach, R. Thibaux and M. I. Jordan. *Computing regularization paths for learning multiple kernels*. Advances in Neural Information Processing Systems (NIPS), 2004.

[Ballard & Brown 1982] Dana Ballard and Chris Brown. Computer vision. Prentice-Hall, 1982.

[Belongie & Malik 2000] Serge Belongie and Jitendra Malik. *Matching with Shape Contexts*. In Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries, page 20, 2000.

[Belongie *et al.* 2002] S. Belongie, J. Malik and J. Puzicha. *Shape Matching and Object Recognition Using Shape Contexts*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pages 509–522, 2002.

[Blei *et al.* 2003] D. Blei, A. Ng and M. Jordan. *Latent Dirichlet Allocation*. Journal of Machine Learning Research, pages 993–1022, January 2003.

[Blei 2004] D. Blei. *Probabilistic Models of Text and Images*. PhD thesis, U.C. Berkeley, 2004.

[Boutell *et al.* 2007] M. R. Boutell, J. Luo and C. M. Brown. *Scene Parsing Using Region-Based Generative Models*. IEEE Transactions on Multimedia, vol. 9, no. 1, pages 136–146, 2007.

[Carson *et al.* 1999] Chad Carson, Megan Thomas, Serge Belongie, Joseph M. Hellerstein and Jitendra Malik. *Blobworld: A system for region-based image indexing and retrieval*. In Third International Conference on Visual Information Systems. Springer, 1999.

[Chua *et al.* 1997] T.-S. Chua, K.-L. Tan and B. C. Ooi. *Fast Signature-Based Color-Spatial Image Retrieval*. In ICMCS 1997, pages 362–369, 1997.

[Comaniciu & Meer 2002] Dorin Comaniciu and Peter Meer. *Mean Shift: A Robust Approach Toward Feature Space Analysis*. PAMI, vol. 24, no. 5, pages 603–619, May 2002.

[Cordella *et al.* 1998] L.P. Cordella, P. Foggia, C. Sansone, F. Tortorella and M. Vento. *Graph Matching: A Fast Algorithm and its Evaluation.* Proc. of the 14th International Conference on Pattern Recognition, Brisbane, Australia, pages 1582–1584, 1998.

[Datta *et al.* 2008] R. Datta, D. Joshi, J. Li and J. Z. Wang. *Image Retrieval: Ideas, Influences, and Trends of the New Age*. ACM Computing Surveys, vol. 40, no. 2, pages 1–60, 2008.

[Deerwester *et al.* 1990] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas and Richard A. Harshman. *Indexing by Latent Semantic Analysis*. JASIS, vol. 41, no. 6, pages 391–407, 1990.

[Duffy & Crowley 2000] N. Duffy and James L. Crowley. *Object Detection Using Colour*. In International Conference on Pattern Recognition, pages 1700–1706, Barcelona, 2000.

[Egenhofer & Herring 1991] M. Egenhofer and J. Herring. *Categorizing binary topological relationships between regions, lines and points in geographic databases*. In A Framework for the Definition of Topological Relationships and an Approach to Spatial Reasoning within this Framework, Santa Barbara, CA, 1991.

[Fei-Fei & Perona 2005] Li. Fei-Fei and P. Perona. *A Bayesian Hierarchical Model for Learning Natural Scene Categories*. CVPR, pages 524–531, 2005.

[Felzenszwalb & Huttenlocher 2004] P. F. Felzenszwalb and D. P. Huttenlocher. *Efficient Graph-Based Image Segmentation*. International Journal of Computer Vision, vol. 59, no. 2, pages 167–181, 2004.

[Feng *et al.* 2004] S. Feng, V. Lavrenko and R. Manmatha. *Multiple Bernoulli Relevance Models for Image and Video Annotation.* CVPR'04, 2004.

[Ferrari *et al.* 2010] Vittorio Ferrari, Frédéric Jurie and Cordelia Schmid. *From images to shape models for object detection*. International Journal of Computer Vision, vol. 87, no. 3, pages 284–303, 2010.

[Flickner *et al.* 1995] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele and Peter Yanker. *Query by Image and Video Content: The QBIC System*. Computer, vol. 28, no. 9, pages 23–32, 1995.

[Freeman 1974] H. Freeman. *Computer processing of line-drawing images*. Computing Surveys, vol. 6, pages 57–97, 1974.

[Gao *et al.* 2004] Jianfeng Gao, Jian-Yun Nie, Guangyuan Wu and Guihong Cao. *Dependence language model for information retrieval*. In Research and Development in Information Retrieval, pages 170–177, 2004.

[Gorban *et al.* 2007] A. Gorban, B. Kegl, D. Wunsch and A. Zinovyev. *Principal Manifolds for Data Visualisation and Dimension Reduction*. Springer, Berlin Heidelberg, vol. 58, 2007.

[Gusfield 1997] D. Gusfield. Algorithms on strings, trees, and sequences. Cambridge Univ. Press, 1997.

[Harchaoui & Bach 2007] Z. Harchaoui and F. Bach. *Image classification with segmentation graph kernels*. Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), page 8, 2007.

[Hare & Lewis 2005] J. S. Hare and P. H. Lewis. *Saliency-based Models of Image Content and their Application to Auto-Annotation by Semantic Propagation (Speech)*. Proceedings of Multimedia and the Semantic Web, 2005.

[Harris & Stephens 1988] C. Harris and M. Stephens. *A combined corner and edge detector*. Proceedings of the 4th Alvey Vision Conference, pages 147–151, 1988.

[Hofmann & Puzicha 1998] Thomas Hofmann and Jan Puzicha. *Statistical Models for Co-occurrence Data*. A.I. Memo No. 1625, page 8, 1998.

[Hofmann 1999] Thomas Hofmann. *Probabilistic latent semantic indexing*. In ACM conference on Research and development in information retrieval, pages 50–57, 1999.

[Iwayama & Tokunaga 1995] Makoto Iwayama and Takenobu Tokunaga. *Hierarchical Bayesian clustering for automatic text classification*. In Proceedings of the 14th international joint conference on Artificial intelligence, pages 1322–1327, 1995.

[Jelinek *et al.* 1991] F. Jelinek, R.L. Mercer and S. Roukos. *Principles of Lexical Language Modeling for Speech Recognition*. In Advances in Speech Signal Processing, pages 651–700, 1991.

[Jelinek 1998] JF. Jelinek. Statistical methods for speech recognition. MIT Press, Boston, 1998.

[Jiang *et al.* 2001] Xiaoyi Jiang, Andreas Müunger and Horst Bunke. *On Median Graphs: Properties, Algorithms, and Applications*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 10, pages 1144–1151, 2001.

[Kostin *et al.* 2005] Alexey Kostin, Josef Kittler and William Christmas. *Object recognition by symmetrised graph matching using relaxation labelling with an inhibitory mechanism*. Pattern Recogn. Lett., vol. 26, no. 3, pages 381–393, 2005.

[Lazebnik *et al.* 2006] S. Lazebnik, C. Schmid and J. Ponce. *Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories*. In CVPR, pages 2169–2178, 2006.

[Lee *et al.* 2006] Changki Lee, Gary Geunbae Lee and Myung Gil Jang. *Dependency structure language model for information retrieval*. In ETRI journal, 2006.

[Li & Wang 2003] J. Li and J. Wang. *Automatic linguistic indexing of pictures by a statistical modeling approach*. In IEEE Trans. on Pattern Analysis and Machine Intelligence, pages 1075–1088, 2003.

[Lienhart *et al.* 2009] Rainer Lienhart, Stefan Romberg and Eva Hörster. *Multilayer pLSA for multimodal image retrieval*. In CIVR '09: Proceeding of the ACM International Conference on Image and Video Retrieval, pages 1–8, 2009.

[Lim & Jin 2005] Joo-Hwee Lim and Jesse S. Jin. *A structured learning framework for content-based image indexing and visual query*. In Multimedia System, volume 10, pages 317–331. Springer-Verlag, 2005.

[Lim *et al.* 2007] J. Lim, Y. Li, Y. You and J. Chevallet. *Scene Recognition with Camera Phones for Tourist Information Access*. In ICME, pages 100 – 103, 2007.

[Lowe 1999] David G. Lowe. *Object Recognition from Local Scale-Invariant Features*. In Proc. of the International Conference on Computer Vision ICCV, Corfu, pages 1150–1157, 1999.

[Lowe 2004] David G. Lowe. *Distinctive image features from scale-invariant keypoints*. In International Journal of Computer Vision, pages 91–110, 2004.

[Lu *et al.* 2010] Zhiwu Lu, Yuxin Peng and Horace Ip. *Image categorization via robust pLSA*. Pattern Recognition Letter, vol. 31, no. 1, pages 36–43, 2010.

[Luo *et al.* 2006] J. Luo, A. Pronobis, B. Caputo and P. Jensfelt. *The KTH-IDOL2 database*. In Technical Report, Kungliga Tekniska Hoegskolan, CVAP/CAS, 2006.

[Maillot 2005] Nicolas Eric Maillot. *Ontology Based Object Learning and Recognition*. PhD thesis, Universite de Nice - Sophia Antipolis, 2005.

[Maisonnasse *et al.* 2008] Loic Maisonnasse, Eric Gaussier and Jean-Pierre Chevallet. *Multiplying Concept Sources for Graph Modeling*. In Advances in Multilingual and Multimodal Information Retrieval, pages 585–592. Springer-Verlag., 2008.

[Maisonnasse *et al.* 2009] Loic Maisonnasse, Eric Gaussier and Jean-Pierre Chevallet. *Model Fusion in Conceptual Language Modeling*. In 31st European Conference on Information Retrieval (ECIR 09), pages 240–251, Toulouse (France), 2009.

[Manning *et al.* 2009] C. D. Manning, P. Raghavan and H. Schtze. *Language Models for Information Retrieval*. In An Introduction to Information Retrieval, pages 237–252. Cambridge University Press, 2009.

[Marr 1982] D. Marr. Vision: A computational investigation into the human representation and processing of visual information. W. H. Freeman, 1982.

[Mikolajczyk & Schmid 2002] Krystian Mikolajczyk and Cordelia Schmid. *An affine invariant interest point detector*. In ECCV, pages 128–142. Springer, 2002. Copenhagen.

[Monay & Gatica-Perez 2003] Florent Monay and Daniel Gatica-Perez. *On image auto-annotation with latent space models*. In ACM Multimedia, pages 275–278, 2003.

[Monay & Gatica-Perez 2004] F. Monay and D. Gatica-Perez. *PLSA-based Image Auto-Annotation: Constraining the Latent Space*. In Proc. ACM Int. Conf. on Multimedia, pages 348–351, 2004.

[Monay & Gatica-Perez 2007] Florent Monay and Daniel Gatica-Perez. *Modeling Semantic Aspects for Cross-Media Image Indexing*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 10, pages 1802–1817, 2007.

[Moore 1998] Andrew Moore. *Very Fast EM-based Mixture Model Clustering Using Multiresolution kd-trees*. In Advances in Neural Information Processing Systems 11, pages 543–549. MIT Press, 1998.

[Mulhem *et al.* 2001] Philippe Mulhem, Wee Kheng Leow and Yoong Keok Lee. *Fuzzy conceptual graphs for matching images of natural scenes*. In Proceedings of the 17th international joint conference on Artificial intelligence, pages 1397–1402, 2001.

[Ounis & Pasca 1998] Iadh Ounis and Marius Pasca. *RELIEF: Combining Expressiveness and Rapidity into a Single System*. In ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia, pages 266–274. ACM, 1998.

[Pham *et al.* 2007] Trong-Ton Pham, Nicolas Maillot, Joo-Hwee Lim and Jean-Pierre Chevallet. *Latent semantic fusion model for image retrieval and annotation*. In CIKM, pages 439–444, 2007.

[Pham *et al.* 2009] T. T. Pham, L. Maisonnasse and P. Mulhem. *Visual Language Modeling for Mobile Localization: LIG Participation in RobotVision09*. In CLEF working notes 2009, Corfu, Greece, 2009.

[Pham *et al.* 2010] Trong-Ton Pham, Loic Maisonnasse, Philippe Mulhem and Eric Gaussier. *Integration of Spatial Relationship in Visual Language Model for Scene Retrieval*. In 8th IEEE Int. Workshop on Content-Based Multimedia Indexing, page 6, 2010.

[Pham 2006] Trong-Ton Pham. *Automatic Image Annotation: Towards a Fusion of Region-based and Saliency-based Models*. Master Thesis, University of Paris 6, 2006.

[Ponte & Croft 1998] J. M. Ponte and W. B. Croft. *A Language Modeling Approach to Information Retrieval*. In Research and Development in Information Retrieval, pages 275–281, 1998.

[Prasad *et al.* 2001] B. G. Prasad, S. K. Gupta and K. K. Biswas. *Color and Shape Index for Region-Based Image Retrieval*. In Proceedings of the 4th International Workshop on Visual Form, pages 716–728, 2001.

[Pronobis *et al.* 2008] A. Pronobis, O. Martnez Mozos and B. Caputo. *SVM-based discriminative accumulation scheme for place recognition*. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA08), pages 522–529, 2008.

[Quelhas *et al.* 2007] Pedro Quelhas, Florent Monay, Jean-Marc Odobez, Daniel Gatica-Perez and Tinne Tuytelaars. *A Thousand Words in a Scene*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 9, pages 1575–1589, 2007.

[Rosenfeld 2000] R. Rosenfeld. *Two Decades Of Statistical Language Modeling: Where Do We Go From Here?* Proceedings of the IEEE, vol. 88, no. 8, pages 1270–1278, 2000.

[Schmid & Mohr 1997] Cordelia Schmid and Roger Mohr. *Local Greyvalue Invariants for Image Retrieval*. Pattern Analysis and Machine Intelligence, vol. 19, no. 5, May 1997.

[Shakhnarovich *et al.* 2005] G. Shakhnarovich, T. Darrell and P. Indyk. Nearest-neighbor methods in learning and vision. MIT Press, 2005.

[Shawe-Taylor & Cristianini 2004] J. Shawe-Taylor and N. Cristianini. Kernel methods for pattern analysis. Cambridge Univ. Press, 2004.

[Shi *et al.* 1998] J. Shi, S. Belongie, T. Leung and J. Malik. *Image and Video Segmentation: The Normalized Cut Framework*. IEEE Int'l Conf on Image Processing, pages 943–947, 1998.

[Shokoufandeh *et al.* 2002] Ali Shokoufandeh, Sven Dickinson, Clas Jnsson, Lars Bretzner and Tony Lindeberg. *On the Representation and Matching of Qualitative Shape at Multiple Scales*. Proceedings of ECCV, pages 759–775, 2002.

[Sivic & Zisserman 2003] J. Sivic and A. Zisserman. *Video Google: A Text Retrieval Approach to Object Matching in Videos*. In Proceedings of the International Conference on Computer Vision, volume 2, pages 1470–1477, October 2003.

[Smeulders *et al.* 2000] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta and Ramesh Jain. *Content-Based Image Retrieval at the End of the Early Years*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 12, pages 1349–1380, 2000.

[Smith & Chang 1996] J. R. Smith and S. F. Chang. *VisualSEEk: a fully automated content-based image query system*. In Proceedings ACM MM, pages 87–98, 1996.

[Souvannavong *et al.* 2004] Fabrice Souvannavong, Bernard Mérialdo and Benoit Huet. *Improved Video Content Indexing by Multiple Latent Semantic Analysis*. In CIVR, pages 483–490, 2004.

[Sowa 1984] J. F. Sowa. Conceptual structures: information processing in mind and machine. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1984.

[Suard *et al.* 2005] F. Suard, V. Guigue, A. Rakotomamonjy, and A. Benshrair. *Pedestrian detection using stereo-vision and graph kernels*. IEEE Symposium on Intelligent Vehicule, 2005.

[Swain & Ballard 1991] Michael J. Swain and Dana H. Ballard. *Color indexing*. International Journal of Computer Vision, vol. 7, pages 11–32, 1991.

[Tabbone *et al.* 2006] S. Tabbone, L. Wendling and J.-P. Salmon. *A new shape descriptor defined on the Radon transform*. Computer Vision and Image Understanding, vol. 102, pages 42–51, 2006.

[Tirilly *et al.* 2008] Pierre Tirilly, Vincent Claveau and Patrick Gros. *Language modeling for bag-of-visual words image categorization*. ACM International Conference on Image and Video Retrieval (CIVR), pages 249–258, 2008.

[Ullmann 1976] J. R. Ullmann. *An algorithm for subgraph isomorphism*. Journal of the ACM, pages 31–42, 1976.

[Vapnik 1995] Vladimir Vapnik. The nature of statistical learning theory. Springer-Verlag, 1995.

[Won *et al.* 2002] Chee Sun Won, Dong Kwon Park and Soo-Jun Park. *Efficient Use of MPEG-7 Edge Histogram Descriptor*. ETRI Journal, vol. 24, no. 1, pages 23–30, 2002.

[Wu *et al.* 2007] Lei Wu, Mingjing Li, Zhiwei Li, Wei-Ying Ma and Nenghai Yu. *Visual language modeling for image classification.* In Proceedings of the international workshop on multimedia information retrieval, pages 115–124, 2007.

[Zhang & Chang 2004] D. Q. Zhang and S. F. Chang. *Detecting Image Near-Duplicate by Stochastic Attributed Relational Graph Matching with Learning*. In ACM conference of Multimedia, pages 877–884, 2004.

[Zhang & Lu 2001] Dengsheng Zhang and Guojun Lu. *Content-Based Shape Retrieval Using Different Shape Descriptors: A Comparative Study*. IEEE International Conference on Multimedia and Expo, pages 1139–1142, 2001.

[Ziou & Tabbone 1998] Djemel Ziou and Salvatore Tabbone. *Edge Detection Techniques - An Overview*. International Journal of Pattern Recognition and Image Analysis, vol. 8, pages 537–559, 1998.