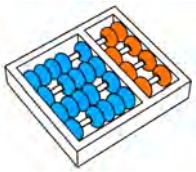
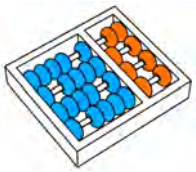



Otávio Augusto Bizetto Penatti

# "Image and Video Representations based on Visual Dictionaries"

# "*Representações de Imagens e Vídeos baseadas em Dicionários Visuais*"

**CAMPINAS**
**2012**

**University of Campinas**
**Institute of Computing**

***Universidade Estadual de Campinas***
***Instituto de Computação***

**Otávio Augusto Bizetto Penatti**

# "Image and Video Representations based on Visual Dictionaries"

Supervisor:
*Orientador(a):* **Prof. Dr. Ricardo da Silva Torres**

# *"Representações de Imagens e Vídeos baseadas em Dicionários Visuais"*

PhD Thesis presented to the Post Graduate Program of the Institute of Computing of the University of Campinas to obtain a PhD degree in Computer Science.

*Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Computação da Universidade Estadual de Campinas para obtenção do título de Doutor em Ciência da Computação.*

This volume corresponds to the final version of the Thesis defended by Otávio Augusto Bizetto Penatti, under the supervision of Prof. Dr. Ricardo da Silva Torres.

Este exemplar corresponde à versão final da Tese defendida por Otávio Augusto Bizetto Penatti, sob orientação de Prof. Dr. Ricardo da Silva Torres.

---

Supervisor's signature / *Assinatura do Orientador(a)*

CAMPINAS
2012

Institute of Computing /*Instituto de Computação*
University of Campinas /*Universidade Estadual de Campinas*

# Image and Video Representations based on Visual Dictionaries

**Otávio Augusto Bizetto Penatti**[1]

November 29, 2012

**Examiner Board/*Banca Examinadora*:**

- Prof. Dr. Ricardo da Silva Torres (Supervisor/*Orientador*)
- Prof. Dr. Siome Klein Goldenstein
  Institute of Computing - UNICAMP
- Prof. Dr. Hélio Pedrini
  Institute of Computing - UNICAMP
- Prof. Dr. Agma Juci Machado Traina
  Institute of Mathematics and Computer Sciences - USP-São Carlos
- Prof. Dr. Marcos André Gonçalves
  Department of Computer Science - UFMG
- Prof. Dr. Alexandre Xavier Falcão
  Institute of Computing - UNICAMP (Substitute/*Suplente*)
- Prof. Dr. Eduardo Alves do Valle Junior
  School of Electrical and Computer Engineering - UNICAMP (Substitute/*Suplente*)
- Prof. Dr. Jussara M. Almeida
  Department of Computer Science - UFMG (Substitute/*Suplente*)

[1]Financial support: FAPESP scholarship (process 2009/10554-8) 2009–2012, CAPES scholarship 2009

# Abstract

Effectively encoding visual properties from multimedia content is challenging. One popular approach to deal with this challenge is the visual dictionary model. In this model, images are handled as an unordered set of local features being represented by the so-called bag-of-(visual-)words vector. In this thesis, we work on three research problems related to the visual dictionary model.

The first research problem is concerned with the generalization power of dictionaries, which is related to the ability of representing well images from one dataset even using a dictionary created over other dataset, or using a dictionary created on small dataset samples. We perform experiments in closed datasets, as well as in a Web environment. Obtained results suggest that diverse samples in terms of appearances are enough to generate a good dictionary.

The second research problem is related to the importance of the spatial information of visual words in the image space, which could be crucial to distinguish types of objects and scenes. The traditional pooling methods usually discard the spatial configuration of visual words in the image. We have proposed a pooling method, named Word Spatial Arrangement (WSA), which encodes the relative position of visual words in the image, having the advantage of generating more compact feature vectors than most of the existing spatial pooling strategies. Experiments for image retrieval show that WSA outperforms the most popular spatial pooling method, the Spatial Pyramids.

The third research problem under investigation in this thesis is related to the lack of semantic information in the visual dictionary model. We show that the problem of having no semantics in the space of low-level descriptions is reduced when we move to the bag-of-words representation. However, even in the bag-of-words space, we show that there is little separability between distance distributions of different semantic concepts. Therefore, we question about moving one step further and propose a representation based on visual words which carry more semantics, according to the human visual perception. We have proposed a bag-of-prototypes model, according to which the prototypes are the elements containing more semantics. This approach goes in the direction of reducing the so-called semantic gap problem. We propose a dictionary based on scenes, that is used

for video representation in experiments for video geocoding. Video geocoding is the task of assigning a geographic location to a given video. The evaluation was performed in the context of the Placing Task of the MediaEval challenge and the proposed bag-of-scenes model has shown promising performance.

# Resumo

Codificar de maneira eficaz as propriedades visuais de conteúdo multimídia é um desafio. Uma abordagem popular para tratar esse desafio consiste no modelo de dicionários visuais. Neste modelo, imagens são consideradas como um conjunto desordenado de características locais e são representadas por um saco de palavras visuais (*bag of visual words*). Nesta tese, trabalhamos em três problemas de pesquisa relacionados ao modelo de dicionários visuais.

O primeiro deles é relacionado ao poder de generalização dos dicionários visuais, que se refere à capacidade de criar boas representações para imagens de uma dada coleção mesmo usando um dicionário criado sobre outra coleção ou usando um dicionário criado sobre pequenas amostras da coleção. Experimentos foram realizados em coleções fechadas de imagens e em um ambiente Web. Os resultados obtidos sugerem que o uso de amostras diversas em termos de aparência é suficiente para se gerar bons dicionários.

O segundo problema de pesquisa é relacionado à importância da informação espacial das palavras visuais no espaço da imagem. Esta informação pode ser fundamental para diferenciar tipos de objetos e cenas. As técnicas mais comuns de *pooling* normalmente descartam a configuração espacial das palavras visuais na imagem. Propomos uma nova técnica de *pooling*, chamada de *Word Spatial Arrangement (WSA)*, que codifica a posição relativa das palavras visuais na imagem e tem a vantagem de gerar vetores de características mais compactos do que a maioria das técnicas de *pooling* espacial existentes. Experimentos em recuperação de imagens mostram que o WSA supera em eficácia a técnica mais popular de *pooling* espacial, as pirâmides espaciais.

O terceiro problema de pesquisa em investigação nesta tese é relacionado à falta de informação semântica no modelo de dicionários visuais. Mostramos que o problema de não haver semântica no espaço de características de baixo nível é reduzido ao passarmos para o espaço das representações baseadas em sacos de palavras visuais. Contudo, mesmo no espaço destas representações, mostramos que existe pouca separabilidade entre distribuições de distância de conceitos semânticos diferentes. Portanto, questionamos sobre passar para um novo espaço e propomos uma representação baseada em palavras visuais que carreguem mais semântica de acordo com a percepção visual humana. Propomos

um modelo de saco de protótipos, segundo o qual os protótipos são elementos com maior significado. Esta abordagem tem potencial para reduzir a chamada lacuna semântica entre a interpretação do usuário sobre uma imagem e a sua representação. Propomos um dicionário baseado em cenas, que é usado para representar vídeos em experimentos de geo-localização. Geo-localização de vídeos é a tarefa de atribuir uma posição geográfica para um dado vídeo. A avaliação foi conduzida no contexto da Placing Task da competição MediaEval e o modelo proposto mostrou resultados promissores.

# Acknowledgements

I would like to thank FAPESP (grant number 2009/10554-8), CAPES, CNPq, AMD, and Microsoft Research for financial support.

I am so glad to have my family by my side, specially my father Vainer and my mother Marlene. They were very important for my emotional support. The weekends at my hometown, Santa Bárbara d'Oeste, were important to refresh my mind and to be ready for going on with my research. The delicious cookings of my mother were a gift. My father's rides to the airport, when traveling to conferences, were also very helpful.

I also would like to thank my advisor, professor Ricardo Torres, for his guidance and patience. Besides being very important for the research itself, his way of supervising my work was very motivational. Meetings with him were able to clarify ideas and to make the work to progress smoothly. He is full of ideas and very inspiring. I am also glad about the supervision from professor Eduardo Valle. I could learn a lot about experimental design, statistics, and how to elaborate a good discussion when writing papers.

I thank professor Valerie Gouet-Brunet for receiving me at the Cedric-CNAM, Paris, France, for about one month in 2011. She was very receptive and opened to discussions. We could talk many times during that month on the work about WSA. She was very important for the experiments in the retrieval scenario. I also thank professors Terry Boult and Walter Scheirer for receiving me at the University of Colorado at Colorado Springs (UCCS), during three months in 2012. I thank them for being patient when receiving me as I was, most of the time, writing this thesis. Their experience in machine learning was important to make me better in this field. They were also very hospitable and we could know a lot of nice places in Colorado.

I am also glad for having good friends during this time. I specially thank the colleagues from the Recod lab, Jefersson dos Santos, Ricardo Panaggio, Fábio Faria, Rodrigo Tripodi, Guilherme Armigliatto, and Felipe Sansão. We could have scientific discussions that were important for clarifying ideas. Friends were also very important outside the lab, where we could have fun and extra activities. I also thank the good research collaboration with Jurandy Almeida and Lin Tzy Li in the work with the bag-of-scenes model.

# Abbreviations

$\alpha$ Assignment vector resulting from the coding step; in some cases, it may refer to the confidence level to compute confidence intervals

**BoW** Bag-of-words (or bag-of-visual-words)

**BIC** Border/interior pixel classification descriptor

**CBIR** Content-based Image Retrieval

**CEDD** Color and edge directivity descriptor

**MAP** Mean average precision

**nTrain** Number of training samples per class used in a classification experiment

**P@N** Precision measure for the top $N$ retrieved images

$\sigma$ Parameter that indicates the softness of a soft assignment; in some cases, it is the scale of a point detected in the image, by sparse or dense sampling

**SPM** Spatial pyramid match

**SIFT** Scale-invariant feature transform

**SVM** Support vector machines

**UNC** Codeword uncertainty scheme of soft assignment

**WSA** Word spatial arrangement

# Glossary

**Assignment** Step of associating the feature vector of a point detected in the image with the visual words in the dictionary; this step is also referred as *coding*

**Bag of prototypes** Image or video representation in which the visual words are elements containing more semantics (prototypes)

**Bag of scenes** Similar representation to a bag of words, except for the fact that the visual words are scenes (whole pictures); this representation is based on a dictionary of scenes

**Bag of words** Image representation containing statistical information about the occurrences of the visual words in an image; this representation is based on a visual dictionary

**Codeword** One element of a visual dictionary; a visual word

**Coding** Process of representing the image descriptions in the visual dictionary space (quantized space); this can be seen as an *assignment* step

**Dense sampling** Sampling scheme where regions in an image are obtained by using a dense grid, discarding its content

**Dictionary of scenes** A visual dictionary where the visual words are scenes (whole pictures)

**Distance function** Function used to compare feature vectors

**Feature space** Space defined by a certain type of feature; each feature vector is a point in that space

**Feature space quantization** Action of reducing variations in a feature space; ranges of values in the original feature space are converted to a single value in the quantized space

**Feature vector** Vector describing a digital element (e.g., image or video); this vector contains information about one or more aspects of the digital element

**Hard assignment** Assignment scheme where a feature vector is assigned to only one visual word in the dictionary

**Image classification** Task of assigning a class/category to a given test image

**Image descriptor** Algorithm used to extract a feature vector from a given image or image region; it is also composed of a distance function suitable to compare feature vectors

**Image retrieval** Task of retrieving a ranked list of relevant images in relation to a given query image

**Interest-point detector** Algorithm to detect regions in an image; usually those algorithms detect points in regions of high differences of contrast and brightness

**Pooling** Strategy for summarizing/selecting the assignment values from the coding/assignment step, generating the image feature vector

**Soft assignment** Assignment scheme where a feature vector can be assigned to more than one visual word in the dictionary

**Sparse sampling** Sampling scheme where regions in an image are obtained by using an interest-point detector

**Video geocoding** Task of assigning geographic locations to videos

**Visual codebook** Other designation for visual dictionary

**Visual dictionary** Result of a feature space quantization; set of regions in the quantized feature space

**Visual word** One element of a visual dictionary; one region in the quantized feature space; a codeword

# Contents

# List of Tables

# List of Figures

# Chapter 1

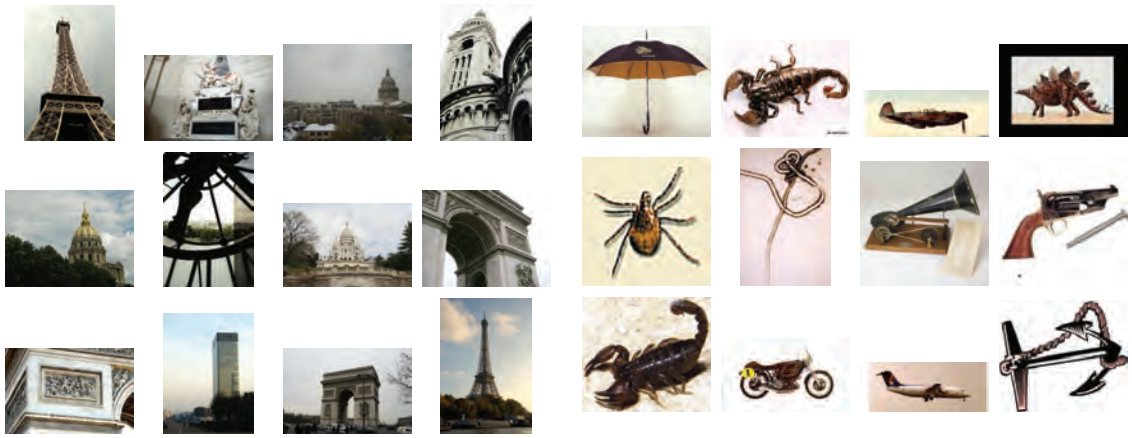
# Introduction

Representing images based only on their content has been challenging researchers and companies for decades. Many steps towards the objective of making a machine able to understand what it sees have been successful, but many others are still necessary in order to obtain satisfactory results in practical situations. This thesis aims at contributing in smoothing the next steps in this direction.

The current advances in technology are changing the way how people live, specially considering the impact brought by the high-speed Internet connections and the image capturing devices. It has become easy to create, share, and access digital information, generating an exponential growth in the availability of visual data. Recently, due to the increasing computational power of digital devices, people are getting in touch with systems based on powerful computer vision approaches. We can notice, for example, the popularity of face recognition algorithms embedded into digital cameras and the trend of mobile applications like Google Goggles[1]. All those kinds of applications employ different types of computer vision techniques and they are very dependent on representing image visual properties effectively.

The challenge of encoding image properties, like color, texture, shape, local properties of objects, and semantic aspects of scenes, for example, has motivated industry and research communities to keep developing new algorithms and methods for representing images. In the beginning of the decade of 1990, several algorithms were proposed to extract color, texture, and shape features from images [8, 18, 71]. Those techniques usually relied on computing a representation that encodes global aspects of images, therefore called *global descriptors*. Global descriptors have the advantage of being simple to compute but they share the deficiency of encoding few local properties of images. They can provide a good general idea of the image content, but for object recognition and more precise applications, they can be less effective, as shown in Figure 1.1. Anyhow, they keep

[1] `www.google.com/mobile/goggles/` (as of February 6th, 2013).



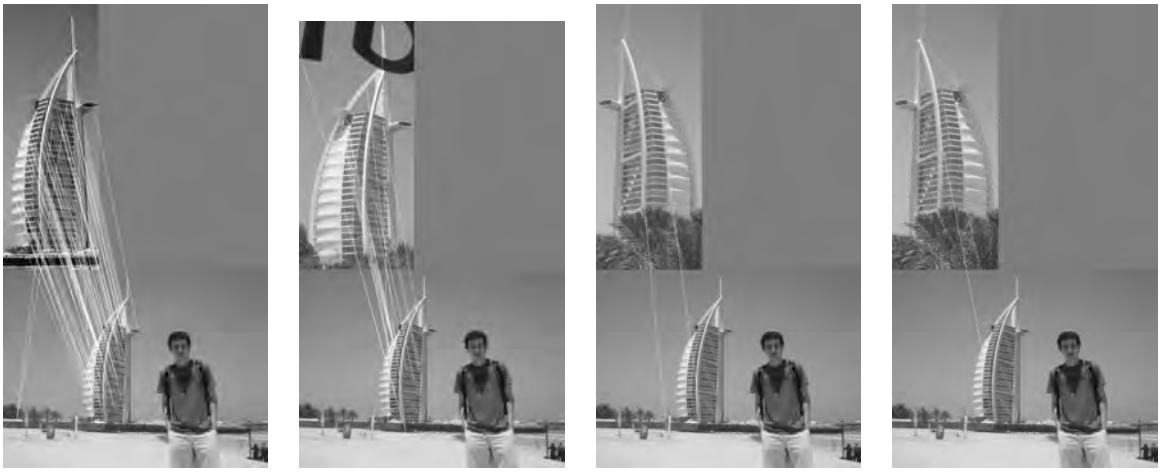
(a) (b)

Figure 1.1: Examples showing the lack of precision in the image representations computed by global descriptors. Images are similar considering color properties but are very different considering their semantics. The examples shown are based on ranking the images which are represented by the BIC global descriptor [17] in the (a) Paris and (b) Caltech-101 datasets.

being a fair choice in modern retrieval and classification systems [7, 19, 40, 58, 68, 84]. For a survey on global color and texture descriptors, readers may refer to [61].

The research community has also developed *local descriptors* [54, 55, 74]. They are usually computed over regions of high differences of contrast and brightness, like corners and edges. Although more powerful to represent local properties, extracting local description from images is more costly and also results in a variable number of feature vectors per image, which makes the comparison between a pair of images more complex. They are also very precise, as we can see in the examples of Figure 1.2: small variations in the objects may avoid similar regions to be considered as a match. Therefore, the use of local descriptors can be limited to some applications like copy detection [42, 72] or object localization [69], for example.

In the year of 2003, a method proposed by Sivic and Zisserman [70] introduced the idea of representing images in a similar fashion as representing text documents. Their approach quickly became a cornerstone for multimedia retrieval and classification systems. As well as a text document is composed of a set of textual words, an image can be analyzed as a set of local appearances. Due to this analogy, they had to change the concept of *word* to a *visual word*. To achieve this, images are decomposed into a set of local patches which are then assigned to a vocabulary of patches, the so-called *visual dictionary*. The visual dictionary is the codebook of the available patches that are used to represent the image content. This approach is based on the use of local descriptors, however, by using



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

Figure 1.2: Local descriptors: example of how the number of matching regions decrease as more transformations are performed in the object of interest, showing the specificity of local descriptors. This example is based on running the matching algorithm of the most popular local descriptor (SIFT [50]) with the default parameters.

the visual dictionary, a single feature vector is generated per image, which is the popular *bag of (visual) words*. Therefore, the visual dictionary model solves the issue of multiple feature vectors per image computed by local descriptors. Another advantage is that the description is more general, eliminating the problem of very precise representations generated by local descriptors, and making the dictionary-based representations useful in a wider range of applications. Figure 1.3 shows the generalization caused by descriptions based on visual dictionaries in relation to pure local descriptions.

The visual dictionary results from the feature space quantization, which is the responsible for increasing the generality of the descriptions. Figure 1.4 shows how a visual dictionary is created. The feature space generated by the local descriptions extracted from images is quantized and each region obtained is a visual word in the dictionary. After that, the local descriptions of an image need to be encoded according to the quantized space, as shown in Figure 1.5. This is performed by assigning to each local descriptor, the label of its nearest region. Then, when all local feature vectors are represented according to the dictionary, the image feature vector is created by summarizing their local vectors.

The process of generating and using the visual dictionary has raised several challenges and this thesis goes in the direction of addressing some of them.

(a) (b)

Figure 1.3: Examples showing the increase in precision in relation to global descriptors and also the increase in generality in relation to local descriptors for the representations obtained when using visual dictionaries. In (a), we show that even when the object of interest suffers large transformations, like illumination, point of view, and scale, the representations remain similar. In (b), we show different instances of objects of the same type, which are considered similar using a visual dictionary representation. The examples shown are based on ranking the images which are represented by the proposed WSA descriptor (see Chapter 4) in the (a) Paris and (b) Caltech-101 datasets.

## 1.1 Hypotheses and research questions

The main hypotheses analyzed in this thesis are the following:

- Visual dictionaries generalize well from one dataset to another, and from a subset of the classes to a whole dataset.
- The spatial information of visual words in the image space is important to distinguish types of scenes and objects.
- The use of semantically enriched dictionaries improves the quality of image and video representations.

The first hypothesis is related to the fact that usually visual dictionaries are generated based on a set of features extracted from images. We question if the set of images used for dictionary generation really makes difference in the dictionary quality. How should we handle the cases in which the set of images is completely different from the images to be represented? What should we do if the image dataset is constantly changing?

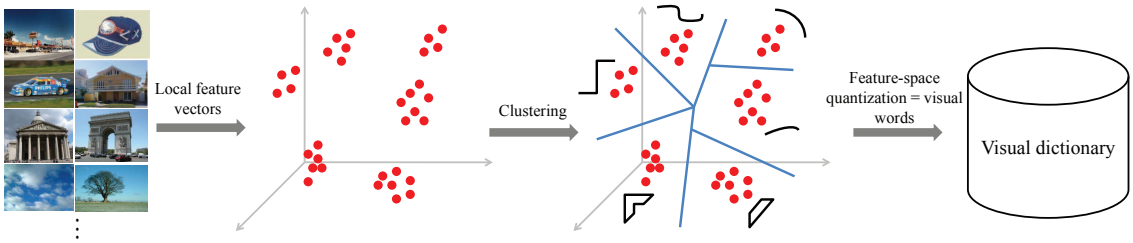



Figure 1.4: Schema to generate a visual dictionary. After extracting local feature vectors from an image dataset, the feature space is quantized and each region corresponds to a visual word.

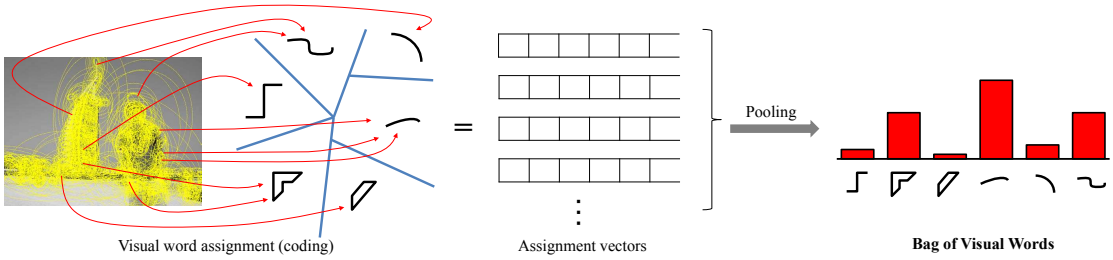



Figure 1.5: Schema to represent an image based on a visual dictionary. Given an input image, its local feature vectors are computed and then assigned to the visual words in the dictionary. Finally, the local assignment vectors are summarized by a pooling strategy, creating the *bag-of-visual-words* representation.

The second hypothesis is based on the fact that if we change the spatial arrangement of image local patches we may also change image semantics. Therefore, how can we encode the spatial arrangement of visual words in an effective and efficient manner?

The third hypothesis relies on the lack of semantics in the visual words of the traditional dictionaries of local patches. We use the term *semantics* to refer to a set of visual properties that carry meanings for humans. Is semantics encoded by image local patches? Furthermore, is there any semantics in the bag-of-words (BoW) representations? And if we have visual words with more semantic information, would the image representations become better?

In the following section, we give the background for the statement of each hypothesis.

## 1.2 Challenges and contributions

During the latest years, the research community has been very active in the field of visual recognition and a plenty of improvements have been made over the visual dictionary model [11, 15, 26, 29, 31, 34, 35, 43, 48, 49, 62–64, 66, 69, 75, 77, 79, 81, 85, 88, 89]. The steps to generate a dictionary and the steps to create a BoW representation are shown in Figures 1.4 and 1.5, respectively, and can be summarized in: low-level feature extraction, feature space quantization, visual word assignment (coding), and pooling.

Our first contribution in this thesis concerns the dictionary creation, which is related to the feature space quantization. The second contribution comprises the pooling strategy, which is based on the results of the assignment step. And the third contribution is based on the challenge of encoding more semantic information into the dictionary.

### 1.2.1 Dictionary creation

The first step to create a visual dictionary is the extraction of low-level features from images, which is usually performed by local descriptors. After extracting such features, the feature space is quantized in order to generate the visual dictionary (codebook). There are many works proposing enhancements in the feature quantization step [34,35,48,79,85], however, k-means is still the most popular choice [9, 14, 26, 58, 75, 77, 84] and some papers use k-means variations to generate better codebooks [15,29]. The use of k-means in a high-dimensional space tends to give no better quantization than a simple random selection of points [35, 79]. This fact challenges us to question about how much effort should be put in the feature space quantization phase. Do we really need to use costly computational techniques to quantize the feature space? Additionally, most of the papers deal with fixed-size and static datasets. In a Web-like environment, how should we deal with the fact that many images are constantly being inserted and removed from the dataset? Does that impact the dictionary quality? The new images would still be well represented by previously created dictionaries?

Our first contribution in this thesis is an analysis of the impact in the image representation when using different sources of information to generate the dictionary. Should a dictionary created on one dataset be good to represent images of another dataset? A similar phenomenon for training and testing learning algorithms is also known as *transfer learning* [56]. We also evaluate the impact of using samples of the dataset in the dictionary quality. Our conclusions point to the direction to alleviate the cost of dictionary generation showing its generalization power and also giving the clues for using visual dictionaries in Web-like environments.

### 1.2.2 Spatial information of visual words

The image representation based on the visual dictionary depends on putting its local descriptions into the quantized feature space. For that, literature presents several coding strategies for assigning a local description to the visual words of the dictionary, like the popular *hard* [70] and *soft* [49, 64, 77] assignments. When all the points in the image have already been assigned to the visual words, a pooling strategy is applied to summarize the set of points in the image into a single feature vector. The traditional BoW vector [70] is simply a histogram of visual words and discards any kind of spatial arrangement regarding the points in the image space. The spatial information of visual words in the image may be crucial to distinguish different types of scenes and objects. In the past, researchers faced the problem of having images with similar color histograms but different semantics [57]. In the BoW representation, we migrate the problem from pixels to local patches. Therefore, literature has a vast range of techniques [11, 15, 26, 29, 43, 62, 66, 88, 89] targeting the encoding of spatial information of visual words in the image space. The most popular approach is based on Spatial Pyramids [43], which simply split the image hierarchically into rectangular tiles. Although they lead to very large improvements on classification experiments, their huge feature vector is a problem in image retrieval applications. Many other approaches suffer from the same problem of generating large feature vectors [15, 66, 69] and some others target specific applications [26, 29, 88, 89].

A second contribution of this thesis is a pooling method that encodes the spatial arrangement of visual words in an image, called *Word Spatial Arrangement (WSA)*. WSA increases the discriminating power of non-spatial pooling approaches keeping one of the BoW strengths, that is the general aspect of the representation. Also, WSA is suitable for both retrieval and classification scenarios and works well in both hard and soft assignments. WSA has the benefits of generating more compact vectors than most of the spatial pooling methods in a compromise of loosing some accuracy in relation to them in the classification scenario. In the retrieval scenario, WSA outperforms the most popular approach to spatial pooling, the Spatial Pyramids [43].

### 1.2.3 Semantic information in visual dictionaries

Another important aspect of visual dictionaries based on local features lies in the fact that visual words carry little or no semantics [34, 44, 48, 74]. Therefore, the term *dictionary* is somewhat misleading, because their words have no meaning for humans. However, the representations based on visual dictionaries are powerful. Thus, when we move from the low-level feature space, composed of local feature vectors, to the mid-level (bag-of-words) space [11], we obtain a semantic separability that makes it possible to distinguish different

types of scenes and objects. What would happen if we move one step further by using a dictionary where the visual words have more semantic information?

A third contribution of this thesis is a study on the semantic separability in the different feature spaces: low-level and mid-level. We analyze the semantic separability between distance distributions considering different semantic classes of points or objects. In the low-level feature space, although we could expect that appearances carry semantics, we show that there is no semantic separability between distance distributions, making it difficult to distinguish local patches by their semantics. In the mid-level space, despite the good results of BoW representations in the literature, we show that the semantic separability between distance distributions in this space is very small, emphasizing the need of having dictionaries based on semantic elements. Finally, we evaluate a representation model based on elements that carry more semantics. We call this model as *bag of prototypes*, according to which the prototypes are visual words containing more semantics. It is a step forward to reduce the semantic gap and to create a representation that is more intuitive for humans [71, 80]. The term *semantic gap* refers to the difference between the user interpretation of an image and the representation computed for that image [71]. Our proposed representation using the bag-of-prototypes model is based on a dictionary of scenes and is called *bag of scenes*. It was evaluated in the context of video geocoding which is the task of assigning a geographic location to videos. The evaluation was performed under the Placing Task [65] of the MediaEval 2011 challenge [40].

## 1.3 Thesis outline

This thesis is organized according to its hypotheses and contributions. Therefore, each contribution is presented in a separate chapter. For the experiments in each chapter, we have selected datasets whose properties make them suitable for the evaluation of each hypothesis. Hence, different datasets were used in the following chapters.

Initially, Chapter 2 gives the background necessary for the understanding of the following chapters. Context-specific related work is covered in each chapter.

Chapter 3 shows the potential generality of visual dictionaries. This important aspect of a dictionary is explored in several experiments, pointing to the feasibility of using visual dictionaries in a Web environment. Experiments in such environment are presented in the last section of the chapter. The analysis of the dictionary generality was reported in an article submitted to the *Image and Vision Computing* journal.

Chapter 4 details the second contribution of this thesis: a pooling method for encoding the spatial arrangement of visual words, called *WSA*. We perform experiments in both retrieval and classification scenarios showing the potential of the proposed method. The initial WSA proposal was published in the *Iberoamerican Congress on Pattern Recognition*

(CIARP) [62], in 2011, receiving the best paper award. The contributions presented in Chapter 4 were submitted to the *Pattern Recognition* journal.

Chapter 5 gives the evidences of why the semantic information is important for visual dictionaries and, consequently, to improve visual recognition. We show experiments to describe the semantic separability between distance distributions in low-level and mid-level feature spaces. We also show the details of the proposed dictionary, which encodes more semantic information than the traditional dictionaries based on local patches. Our *dictionary of scenes* is evaluated in a video geocoding task [65] under the MediaEval 2011 challenge [40]. The bag-of-scenes model presented in that chapter was published in the *ACM International Conference on Multimedia Retrieval* (ICMR), in 2012 [59]. An extension of that paper, which includes the results presented in Chapter 5, was submitted to the *Journal of Visual Communication and Image Representation*.

Chapter 6 presents the conclusions of the thesis and shows the opportunities for future work.

# Chapter 2

# Background

The visual dictionary model is one of the most effective approaches to represent visual content nowadays. The popular bag-of-(visual)-words representation has the ability to encode local properties while still generating a single feature vector per image.

This chapter details the main concepts related to the visual dictionary model. We give background information about each of the steps necessary to create a visual dictionary and then to represent images based on it. We focus on the techniques used throughout this thesis. The main steps are summarized in Figures 1.4 and 1.5.

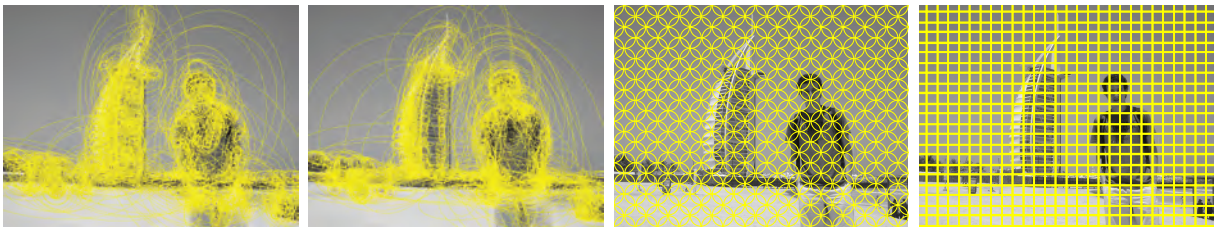
## 2.1 Low-level feature extraction

The creation of a visual dictionary is based on the quantization of a feature space. Thus, the first step to generate such dictionary is the extraction of low-level features from images. Those features are normally computed by local descriptors, which extract feature vectors from image regions.

A common approach to obtain regions of interest from images consists in using interest-point detectors [55]. Sampling images employing such detectors is often called *sparse sampling* [35]. Figure 2.1(a) shows some examples of the regions detected by two different detectors. The advantage of that sampling method is that the points detected are usually invariant to transformations like scale and rotation. However, interest-point detectors are computationally expensive and do not detect points in homogeneous regions. As they analyze differences in contrast, for example, points are normally detected in edges and corners. Therefore, some image parts can stay without a representation.

Another approach to image sampling is *dense sampling*. This sampling scheme simply uses a dense grid with rectangles or circles over the image, as shown in Figure 2.1(b).

Its strengths are the low computational cost and the ability to capture regions in every part of an image. However, it is usually applied in one single scale, making it not scale



Harris-Laplace Hessian-Affine Dense (circles) Dense (grid)

(a) Sparse sampling (b) Dense sampling

Figure 2.1: Examples of low-level image sampling. The two images on the left show the results of using *sparse sampling* (interest-point detectors) while the two on the right show the results of using *dense sampling*.

invariant. For classification experiments [35], the dense sampling approach outperforms interest-point detectors, specially because it generates a representation for every part of an image. Even homogeneous regions, which are not detected by sparse sampling, can be important to distinguish classes of objects and scenes.

The sampled image is described by image descriptors. SIFT [50] is the most popular descriptor used in those cases, but descriptors of any kind are also suitable. One can use simple global descriptors over each region of the dense sampled image, for instance. Van de Sande et al. [75] investigate the variations in performance when using different image descriptors over distinct sampling approaches.

The choice of the low-level description approach depends on the application. More precise local representations may be necessary in the case of applications like partial-duplicate image search, for example. In some scenarios, color information may be important, thus, the use of color descriptors should be considered.

The implementation of sampling schemes and image descriptors are available for research purposes. Popular softwares to perform the low-level feature extraction are the one provided by Mikolajczyk et al. [55][1], which supports many different sparse sampling methods and gray-level descriptors, and the one by van de Sande et al. [75, 76][2], which implements dense sampling (by circles) and several color descriptors.

[1] `http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html` (as of February 6th, 2013.)

[2] `http://koen.me/research/colordescriptors/` (as of February 6th, 2013.)

(a) 15-Scenes (b) Caltech-101

Figure 2.2: Examples of 50 visual words obtained from sparse sampling (Harris-Laplace detector) in a dictionary of 1 000 words computed for the (a) 15-Scenes and (b) Caltech-101 datasets.

## 2.2 Feature space quantization

The quantization of the space of low-level descriptions is responsible for the dictionary generation. Although the designation *visual dictionary* is popularly used in the literature [26, 27, 83, 86, 87], we can also refer to the quantization of the feature space as *visual vocabulary* [9, 15, 32], *vector quantized space* [75, 81], and *visual codebook* [11, 49, 77], for example.

The process of quantizing the feature space is responsible for making the local descriptions less precise. This is a desired effect considering the use of the dictionary in more general applications. Figures 1.2 and 1.3 in the Introduction of this thesis show examples of how the visual dictionary can increase the generality of pure local descriptions. Nevertheless, the quantization level is chosen according to the application. In applications for which small differences between vectors should be detected, like partial-duplicate image search or copy detection, less quantization is necessary. On the other hand, if the representation should be robust to intra-class variations, like in semantic-search applications, the feature space can be largely quantized. Therefore, in the first case, large dictionaries should be used, while smaller dictionaries are recommended for the latter applications.

Each region in the quantized feature space is considered a *visual word* [70]. Visual words tend to represent a certain type of visual appearance. Figures 2.2(a) and (b) show examples of 50 visual words taken from a dictionary of 1 000 words created based on sparse sampling (Harris-Laplace detector) over 15-Scenes and Caltech-101 datasets, respectively.

We can see that, as the sampling approach used in the example obtains very local regions, the visual words are small parts of scenes or objects.

To implement the feature space quantization, k-means is the most popular approach used nowadays [9, 14, 26, 58, 75, 77, 84]. However, in high dimensional spaces, as the ones created by the low-level descriptions, k-means tends to have low effectiveness [35, 79]. As k-means computes distances between vectors, it is subject to the effects of the *curse of dimensionality*. This phenomenon refers to the problem that, as dimensionality grows, the distribution of distances between features tends to become narrowly concentrated around an average value, reducing the contrast between similar and dissimilar features. Hence, some works [35, 79] show that k-means produces dictionaries no better than dictionaries created by a simple random selection of vectors in the feature space. Additionally, the computational cost to compute random dictionaries is extremely lower than by using k-means. Those facts motivated us to employ random dictionaries in this thesis.

The good results of random dictionaries also motivated us to elaborate our first hypothesis in this thesis. This hypothesis is concerned with the dictionary generality, that is, the possibility of creating a dictionary on one dataset and using it for other datasets, as well as creating a dictionary based on very small samples of a dataset. Our analysis over this topic is presented in Chapter 3.

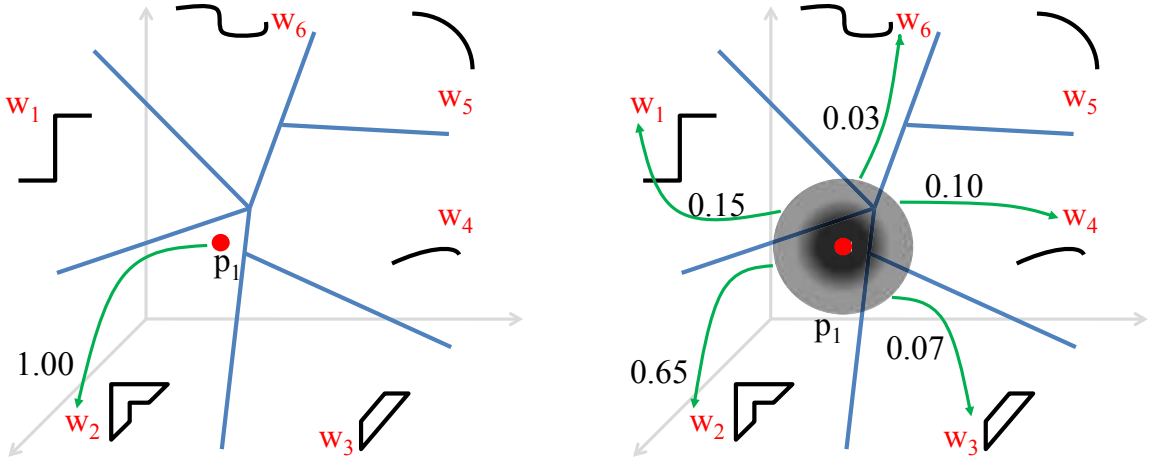
In literature, we can find other works proposing enhancements in the feature quantization step [34, 35, 48, 79, 85]. In this thesis, we are not aiming at proposing improvements in this step and we keep this phase as simple as possible.

## 2.3 Visual word assignment (coding)

After creating the visual dictionary, the image descriptions need to be coded according to the quantized feature space to make them comparable. This step is often called *visual word assignment* or simply *coding*. The coding phase must consider how the low-level features in the image are distributed according to the new quantized space. This can be performed by simply assigning the image local features to the visual words in the dictionary.

The first assignment scheme proposed, called *hard* assignment [70], consists in labeling a local patch with its closest region in the quantized feature space. That could be implemented by computing the distances from the vector of a patch to all the vectors corresponding to the visual words and assigning the label of the closest visual word to the patch. Equation 2.1 formally describes the hard assignment for the vector of local patch $i$:

$$\alpha_{i,j} = \begin{cases} 1 & \text{if } j = argmin\ D(v_i, w_j) \\ 0 & otherwise \end{cases} \tag{2.1}$$




(a) hard (b) soft

Figure 2.3: Toy example of (a) hard and (b) soft assignment for a given point $p_1$ (red circle). Green arrows indicate the visual words assigned to $p_1$ and the corresponding assignment value.

where $j$ varies from 1 to the dictionary size ($k$), $v_i$ is the feature vector of patch $i$, $w_j$ is the vector corresponding to visual word $j$, and $D(a, b)$ is the distance between vectors $a$ and $b$. Figure 2.3(a) shows a toy example of hard assignment.

Hard assignment is still commonly used [26, 29, 88, 89], but there are some known problems in this approach. In a high-dimensional feature space, a vector tends to be in the frontier of many regions of the quantized space, thus, assigning only the label of its closest region may discard important information about the vector description [77]. Van Gemert et al. [77] have also shown that the hard assignment has poor performance in very large dictionaries. Van Gemert et al. [77] call the phenomenon of having a given vector close to several regions in the quantized space as *codeword uncertainty*.

Capturing the information of the neighboring regions of a vector in the space should improve the coding phase. A popular approach to encode such information is called *soft* assignment [49, 64, 77]. Soft assignment tags a vector with the labels of its most activated regions in the quantized feature space. Thus, besides discarding less information about the vector description than hard assignment, soft assignment reduces the effect of poor feature space quantization during the dictionary creation. Figure 2.3(b) shows a toy example of soft assignment.

The implementation of soft assignment can be performed in different ways. In this thesis, we are using the *codeword uncertainty (UNC)* presented in [77]. In that work, the UNC implementation has the ability to deal with the codeword uncertainty phenomenon. UNC has presented the highest performance in relation to the other soft assignment mod-

els evaluated and it was more robust to the variations in the dictionary sizes. The equation presented in [77] also comprises the pooling phase (represented by the sum). Nevertheless, we are separating those phases because we can apply different pooling strategies over the assignment results. In that assignment scheme, the distances are smoothed by a Gaussian, which gives less weights for farther regions and higher importance for closer ones. The equation for soft assignment implemented and used in this thesis is the following:

$$\alpha_{i,j} = \frac{K_\sigma(D(v_i, w_j))}{\sum_{l=1}^{k} K_\sigma(D(v_i, w_l))}, \tag{2.2}$$

where $j$ varies from 1 to the dictionary size ($k$), $v_i$ is the feature vector of patch $i$, $w_j$ is the vector corresponding to visual word $j$, $K_\sigma(x) = \frac{1}{\sqrt{2\pi}\times\sigma} \times exp(-\frac{1}{2}\frac{x^2}{\sigma^2})$, and $D(a, b)$ is the distance between vectors $a$ and $b$. The $\sigma$ parameter indicates the smoothness of the Gaussian function: the higher the value, the larger the number of neighboring regions considered.

Literature also presents other techniques to improve the assignment step [31, 49, 63, 64, 77, 81]. As we are not proposing improvements in this step, we are using the current most solid schemes presented in Equations 2.1 and 2.2.

## 2.4 Pooling

The coding phase produces an assignment vector $\alpha_i$ for each of the points detected in the image. Over those vectors, a pooling strategy is employed. The pooling step aims at maintaining the properties encoded in the coding phase, or at least, discarding the least important ones, generating a single feature vector for the image.

In the initial BoW representations, usually based on hard assignment, the pooling method was employed by simply counting the number of occurrences of each visual word in the image. This generates exactly a histogram of visual words. However, as more elaborated coding schemes were being proposed, the pooling strategy also changed [11,26].

One of the most popular pooling approaches is based on computing the average assignment value of each visual word in the image. This is exactly the normalized histogram of visual words if it is used over hard assignment. Often called as *average (avg)* pooling, it can be formally defined by Equation 2.3:

$$h_j = \frac{\sum_{i=1}^{N} \alpha_{i,j}}{N}. \tag{2.3}$$

Another popular pooling approach, which presents better results than average pooling in classification experiments [11], is called *max* pooling. It is based on considering only the

| | Visual words | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Points** | $w_1$ | $w_2$ | $w_3$ | $w_4$ | $w_5$ | $w_6$ | $w_7$ | $w_8$ | $w_9$ | $w_{10}$ |
| **A** | 0.02 | 0.30 | 0.10 | 0.00 | 0.00 | 0.00 | 0.00 | 0.58 | 0.00 | 0.00 |
| **B** | 0.00 | 0.00 | 0.00 | 0.00 | 0.90 | 0.00 | 0.00 | 0.00 | 0.00 | 0.10 |
| **C** | 0.40 | 0.00 | 0.00 | 0.20 | 0.00 | 0.10 | 0.10 | 0.20 | 0.00 | 0.00 |
| **D** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.50 | 0.40 | 0.00 | 0.05 | 0.05 |
| **E** | 0.05 | 0.05 | 0.00 | 0.10 | 0.00 | 0.00 | 0.00 | 0.00 | 0.80 | 0.00 |
| **F** | 0.00 | 0.95 | 0.05 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **G** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.20 | 0.20 | 0.00 | 0.60 |
| **H** | 0.00 | 0.30 | 0.30 | 0.30 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.10 |
| **avg** | 0.06 | 0.20 | 0.06 | 0.08 | 0.11 | 0.08 | 0.09 | 0.12 | 0.11 | 0.11 |
| **max** | 0.40 | 0.95 | 0.30 | 0.30 | 0.90 | 0.50 | 0.40 | 0.58 | 0.80 | 0.60 |

Table 2.1: Example of *avg* and *max* pooling for an image with 8 points (A to H) and a dictionary of 10 visual words ($w_1$ to $w_{10}$). Each row represents the results of soft assignment for the corresponding point.

maximum activation value of each visual word in the image. Its good performance may be related to the fact that, even if only one point detected in the image highly activates certain visual word, this activation is kept in the final feature vector. On the other hand, by using average pooling, one isolated high assignment value would be divided by the number of points in the image, making it very small if the image has many points. By using max pooling, this value is preserved. The following example does not reflect the reality but is didactic to show why max pooling is good. Considering that each point in the image is a whole object (not a local patch) and that the visual dictionary is composed of objects, we need only one good object activation to have the information that the image contains that object. The idea would be the same for the dictionary of local patches. Max pooling is given by Equation 2.4 [11]:

$$h_j = \max_{i \in N} \alpha_{i,j} \tag{2.4}$$

in both Equations 2.3 and 2.4, $\alpha_i$ is the assignment vector, $N$ is the number of points in the image, and $j$ varies from 1 to the dictionary size ($k$). Table 2.1 presents an example of using avg and max pooling in a dictionary of 10 visual words considering an image with 8 points.

The average and max pooling strategies do not consider the spatial information of visual words in the image space. Therefore, they discard important information that could be crucial to distinguish types of scenes and objects. Literature presents several methods for encoding the spatial information of visual words in the image space [11, 15, 26, 29, 43, 62, 66, 88, 89]. However, most of them suffer from the problem of generating very large feature vectors or they are suitable only for specific applications. As mentioned

previously, our second contribution in this thesis is a pooling method for encoding the spatial arrangement of visual words, called *Word Spatial Arrangement (WSA)*. WSA is presented in Chapter 4 and has the benefits of generating more compact feature vectors than most of the existing spatial pooling approaches.

# Chapter 3

# Are visual dictionaries generalizable?

This chapter explores the first hypothesis presented in this thesis, which states that visual dictionaries are generalizable. The two main questions addressed are:

- are dictionaries created over certain images generalizable to images of other nature?
- do we need a representative subset of the whole collection to create a good dictionary?

To answer those questions, in Section 3.3, we first conduct experiments in closed datasets, creating dictionaries in one of them and representing images of the other dataset. Then, we create dictionaries based on samples of a dataset, aiming at verifying their quality in comparison to a dictionary based on the whole dataset. Finally, in Section 3.4, we perform similar experiments now considering the use of visual dictionaries in a Web environment.

## 3.1 Introduction

The whole process of dictionary creation is normally based on images from the same collection that will be represented. In the closed datasets popularly used in the literature [73], like the 15-Scenes, Caltech-101 and 256, and Pascal VOC, the amount of images is fixed, therefore no new content is added after the dictionary is created. However, in a large-scale dynamic scenario, like the Web, images are constantly inserted and deleted. In order to represent well those collections, how should a dictionary be created? This dynamic property of Web scenarios may cause what is called *concept drift*, which refers to the change in definitions over time [21].

The term "dictionary" is somewhat a misnomer, because it is not concerned with semantic information. More often than not, the creation of the visual dictionary ignores

completely the image labels, which capture the users' conceptual view of the images, and uses only the low-level features. Provided that the selected sample represents well enough that low-level feature space (being, for such, diverse in terms of appearances), the dictionary obtained will be sufficiently accurate, even if based on a small subset of the collection, or even on a completely different collection.

We have used the Caltech-101 and the 15-scenes datasets in order to evaluate the impact of using "cross-base" dictionaries, i.e., dictionaries created from samples of one dataset are used to create the bags-of-words of the images in another dataset. We have also used the Caltech-101 dataset alone to evaluate the impact of diversity on the quality of the dictionary used. After that, we have performed similar experiments in a Web scenario, aiming at verifying if the conclusions for the closed datasets also apply in such scenario.

Torralba and Efros [73] show the dataset bias in most of the popular datasets by a classification setup, training on images from one dataset and testing on images of another dataset. Those experiments are somewhat similar to our experiments in this chapter, however they work on the classification level, while we focus on the representation level. Our focus is to evaluate how the source of information impacts the quality of dictionaries.

## 3.2 Experimental setup

As pointed out before, the traditional datasets used in the experiments of literature are static, which means that no new images are inserted or removed after the dictionary creation. However, in a Web scenario, where new content is constantly being indexed (while others are being deleted), is the previously created visual dictionary still good for representing the new images? Of course, regenerating the dictionary whenever the database changes is unfeasible. Therefore, we explore those aspects going in the direction to evaluate if it is feasible to use visual dictionaries in this dynamic environment.

In all experiments, the parameters for dictionary generation and image representation are the same: dense sampling (6 pixels) [75] and SIFT descriptor, 1000 visual words selected by random, and soft assignment ($\sigma$=60) with max pooling (Equations 2.2 and 2.4 from Sections 2.3 and 2.4, respectively). Those are one of the best parameter configurations found in literature [11].

We have initially used two popular closed datasets, 15-Scenes and Caltech-101. We have conducted experiments in a classification protocol, with SVM using linear kernel ($c$=1.0). A balanced validation was performed, varying the number of training samples per class and using the rest of images in the test set. We have evaluated the results in terms of classification accuracy.

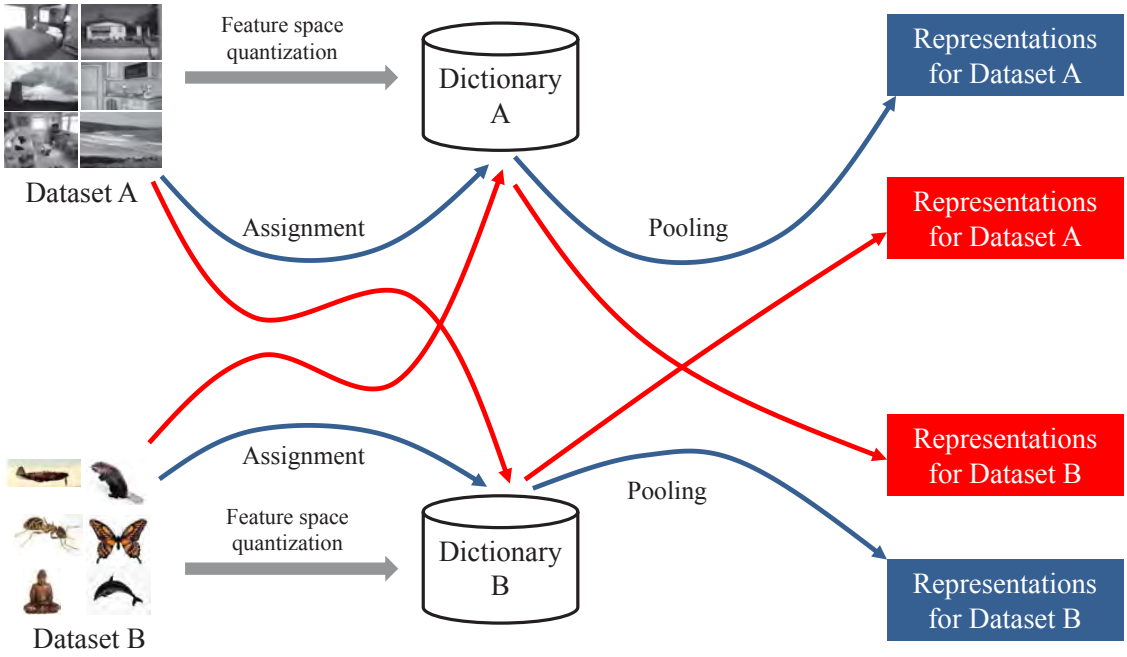



Figure 3.1: Schema of the experimental setup used to create the dictionaries and the cross-base image representations.

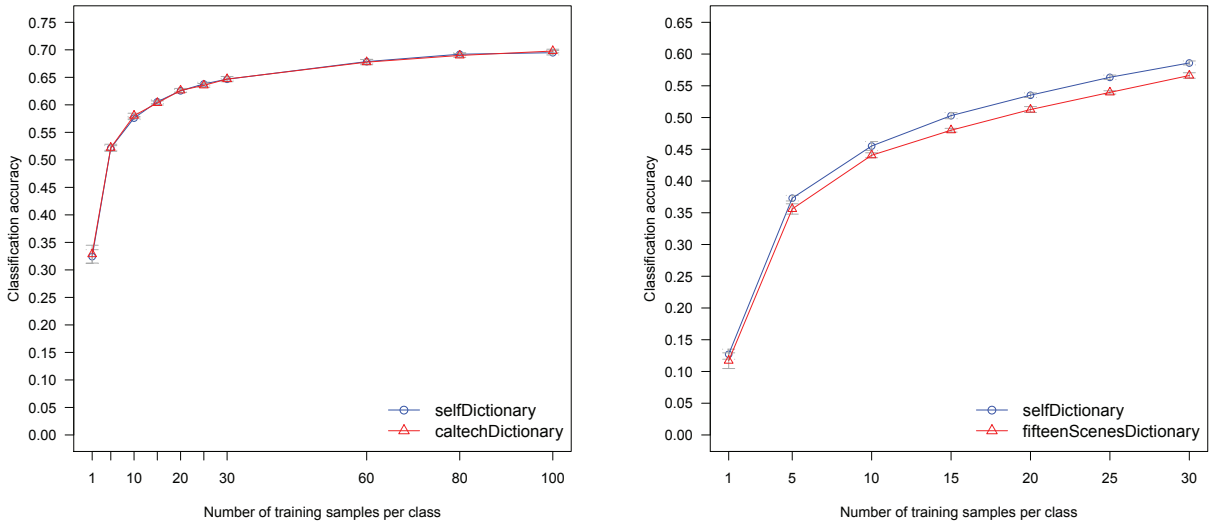
For the experiments in the Web scenario, due to the dataset used, we have performed image retrieval instead of image classification. Details are presented in Section 3.4.

## 3.3 Closed datasets experiments

The experimental results and discussion are shown considering each of the two questions presented.

### 3.3.1 Are dictionaries generalizable?

To answer the first question, we have created 5 dictionaries based on each of the two datasets. Then, we have used each dictionary to represent images from the same dataset and images from the other dataset. For example, in one of the cases, we have represented images of 15-Scenes using a dictionary created over the Caltech-101 images. Figure 3.1 shows the schema used to create the dictionaries and the cross-base image representations. It should be more natural to expect that images whose representations are based on a dictionary created over images of the same dataset are better than representations based on dictionaries created over images of the other dataset.




(a) 15-Scenes (b) Caltech-101

Figure 3.2: Classification accuracies on the datasets using dictionaries based on the same dataset (blue circles) and on the other dataset (red triangles). The confidence intervals (error bars) are for $\alpha$=0.05, on an average of 5 runs obtained on different dictionaries. In (a), the 15-Scenes dataset with its own dictionary is not significantly better than that using the Caltech-101 dictionary. The opposite configuration (b), using 15-Scenes dictionary on Caltech-101 dataset, shows some loss of accuracy. Contrarily to Caltech-101, the visual diversity of 15-Scenes is more limited.

The creation of the training and test sets were made by randomly selecting *nTrain* images of each class to compose the training set and using the rest in the test set. This was performed 5 times. We have also varied *nTrain* from 1 to 100 in the case of 15-Scenes dataset, and, from 1 to 30 for Caltech-101.

Figure 3.2(a) compares the classification accuracies when the images from 15-Scenes dataset are represented by dictionaries either based on their own images or based on Caltech-101 images. We can see that the results obtained with the dictionary based on Caltech-101 images are as good as those obtained with the dictionary based on 15-Scenes images. The results are, in fact, so close, that they fail a significance test of difference.

Figure 3.2(b) compares the classification accuracies when the images from Caltech-101 dataset are represented by dictionaries based on their own images and by dictionaries based on 15-Scenes images. The representations of Caltech-101 images are slightly better if they use dictionaries based on their own images. The difference is small, but enough

to pass a significance test. This contrasts with the results obtained previously, where the differences were, for all practical effects, non-existent.

We can conclude that 15-Scenes images are less variable than Caltech-101 images in terms of SIFT descriptions. The SIFT descriptions of Caltech-101 seem to comprise more of the whole SIFT space, while the SIFT descriptions of 15-Scenes may concentrate only on portions of that space. Another possibility is that the space comprised by Caltech-101 descriptions is larger and covers the space of 15-Scenes descriptions. Therefore, the dictionary based on Caltech-101 is more general than the dictionary based on 15-Scenes images.

Those results answer our first question. The variability of the SIFT descriptions of a dataset is important to indicate how general is a dictionary created over its images. A stereotyped dataset will probably generate good dictionaries only for itself or for other datasets with the same characteristics. A heterogeneous dataset in terms of feature descriptions can generate dictionaries which could be used effectively in a wider range of other datasets.

It is important to highlight that we are not analyzing if any of the datasets is biased in terms of classes or images. We are providing results indicating the dataset variability in terms of the feature space of local descriptions.

With the results presented, we can say that if we use a good dataset in terms of visual variability, we can generate a dictionary able to represent well many different types of images, even images that are not known yet, like in a Web scenario. Therefore, this is an indication that visual dictionaries can be used in heterogeneous and dynamic environments.

### 3.3.2 Do we need to have a representative subset of the whole collection to create a good dictionary?

To answer our second question, that raises the need of having or not a substantial part of the dataset to generate a good dictionary for representing images, we have prepared an experimental setup varying the number of image classes used for dictionary generation. We have used Caltech-101 as the source of features, due to its variability presented in the experiments described in the previous section.

We have performed random selections of classes from Caltech-101. For each selection, we have taken a variable number of classes to be the source of the dictionary, generating 9 dictionaries. The first dictionary was generated based on images of only 1 class. The second dictionary was based on images of 3 classes, including the class used in the first dictionary. The following dictionaries kept the incremental aspect, increasing number of classes to 6, 12, 25, 50, 67, 84, and 101. Therefore, we could evaluate what is the impact

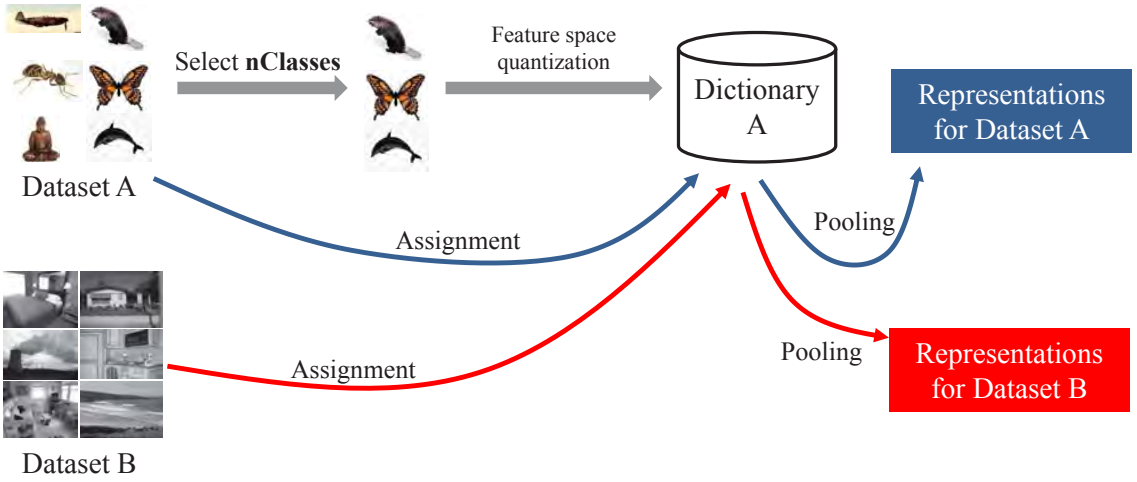



Figure 3.3: Schema of the experimental setup used to create the dictionaries based on parts of a dataset. The BoW representations were based on the partial dictionaries.

in the dictionary quality when using parts of the dataset. As we have selected the classes randomly, there may be different dictionary qualities depending on the classes selected in each case. For example, if for dictionary based on 1 class, the class selected is poor in terms of visual diversity, its dictionary tends to be bad; on the other hand, if the class is visually diverse, its dictionary could be good. To also evaluate this phenomenon, we have performed 5 different random selections for each number of classes. Figure 3.3 shows the schema used to create the image representations based on dictionaries created on parts of a dataset. Table 3.1 shows a summary of the classes selected and the number of images and points in each random selection of classes from Caltech-101.

For each dictionary, we have represented the whole Caltech-101 dataset and also the whole 15-Scenes dataset and have conducted classification experiments.

The training and testing phases in the experiments of this section are slightly different from the ones presented in Section 3.3.1. In those experiments, the training phase used 5 random sets of training samples per class. Aiming at eliminating the training set variability from the results presented in this section, we have randomly pre-selected 8 training sets to be used for all classification setups.

The training sets were created by randomly selecting *nTrain* images from each class for the training set. The images that were not selected for training are used for testing. We have made this 8 times, generating the training and test sets for *all* of the following experiments. Thus, the training and testing phases for all the representations use the same samples. For Caltech-101, we have used *nTrain* equal to 30 and for 15-Scenes, *nTrain* equal to 100.

Figure 3.4 shows the average accuracies for 15-Scenes and Caltech-101 datasets when

| nClasses | Selection | nImgs | nPoints | Classes selected (incremental) |
|---|---|---|---|---|
| 1 | 1 | 35 | 14 878 | strawberry |
| | 2 | 54 | 21 781 | hedgehog |
| | 3 | 59 | 21 551 | rhino |
| | 4 | 34 | 13 819 | gerenuk |
| | 5 | 56 | 20 171 | windsor_chair |
| 3 | 1 | 152 | 66 288 | umbrella, anchor |
| | 2 | 136 | 52 620 | snoopy, ceiling_fan |
| | 3 | 183 | 73 632 | okapi, sunflower |
| | 4 | 149 | 55 641 | minaret, wrench |
| | 5 | 156 | 64 266 | lobster, wheelchair |
| 6 | 1 | 358 | 157 421 | ceiling_fan, yin_yang, grand_piano |
| | 2 | 349 | 144 498 | dragonfly, chandelier, panda |
| | 3 | 319 | 130 129 | accordion, barrel, gerenuk |
| | 4 | 1 064 | 299 545 | Motorbikes, crocodile_head, lotus |
| | 5 | 323 | 130 284 | sunflower, wrench, brontosaurus |
| 12 | 1 | 644 | 269 602 | wild_cat, revolver, binocular, cougar_body, snoopy, accordion |
| | 2 | 1 025 | 694 474 | crayfish, dollar_bill, saxophone, beaver, Faces, binocular |
| | 3 | 693 | 283 820 | kangaroo, water_lilly, crab, umbrella, elephant, wrench |
| | 4 | 1 504 | 474 866 | ibis, rhino, chandelier, helicopter, sea_horse, tick |
| | 5 | 656 | 253 229 | anchor, lamp, ant, crocodile_head, dollar_bill, ewer |

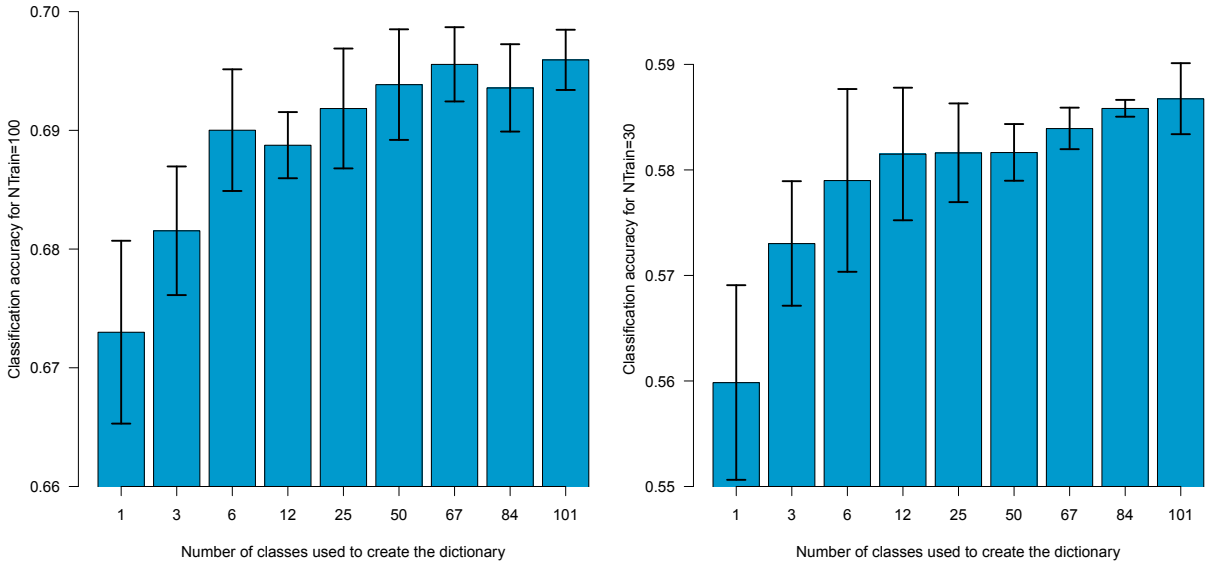
Table 3.1: Summary of the smaller partial datasets (1 to 12 classes) used in the selections performed over Caltech-101 when evaluating the impact of creating visual dictionaries based on parts of the whole dataset.

using the dictionaries created over a variable number of classes from Caltech-101. The confidence intervals were computed based on the 5 random selections of classes and $\alpha$=0.05.

We can see that the largest difference occurs for the dictionary based on 1 class, and this difference is still very small (around 2% in relation to the dictionary based on all 101 classes, without considering the confidence intervals). For most of the dictionaries based on more than 1 class the confidence intervals intersect and we cannot say that one is better than the other.

Therefore, we can also answer our second question. The results just presented are a good indication that, even with a small portion of the dataset, we can generate a good dictionary. As the low-level descriptor (SIFT) is based on image local textures and not on semantics, the fast dictionary generalization occurs if we have a set of images rich enough in terms of textures, which will cover all the feature space without requiring the use of all image classes. To verify if the same conclusions can be made in a Web scenario, we present in the next section such kind of experiments.

In Chapter 5, we revisit this discussion upon semantics in representations based on low-level features and visual dictionaries.




(a) 15-Scenes (b) Caltech-101

Figure 3.4: Classification accuracy on (a) 15-Scenes and (b) Caltech-101 datasets using the 9 different dictionaries created over a variable number of classes from Caltech-101. Although the results show some random fluctuation, it is clear that as soon as we have higher *visual* diversity, the accuracy reaches its asymptotic value, even if *semantically* (in terms of label diversity), the sample is still very poor.

## 3.4 Web-environment experiments

To evaluate the dictionaries in a Web environment considering their generality, we have performed experiments in a dataset with more than 230 thousand images. This dataset was also used to evaluate global image descriptors in a Web environment in previous works [38, 61]. Called *WebSample* dataset, it has very heterogeneous content and has no categorization. The dataset was collected by researchers from Federal University of Amazonas (UFAM), Brazil, with the objective to create a collection with representative data from the Web. The data gathering started recursively from the Yahoo directory[1] and generated a database with more than 230 thousand images (excluding icons and banners) and 1.2 million HTML documents. After that, further work in the WebSample dataset [38] created a set of 30 query images with their respective pool of relevant images. The pool was created by real users annotating retrieved images [38].

Therefore, instead of performing experiments for image classification, we have performed experiments for image retrieval using the just mentioned pool of relevant images

[1] `http://dir.yahoo.com/` (as of February 6th, 2013).

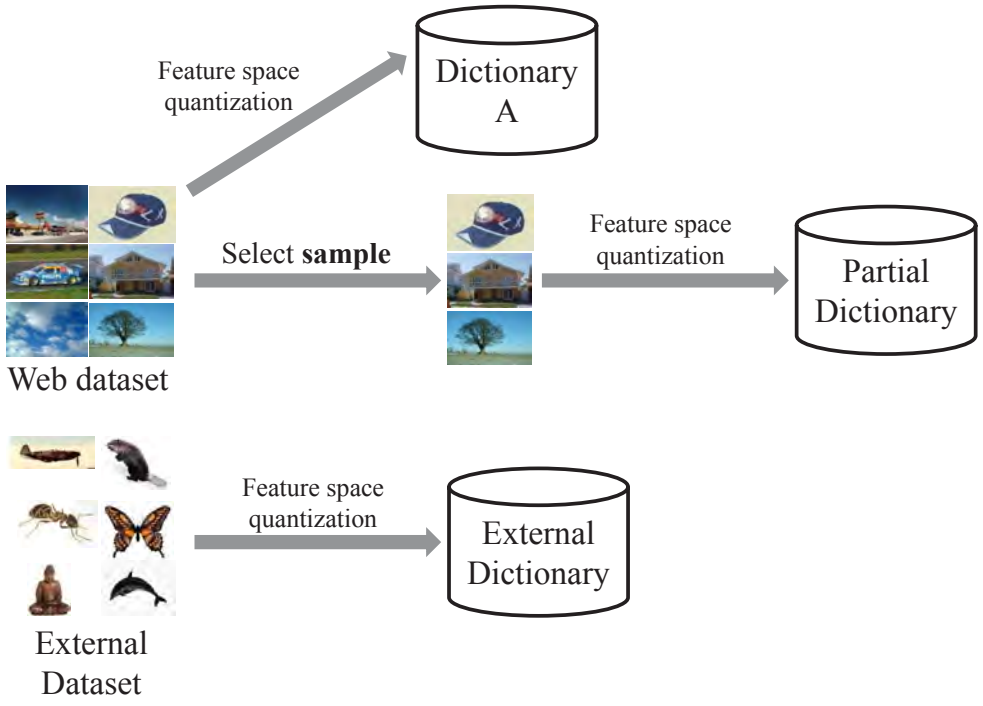



Figure 3.5: Schema of the experimental setup used to create the dictionaries in the Web environment. The whole and samples of the Web dataset, as well as an external dataset, were used to create the dictionaries.

for all the 30 query images. The results are based on effectiveness measures, including mean average precision (MAP) and precision at different number of retrieved images (P@N).

The objective of these experiments is to evaluate how different dictionaries could change the quality of the representation. As well as we have done previously for the 15-Scenes and Caltech-101 datasets, we created dictionaries using several sources of information and used them to represent the WebSample images. Results show what is the impact in the dictionary quality when using external sources or when using the dataset partially. Figure 3.5 shows the schema used to create the dictionaries.

Table 3.2 summarizes the datasets used to generate the dictionaries evaluated in the experiments. We have used the complete WebSample dataset and also two partial random samples containing 1 thousand and 1 hundred images. The use of part the dataset that is being represented will tell us if we really need a representative amount of images to generate a good dictionary or not. We have also used Caltech-101 to generate the dictionary. This will tell us if in a Web environment, we can generate a good dictionary even creating it with a different dataset. This phenomenon has already been presented in this chapter (see Section 3.3.1) where we show the effects of creating a dictionary on

| Dataset | number of images | number of points |
|---|---|---|
| WebSample | 235 063 | 1 337 744 530 |
| WebSample (partial 1k) | 1 000 | 5 761 887 |
| WebSample (partial 100) | 100 | 582 939 |
| Caltech-101 | 9 144 | 4 249 909 |

Table 3.2: Datasets used to generate the different dictionaries evaluated in the experiments.

the 15-Scenes dataset and representing Caltech-101 images, and vice-versa. If the same phenomenon appears in the Web environment, the dictionary based on Caltech-101 will be as good as dictionaries created in the WebSample dataset.

To represent the images, we have used the same configuration presented in Section 3.2.

To compute the effectiveness measures, each of the 30 query images was compared to all the images in the WebSample dataset (by Euclidean distance) and then ranked. It is important to note that we have considered the query itself as being in the dataset.

The results are presented in Table 3.3, where confidence intervals are based on $\alpha$=0.05 and in the 30 queries used. We can see that the average values are very similar, both for MAP and P@10. Due to the large variation in the queries, the confidence intervals are large, therefore, there is no statistical difference between the average values. To have a better comparison considering the inter-query variation, we have also conducted a paired-test analyzing the differences for each query. The results are presented in Figure 3.6.

In a paired-test, we compute the differences of MAP values (or P@N values) for two methods for all corresponding pair of queries. Then, we compute the average and the confidence intervals of those differences. If the confidence interval includes the zero, the two methods are equivalent at that confidence level. Otherwise, the sign of the difference indicates the best method. We can see in Figure 3.6 that for all the dictionaries used, the confidence interval includes the zero, therefore, there is no statistical difference between any of them.

The results presented in this section agree with the results presented in Section 3.3, showing that we can use a very small part of the data or we can use a completely different dataset to create good dictionaries for representing a given dataset. We can conclude that if the sample used to generate the dictionary is diverse enough in terms of local appearances, it is enough to create a good dictionary. This was observed in all of the datasets used as sources for the dictionaries evaluated in these experiments.

Therefore, our final conclusion for the experiments presented in this section is that visual dictionaries can be used in a Web environment even considering the fact that the Web is very dynamic and heterogeneous.

| Dictionary based on | MAP | P@10 |
|---|---|---|
| WebSample | 14.60 ± 6.16 | 23.67 ± 7.83 |
| WebSample (partial 1k) | 13.54 ± 5.90 | 22.67 ± 8.24 |
| WebSample (partial 100) | 14.92 ± 6.00 | 23.67 ± 7.60 |
| Caltech-101 | 14.78 ± 6.02 | 21.33 ± 7.44 |

Table 3.3: Retrieval results for the representations based on each of the 4 dictionaries tested. We can see that there is no statistical difference between them.

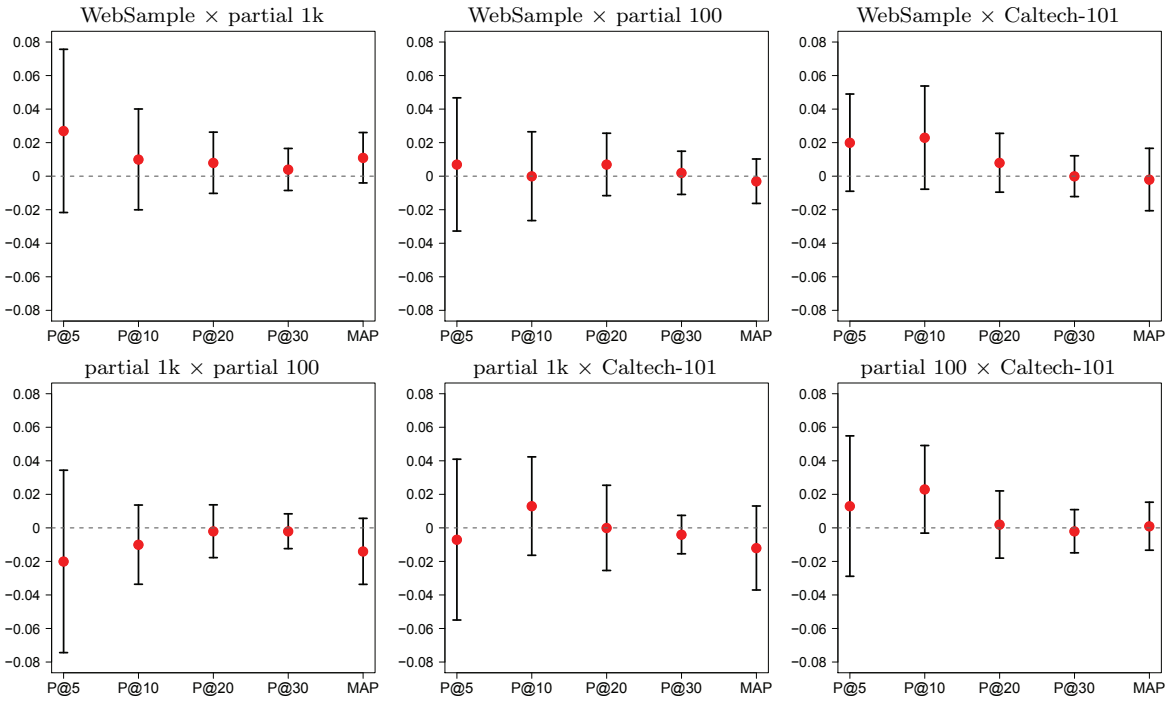



Figure 3.6: Retrieval results in the WebSample dataset: paired-test for the per-query comparison showing that no statistical differences exist for all the dictionaries (intervals of the average of the differences include the zero). The vertical axis is the average of the differences for the corresponding evaluation measure in the horizontal axis.

## 3.5 Discussion

This chapter evaluated the impact and the feasibility of using visual dictionaries in scenarios where the entire dataset is not available for the dictionary construction as, for example, in large-scale dynamic datasets, like the Web. The experiments conducted show that dictionaries based on a subset of the collection, or even on an entirely different collection, may still provide good performance, on the condition that the selected sample is visually diverse.

Therefore, we could confirm the first hypothesis presented in this thesis: visual dictionaries are generalizable. They generalize among datasets with similar characteristics, that is, similar datasets in terms of visual diversity may be used to generate good dictionaries for other datasets of the same kind. However, for special-purpose datasets, like medical images, for instance, this might not be true.

Those findings open the opportunity to greatly alleviate the burden in generating the codebook, since, at least for general-purpose datasets, we show that the dictionaries do not have to take into account the entire collection, and may even be based on another small collection of well-chosen visually diverse images.

# Chapter 4

# Encoding spatial arrangement of visual words

This chapter presents our approach to encode the spatial arrangement of visual words in the image space. It is related to our second hypothesis in this thesis, which says that the spatial information of visual words is important to distinguish types of scenes and objects.

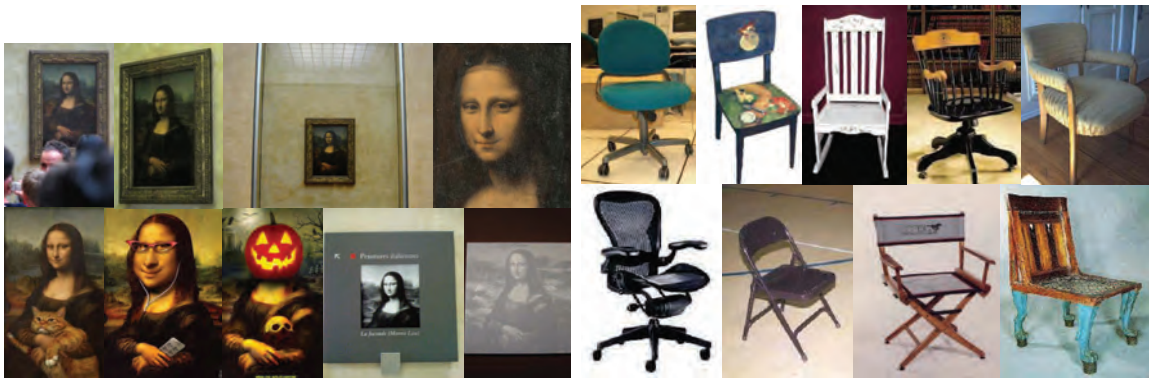
We first give in Section 4.1 an overview of the challenges in designing image representations to consider the spatial configuration of visual words in the image space. Next, in Section 4.2, we describe related work highlighting the differences between existing approaches and the proposed method. Then we present the proposed spatial pooling method in Section 4.3 and show experiments for image retrieval and classification in Sections 4.4 and 4.5, respectively.

## 4.1 Introduction

When designing an image representation, one must be aware of its target application. Applications like copy-detection or partial-duplicate image search, as shown in Figure 4.1(a)[1], require the creation of really discriminating representations. Very small differences between images or objects must be encoded, while still being robust to specific photometric/geometrical transformations related to the domain. Therefore, the representation must be very precise. The semantic-search application, as shown in Figure 4.1(b)[2], requires precise representations but, at the same time, general enough to comprise the intra class variations. One may be interested in finding different types of the same object, like, for example, retrieving different types of chairs, instead of finding exactly the same chair.

[1]CreativeCommons images downloaded from Flickr (as of July 9th, 2012).

[2]Chairs from Caltech-101 dataset [25].



(a) (b)

Figure 4.1: Application examples: (a) retrieval of partial duplicates, where (parts of) the same object or scene are shared between the query and target images, possibly with transformations and noise; (b) semantic search, where query and target images share concepts (e.g., different instances coming from the same class of objects), but not necessarily objects or scenes.

The research community has been very active in the areas of computer vision in the latest years and many new proposals over the visual dictionary model constantly appear. Special attention has been given to the lack of geometrical information encoded by the traditional bag-of-words representation [15, 26, 29, 32, 43, 62, 89]. The spatial arrangement of visual words in images is important to understand image semantics and is often crucial to distinguish different classes of scenes or objects. In that direction, approaches are proposed for image classification [26, 43] and retrieval [15, 29, 32, 89].

In the classification scenario, usually relied on Support Vector Machines (SVMs), the high dimensionality of vectors do not degrade effectiveness, because SVMs suffer less from the curse of the dimensionality. The popular Spatial Pyramids [43] are very successful for image classification and their vectors have high dimensionality. However, for retrieval experiments, which are generally based on computing distances between vectors, with the Euclidean distance, for example, vectors should be compact, or embedded in an index structure, to avoid the curse of the dimensionality [13, 29, 36, 82]. As dimensionality grows, the distribution of distances between features tends to become narrowly concentrated around an average value, reducing the contrast between similar and dissimilar features. Therefore, to create an image representation that works well in both classification and retrieval scenarios, one must be aware of the feature vector size. There are also some alternatives to the direct use of distance computations for ranking, which are referred to as *learning to rank* [24].

Many of the existing approaches to spatial pooling which are employed in the retrieval

scenario leave the spatial verification as a post-processing step [32, 89]. They compute a simple representation and then, after finding the matching visual words between images, they compute the spatial representation and perform a spatial consistency verification, before reranking the images. Furthermore, some of the existing approaches used in the retrieval scenario are very precise and suitable for partial-duplicate image search [32, 89], thus their use for the semantic-search application is challenging.

In this chapter, we present *Word Spatial Arrangement* (WSA), a spatial pooling approach to both image retrieval and classification. Our approach adds spatial information into the feature vector having the advantages of generating more compact vectors than the popular approaches to spatial pooling. It is also more precise than the traditional bag of words but keeps the generality useful for the semantic-search application. Our approach aims at addressing both the retrieval and classification scenarios. In the retrieval environment, WSA encodes the spatial information of visual words into a single feature vector prior to any filtering step with matching visual words. Most of the approaches that encode spatial information of visual words in the retrieval scenario [29, 89] works solely with the assignment of a unique visual word to a point (hard assignment). WSA, however, also works with soft assignment, taking advantage of the good performance of soft assignment in classification experiments [49, 64, 77]. We also provide an online interface to show the experiment results in the retrieval scenario[3] [4].

The spatial arrangement of visual words encoded by WSA is based on a sliding quadrant partition in the image space considering each point in the image as the origin of the quadrants and counting the visual words occurrences in each quadrant [62]. Some attempts to improve the WSA algorithm were performed during its development phase, however, in this chapter, we report only the approach which obtained the best results.

## 4.2 Related work

In this section, we present some of the recent advances on encoding spatial information of visual words [15, 26, 29, 32, 43, 62, 89].

In the early days of the content-based image retrieval (CBIR) area [71], researchers faced the problem of having many different images with identical or very similar color histograms, motivating the creation of new methods for encoding the spatial arrangement of colors, like, for example by using color correlograms [30] or color-coherence vectors [57]. This issue is being revisited nowadays with the visual dictionary model. However, the element under analysis moved from single pixel values to local patches.

[3] `http://www.recod.ic.unicamp.br/eva/view_images_base600.php` (as of February 6th, 2013).
[4] `http://www.recod.ic.unicamp.br/eva/view_images_paris.php` (as of February 6th, 2013).

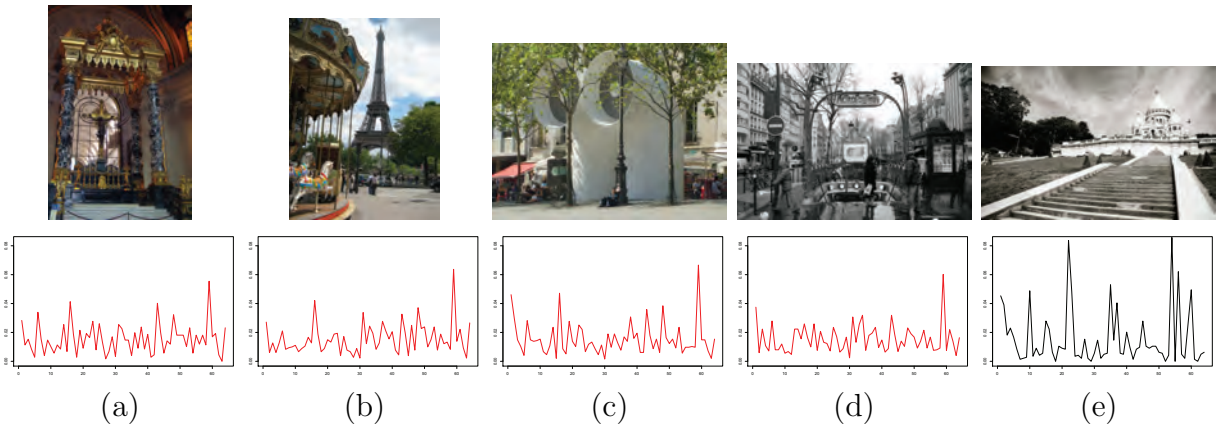



Figure 4.2: Examples of images (a–d) with different semantics but similar bags of visual words (BoW). The graph below each image shows its BoW, created using a dictionary of 64 words, hard assignment, and average pooling. The horizontal axis is the label of the word (1–64) and the vertical axis is the frequency of occurrence of each word. Due to the loss of spatial information, unrelated images (a–d) may end up sharing very similar BoWs. For sake of comparison, we also show an image with a dissimilar BoW (e).

Spatial information of visual words, usually lost by the traditional pooling techniques like *average* and *max* pooling [11], may be very important for discriminating image content and for encoding image semantics. Consider the images shown in Figure 4.2. They have different semantics but their BoW representations are very similar.

The development of methods for encoding spatial information of visual words may take into account several aspects depending on the target application. Considering the semantic-search application, where we would like to be able to find different types of the same object or image, as shown in the example in Figure 4.1(b), the representation needs to be specific enough to distinguish one class of objects from the others, but not too precise, otherwise only the same object instance will be considered similar. Therefore, capturing spatial information for semantic search must be planed carefully for not loosing generality, which is one of the main strengths of the BoW representation. On the other hand, in the partial-duplicate search application, where the changes among images exist but images still share some duplicate patches [89], the representation must be very precise. Several approaches include the geometrical verification as a post-processing step, keeping the representation simple and applying the geometrical constraints on a subset of matched visual words [32, 89].

Another important issue when developing a new method for encoding the spatial information of visual words is related to the compactness of the representation. In the classification scenario, which is popularly based on SVMs, the curse of the dimensionality

does not impact considerably the effectiveness of the methods, because SVM usually deals well with very large feature vectors [43, 63]. However, considering the retrieval scenario, the feature vector size considerably impacts the effectiveness of search approaches. The curse of the dimensionality is closely related to the action of computing distances between vectors, a frequent operation in retrieval systems. Therefore, some representations which work well for image classification may not work for image retrieval. Our approach aims at encoding the spatial arrangement of visual words being compact to be useful for both classification and retrieval scenarios.

The most popular approach to encode the spatial information of visual words is the *Spatial Pyramid* [43]. A spatial pyramid hierarchically splits the image into fixed-size tiles and generates one BoW representation for each tile. For a pyramid level of 2, for example, 21 bags are generated. The first bag comes from the image without splitting. In the next level, the image is split into 4 tiles of the same size. The next level splits each of the 4 tiles into another set of 4 tiles. Therefore, there is 1 bag for level 0, 4 bags for level 1 and 16 bags for level 2. All the bags are concatenated to create the image feature vector. The main advantage of pyramids is their simplicity. Other advantage is that the hierarchical splitting tends to create a multi-scale image representation. Their main drawbacks are related to the large feature vector size, to the fact that no information regarding the image scale is taken into account, and that no spatial relationship among visual words is encoded.

Other recent approach to spatial pooling of visual words [15] is based on creating linear and circular projections of the image. The linear projections consider the horizontal axis as reference. The image is split into $L$ vertical tiles and a BoW representation is generated for each tile. The axis is then rotated by an angle of $\theta$ and each of the $L$ tiles generates another set of bags. This is performed by a predefined number of angles. The circular projections consider a set of points to be the center of the image splitting and then splits the image into $L$ sectors. A BoW representation is computed for each sector. The final feature vector is a concatenation of all bags generated by linear and circular projections. The method also conducts reordering of bags in the feature vector to achieve rotation, translation, and scale invariance. Its main advantage lies in capturing more spatial configurations than the Spatial Pyramids, as these last ones could be considered particular cases of linear projections. Its main disadvantage is the large feature vector size. Moreover, no spatial relationship information among visual words is explicitly encoded.

Another recent approach encodes the spatial relationship of visual words by using triangular relations among neighboring words [29]. All the triangular relationships between 3 points in the image are computed and, for each relationship, a set of signatures is created. There are signatures which depend on point labels, signatures considering the angles among points, and signatures considering point scales. Each relationship is

indexed independently and is composed of a maximum of 7 signatures (7-D vector). The signatures maintain invariance to translation, rotation, scale, and flipping. To avoid a large number of triangular relationships, pruning strategies are employed. This method explicitly encodes the spatial relationship among visual words, however, the description and its similarity measure were not designed for kernels, making it challenging to use in classification scenarios.

A recent spatial coding technique for partial-duplicate image search encodes the spatial relationship among every pair of points in the image by using binary spatial maps [88,89]. The spatial verification is a post-processing step in the retrieval framework, applied only for matching visual words between query and database images. A horizontal spatial map is an $N \times N$ binary matrix, where each row $i$ says if the feature $i$ is at right (1) or at left (0) of each other feature. The vertical spatial map is analogous, having value 1, in row $i$, for points which $i$ is above and value 0, otherwise. The effect of the spatial maps calculations consists in splitting the image into 4 quadrants, using each point in the image as the origin. The method also considers rotation and scale issues, by rotating the image according to the orientation of the origin SIFT point [88] and by considering the distance between points (square maps). This method explicitly encodes the spatial relationship among visual words, but its representation is very precise making it not suitable for the semantic-search application. The spatial maps are computed only for matching words, therefore, changes in the representation are necessary to allow its use in classification scenarios. Our approach uses a similar idea of the image space splitting, however, our representation embeds the spatial information into the feature vector and works both for classification and retrieval scenarios. Furthermore, the applications considered are different and we intended to keep our representation more general.

Another recently proposed approach works specifically for image classification [26]. It is a geometric $l_p$-norm pooling method that learns the positions of visual words occurrences in an image dataset. For that, the method first puts all the images into the same resolution, discarding their aspect ratio, and uses a regular (dense) grid for image sampling. Therefore all the images will have the same number $M$ of points. At the end, each visual word $k$ has a vector of dimension $M$, where each vector position $m$ corresponds to the activation of the visual word $k$ in the $m^{th}$ position of the dense grid. This approach can effectively learn the positions of visual words in the images, however it greatly depends on putting all the images into the same resolution and using the dense sampling. Additionally, the encoded properties represent the absolute visual word position in the image and objects translation inside the images will change considerably the final representation. Our proposed approach has some relation to the geometric $l_p$-norm pooling just presented [26]. The geometric $l_p$-norm pooling method encodes the *absolute* position of visual words in the images. Our method, on the other hand, by counting visual word

positions in relation to all the other points in the image, discarding their visual word assignments, encodes the *relative* position of each visual word in the image. Our method is based on image sparse sampling (by interest point detectors) and geometric $l_p$-norm pooling uses dense sampling. If the majority of points detected in the image are in the object of interest, our approach does not suffer from the translation problem mentioned for the geometric $l_p$-norm pooling. Other advantage of our method is that it also works in the retrieval scenario.

There are many other proposals for encoding the spatial information of visual words, like, for example, by using the co-occurrence of pairs visual words [69], by using correlograms [66], or by appending the point coordinates to their feature vectors before creating the dictionary [53]. Many of those methods face the problem of generating high-dimensional feature vectors, since including all the possible spatial configurations into the feature vector and keeping compactness is challenging. This is one reason that leads some approaches to leave the spatial verification as a post-processing step [32, 88, 89].

The next section details the proposed WSA representation.

## 4.3 Word Spatial Arrangement (WSA)

This section presents our approach to encode the spatial arrangement of visual words, which is called Word Spatial Arrangement (WSA). The main goal when designing WSA was to include the spatial information of visual words, aiming at increasing the precision of the traditional BoW representation but keeping the generality which can make it also useful for the semantic-search application. WSA was also designed to be able to work in both retrieval and classification scenarios.

As mentioned previously in Section 4.2, WSA presents some similarities with other methods from the literature. Other important aspects of WSA are the following:

- the spatial information of visual words is embedded into the feature vector, therefore, in the retrieval scenario, no post-processing is required;
- WSA encodes the relative position of visual words in the image space;
- WSA representation is more compact than many of the spatial pooling approaches in the literature;
- WSA works with soft assignment as well as with hard assignment;
- WSA works with sparse sampling (interest-point detectors);

WSA is based on the idea of dividing the image space into quadrants [62] using each point as the origin of the quadrants and counting the number of points that appear in

each quadrant. We count how many times a visual word $w_i$ appears in each quadrant in relation to all other points in a specific image. This counting will tell us the *spatial arrangement* of the visual word $w_i$. Intuitively, the counting will measure the positioning of a word in relation to the other points in the image. It reveals, for example, that a word $w_i$ tends to be below, at right, or surrounded by other points. By counting $w_i$ position in relation to the other points in the images, without considering the labels of other points (visual words assigned to them), we generate a not-too-precise representation, which is interesting for the semantic-search application.

Figure 4.3 shows an example of partitioning the image space and counting. To generate the WSA vector, the image space is divided as follows: for each point $p_i$ detected in the image, we divide the space into 4 quadrants, putting the point $p_i$ in the quadrant's origin; then, for every other detected point $p_j$, we increment the counters of the visual word associated with $p_j$ in the position that corresponds to the position of $p_j$ in relation to $p_i$. For example, if $w_j$ is the visual word associated with $p_j$ and $p_j$ is at top-left from $p_i$, the counter for top-left position of $w_j$ is incremented. After all points are analyzed in relation to $p_i$, the quadrant's origin goes to the next point $p_{i+1}$, and the counting in relation to $p_{i+1}$ begins. When all points have already been the quadrant's origin, the counting finishes.

Each visual word will be associated with 4 numbers, which tell the spatial arrangement of the visual word in the image. The same visual word can appear in several different locations in an image, however, there is only one set of 4 counters for each visual word. The complexity of this method for generating the feature vector is $O(n^2)$, while the traditional bag is $O(n)$, where $n$ is the number of points in the image.

When the counting is finished, each 4-tuple is normalized by its sum. If the word $w_i$ has most of the counting values in its bottom-right counter, for instance, we can say that $w_i$ is a bottom-right word, as the word $w_4$ in Figure 4.3(c). If $w_i$ has top-left and top-right counters with high values, we can say that $w_i$ is a word that usually appears above other points. If all counters of $w_i$ are equally distributed, $w_i$ is surrounded by other points (middle-word) or it is a word that repeatedly surrounds other points (border-word).

Another advantage of WSA is that we do not need to tune parameters for better performance, as no parametrization is necessary. Furthermore, the WSA implementation is flexible to use either *hard* or *soft* assignment. In some methods of the literature, which are employed in the retrieval scenario using inverted files, only *hard* assignment is used [29, 89]. In WSA, when using hard assignment, the increment in the visual word counters is always by 1. On the other hand, when using soft assignment, the increment is proportional to the activation of the point to every visual word. For example, considering that $p_i$ activated $w_1$ in 0.8 and $w_2$ in 0.2, we increment the corresponding counters of $w_1$ by 0.8 and the corresponding counters of $w_2$ by 0.2.

The final WSA feature vector is the concatenation of all 4 counters of each visual

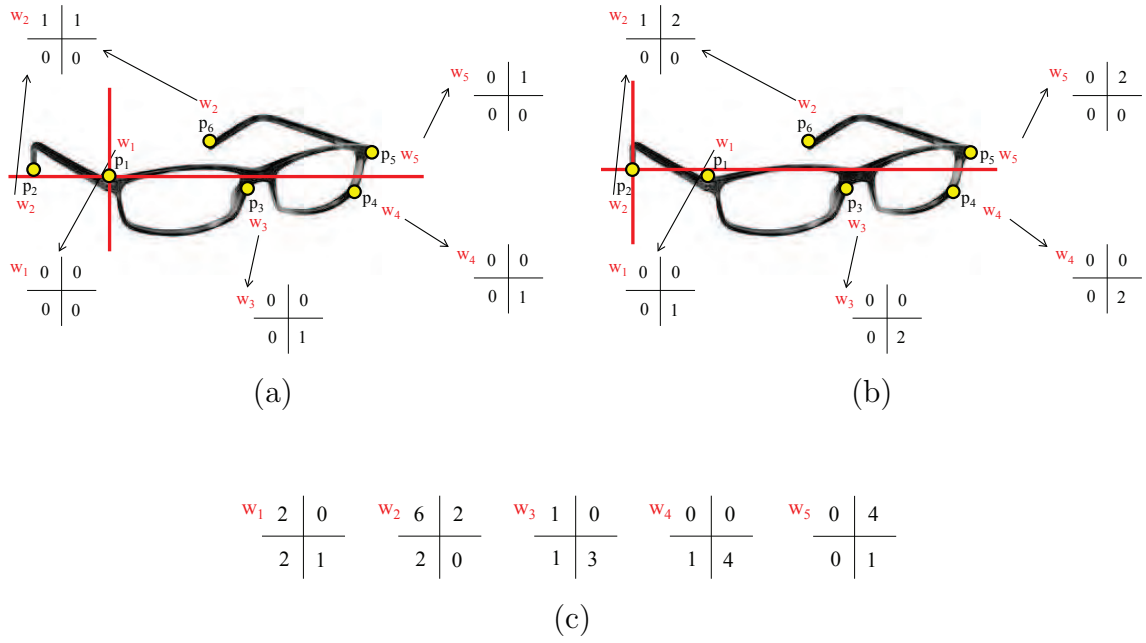



Figure 4.3: Example of partitioning and counting. The small circles are the detected points, tagged with their associated visual words ($w_i$'s). We start in (a), putting the quadrant's origin at $p_1$ and counting in the visual word associated with each other point, where the point is in relation to $p_1$. On the second step (b) the quadrant is at $p_2$; we add again the counters of the words associated with each other point in the position corresponding to their position in relation to $p_2$. We proceeded until the quadrant has visited every point in the image. Final counter values are shown in (c).

word, resulting in a feature vector of dimension 4×k, where $k$ is the dictionary size. The concatenation order is from the top-right to the bottom-right counter in counterclockwise direction.

### 4.3.1 WSA-window-weighted

As the WSA counting process considers all the points in the image, points that are far from the origin point and that possibly belong to background or to other objects will also be considered. Therefore, it would be better to consider in the counting process only points from the object where the origin point is located. We have implemented the use of windows around each origin point, aiming at capturing those points. The window size is determined by the scale of the origin point (the scale of a point is computed by the interest-point detector). Consequently, the approach keeps scale invariance.

In addition, the window has a Gaussian behavior over the counting process. Points

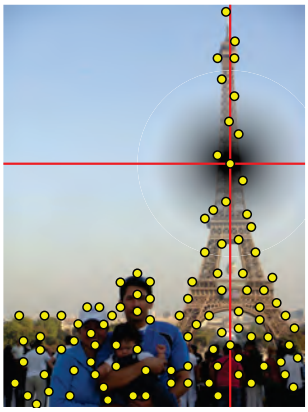


Figure 4.4: Toy example showing the use of a weighted window around the point during the WSA counting process. The window size is determined by the scale of the point and avoids considering points that are too distant in the counting process.

near the origin have higher weight in the counting than points far from it. Figure 4.4 shows an example of a weighted window around the origin point, avoiding considering distant points in the counting process. The equation to compute the weight $w$ when $p_i$ is the origin is:

$$w = \frac{1}{\sqrt{2\pi} \times \sigma} \times exp(-\frac{1}{2} \times \frac{d^2}{\sigma^2}) \tag{4.1}$$

where $d = D_{L2}(p_i, p_j)$ is the Euclidean distance between $p_i$ and $p_j$ and $\sigma$ is the scale of $p_i$ (determined by the interest-point detector).

In the experiments, we call *WSA-ww* the version that uses the Gaussian behavior of the window.

### 4.3.2 Distance function

In the retrieval scenario, a distance function is required to compare feature vectors. Therefore, we present here the distance function to be used with WSA.

The idea behind this function is somehow to assess if images contain the same visual words with the same spatial arrangement. Therefore, distances among points are computed only between corresponding visual words that present similar spatial arrangement. The effect is the same as first finding the matching visual words and then applying the spatial verification. However, as pointed in the beginning of Section 4.3, WSA does not require a post-processing step for spatial verification in retrieval scenarios. The reason is that, as the spatial information is already embedded into the feature vector, the

spatial verification can be performed while going through the feature vector. By "post-processing", we understand that, after finding the matching visual words, one is able to compute the spatial information and then perform the spatial verification, as it occurs with the methods presented in [88, 89], but not with WSA.

The retrieval scheme is based on the following distance function:

$$D_{Q,I} = \frac{\sum_{j=1}^{N_{WC}} dist_j(WSA_j^{(Q)}, WSA_j^{(I)})}{(distMax \times N_{WC})} \tag{4.2}$$

where

$N_{WC}$ is the number of visual words in common between the query image $Q$ and the database image $I$,

$dist_j$ is a distance function for the WSAs of common words,

$WSA_j$ is the WSA (4-values set) of word $j$,

$distMax$ is the maximum distance for one pair of WSAs.

The number of words in common $N_{WC}$ depends on the images. The distance function $dist_j$ for each pair of WSAs can be any, like the popular Euclidean (L2) or Manhattan (L1) distances. The maximum distance $distMax$ between a pair of WSAs depends on the distance function used. For the Euclidean distance, for example, it is $\sqrt{2}$, while for Manhattan distance, it is 2.

To consider a pair of corresponding visual words as a match (words in common), the distance between their WSAs needs to be lower than or equal to $\epsilon$. Otherwise, it is likely that the respective visual word is not present in both images. In the experiments, tests have been made with L1 and L2 distances, using $\epsilon$ equal to $\frac{1}{4}$, $\frac{1}{3}$, and $\frac{1}{2}$ of the maximum WSA distance ($distMax$).

## 4.4 Experiments for image retrieval

To evaluate the proposed approach considering the retrieval scenario, we have used two datasets. One dataset is composed of 600 synthetic images and the other collection is the popular Paris dataset[5]. Both datasets can be classified in the partial-duplicate search application because, for each category, the same object appears in different rotation and viewpoints.

The main questions to be answered by these experiments are:

[5] `http://www.robots.ox.ac.uk/~vgg/data/parisbuildings/` (as of February 6th, 2013).

| Pooling method | Acronym | Feature vector size |
|---|---|---|
| Average | avg | 1k |
| Max | max | 1k |
| Max pooling with Spatial Pyramids | max-SPM | 21k |
| Word Spatial Arrangement | WSA | 4k |
| WSA using weighted windows | WSA-ww | 4k |
| WSA using weighted windows and half of the original window size | WSA-$\frac{1}{2}$ww | 4k |
| WSA using weighted windows and a quarter of the original window size | WSA-$\frac{1}{4}$ww | 4k |

Table 4.1: Acronyms and feature vector sizes for the pooling methods being evaluated in the experiments for image retrieval. $k$ is the dictionary size.

- is the accuracy of WSA comparable to the best methods from literature?
- what is the impact of the soft assignment in WSA?

In our experimental setup, the images were represented by different methods based on the BoW approach. First, the Harris-Laplace detector [55] and the SIFT descriptor [50] were used to extract local feature vectors from images. Dictionaries of 15 000 and 8 000 visual words were constructed by randomly selecting points in the feature space [79] and they were used in the Base-600 and Paris datasets, respectively. As the datasets used here are related to the partial-duplicate search application, larger dictionaries are recommended [29]. We have varied the assignment method, using hard and soft assignment (according to Equations 2.1 and 2.2 presented in Section 2.3), the last with $\sigma$ varying in 30, 60, 90, and 150. The following pooling methods were compared: average pooling, max pooling, and max pooling with Spatial Pyramids. For WSA, we have used the standard version (WSA) and three versions that use the window around the origin point during the counting process: WSA-ww, WSA-$\frac{1}{2}$ww, and WSA-$\frac{1}{4}$ww. The last two versions use half and one quarter of the original window size presented in Section 4.3.1, respectively.

Table 4.1 summarizes the pooling methods and the size of their feature vectors. We have not used WSA with Spatial Pyramids because Spatial Pyramids enlarge the feature vectors and large vectors suffer from the curse of the dimensionality when computing distances. This is noticed when using the max pooling with Spatial Pyramids in the following experiments. We also have not used as baselines the other spatial pooling methods presented in Section 4.2 because the ones that are suitable for retrieval scenarios depend on performing the spatial verification as a post-processing step [88, 89] or they generate variable number of feature vectors per image [29].

The retrieval scenario requires distance computations between image representations. For the non-WSA representations, the Euclidean distance (L2) was used to compare the vectors. For WSA, we have used the distance function presented in Section 4.3.2. As

Figure 4.5: Sample images from the Base-600 dataset, highlighting 3 categories (one per row). There are 20 categories, each containing a particular object in different poses and orientations, and a random background.

the proposed distance function has some parameters, we have tested the variation of them (see Equation 4.2) and the results presented in this section consider one of the best configurations: $\epsilon=\frac{1}{2}distMax$ and $dist_j=L1$. In Appendix A, we explicitly show the results for all parameter combinations in both datasets.

The results are presented in terms of mean average precision (MAP) and precision for the top N retrieved images (P@N). It is important to highlight that, although MAP is a very popular measure to assess the effectiveness of CBIR methods, it does not reflect the ranking quality in the first positions. It only says how good a method is to retrieve all the relevant images. Considering an environment where the user analyzes the retrieved images visually, like the Web, it is crucial to have a good set of 10 or 20 retrieved images even if the MAP value is not good. Therefore, in that case, we are more interested in good P@N values than good MAP values. Other measures, like the *Normalized Discounted Cumulative Gain (NDCG)*, aim at also taking into account the ranking order, phenomenon that is not considered by P@N measures [1]. Results are reported with confidence intervals for $\alpha$=0.05 and are based on the number of query images used.

**Base-600** The first dataset used is composed of 600 synthetic images where there is a main object in the center over a heterogeneous background. This dataset, here called Base-600, simulates the partial-duplicate application and has 20 categories, each one containing 30 images. Each category refers to an object taken from the Coil-100 dataset[6] and each view of it was inserted in a different background, while keeping it in the center

[6] `http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php` (as of February 6th, 2013).

(a) WSA: L2 distance × proposed distance function

| | L2 distance | | | Proposed distance | | |
|---|---|---|---|---|---|---|
| **Pooling** | **MAP (%)** | **P@10 (%)** | **Assignment** | **MAP (%)** | **P@10 (%)** | **Assignment** |
| WSA | 12.56 ± 0.22 | 21.88 ± 0.67 | Soft ($\sigma$=60) | 26.46 ± 0.83 | 55.63 ± 2.05 | Soft ($\sigma$=30) |
| WSA-ww | 13.76 ± 0.47 | 24.35 ± 1.23 | Soft ($\sigma$=150) | 31.24 ± 0.85 | 68.47 ± 2.29 | Soft ($\sigma$=30) |
| WSA-$\frac{1}{2}$ww | 14.49 ± 0.51 | 25.73 ± 1.34 | Soft ($\sigma$=150) | 34.29 ± 0.67 | 75.03 ± 1.85 | Soft ($\sigma$=30) |
| WSA-$\frac{1}{4}$ww | 16.86 ± 0.62 | 30.95 ± 1.61 | Soft ($\sigma$=150) | **34.36 ± 0.60** | **76.15 ± 1.79** | Soft ($\sigma$=60) |

(b) Baselines

| **Pooling** | **MAP (%)** | **P@10 (%)** | **Assignment** | - | - | - |
|---|---|---|---|---|---|---|
| Avg | 20.89 ± 0.78 | 44.32 ± 2.15 | Soft ($\sigma$=90) | - | - | - |
| Max | **33.41 ± 0.67** | **74.73 ± 1.99** | Soft ($\sigma$=150) | - | - | - |
| Max-SPM | 25.87 ± 0.67 | 53.33 ± 2.10 | Soft ($\sigma$=150) | - | - | - |

Table 4.2: Base-600: We can clearly see that the proposed distance function is more adequate for WSA than L2. The parameter values for the proposed distance function are: $\epsilon$=$\frac{1}{2}distMax$ and $dist_j$=$L1$. Comparing the best WSA with the proposed distance in (a) to the best baseline in (b), we can see a similar performance. The best results in each table are shown in boldface. For each method, it was chosen the best assignment scheme (shown in the Assignment column).

of the images. The goal when using this dataset is to verify if the image representation is robust enough to encode the object properties without mixing background information. Good precision values are obtained when images containing the same main object are retrieved, disregarding their background. Figure 4.5 shows some images from Base-600.

For Base-600, we used a dictionary of 15 000 visual words and all images were used as queries.

Table 4.2(a) shows how the proposed distance function improves the performance of WSA in relation to L2 distance. For all WSA variations, the improvement is remarkably good. WSA presents MAP values around 12% for the L2 distance while for the proposed distance function, its MAP increases to more than 25%. WSA-$\frac{1}{4}$ww has its best P@10 of almost 31% for L2 distance and it increases to more than 75% with the proposed distance. We can also note that the smaller the window, the better for WSA.

The results for the baselines are presented in Table 4.2(b). We can see that max-SPM does not improve the performance over max pooling, giving a clear indication of the curse of dimensionality. Max-SPM presents one of the best results in the classification experiments (see Section 4.5), however, in the retrieval scenario its performance is degraded due to its large feature vector. Max pooling shows the best MAP and precision values when the assignment is very soft.

Comparing the best WSA configuration (WSA-$\frac{1}{4}$ww with soft assignment $\sigma$=60, using the proposed distance function) with the best baseline (max pooling with soft assignment $\sigma$=150), we can see a similar retrieval quality. Although there is a difference in favor of WSA in the average value, MAP and P@10 values are statistically equivalent. But considering Spatial Pyramids as a baseline in this scenario, using WSA with the proposed

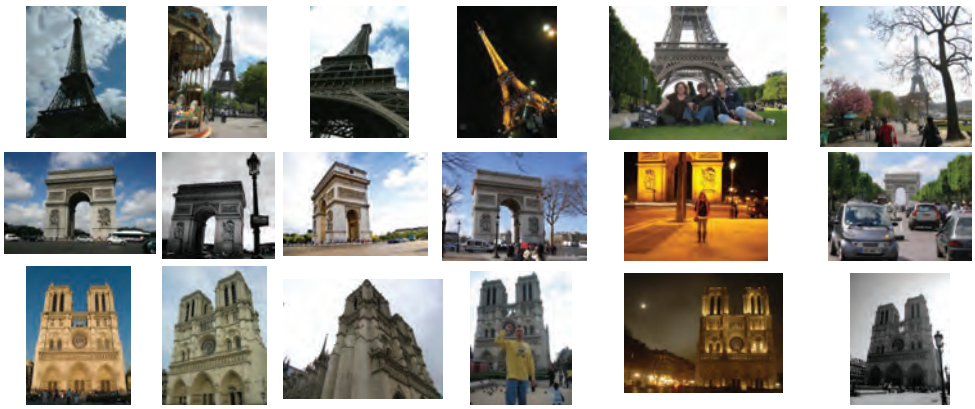


Figure 4.6: Sample images from the Paris dataset, highlighting 3 categories (one per row). There are 9 categories, each showcasing a landmark of the city of Paris, France.

distance function can improve the retrieval results by a difference of almost 10% in MAP and almost 20% in P@10.

We have created an interface based on Eva tool [60] to show the retrieved images of each pooling method and this interface is available online[3].

**Paris** Paris dataset is composed of more than 6 000 images divided into 9 categories of different sizes. Each category represents a monument in the city of Paris, France. Although divided into 9 categories, the relevance between images are not necessarily based on the categories. A set of 55 query images was specifically released by dataset creators for standard evaluation purposes. Each query has its own pool of relevant images. We have computed our MAP and P@N measures using the 55 query set and their respective pool. Figure 4.6 shows examples of Paris dataset images.

For the Paris dataset, we have used a dictionary of 8 000 visual words as it presented better performance in [29].

Table 4.3(a) shows the large improvement in results caused by using the proposed distance function with WSA. WSA without windows and soft assignment ($\sigma$=60) presents P@10 around 53% for L2 distance but, for the proposed distance, the P@10 value increases to almost 89%. A large improvement is also observed for WSA versions with windows. Contrarily to the results obtained for Base-600, in the Paris dataset, the larger the window, the better. WSA without the window was clearly superior to the other WSA versions that use the window, even with the L2 distance. The reason is that in Base-600 the main object, which is responsible for the dataset categorization, appears only in smaller size in relation to the image. Therefore, the use of windows was able to separate object and background information into the feature vector. In the Paris dataset, the monument of

(a) WSA: L2 distance × proposed distance function

| | L2 distance | | | Proposed distance | | |
|---|---|---|---|---|---|---|
| **Pooling** | **MAP (%)** | **P@10 (%)** | **Assignment** | **MAP (%)** | **P@10 (%)** | **Assignment** |
| WSA | 14.11 ± 2.11 | 53.27 ± 8.59 | Soft ($\sigma$=60) | **33.43 ± 4.17** | **88.91 ± 4.71** | Soft ($\sigma$=60) |
| WSA-ww | 6.77 ± 0.98 | 23.45 ± 4.75 | Soft ($\sigma$=60) | 21.00 ± 4.60 | 55.82 ± 9.08 | Hard |
| WSA-$\frac{1}{2}$ww | 5.92 ± 0.83 | 16.55 ± 3.19 | Soft ($\sigma$=60) | 17.06 ± 3.85 | 48.91 ± 8.18 | Soft ($\sigma$=30) |
| WSA-$\frac{1}{4}$ww | 5.97 ± 0.91 | 16.00 ± 2.52 | Soft ($\sigma$=60) | 14.72 ± 2.76 | 48.55 ± 8.02 | Soft ($\sigma$=60) |

(b) Baselines

| **Pooling** | **MAP (%)** | **P@10 (%)** | **Assignment** | - | - | - |
|---|---|---|---|---|---|---|
| Avg | 15.03 ± 3.64 | 58.18 ± 9.30 | Soft ($\sigma$=90) | - | - | - |
| Max | **28.68 ± 5.03** | **79.64 ± 7.08** | Soft ($\sigma$=150) | - | - | - |
| Max-SPM | 20.74 ± 3.64 | 69.64 ± 8.65 | Soft ($\sigma$=150) | - | - | - |

Table 4.3: Paris: The proposed distance function boosts WSA effectiveness in relation to L2. The parameter values for the proposed distance function are: $\epsilon$=$\frac{1}{2}distMax$ and $dist_j$=$L1$. Comparing the best WSA with the proposed distance in (a) to the best baseline in (b), we can see a similar performance. The best results in each table are shown in boldface. For each method, it was chosen the best assignment scheme (shown in the Assignment column).

interest has different sizes and appears in different positions into the images, therefore, a more general representation is necessary and was obtained by the WSA version without the windows.

Table 4.3(b) shows the results for the baselines using L2 distance. We can see that max pooling has the best effectiveness. As observed for Base-600, the use of Spatial Pyramids (max-SPM) does not improve the results of max pooling, giving an indication of the curse of dimensionality. Comparing the best WSA configuration to the best baseline, WSA presents the highest average MAP and P@10 values. Comparing the results of max-SPM and WSA, we can see that WSA is very superior both in terms of MAP and P@10. Therefore, considering the use of a spatial pooling method in retrieval experiments, WSA shows to be a promising choice, being more recommended than Spatial Pyramids because of its compact feature vector.

WSA has the best effectiveness than the best baseline (max pooling) in the average, but confidence intervals intersect. Therefore, we have performed a per-query analysis to better understand the difference between the methods. This kind of analysis puts into the statistical model the query variability, oppositely to the analysis shown in previous tables. The previous analysis excludes the query variability considering that their differences are noise in the statistical model. This is one of the reasons for the large confidence intervals presented previously. However, the previous analysis is useful to have a general idea of the performance of the methods evaluated. The per-query analysis solves this problem and gives a deeper understanding of how methods differ from each other. It is important to mention that for Base-600, as the main object for each category is always the same in

the middle of the image and has only few variations, a per-query analysis is not necessary, and the intra-class differences can be considered noise.

We have selected the best WSA configuration to compare with the best baseline configuration. The best WSA performance considering P@10 values was obtained when using soft assignment ($\sigma$=60) and the proposed distance function with parameters $\epsilon=\frac{1}{2}distMax$ and $dist_j=L1$. The best baseline performance was obtained by max pooling with soft assignment ($\sigma$=150) and the L2 distance.

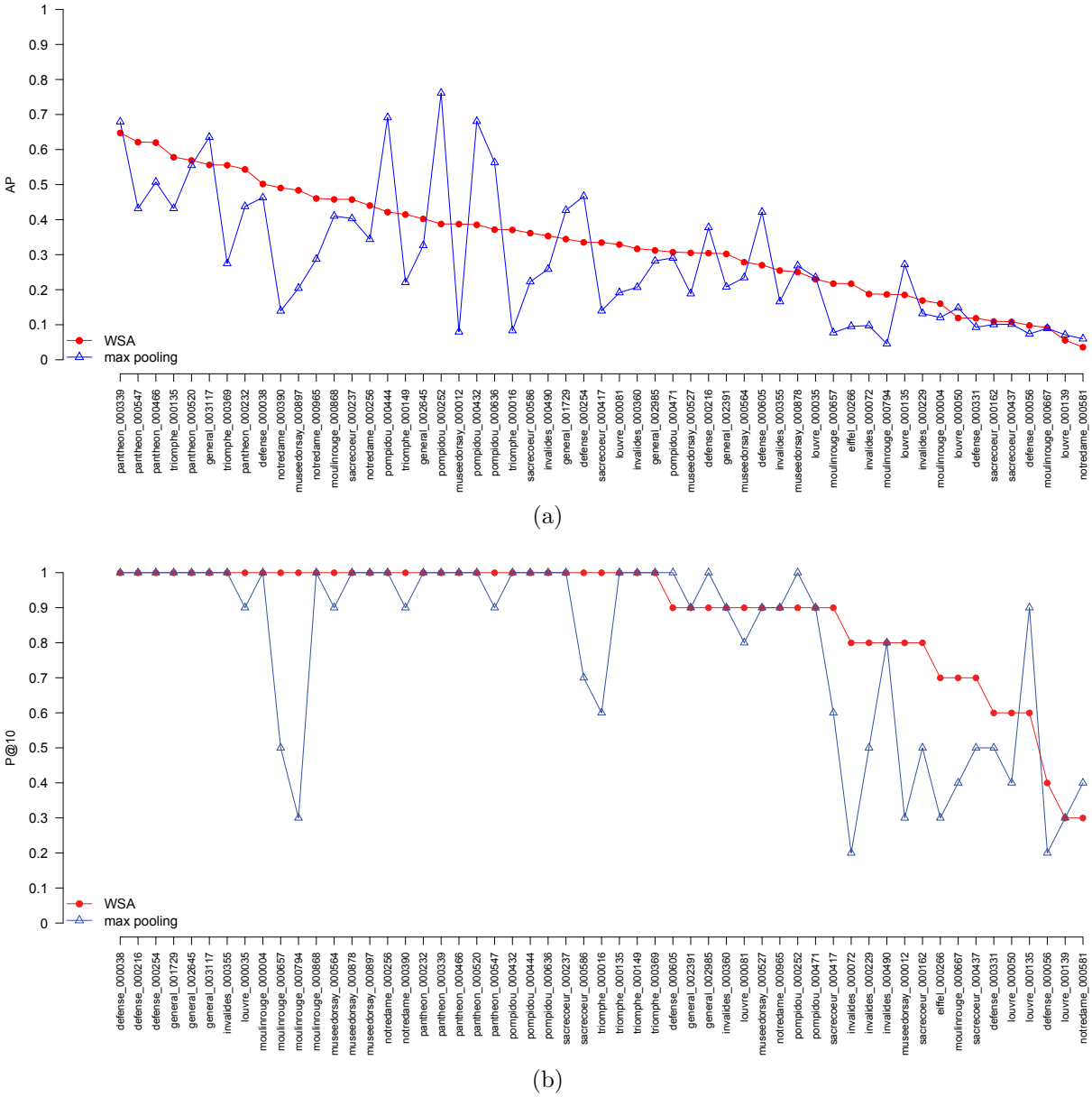
Our analysis uses S-curves and a paired-test. S-curves put in comparison the effectiveness measures obtained for each of the 55 query images. To plot a S-curve, we have selected WSA as the reference method, sorted the precision values of each query in decreasing order, and plotted them into the graph. Using the same query order obtained, we plot the precision values for the max pooling method. Figure 4.7 shows the results.

Analyzing the S-curves, we can see that WSA is better than max pooling for most of the queries. For AP values (Figure 4.7(a)), max pooling is better in only 16 queries (less than 30% of the total number of queries). For P@10 (Figure 4.7(b)), max pooling wins for only 5 queries, while WSA wins for 17. The largest precision difference in favor of WSA is around 70%. On the other hand, when max pooling is better than WSA, the differences in precision are smaller, being at most 40%. This means that, when WSA is less effective than max pooling, it is not so bad.

Figure 4.8 shows the results for the paired-test. As explained in Section 3.4, we compared the MAP or P@N values of two methods for all corresponding queries. The average and confidence intervals of those differences are used to indicate the best method. In case the confidence interval includes the zero, there is not statistical difference between them. Otherwise, the sign of the difference indicates the best method. In Figure 4.8, max pooling is the first method and WSA is the second, therefore, a positive value would indicate that max pooling is better and, a negative value, that WSA is better. Thus, for a confidence of 95%, WSA is better than max pooling for P@5, P@10, P@20, P@30, and MAP. Min and max show the extreme values for the average of the differences considering the confidence interval obtained.

An online interface is available to show the retrieved images of some pooling methods[4].

In this experimental setup, we focus on evaluating the retrieval effectiveness, therefore, we are not providing experiments measuring the efficiency of methods. Literature has shown works aiming at compacting the image representation in order to obtain scalability [29,33]. Therefore, the matter of small feature vectors for image retrieval is important. We can point that as WSA has a larger feature vector than avg and max pooling, it will be less efficient. However, considering most of the spatial pooling approaches and specially the Spatial Pyramids, WSA has a more compact feature vector, which makes it more efficient. Additionally, as WSA computes a vector of 4 dimensions for each visual word,




(a)

(b)

Figure 4.7: S-curves for the best WSA configuration (red line) and the best max pooling configuration (blue line) in the Paris dataset. The S-curves highlight the performance (vertical axis) of the two methods for each of the queries (horizontal axis), allowing to appreciate visually how often one outperforms the other. Performance measurements: (a) AP and (b) P@10.

indexing them to accelerate retrieval does not represent a challenge. Tools like inverted files or customized trees such as in [29] should be considered as relevant solutions.

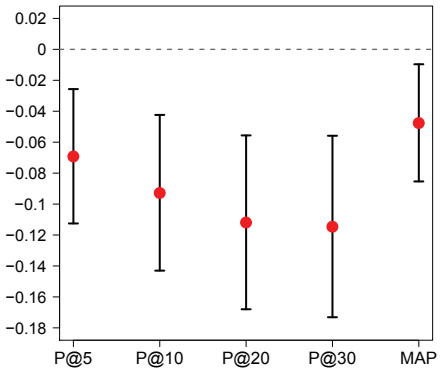



Figure 4.8: Paris dataset: paired-test comparing max pooling and WSA. As the min and max values are always negative (do not include the zero), the test indicates a superiority of WSA.

**Conclusions** Considering the questions presented in the beginning of this section, we can point that WSA has better effectiveness than the most popular approach to spatial pooling, the Spatial Pyramids. WSA has also shown comparable performance to max pooling in Base-600 and better performance in the Paris dataset, which represents a more real scenario of use. The use of the proposed distance function to be used with WSA has shown large improvements in effectiveness when compared to the L2 distance.

WSA has shown some improvements when using soft assignment, nevertheless, it does not work well with very soft assignments. The reason is that many words are assigned to each point, resulting in the increment of counters of too many words during the WSA counting process.

The results presented also indicate the importance of compact feature vectors in the retrieval scenario. We could observe that the use of Spatial Pyramids did not improve the performance of max pooling, having, in fact, reduced its discriminating power. This is an indication of the curse of the dimensionality.

We could observe that the use of the weighted window in the counting process was good only for Base-600, where the main object is small and centrally located in all the images. For the Paris dataset, WSA without windows had better effectiveness.

As summary, we conclude that the spatial information encoded by WSA can improve the effectiveness of retrieval systems without suffering from large feature vectors, usually generated by many spatial pooling methods.

| Pooling method | Acronym | Feature vector size |
|---|---|---|
| Average | avg | 1k |
| Max | max | 1k |
| Max pooling with Spatial Pyramids | max-SPM | 21k |
| Word Spatial Arrangement | WSA | 4k |
| Word Spatial Arrangement with Spatial Pyramids | WSA-SPM1 | 20k |

Table 4.4: Acronyms and feature vector sizes for the pooling methods being evaluated in the experiments for image classification. $k$ is the dictionary size.

## 4.5 Experiments for image classification

The experiments in the classification scenario are based on traditional image datasets which comprise the semantic-search application. We focus our experiments on evaluating scene categorization using the 15-Scenes dataset [43] and object categorization using the Caltech-101 dataset [25].

The main questions to be answered by these experiments are:

- is the accuracy of WSA comparable to the best methods from literature?
- what is the impact of soft assignment in WSA?
- can WSA performance be improved by combining it with spatial pyramids?

The images were represented using the same configurations presented in the retrieval experiments in Section 4.4: dictionaries based on the Harris-Laplace detector and the SIFT descriptor, combined with several assignment and pooling strategies. However, dictionaries of 1 000 words were used because, in the 15-Scenes and Caltech-101 datasets, small dictionaries are commonly used [11, 26]. The following pooling strategies were employed: average, max, max with Spatial Pyramids (max-SPM), WSA, and WSA with Spatial Pyramids (WSA-SPM1). For WSA, we have used Spatial Pyramids of level 1 (5 WSA vectors concatenated). We have not used Spatial Pyramids of level 2 for WSA, because this would make the feature vector larger than max-SPM. WSA-SPM1 (5×4×k) is still more compact than max-SPM (21×k).

Table 4.4 summarizes the pooling methods and their feature vector sizes. Spatial Pyramids (SPM) were used as our main baseline for spatial pooling of visual words, because, although there are many new approaches with better and comparable results to SPM, SPM still are the most widely used. Another advantage is that SPM can be used together with many new methods, as well as with WSA. The other spatial pooling methods adequate for the classification scenario presented in Section 4.2 were not used because they present limitations. The spatial-bag-of-features [15] generates extremely large feature vectors and the geometric $l_p$-norm pooling [26] depends on resizing all the

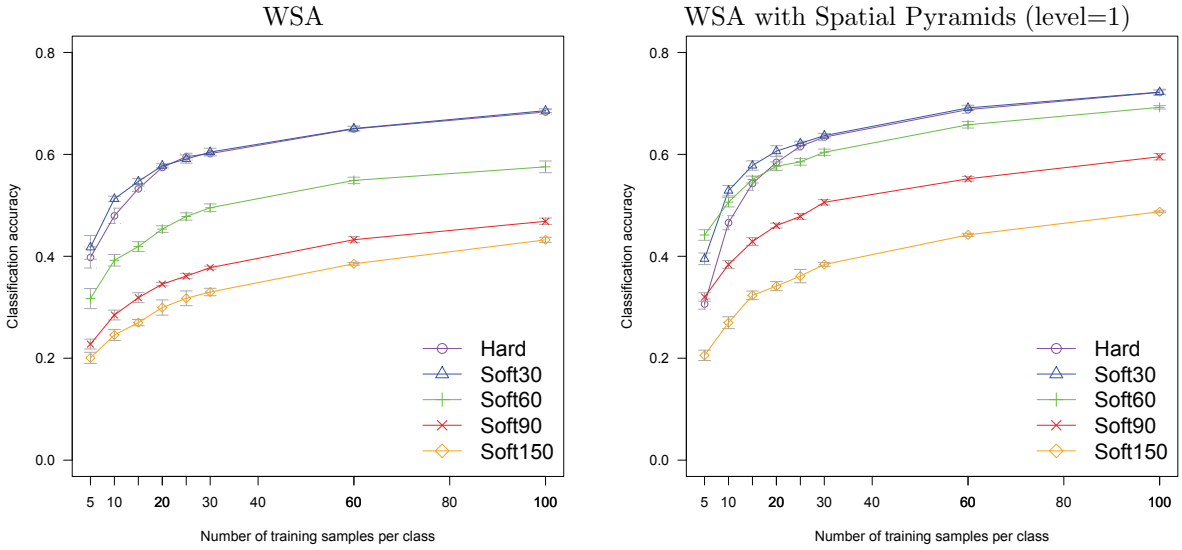



Figure 4.9: Evaluating the effect of soft assignment for WSA and WSA-SPM1 in the 15-Scenes dataset for variable training set sizes. WSA-SPM1 suffers less than WSA with the increase of the assignment softness. However, both methods have a decrease in performance for $\sigma \geq 60$.

images to the same size. Using dimension reduction techniques or special treatments for individual methods were not in the scope of our experiments, because they can create advantages for a specific method and make the comparison unfair.

We are also not showing the results of WSA-ww, because it presented inferior accuracy than the WSA version that does not use windows. This also happened in the retrieval experiments on the Paris dataset (see Section 4.4). WSA-ww was good only in the retrieval experiments on Base-600, where the main object was in the middle of the image and in small size in relation to the whole image. These characteristics are not present in the 15-Scenes dataset neither in the Paris dataset, therefore we would expect that WSA without windows would perform better than WSA-ww. In relation to Caltech-101, many categories contain the object of interest in the middle of the image as in Base-600, however, in Base-600 the object is exactly the same for a given class while this is not true for Caltech-101.

For the classification setup, we have employed SVMs with linear kernel ($c$=1.0) and a balanced validation. A number of samples per class ($nTrain$) was taken for training and the rest were used for testing. We have varied $nTrain$ from 5 to 100 in the 15-Scenes dataset and from 5 to 30 in the Caltech-101 dataset. Results are reported with confidence intervals for $\alpha$=0.05 for the 5 runs of each balanced validation.

**15-Scenes** Figure 4.9 shows how the WSA descriptors react to different assignment softness. We can see that both WSA and WSA-SPM1 have a decrease in accuracy when

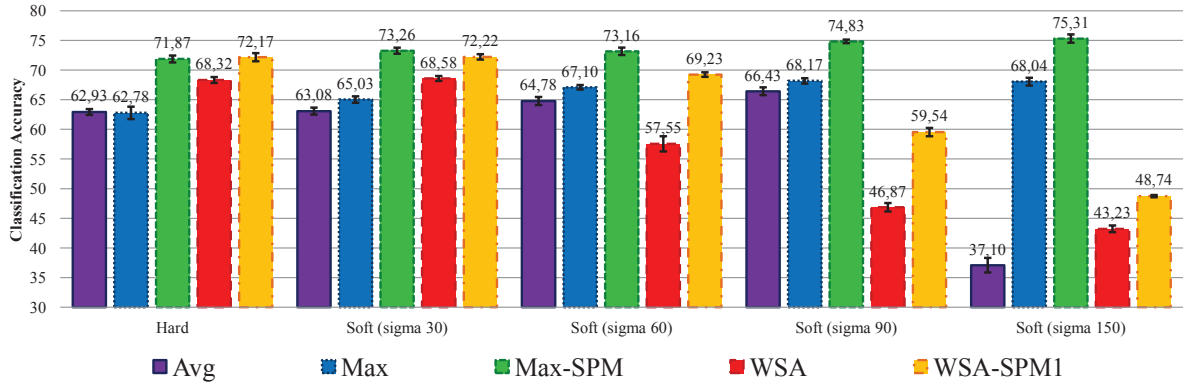



Figure 4.10: 15-Scenes: average classification accuracies with confidence intervals for $nTrain$=100.

the assignment becomes softer. However, WSA-SPM1 suffers less than WSA. We can also note that WSA-SPM1 has a larger increase in accuracy as the training set grows. This means that WSA alone is more robust in conditions of smaller training sets.

Figure 4.10 aggregates all the results for $nTrain$=100. The graph shows how each method performs when changing the assignment softness. Average pooling, as well as WSA methods, suffers more when the assignment becomes softer, while max pooling benefits from this phenomenon. We can also compare the methods in each assignment schema. For harder assignments (hard and soft $\sigma$=30), we can note that WSA outperforms avg and max pooling. In relation to max pooling, the differences in accuracy in favor of WSA, considering the confidence intervals, are around 4% and 2.5% for the above mentioned assignments, respectively. Although WSA is outperformed by max-SPM in those assignments, the differences in favor of max-SPM, considering the confidence intervals, are around only 2% and 3.5%, respectively. As the assignment increases, max pooling and max-SPM tend to benefit from that while WSA is harmed. Therefore, in very soft assignments, WSA presents low accuracies. Considering the use of Spatial Pyramids with WSA (WSA-SPM1), we can see a great improvement in accuracy. For harder assignments, the gain is around 4%, while for softer assignments ($\sigma$=60 and $\sigma$=90), the gain is sometimes greater than 10%. Also, WSA-SPM1 has equivalent accuracy to max-SPM for harder assignments.

Table 4.5 shows a comparison for individual classes of the 15-Scenes dataset considering the best non-spatial baseline configuration (max pooling with soft assignment ($\sigma$=90)) and the best WSA configuration (WSA with soft assignment ($\sigma$=30)). Methods are equivalent in most of the classes, but in some of them there is statistical difference. A paired-test comparing the results per class shows that, for $nTrain$=100, the methods are equivalent.

| Class | Max(soft-$\sigma$=90) | WSA(soft-$\sigma$=30) | Winner |
|---|---|---|---|
| bedroom | 49.66 ± 2.70 | 53.79 ± 3.61 | - |
| CALsuburb | 93.90 ± 2.35 | 90.07 ± 2.11 | - |
| industrial | 37.16 ± 3.16 | 37.54 ± 3.35 | - |
| kitchen | 38.00 ± 3.21 | 49.64 ± 1.45 | WSA |
| livingroom | 45.50 ± 1.80 | 50.05 ± 4.30 | - |
| MITcoast | 76.77 ± 3.94 | 81.77 ± 1.54 | - |
| MITforest | 92.72 ± 1.38 | 92.72 ± 1.04 | - |
| MIThighway | 77.63 ± 3.47 | 76.00 ± 2.11 | - |
| MITinsidecity | 65.77 ± 1.09 | 61.73 ± 3.25 | - |
| MITmountain | 79.05 ± 3.24 | 71.68 ± 2.56 | Max |
| MITopencountry | 67.48 ± 1.41 | 67.03 ± 2.21 | - |
| MITstreet | 67.92 ± 2.39 | 72.81 ± 1.56 | WSA |
| MITtallbuilding | 78.52 ± 1.39 | 73.36 ± 2.49 | Max |
| PARoffice | 69.57 ± 2.22 | 69.91 ± 3.68 | - |
| store | 60.28 ± 3.47 | 66.51 ± 3.95 | - |

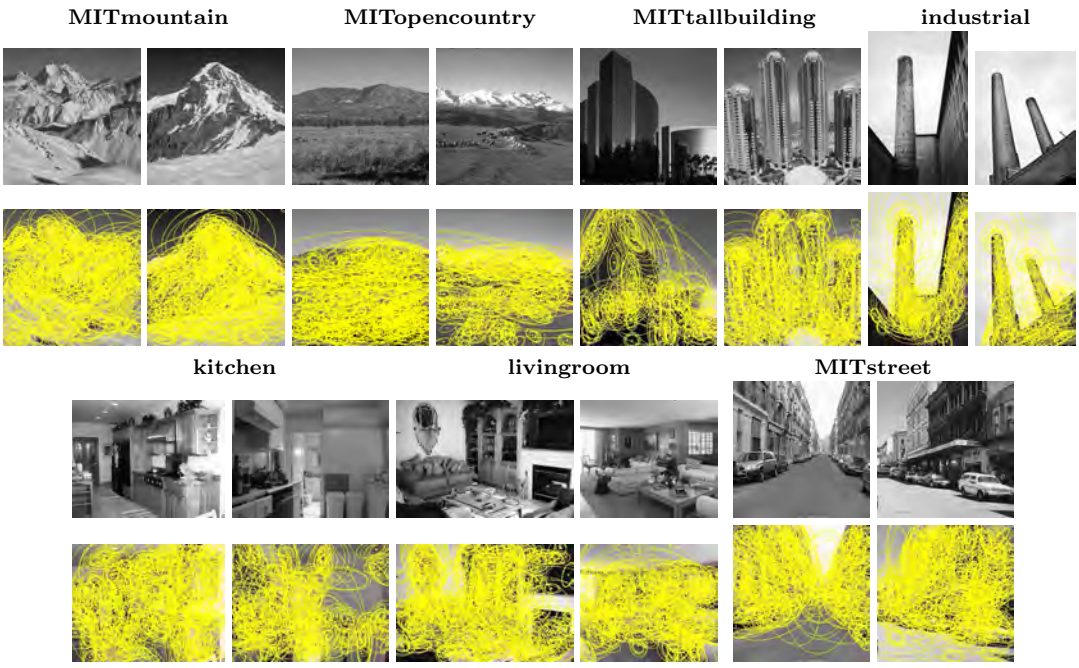



Table 4.5: Contrasting the performance of WSA and max pooling in the classes of 15-Scenes dataset for *nTrain*=100. In *kitchen* and *MITstreet*, WSA significantly outperforms max pooling, while in *MITmountain* and *MITtallbuilding*, the opposite happens. We show examples of images from classes where there is a meaningful difference between WSA and max pooling. There are also images from the classes which are confused by the methods. Those images were obtained by analyzing the confusion matrices of the results. Below each image, we show the points detected by using the Harris-Laplace detector.

Table 4.5 also shows images from the classes where there is a meaningful difference between WSA and max pooling. We are also showing images from classes which are confusing for the methods. They were obtained from an analysis in the confusion ma-

trices of each method. WSA is worse than max pooling in classes *MITmountain* and *MITtallbuilding*.

For *MITmountain*, when WSA is wrong, it confuses *MITmountain* with *MITopencountry*. We could note that many images from both classes have clear sky, as the ones shown in Table 4.5, which means that no points are detected in the top part of the images, but many points appear the lower part (see the images just below each original image in Table 4.5). Therefore, the spatial relationship between the lower parts of those images were probably not enough to distinguish between the two classes.

For the class *MITtallbuilding*, WSA is confusing with the class *industrial*. We can suggest that the spatial relationship between the tall structures are generating similar WSA representations.

When WSA wins, max pooling makes confusion between *kitchen* and *livingroom* and also between *MITstreet* and *industrial*. For class *kitchen*, there must be a large intersection between their visual words and the visual words present in *livingroom*. Therefore, their spatial relationship is more important to distinguish between those classes. For class *MITstreet*, the spatial arrangement of visual words present in the tall structures (buildings for *MITstreet* and chimneys for *industrial*) and the other structures could improve significantly the discrimination between those classes, and this information was not captured by max pooling.

To summarize the results in the 15-Scenes dataset, WSA is worse than max-SPM for softer assignments, but for harder assignments WSA outperforms max pooling and is only a bit below max-SPM. As WSA has a vector more than 5 times smaller than max-SPM, it would be more efficient in terms of time and space. Therefore, WSA is a good option to encode the spatial arrangement of visual words for scene categorization while saving storage space and classification time.

**Caltech-101** Figure 4.11 shows how WSA methods perform in different assignment schemes when varying the training set size. We can notice the same aspects when using the 15-Scenes dataset: WSA and WSA-SPM1 have a decrease in accuracy when the assignment becomes very soft. However, both methods benefit from the soft assignment at a certain amount. WSA-SPM1 is again more robust to softer assignments increasing its accuracy for assignments with $\sigma$ up to 90. WSA has an increase in accuracy for $\sigma$ up to 60.

The graph in Figure 4.12 shows the overall results for all methods in the different assignment schemes tested for the Caltech-101 dataset, using $nTrain$=30. We can see that WSA outperforms both avg and max pooling for harder assignments (hard, soft $\sigma$=30, and $\sigma$=60). In relation to max pooling, the differences in accuracy in favor of WSA, considering the confidence intervals, are around 2.5%, 4.8%, and 2.8% for the above men-

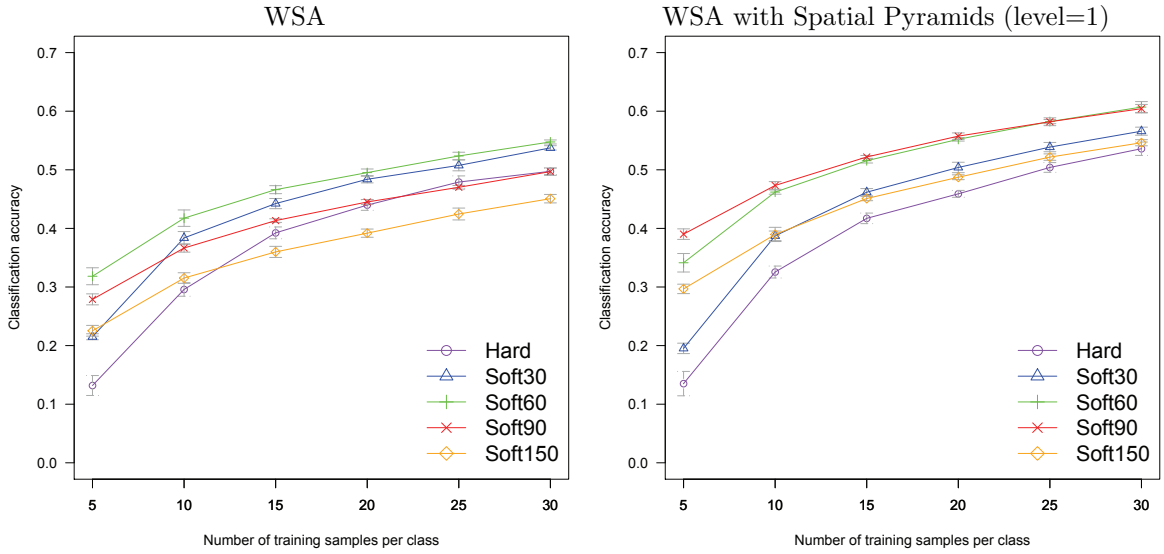



Figure 4.11: Evaluating the effect of soft assignment for WSA and WSA-SPM1 in the Caltech-101 dataset for variable training set sizes. Both methods have a decrease in accuracy when increasing the value of $\sigma$ in the soft assignment, however, WSA-SPM1 suffers has than WSA.

tioned assignments, respectively. Again, WSA is outperformed by max-SPM, but the differences are still only around 2.7% and 1.1%, for hard assignment and soft assignment ($\sigma$=30), respectively. For assignments softer than those, max-SPM keeps increasing its accuracy, while WSA suffers from that. Considering WSA combined with Spatial Pyramids (WSA-SPM1), we can note again a great improvement in the performance in relation to WSA alone. The accuracy difference increases from around 3% for harder assignments to around 10% for softer assignments. Besides that, we can see that WSA-SPM1 has comparable accuracy to max-SPM for assignments until $\sigma$=90.

A per-class analysis was also performed for Caltech-101 and the results were very similar to the results presented for 15-Scenes. Considering the best configurations of the non-spatial baseline (max-soft($\sigma$=150)) and the best WSA configuration (WSA-soft($\sigma$=60)), they are equivalent for most of the classes and a paired-test shows that they are equivalent in general.

As summary, the experiments in Caltech-101 show that WSA does not win in classification accuracy in relation to max-SPM. However, WSA improves accuracy over max pooling for harder assignments and is sometimes (soft $\sigma$=30) very close to max-SPM. This means that WSA can improve object categorization by including spatial information of visual words in a compact feature vector, being an interesting alternative to save storage space and classification time in relation to Spatial Pyramids. If accuracy is more important than efficiency, WSA can be used together with Spatial Pyramids (WSA-SPM1)

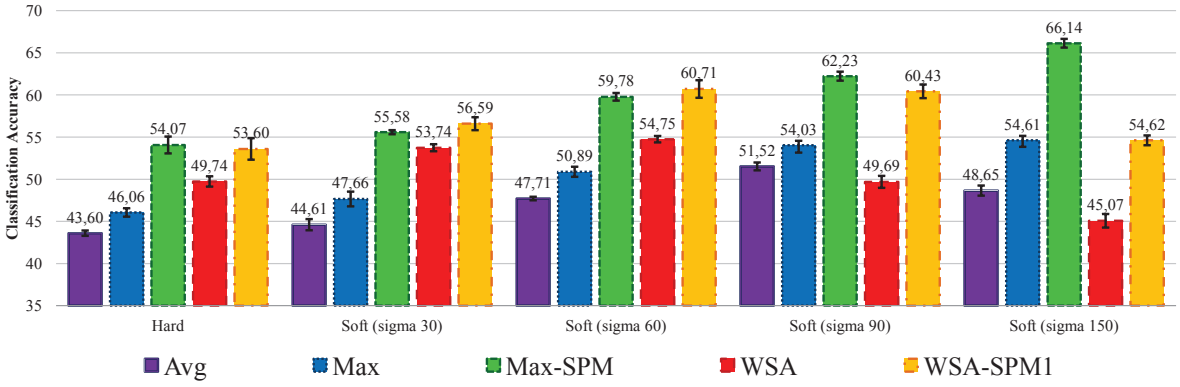



Figure 4.12: Caltech-101: average classification accuracies with confidence intervals for $nTrain$=30.

to achieve comparable accuracy to max-SPM, yet saving some space. Therefore, if storage is a constraint in the classification system and classification time is important, WSA can include spatial information of visual words keeping compact feature vectors and still increasing accuracy rates over non-spatial methods.

**Conclusions** Considering the questions presented in the beginning of this section, we can draw our conclusions about the performance of WSA in the classification scenario:

- WSA has better performance than the non-spatial baselines and has comparable performance to Spatial Pyramids, specifically for harder assignments ($\sigma \leq 60$).
- Soft assignment brings some improvements for WSA, nevertheless, WSA does not perform well with very soft assignments ($\sigma \geq 90$).
- Considering the use of WSA in combination with Spatial Pyramids, we could note improvements in the classification accuracy.

The large improvement when using Spatial Pyramids with WSA on softer assignments indicates that, when many visual words are activated by each point in the image, WSA alone tends to increment the counters of too many words and this fact reduces its discriminating power (as observed in the low accuracies for softer assignments). The improvements in accuracy, for WSA-SPM1 over WSA, are around 4% for harder assignments and increases to around 10% for softer assignments in both datasets used. When the image is split (using Spatial Pyramids), there are less points to be considered and less counters to increment. This shows a way of potentially improving WSA.

Our classification experiments show that WSA is a good option for classification systems requiring better accuracies than traditional non-spatial pooling methods. WSA is

recommended in place of max-SPM if storage is an important constraint for the system, because it saves space in a compromise of loosing a little accuracy in some cases in relation to Spatial Pyramids, but having comparable performance in others, specially for harder assignments. Smaller feature vectors also lead to faster classification, which is another advantage of WSA.

## 4.6 Discussion

This chapter presented WSA (Word Spatial Arrangement), a spatial pooling approach to encode the spatial arrangement of visual words. WSA has the advantage of working both in retrieval and classification scenarios. WSA encodes the relative position of visual words in the image by splitting the image space using each point as the origin of a four-quadrant structure and counting the number of points in each quadrant.

To work in the retrieval scenario, we have also proposed a distance function to be used with WSA. Experimental results show that the proposed distance function remarkably improves WSA effectiveness over the Euclidean distance. Experiments in the retrieval scenario also show that WSA outperforms the most popular approach to spatial pooling, the Spatial Pyramids. The latter degraded the performance of max pooling, giving a clear indication of the curse of the dimensionality in scenarios where distance computations are required. A per-query analysis by S-curves and a paired-test has shown that WSA is also superior than max pooling, the best baseline in our retrieval experiments. We also provide an online interface to navigate through the results.

Experiments in the classification scenario have shown that WSA has close accuracy to max pooling with Spatial Pyramids (max-SPM) in harder assignments. For configurations of very soft assignments, max-SPM is superior. However, WSA computes vectors more than 5 times smaller than max-SPM, which is a clear advantage considering efficiency, both in terms of time and space. Anyhow, if accuracy is priority, WSA can also be combined with Spatial Pyramids, boosting its performance.

By using WSA, we could show the power of the spatial information to differentiate types of scenes and objects. It is important to highlight that WSA encodes *only* the spatial information of visual words, that is, it does not encode the frequency of occurrence of visual words in the image. The spatial information has shown to be very discriminating, being, in some cases, more discriminating than the information of frequency of occurrence.

# Chapter 5

# Semantic information in visual dictionaries

This chapter presents our studies concerning the third hypothesis under analysis in this thesis, which is related to the fact that the traditional visual dictionaries do not contain semantic information. We first present in Section 5.2 an analysis on the semantic separability between distance distributions for the feature spaces involved in the visual dictionary model: the space of low-level descriptions and the BoW (mid-level) space. The results of the analysis motivate the use of dictionaries based on elements containing more semantics. Then, in Section 5.3, we present the proposed dictionary which is based on visual words which carry more semantics.

## 5.1 Introduction

The idea of using visual dictionaries to describe images has already shown its potential, as we could see throughout the previous chapters in this thesis and in the vast literature in the visual recognition area. However, the traditional visual dictionaries are based on low-level features extracted from local patches, which have no semantic information [11, 34, 44, 48, 80]. The results presented in Chapter 3 also show that, for generating a good visual dictionary, we do not need to have a sample with high variability in terms of semantics, but we need a sample with high variability in terms of visual appearances. Therefore, we could say that the term *dictionary* is somewhat misleading, because their visual words carry no semantics. We could also question why the BoW model works so well in different scenarios, even with this peculiarity. And what would happen if we could use visual words which carry more semantic information?

There are some recent papers in literature that explore the potential of using dictionaries containing more semantic information [11, 12, 34, 44, 48, 52, 80]. The semantic-aware

dictionaries presented in [11, 34, 48] use class labels to supervise the dictionary creation, although their dictionaries are still based on local descriptions.

There are other works which use dictionaries based on elements containing more semantics, like objects or parts of people. The use of semantically enriched elements to compose the dictionary can be simply understood as the use of visual words that are more representative for humans. For example, a whole scene has more semantics than a small corner detected in an image. Li et al. [44] work on a model where an image is represented as a scale-invariant response map of a large number of pre-trained generic object detectors. Bourdev and Malik [10] perform people detection by using poselets. Poselets are parts of human poses under different viewpoints. Their representation is an activation vector of such poselets. Their work is also extended by Brox et al. [12] to deal with other elements besides people poses.

Those works above mentioned show a trend in visual representation, where the concept of visual word is modified. We call this model as the *bag-of-prototypes* model, according to which the prototypes are elements containing semantics. We defend that this model is promising to improve the quality of image and video representations, reducing the semantic gap [71].

To have a better understanding on how the semantic information is involved in the current visual dictionary of local features, we first present in the following sections an analysis on the semantic separability based on distance distributions considering the feature spaces comprised by such model, that are the low-level and the BoW (mid-level) feature spaces. Then, in Section 5.3, we present our proposal of a specific case of a dictionary of prototypes. Our proposed representation, called *bag-of-scenes*, is based on a dictionary of scenes and is evaluated in the context of the video geocoding problem [65].

## 5.2 Semantic analysis

The clustering step used to quantize the feature space during the dictionary creation splits the feature space into regions containing patches with similar appearance. Thus, visual dictionaries based on local patches are composed of visual words which are very local. We would expect that a visual dictionary carries a little semantics, as those descriptions are based on appearance.

We prepared an experimental setup to evaluate if the semantic information is encoded considering two different feature spaces: the low-level feature space and the BoW (mid-level) feature space. Although the fact that visual dictionaries have no semantics must be already known by most of the research community [34, 44, 48, 80], to the best of our knowledge, there are no objective experiments showing that.

Our experiments use the Pascal VOC 2010 dataset [22] which contains 11 321 images

with one or several objects per image. There are 20 different types of objects which could appear one or multiple times in each image. We have used Pascal VOC 2010 because it has bounding boxes for the objects, making it easier to distinguish points detected in objects and points detected in the background.

We analyze the separability between distance distributions of different semantic classes of points or objects. Section 5.2.1 explores the semantic separability in the low-level feature space and Section 5.2.2 shows the semantic separability in the BoW feature space.

### 5.2.1 Semantic separability in low-level space

The hypothesis to be evaluated in this section is that there is semantic separability between distance distributions considering different semantic classes of points in the low-level feature space. To evaluate that, we have computed histograms of distances between points, considering points inside objects and points in the background. We have performed experiments that reflect the behavior of average and max pooling, as an analogy of what they compute during their pooling steps. To easily distinguish between them, we will call them as *average pooling* analysis and *max pooling* analysis.

For the average pooling analysis, experiments were configured to answer the following questions:

- how is the distribution of distances between random pairs of points from objects of the same category?
- how is the distribution of distances between random pairs of points, being one point from an object and the other from the background?

These experiments generate two distance distributions: $Hist^{avg}_{obj \times obj}$ and $Hist^{avg}_{obj \times bg}$. Figure 5.1 shows toy examples of pairs of points considered in each of the two distance distributions. The reason to call this as an average pooling analysis is that, by considering the distances between all pairs of points to compute the histograms, we have a similar behavior of considering all assignment values in an image when using average pooling.

We expect to have distances between pairs of points from objects smaller than distances between points from object and points from background. It is intuitive to believe that local visual appearances of objects from the same category are more similar to themselves than to local appearances from the background.

For the max pooling analysis, we designed experiments to answer the following questions:

- how is the distribution of distances between a point from an object and its most similar point in another object of the same category?

(a) Object to Object (b) Object to Background

Figure 5.1: Toy example based on the *person* category showing pairs of points considered to compute (a) $Hist^{avg}_{obj \times obj}$ and (b) $Hist^{avg}_{obj \times bg}$.

- how is the distribution of distances between a point from an object and its most similar point in the background?

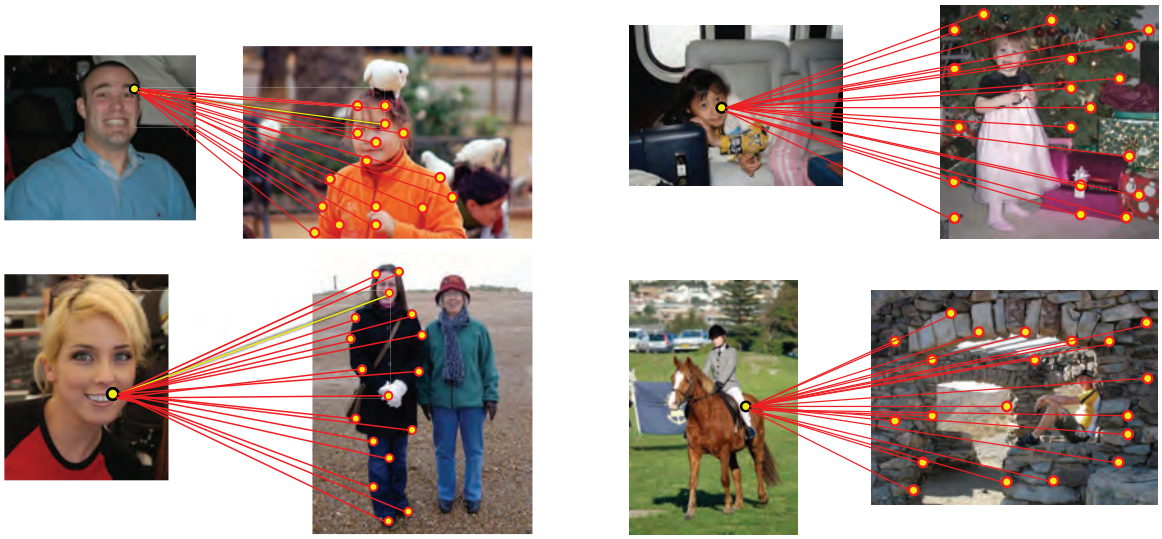
These experiments generate two new distances distributions: $Hist^{max}_{obj \times obj}$ and $Hist^{max}_{obj \times bg}$. Figure 5.2 shows toy examples of points considered in each of the two distance distributions. This configuration reflects the behavior of *max pooling* because, as we analyze the distances for the most similar points in other objects or in background, we have an analogous effect of selecting the maximum assignment value of a point to a visual word.

We expect that the results for the max pooling experiments show that the distances between pairs of points from objects are smaller than distances of points from objects and points from background. Again, this is expected because it is intuitive to believe that patches from objects of the same category are more similar to themselves than to patches of the background.

The details of the experimental setup are the following:

- low-level features extraction using dense sampling (6 pixels) [75] and the SIFT descriptor [50];

- classification of points, separating them into object points and background points:



(a) Object to Object (b) Object to Background

Figure 5.2: Toy example based on the *person* category showing pairs of points considered to compute (a) $Hist^{max}_{obj \times obj}$ and (b) $Hist^{max}_{obj \times bg}$.

  - if the point is inside any bounding box, it is an object point[1];
  - if the point is outside all bounding boxes, it is a background point.

- *average pooling* analysis:
  - select two sets of 10 thousand object points, with no intersection between them;
  - select one set of 10 thousand background points;
  - compute distances, in an aligned fashion[2], between two sets of object points, generating 10 thousand distance values;
  - compute distances, in an aligned fashion, between one set of object points and the set of background points, generating 10 thousand distance values;
  - compute a histogram for the object-to-object distances ($Hist^{avg}_{obj \times obj}$);
  - compute a histogram for the object-to-background distances ($Hist^{avg}_{obj \times bg}$).

- *max pooling* analysis:

[1]We know that some points are inside the bounding boxes but outside the real objects, because bounding boxes cover an area larger than the object. However, we believe that most of the points should appear in the object.

[2]First point of the first set is compared to the first point of the second set, then, the second point of the first set is compared to the second point of the second set, and so on.

- select object points from 10 thousand random objects;
- select background points from 10 thousand random images;
- compute the minimum distance between one object point to all the object points of another object of the same class, generating 10 thousand distance values (1 distance value for each query point);
- compute the minimum distance between one object point to all background points of a given image, generating 10 thousand distance values (1 distance value for each query point);
- compute a histogram of the minimum object-to-object distances ($Hist^{max}_{obj \times obj}$);
- compute a histogram of the minimum object-to-background distances ($Hist^{max}_{obj \times bg}$).

We have used the Euclidean distance as it is adequate for the SIFT feature space.

Figure 5.3 shows the distance histograms for the 5 easiest and the 5 hardest classes. The easiest and hardest classes were chosen according to the results of the classification task in [22]. The histograms for the other 10 classes not shown here were similar.

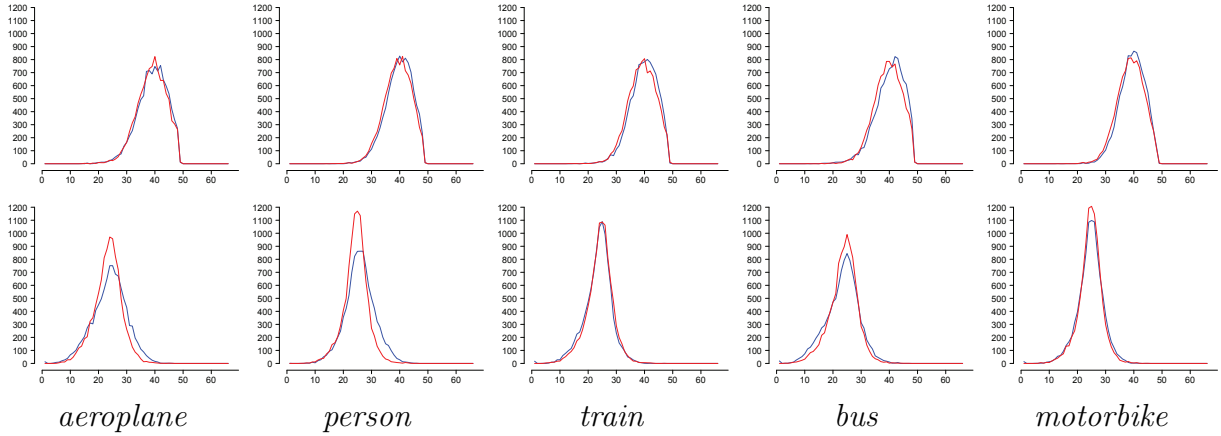
Observing the general aspects of the curves, we can note that, in all classes, the average pooling histograms are very similar. The distribution of object-to-object distances (in blue) always overlaps completely the distribution of object-to-background distances (in red). This means that, on average, object local patches have no difference to background local patches. That is, there is no separability between distance distributions of object points and background points in the SIFT feature space.

Analyzing the max pooling curves, we can note a little separability only for classes *aeroplane*, *person*, *chair*, *bottle* e *potted plant*. However, the separability is opposed to the expected. The object-to-background distances are smaller than the object-to-object distances. This means that the object points are more similar to background points than to points of other objects of the same class.

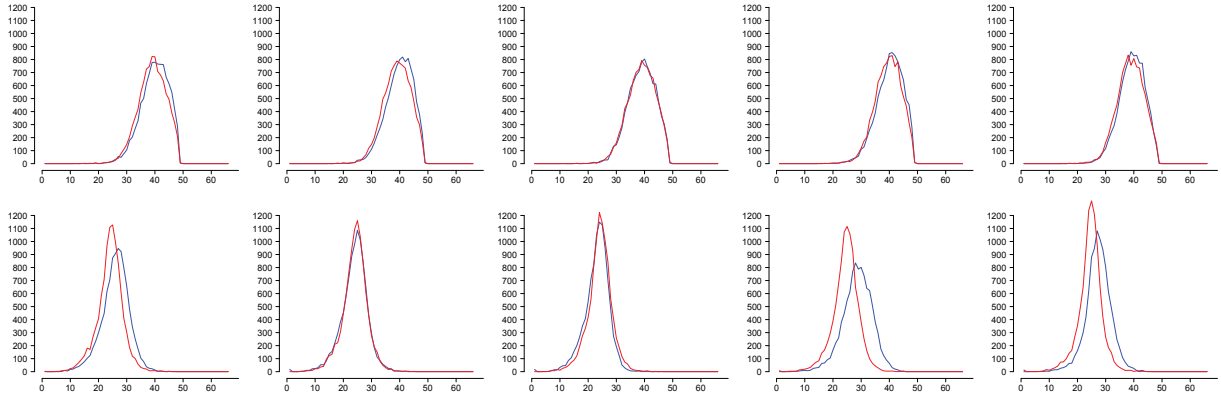
The results go against our hypothesis. Therefore, we can say that there is no semantics in the low-level feature space. This makes it difficult to discriminate classes of objects in this space. The reason is that the local descriptions are too local, based on very small regions, which makes it difficult to give them a semantic meaning. Next, we verify the semantic separability of distance distributions in a higher-level feature space, the BoW space, which is called the mid-level space.

### 5.2.2 Semantic separability in mid-level space

The experiments in Section 5.2.1 show that there is really no semantics in the low-level feature space, making it difficult to semantically separate descriptions in that space.




(a) 5 easiest classes




(b) 5 hardest classes

Figure 5.3: Distance histograms for the (a) 5 easiest and the (b) 5 hardest classes. The top line of each group has the histograms for *average pooling* and the bottom, the histograms for *max pooling* setup. The blue curve corresponds to distances from object to object and the red curve, to distances from object to background. Horizontal axis is the histogram bin and the vertical axis is the frequency of occurrence of the corresponding bin.

However, the use of the mid-level representations (bag of visual words) based on local dictionaries is successful. Therefore, there must be a phenomenon in the mid-level space that creates the semantic separability making it possible to distinguish classes of images and objects.

To evaluate the existence or not of the semantic separability in this feature space, an experimental setup similar to the presented in the previous section was used. We also

used low-level features with dense sampling (6 pixels) and SIFT descriptor. We have generated the dictionary and the BoW representation as follows:

- dictionary:
  - *large-random*: 1 000 visual words randomly chosen from all the dataset points;
  - *large-partially-random*: 1 000 visual words, being 50 random points of each class.
- bag of words:
  - *hard-avg*: performs *hard assignment* and *average pooling*;
  - *soft-max*: performs *soft assignment* ($\sigma$=30) and *max pooling*.

Therefore, we have 4 bags, 2 for each dictionary type.
The histograms of distances were computed as follows:

- select 500 random objects of each class;
- compute the distance between two bags of objects from the same class;
- compute the distance from a bag of an object to a bag of another object from a different class; this last bag is randomly selected from the bags of all classes except the class of the query object.

In the end, for each class, we will have two sets of 500 distances for each bag type (hard-avg or soft-max). We then compute the histograms of those distances.

Our hypothesis is that the distances of objects from the same class are smaller than the distances between objects from different classes. This is intuitive, because we believe that an object is more similar to another object of the same class than to another object of a different class. For example, a motorbike should be more similar to another motorbike than to a chair.

The curves for the 5 easiest and the 5 hardest classes are shown in Figures 5.4 and 5.5, respectively. The histograms for the other 10 classes were similar and are not shown here.

Comparing the first (*large-random-hard-avg*) and second (*large-partially-random-hard-avg*) rows and also the third (*large-random-soft-max*) and fourth (*large-partially-random-soft-max*) rows, it is difficult to see a difference between them. We observe that there is almost no difference in the image representations by using a random dictionary (*large-random*, first and third rows) and by using a partially random dictionary (*large-partially-random*, second and fourth rows).

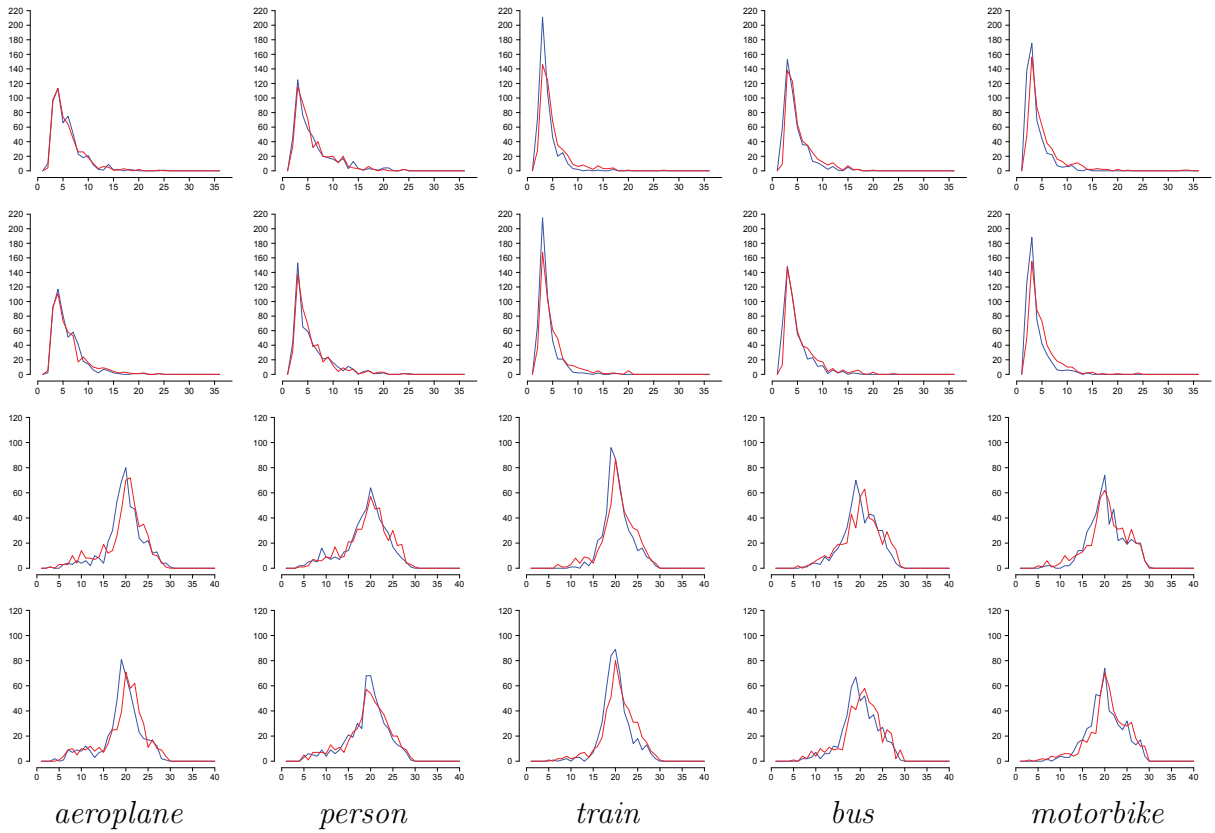



Figure 5.4: Distance histograms for the 5 easiest classes. Each row has the histograms of one type of bag, in the following order: *large-random-hard-avg, large-partially-random-hard-avg, large-random-soft-max, large-partially-random-soft-max*. Each column corresponds to one class. The blue curve refers to distances between objects of the same class and the red curve refers to distances between objects from different classes. Horizontal axis is the histogram bin and the vertical axis is the frequency of occurrence of the corresponding bin.

However, when we compare the *hard-avg* bags to the *soft-max* bags, the differences are clearer. For *hard-avg* bags, the distances between objects of the same class are almost the same as the distances between objects of different classes. Note that the blue and red curves are almost completely overlapped in first and second rows of Figures 5.4 and 5.5.

Considering the *soft-max* bags, we can see a small separation between blue and red curves. This indicates that the distances between objects of the same class are a bit smaller than distances between objects of different classes. We can see this phenomenon in Figures 5.4 and 5.5, in which the blue curves are a little more to the left than the red curves. Although the separability exists in this feature space, it is very small.

It is important to mention that, given the random factor in selecting the objects,

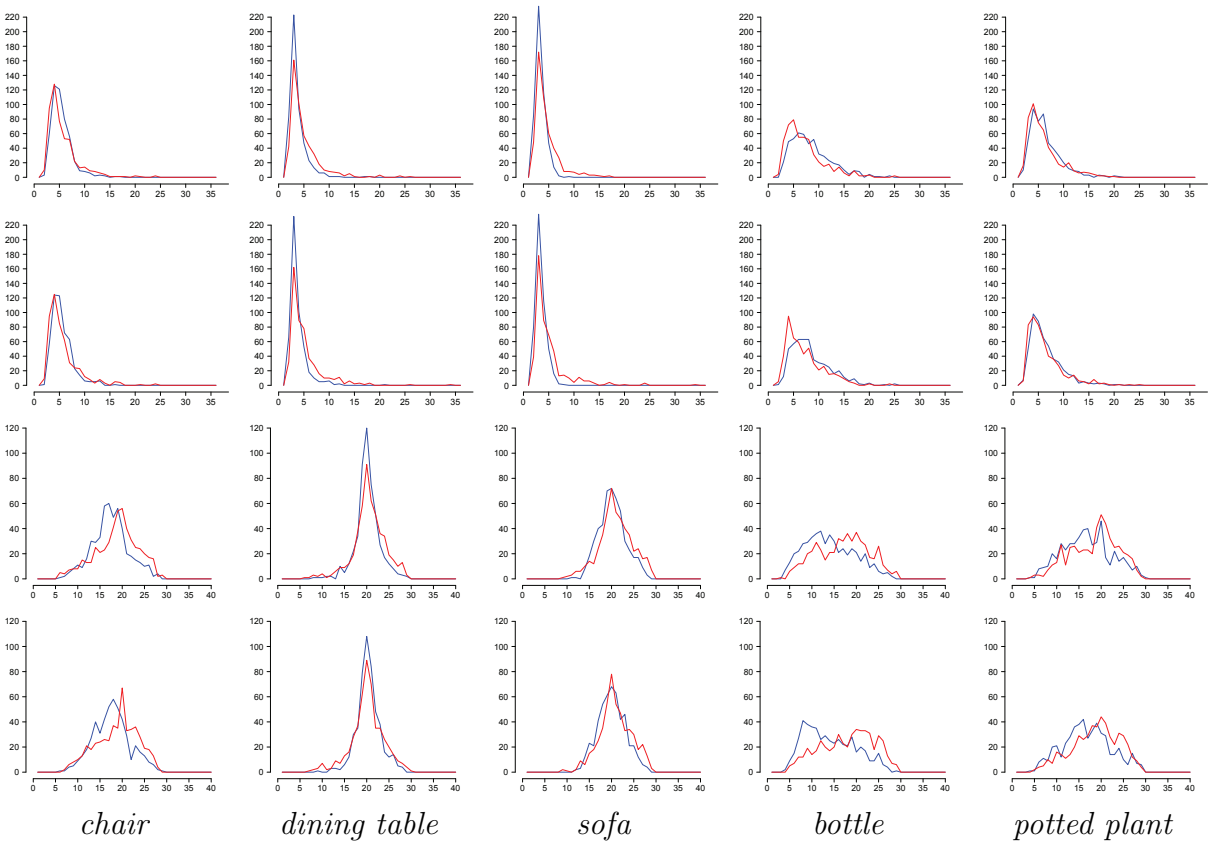



Figure 5.5: Distance histograms for the 5 hardest classes. Each row has the histograms of one type of bag, in the following order: *large-random-hard-avg, large-partially-random-hard-avg, large-random-soft-max, large-partially-random-soft-max*. Each column corresponds to one class. The blue curve refers to distances between objects of the same class and the red curve refers to distances between objects from different classes.

the distances could be affected. Nevertheless, for some classes, the experiments were performed more than once and the curves were almost the same, not affecting the analysis just presented.

### 5.2.3 Conclusions

Based on the experiments in the low-level and in the mid-level feature spaces, we can conclude that:

- there is no semantic information in the low-level feature space, making it challenging to separate classes of objects in this space;

- there is a little separability between distance distributions of different semantic classes in the mid-level space, however, it is too small:
  - completely random or partially random dictionaries do not affect the image representations;
  - bags generated by hard assignment and average pooling carry almost no semantics;
  - bags generated by soft assignment and max pooling embed a little semantics.

The conclusions presented for the mid-level feature space confirm the results presented in literature. Soft assignment combined with max pooling tends to be better than hard assignment and average pooling for classification and retrieval tasks. This is a reflect of the little semantic separability observed in the distance distributions in the soft-max mid-level feature space, in contrast with the lack of separability observed in the hard-avg space.

Additionally, our results indicate that the current mid-level representations are not enough for generating a feature space that encodes semantic information. We address this problem by proposing a new representation in Section 5.3.

## 5.3 Bag-of-Scenes representation

Although the BoW model is successful for visual recognition, its feature space does not have a clear separability when we analyze distance distributions between different semantic classes of objects, as we show in Section 5.2.2.

What we question in this chapter is: if we have a dictionary based on elements (*prototypes*) which contain semantic information, would a better semantic separability be observed in this new feature space?

We have then proposed a new visual representation which goes in the direction to create a *bag-of-prototypes* model, according to which the prototypes are elements containing semantic information. The proposed representation is based on a dictionary of scenes. Scenes are elements with more semantics than local descriptions, therefore, our dictionary comprises more semantic information.

Due to the nature of the dictionary of scenes, its evaluation was performed in a video geocoding scenario. Video geocoding is the task of assigning a geographic location to a video. To create a suitable scenario for that evaluation, we have performed experiments in the Placing Task [65] of the MediaEval 2011 challenge [40].

In Section 5.3.1, we introduce the video geocoding problem and also give background information about related topics. Then, we present our *bag-of-scenes* model in Section 5.3.2 and show experiments in Section 5.3.3.

### 5.3.1 Video geocoding

Video geocoding refers to the task of assigning a geographic location to a video or image. Current solutions for geocoding multimedia material are usually based on textual information [40, 51]. Such a strategy depends on the human intervention to associate textual descriptions with multimedia data. Thus, there is a lack of objectivity and completeness in those descriptions, since the understanding of the visual content of multimedia data may change according to the experience and the perception of each subject. Other issues are related to lexical and geographical problems in recognizing place names [41]. Those limitations open new venues for the investigation of methods that use image/video content in the geocoding process.

Some of the current visual-based approaches to video representation are based on dictionaries of local features, like SIFT or Space-time interest points (STIP) [39]. Despite their good performance, these models are based on elements with very little or no semantic information, like corners and edges.

Our proposed dictionary of scenes provides a higher-level representation for videos. As we explained in Section 5.1, by *higher-level* we mean more intuitive for humans and, therefore, that representation has more semantics considering the human visual perception. Scenes are elements with much more semantic information than local features, specially for geocoding videos using visual content. Our *bag-of-scenes* video representation works like a *place activation vector* because each scene in the dictionary can be seen as a representative picture from a place. In this way, each component of the feature vector has semantics and, hence, it can be directly related to a specific place of interest.

Next, we detail the environment of evaluation used in this chapter, which is the Placing Task at MediaEval 2011. We also present the approaches used by the other teams which participated in the task.

#### Placing task at MediaEval

Placing Task requires participants to automatically assign latitude and longitude coordinates to each of the provided test videos. The most recent approaches to video geocoding were submitted to the Placing Task of MediaEval 2010 and 2011. They can be basically divided into methods based on textual information and methods based on visual information. Our interest in this chapter is to compare with the methods based only on visual information, which were more frequent in the Placing Task of 2011 than 2010.

In the Placing Task of 2010, just one team reported results using only visual content [37]. Their approach was based on predicting keyframe locations and using a voting scheme to assign the final video location. They had first divided the world into regions using k-means clustering over the geographical information of the training data. Then,

they trained a SVM classifier based on the visual features of the development set. Each keyframe was then assigned to a location using the SVM model.

In 2011, four groups submitted results for a run in which only visual features could be used to predict the location of the test videos. Most of them considered visual features as a backup predicting approach to the cases in which no tags or textual description is associated with a test video.

Using an algorithm to compare video sequences [2], Li et al. [45] (UNICAMP team), concentrated only on visual features of a video to predict its location. None of the photos or keyframes were used in this case. Videos were compared by taking into account their motion features. Each video in the test set was compared with those in the development set. Then, for each test video, an ordered list of similar videos from the development set was produced and the geographic information of the most similar video was assigned to the test video.

Choi et al. [16] (ICSI team) proposed an approach based on the visual similarity between query video and items in development set, either video keyframes or Flickr photos. They extracted GIST features of frames and photos and ran an 1-nearest-neighbor search to match each test video against the whole development set. The most similar video, according to the Euclidean distance, was selected and its latitude/longitude was assigned to the query video.

Hauff and Houben [28] (WISTUD team) divided the world globe in cells of variable size (small for dense data area and larger if sparse data) and assigned items from development set to their respective cells. For the visual approach, only 10% of the set was used. Matches between the query video and the videos of the training set work as follows: first, the cell with the highest probability to contain a test video is identified ($C_{max}$). Then, they identify inside $C_{max}$ the closest match to the test video and assign its location. A Naïve-Bayes nearest neighbor approach with all visual features was used.

The strategy of van Laere et al. [78] (UGENT team) was based on comparing photos from the development set to keyframes of query videos, both represented by Color and Edge Directivity Descriptor (CEDD). Once the most similar photo ($p$) to the query video ($v$) is found, the location of $p$ is transferred to $v$.

### 5.3.2 Bag of Scenes

In this section, we describe a novel model for video representation that is based on a dictionary of scenes[3]. In the scenario of video geocoding, the motivation for using this approach is that video frames are like pictures from places and these pictures have important information regarding the place location. If we have a dictionary of representative

[3]The term *scene* refers to images (photos), differently of its designation in video segmentation tasks.

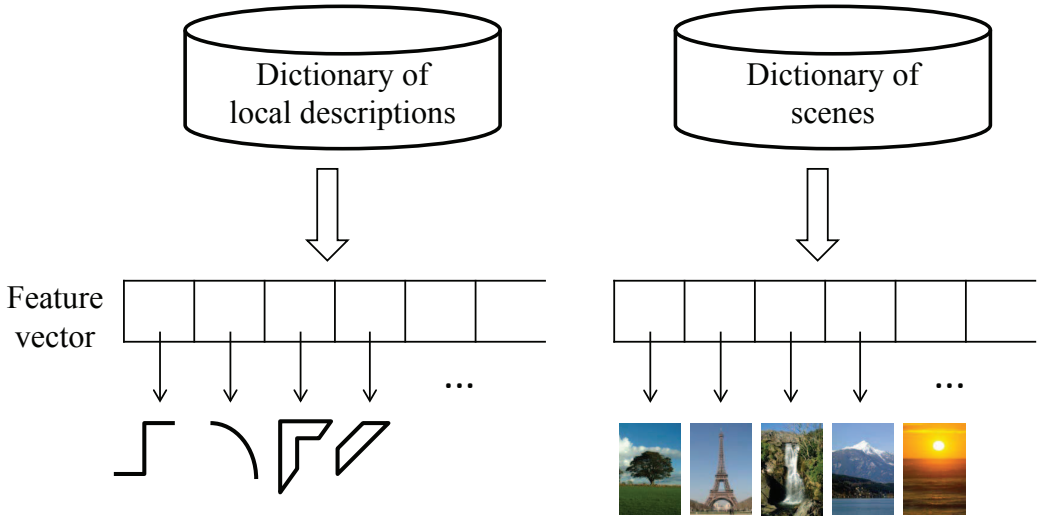



Figure 5.6: Comparison between the proposed dictionary of scenes to a dictionary based on local descriptions. We can notice that the representation based on the local dictionary relies on elements without clear semantics, like small corners and edges, while, the representation based on the dictionary of scenes carries more semantics. In addition, the feature space for the dictionary of scenes has semantics in each dimension independently.

pictures from different places, we can describe video frames by considering their similarities to the representative pictures. Therefore, if a video has frames similar to photos taken in certain locations, we can infer that it is from such a location, facilitating the geocoding task. Given an input video, we create a vector of activations of video frames to each of the scenes in the dictionary: the *bag-of-scenes* representation.

One important advantage of the representation based on the dictionary of scenes is that it relies on elements that have more semantics according to the human visual perception. Traditional dictionaries of local low-level descriptions, like SIFT or STIP, are composed of visual words based on very punctual elements, like small corners and edges, which carry no semantic information, as we have shown in Section 5.2.1. The dictionary of scenes is composed of pictures and they have more semantic information than corners and edges. Therefore, our final video representation is an activation vector to "higher-level" elements, resulting in a representation space where each vector dimension has semantics by itself. Figure 5.6 shows the differences between those types of dictionaries.

To generate a dictionary of scenes, we first need to compute a representation for each scene. Given a set of scenes which may come from frames of training set videos or from an arbitrary collection of images, each scene can be represented by a certain type of

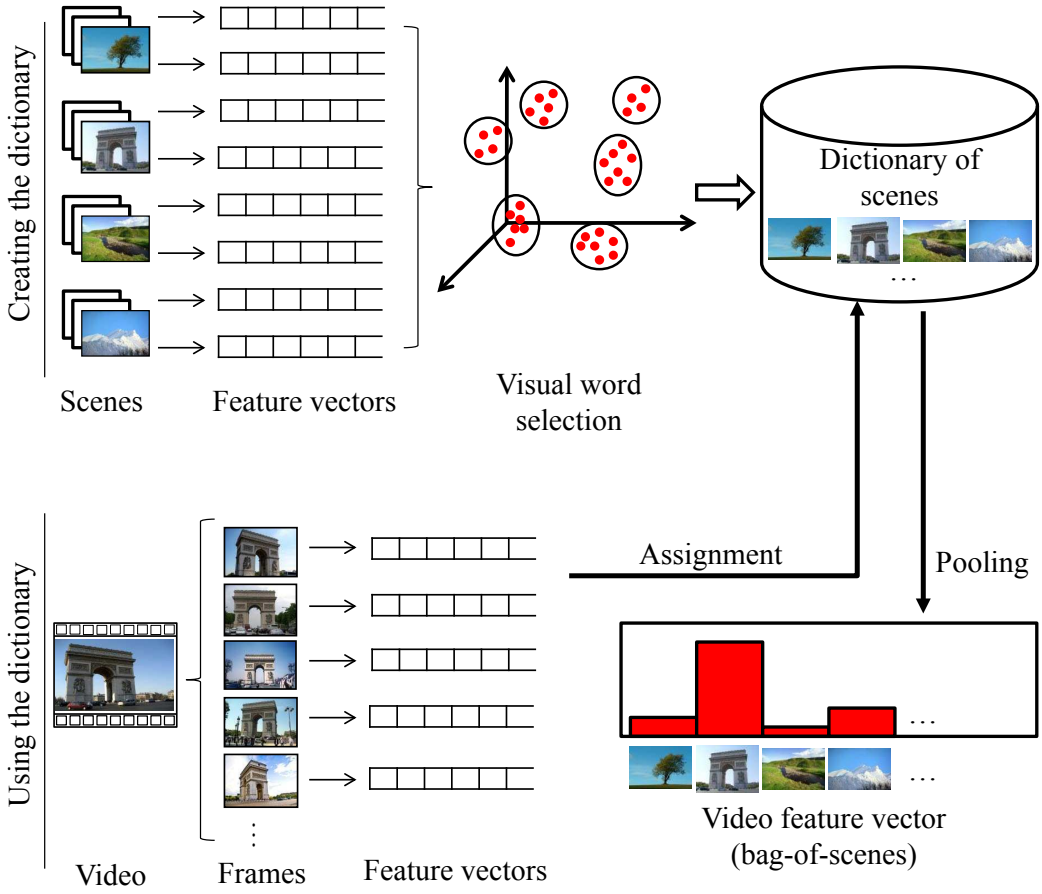



Figure 5.7: The schema for generating and using a dictionary of scenes. The dictionary is created based on a given collection of scenes, which may come from an image dataset or from video frames. After representing each image with any kind of feature vector, some of them are selected to compose the dictionary. Given an input video to be represented, its frames are assigned to one or more of the scenes in the dictionary. A pooling strategy is then applied to generate the video feature vector (*bag of scenes*).

low-level feature, like color histograms or bag of visual words, for example. Figure 5.7 illustrates the steps for generating a dictionary of scenes and the steps to represent a video using the dictionary. The visual dictionary is created by selecting feature vectors of the scenes according to some criteria. One can cluster the feature space in the same fashion it is performed for SIFT dictionaries [70, 75, 77, 79]. Other possibilities rely on a random selection of scenes or even on a manual selection of the most important scenes for the target application. In our application scenario, a guided selection of representative scenes from places of interest may be more promising. For example, if we have videos of a specific city and we want to differentiate videos recorded in different locations of this city,

we can select scenes from those specific locations to compose the dictionary. Algorithm 1 presents the steps to generate a dictionary of scenes. It is also related to the first part of Figure 5.7.

**Algorithm 1:** Algorithm to create a dictionary of scenes.

**Input**: Dataset $D$ of images/frames to be used to create the dictionary; image descriptor $d$
**Output**: Dictionary $W$ with $k$ scenes (visual words)

**foreach** $e$ *in* $D$ **do**
    Compute a feature vector $d_e$ for $e$ using descriptor $d$;
Quantize the feature space of $d$ into $k$ regions ; `/* or supervise the dictionary creation, for instance */`

It is important to highlight that any technique can be used for frame extraction from videos, like sampling at fixed-time intervals or by employing summarization methods [3, 4, 6].

Another important aspect of the description based on dictionaries, and also valid for the dictionary of scenes, is that the feature vectors of each scene and the feature vectors of each visual word need to be of the same nature. In our case, a visual word is also a scene. For example, if we generate the dictionary by representing the scenes with a 64-bin color histogram, each video frame considered in the dictionary also needs to have a 64-bin color histogram representation.

Once the dictionary is generated, we are able to create the video representation. Coding approaches are used to describe the feature vector of each frame according to the dictionary. The *hard* and *soft* assignment methods, popularly used with SIFT dictionaries [49, 64, 77] are suitable for this step. To generate the final *bag-of-scenes* representation for a video, we can employ pooling strategies, like the popular *average* and *max* pooling [11]. The second part of Figure 5.7 and Algorithm 2 show the steps for computing the bag-of-scenes representation.

**Algorithm 2:** Algorithm to compute the bag-of-scenes vector.

**Input**: Dictionary of scenes $W$; video $v$; image descriptor $d$
**Output**: Bag-of-scenes vector for $v$

Splits $v$ in $f$ frames;
**foreach** $f$ *of* $v$ **do**
    Compute a feature vector $d_f$ for $f$ using descriptor $d$ ; `/* the same` $d$ `used in Algorithm 1 */`
Compute $\alpha_v$: coding of $d_f$ to $W$;
Pooling over $\alpha_v$;

The bag-of-scenes representation has some interesting properties. As the visual words are scenes, which tend to carry semantic information according to the human visual per-

ception, the activation vector has one position for each concept, making it simple to analyze the presence or absence of each concept into a video. This is a step forward to reduce the semantic gap and create a representation that is more intuitive for humans [80]. In the video geocoding scenario, the feature vector is a *place activation vector*, because each visual word is a picture of some specific place. Mathematically speaking, the dictionary of scenes creates a vector space where each dimension represents a specific semantic concept. It is important to realize that, despite our dictionary of scenes is being originally proposed and validated for video geocoding, it can be applied to many other applications, like video categorization or video retrieval, for instance.

### 5.3.3 Experiments

The goal of the experiments is to evaluate the dictionary of scenes for video geocoding. To create a suitable scenario, we have worked under all the specifications of the Placing Task of MediaEval 2011 [65].

We have divided our experiments into two phases. The first phase is based on a very simple way to create the dictionary of scenes: selecting random scenes from the dataset. The second phase performs a guided selection of scenes, which is based on the results of the random dictionary.

#### Datasets and evaluation criteria

Participants in the Placing Task at MedialEval 2011 were allowed to use image/video metadata, audio and visual features, as well as external resources, depending on the run submitted. The organizer of this task released two sets of data [65]. The first set is meant to the development and training of algorithms, thus called development set[4]. It is comprised of 10 216 geocoded videos and 3 185 258 CC-licensed geocoded photos from Flickr with corresponding metadata, such as title, tags, and descriptions provided by the owner of the resource, comments of her/his friends, users' contact lists, and other uploaded resources on Flickr. Videos come with their extracted keyframes and both keyframes and photos have a set of pre-extracted low-level visual features. The photos were uniformly sampled from all parts of the world.

The second set, called test data, is composed of 5 347 videos, their keyframes with extracted visual features and related metadata (without geographic location).

Keyframes were extracted at each 4 second intervals from videos and saved as individual JPEG-format images. The following visual feature descriptors for keyframes and photos were provided: Color and Edge Directivity Descriptor (CEDD), Gabor Texture,

[4]The designation *training set* is more common for this kind of set. However, to keep correspondence to the name used in the Placing Task, we have used its original designation, i.e., *development set*.

Fuzzy Color and Texture Histogram (FCTH), Color Histogram, Scalable Color, Auto Color Correlogram, Tamura Texture, Edge Histogram, and Color Layout.

Participants in Placing Task were required to submit at least one run that uses only audio/visual features. The result evaluation was based on the distance to the ground truth geographic coordinate point, in a series of widening circles of radius (in km): 1, 10, 20, 50, 100, 200, 500, 1 000, 2 000, 5 000, 10 000. Thus, an estimated location is counted as correct at a particular circle size, which can be seen as quality or precision level, if it lies within a given circle radius.

More details about the Placing Task at MediaEval 2011 are given at the working notes of the organizers [65].

### Experiments with random dictionaries

The experiments with random dictionaries are good to illustrate the potential of the bag-of-scenes approach. If results are good even with this simple way to select the scenes to compose the dictionaries, we are able to show that the bag-of-scenes approach is promising for video geocoding.

Next, we explain how we have created the dictionaries and the video representations, as well as we show the results in the development and test sets.

**Experimental setup** Our experiments with the random dictionaries are divided into two stages. The first stage comprises the parameter adjustments using the development set. The second stage employs the best dictionary configurations for representing and geocoding videos from the test set. In each of the stages, we have used two sources for the scenes to generate the dictionary: video frames from the development set and Flickr photos. To easily distinguish between them, in the remainder of this section, we call the former as *dictionary of frames* and the latter as *dictionary of scenes*.

To represent each video frame, we have used many of the low-level global descriptions provided with the datasets aiming at discovering which of them are better for the placing task. After that, we have created the dictionary by randomly selecting their feature vectors in the feature space of global descriptions. A first motivation to use the random dictionaries is related to their similar quality to dictionaries computed by k-means in high-dimensional spaces [35, 79]. As we have already pointed throughout this thesis, for SIFT-based dictionaries, a random selection of visual words has similar performance to clustering techniques, due to the curse of the dimensionality [79]. In the dictionary of scenes, the dimensionality is still an important issue.

To represent videos by a given dictionary of scenes, we have employed some of the state-of-the-art assignment and pooling techniques of the image representation community [11,

| Dictionary | % 1km | % 10km | % 100km |
|---|---|---|---|
| Frames | 14.59 | 15.69 | 17.23 |
| Scenes | 13.60 | 14.62 | 16.15 |

Table 5.1: Experiment results showing small performance difference between dictionary of *frames* and dictionary of *scenes* in the development set. The values are the percentage of videos from the development set that were correctly geocoded in the radii 1km, 10km, and 100km.

75, 77]. Hard and soft assignment as well as average and max pooling were used. Details of these techniques are presented in Chapter 2.

After computing the bag-of-scenes representation for each video, our strategy to assign a geographic location to a given video is based only on the visual information. We have computed the Euclidean distance from a query video to all the remaining videos in the development set and estimated its latitude/longitude by assigning those from the nearest video. The evaluation measures were computed using the distance circles to the correct coordinate point, as explained previously. Our results were not submitted to the Placing Task at MediaEval 2011, however, comparisons with other approaches were possible by running the official evaluation program, which was released for participant groups after the event.

**Results on the development set** The experiments in the development set combine different parameters for creating and using the dictionary. To evaluate the parameters, we have used all the videos from the development set as queries and, when estimating their latitude/longitude by assigning the location of the nearest video, we considered that the query video was not part of the development set. Our analysis using the *dictionary of frames* has shown that a good configuration for the visual dictionary uses CEDD descriptor, soft assignment ($\sigma$=3), and max pooling. Although other $\sigma$ values were also tested, $\sigma$=3 was selected because it makes a frame to be assigned to a fair number of visual words, considering the CEDD feature space. There was little impact when changing the dictionary size. A meaningful difference occurred when using a very small or a very large dictionary (30 or 50 000 visual words), but they were worse than dictionaries of sizes 50, 500, and 5 000. The experiments with the *dictionary of scenes* in the development set also shows that CEDD descriptor, soft assignment ($\sigma$=3), and max pooling achieve the best results. We have tried dictionaries up to 50 000 visual words, but the results were better with smaller dictionaries.

Table 5.1 presents those results and compares the two types of dictionary. We can note that there is a little difference between the *dictionary of frames* and the *dictionary of scenes*. This is an interesting result, because frames are clearly elements that came from

the same dataset, while the scenes came from a completely different source. This shows that we can create a good dictionary even with a kind of information that comes from a completely unrelated source. We have also noticed the effect of using different sources of information to create SIFT-based dictionaries in Chapter 3. Therefore, the phenomenon seems to happen also in different feature spaces. In the machine learning community, similar behavior in the classification level is known as *transfer learning* [56].

**Results on the test set** According to the experimental results on the development set, we have used CEDD descriptor, soft assignment ($\sigma$=3), and max pooling to run the experiments on the test set. We have tested 3 different dictionary sizes: 50, 500, and 5 000. The dictionaries were created using frames from the development set, in the case of the *dictionary of frames*, and using Flickr images for the *dictionary of scenes*.

The results for the *dictionary of frames* and the *dictionary of scenes* in the test set were very similar, as well as in the development set. We could also note that the variation in the dictionary sizes has little impact in the results. One possible reason is that the random selection of visual words (both frames or scenes) may have taken many images with little information about place location. Hence, the small portion of representative visual words helped the geocoding of only some of the test videos.

To evaluate the quality of the representation when using the *dictionary of scenes*, we have verified the visual words activated by the videos that we geocoded correctly. The most activated scenes by the best geocoded videos are shown in Table 5.2. Notice that, despite those videos were geocoded really close to the correct location, the scenes activated by them are not necessarily representative from the location. It is important to note that, the scenes themselves do not need to be specifically from a location. However, videos that are specifically from a certain location should activate the same scenes. What might have happened in the case of the best geocoded test videos is that there are videos in the development set which are from the same location and have activated the same scenes from the dictionary.

Table 5.3 compares the results obtained by the proposed method with those reported by four participants of the MediaEval 2011 Placing Task: UGENT [78], UNICAMP [45], ICSI [16], and WISTUD [28]. They are the methods based only on the visual information. We can see that our approach performs better than most of the compared methods, except for that of the UNICAMP team [45]. This method is based on motion information and, hence, it does not consider visual properties of video frames in an independent manner. Such a method has geocoded correctly videos that our approach geocoded wrongly and vice versa. Recent studies show that both methods are little correlated [47].

Although the proposed method is not superior to all approaches to video geocoding, the results obtained show the potential of the idea. Observe that, by generating a video

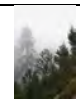
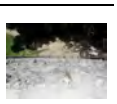
| Video | Most activated scenes 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
|  |  |  |  |  |  |  |  |  |  |
| 0.004 | 12.5 | 6.4 | 6.4 | 6.0 | 5.8 | 5.8 | 5.7 | 4.3 | 4.3 | 3.6 |
|  |  |  |  |  |  |  |  |  |  |
| 0.012 | 1.7 | 1.7 | 1.6 | 1.3 | 1.3 | 1.2 | 1.1 | 1.1 | 1.0 | 1.0 |
|  |  |  |  |  |  |  |  |  |  |
| 0.516 | 4.1 | 3.4 | 2.5 | 1.8 | 1.7 | 1.7 | 1.6 | 1.5 | 1.5 | 1.5 |
|  |  |  |  |  |  |  |  |  |  |
| 0.603 | 0.9 | 0.9 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.7 | 0.7 |
|  |  |  |  |  |  |  |  |  |  |
| 0.861 | 1.3 | 1.2 | 1.1 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 0.9 | 0.8 |

Table 5.2: Ten most activated visual words by some of the best geocoded videos when using the dictionary of 5 000 scenes. The value below the video thumbnail is its distance to the correct location, while the value below each visual word is its activation value, in percentage, by the corresponding video.

representation based only on pictures, which come from a completely different source in the case of the *dictionary of scenes*, it is still enough to provide a good representation for video geocoding. Despite our very simple way to generate the visual dictionary, which has taken photos at random, the results are comparable to (or even better than) some of the methods presented in Table 5.3.

Our random selection of pictures to compose the dictionary may take pictures with very little information regarding the place location and, thus, being not informative for the placing task. Notice that some of those non-informative pictures were activated even in our best geocoded videos, as shown in Table 5.2. A smarter selection of scenes may be able to create more informative dictionaries and, hence, improve the video representation for geocoding. Therefore, in the following section we present another strategy to create dictionaries, which is guided by the video locations.

### Experiments with guided dictionaries

Our results presented in the previous experiments with the random dictionary have shown that, despite the good results considering the geocoding task, many of the scenes that compose the dictionary are not really meaningful. Therefore, we have tried a new scheme

| Radius | Other teams | | | | Bag of Scenes | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Dict. of Frames | | | Dict. of Scenes | | |
| (km) | UGENT [78] | UNICAMP [45] | ICSI [16] | WISTUD [28] | 50 | 500 | 5 000 | 50 | 500 | 5 000 |
| 1 | 2 | 11 | 5 | 0 | 9 | 7 | 7 | 11 | 9 | 6 |
| 10 | 6 | 60 | 16 | 5 | 35 | 36 | 37 | 35 | 40 | 32 |
| 100 | 49 | 145 | 67 | - | 109 | 90 | 96 | 100 | 105 | 95 |
| 1 000 | 624 | 650 | 598 | 583 | 649 | 624 | 614 | 611 | 646 | 610 |
| 10 000 | 4 332 | 4 248 | 4 234 | - | 4 312 | 4 299 | 4 308 | 4 257 | 4 316 | 4 353 |

Table 5.3: Comparison of the results obtained by the proposed approach with those reported by four participants of the MediaEval 2011 Placing Task. The values are the number of test videos correctly geocoded at different distances from the correct video location.

to generate the dictionary of scenes which is not random anymore. The idea is to build a dictionary which is composed of more meaningful scenes for the placing task.

Next, we present how we have created the new guided dictionaries and also the results obtained.

**Experimental setup** We have made a guided selection of scenes based on the worst geocoded videos according to the previous random dictionary. The reason is that, possibly, those videos had no pictures representing their location in the dictionary, therefore they were incorrectly geocoded. It is important to note that, despite this assumption, we know that the dataset has many indoor videos and other videos which have very little visual information about the place where they were recorded (see examples in Figure 5.8). Those videos will be always hard to geocode even if the dictionary has scenes from their locations.

The scenes (photos) were selected from the Flickr dataset provided with the task dataset. To create the new dictionaries, we have first computed a list containing all the development videos in ascending order of geocoding results, that is, from the worst to the best geocoded. This list is based on the results in the development set when using the random dictionary of 5 000 scenes. We have then selected from the beginning of this list, videos that have at least 10 photos that are at most 18km far from its location (0.1 difference in latitude or longitude). If a video does not have the 10 photos, we skip it and use the next one.

This selection scheme finishes when 100 videos are selected. We have then performed it several times, taking at each time, the next 100 worst geocoded videos. To avoid taking videos of similar locations, we have considered another selection restriction: the new selected video should be at least 36km far from any of the videos that were already selected in previous steps. This restriction also avoids problems considering the 18km restriction when selecting photos close to the videos. In the end of this process, we have

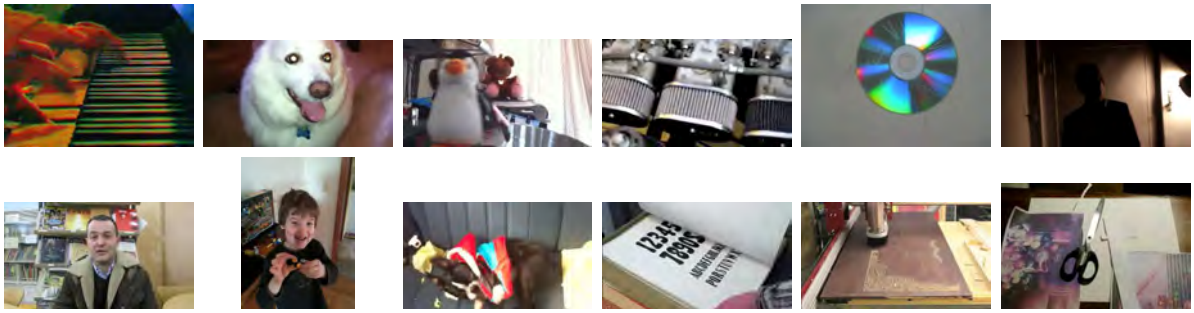


Figure 5.8: Examples of videos with very little visual information about the place where they were recorded.

several sets of 100 videos. Due to the distance restrictions for the photos (18km), we could not select 1 000 videos. Therefore, we have worked over 9 sets of 100 videos.

The next step was concerned with the selection of Flickr photos. We have fixed a number of 10 photos per video, which also means 10 pictures from each location. Hence, each of our 9 groups of videos generated a dictionary of 1 000 scenes. The selection of photos close to the video locations respects the 18km restriction.

For the experiments presented in the following sections, we have concatenated the dictionaries incrementally. This means that we have now 9 new dictionaries of 1 000, 2 000, 3 000, and so on, until 9 000 scenes. The dictionary of 2 000 scenes is composed of the 1 000 scenes from the first group of videos plus the 1 000 scenes from the second group of videos. For each new group of videos added, the dictionary increases in 1 000 visual words. Therefore, for each new dictionary we expect to increase its quality, because we have pictures representing more places. Algorithm 3 summarizes the steps used to create the guided dictionaries.

It is important to highlight that, we are including the pictures in the dictionary considering only their geographic information. That does not guarantee that the pictures really represent places visually and then, we still can have scenes that are not representative for the geocoding task. However, our hypothesis is that this process generates better dictionaries than the completely random dictionary used previously.

The experimental setup for these experiments follows the best configurations observed for the random dictionaries, representing each video based on CEDD descriptor and using soft assignment ($\sigma$=3) and max pooling. We use the same geocoding strategy, which assigns to the test video the latitude/longitude information from its most similar video in the development set.

We are comparing the guided dictionaries only with the random dictionary based on scenes, not the one based on video frames.

**Algorithm 3:** Algorithm to create the *guided* dictionaries of scenes.

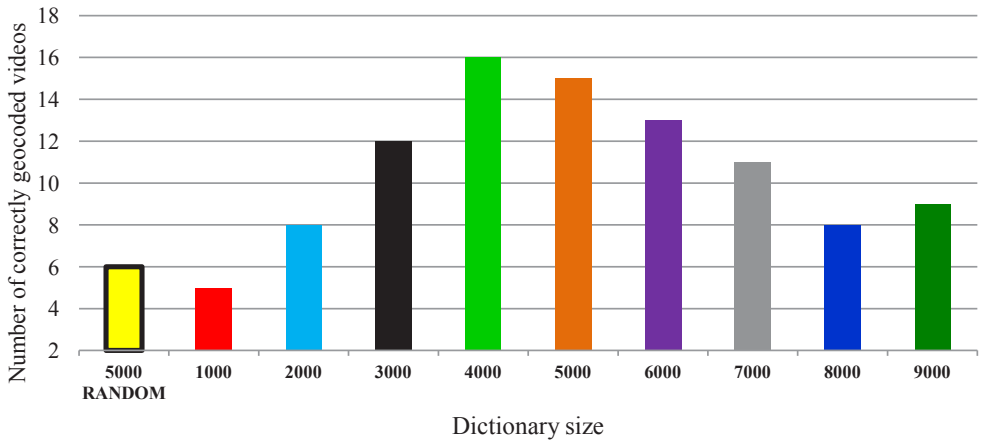
**Input**: dataset $D$ of images to be used to create the dictionary; list $L$ of development videos sorted by ascending order of geocoding results (from the worst to the best geocoded video when using the random dictionary of 5 000 scenes)

**Output**: Dictionaries $W_i$

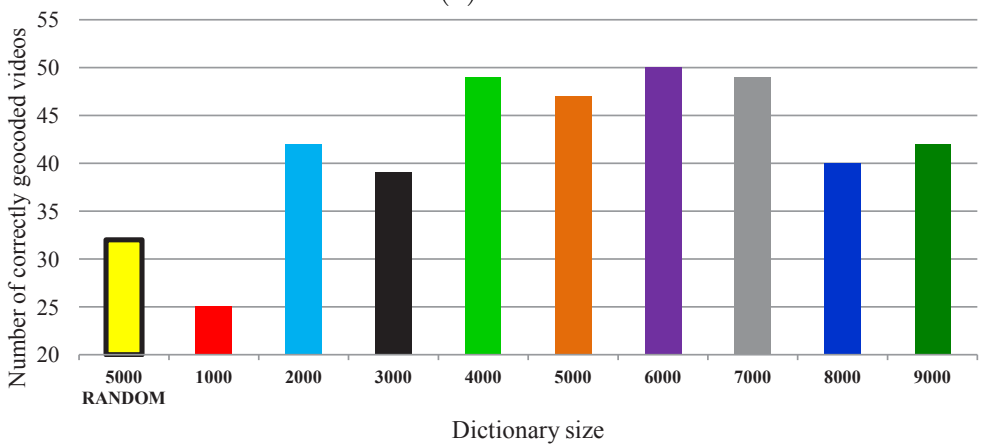
```
foreach v in L do
    i = 1;
    Select n scenes in D which are geographically closer than 18km to v;
    if n ≥ 10 ;                /* video has at least 10 scenes close to its location */
    then
        if v is 36km far to all other videos in V (list of selected videos) then
            Add v to V;
            Add the top 10 scenes from n in W_i;
        else
            Skip v ;        /* skip the video if it is too close to the other selected
            videos */
        if size of V = 100 × i ;                        /* create blocks of 100 videos */
        then
            i = i + 1 ;        /* W_i is ready and has i × 1000 scenes; move to the next
            dictionary */
            W_i = W_{i-1} ;                /* concatenate to the previous dictionaries */
    else
        Skip v ;        /* skip the video if it is not close to at least 10 scenes */
```

**Results** Our first analysis considers the global performance of each dictionary in the placing task, which means that we are reporting the results considering the whole test set. Additionally, as our criterion to select the scenes for the dictionary is very precise (18km away from the video location), we are focusing our analysis on the widening circles with radii closer than that, i.e., 1km and 10km.

Figure 5.9 shows that the guided dictionaries are better than the random dictionary. Except for the first dictionary (1 000 scenes), all the other dictionaries increase the quality of the video geocoding. The dictionary of 1 000 was not good, not because of its size, but mainly due to the fact that it contains only scenes from 100 specific places. Therefore, frames that are not from those places had no other places to be assigned to. As the dictionaries were getting more variable in terms of places, their quality increased, as we can see in the larger number of videos correctly geocoded, both for 1km and 10km. However, there was a saturation in the dictionary quality. For the radius of 1km, this saturation occurs from the dictionary of 5 000 scenes on. For the radius of 10km, it occurs from the dictionary of 7 000 scenes on. One possible reason for observing the saturation in different dictionary sizes when analyzing different radii, is the curse of dimensionality. As our geocoding strategy is based on a 1-nearest-neighbor (1-nn) approach, the dimensionality




(a) 1km




(b) 10km

Figure 5.9: Comparing the overall results of all the guided dictionaries to the random dictionary for widening circles of (a) 1km and (b) 10km. We can see that except for the 1 000 dictionary, all the other dictionaries outperform the random dictionary. We can also note that there is a saturation in performance at a certain dictionary level.

effect may appear first when we analyze using the 1km radius than when using the 10km radius.

Another reason might be the higher number of non-representative scenes in the larger dictionaries. As the scenes are added to the dictionary considering only their geographic location, we might include scenes with little visual information about the place location.

**Analysis on the quality of the representation** Although we have noticed a great improvement for the guided dictionaries, the previous analysis is very dependent on the geocoding scheme employed. We have used a ranked list (1-nn) approach to geocode a given test video, as explained previously. Therefore, the analysis just performed gives

a good insight about the improvement in the dictionary quality but it is made over the geocoded scheme itself and not directly on the video representation.

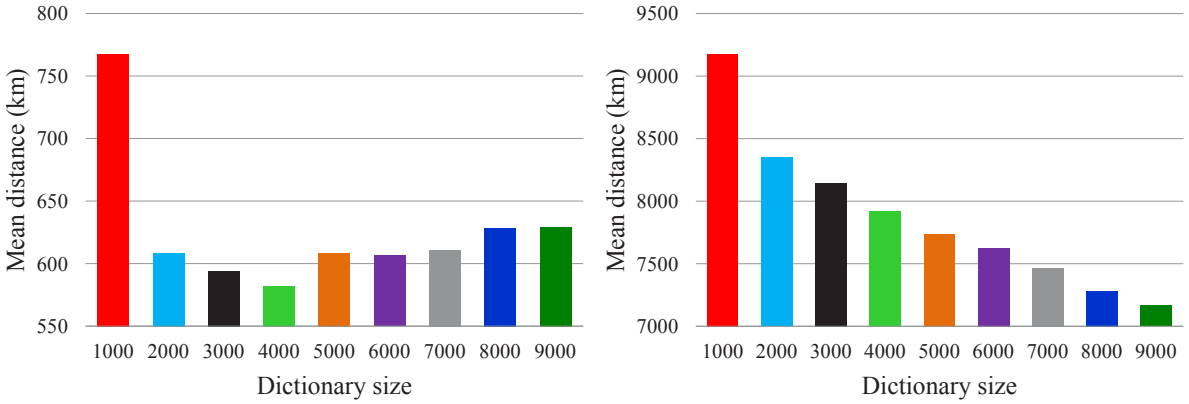
To investigate if the bag-of-scenes vector is representative for the video location, we have also analyzed if the most activated scenes in the dictionary are from places close to the video. We can expect that, as the scenes from a certain video location are inserted into the dictionary, the most activated scenes become closer to the video location.

For that, we have computed the geographic distances from the video location to each of its 50 most activated scenes. We have then computed, for each video, the minimum, the average, and the maximum distances among the 50 distances. It is important to differentiate in these experiments the issue of *most activated scenes* and *closest scenes*. Here, we are analyzing the distances of the 50 most activated scenes, however, we are not taking into account their ranking order. Therefore, when we analyze, for example, the minimum distance among the 50 most activated scenes, we are not necessarily analyzing the most activated scene (the closest scene to the video location is not necessarily the most activated one).

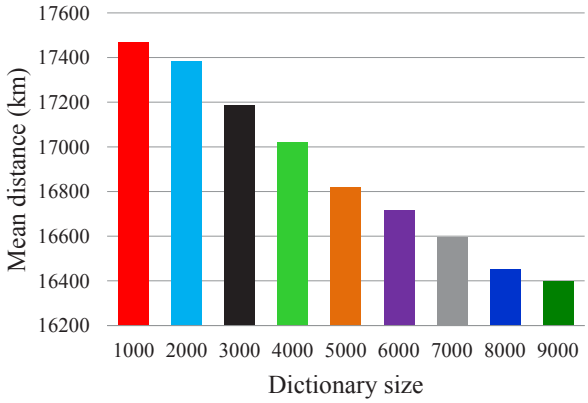
Figure 5.10 shows an analysis looking at a summary (average) of the minimum, average and maximum distances for all of the test videos in the dataset considering their 50 most activated scenes in each of the 9 dictionaries. We can see that the quality of the dictionary increases with its size. Considering the closest scene activated among the 50 most activated ones, it is getting closer until the dictionary of 4 000 scenes (see Figure 5.10(a)). This result agrees with the results for 1km presented in the previous section. The dictionary of 4 000 scenes was the best globally. However, the most interesting result is that the average and the maximum distances of the most activated scenes keep decreasing as the dictionary grows (see Figures 5.10(b) and (c)). This means that the most activated scenes are always getting closer to the video location, giving an indication of the improvement in the dictionary quality.

Additionally, we have performed a more precise analysis considering the distances computed for each of the test videos, without summarizing them by the average as we have just presented. We have computed histograms of distances. The idea is that better dictionaries will present more small distances than bad dictionaries, which means that better dictionaries will make the videos to activate scenes closer to their locations. Therefore, the curve of a histogram of distances would be more to the left (more small distances) for better dictionaries, while the histogram for bad dictionaries would be more to the right (more large distances). To compute those histograms, we first had to select a quantization scheme for the distances. We have used quantizations of 1km, 10km, and 100km.

Analyzing the *minimum* distances in Figure 5.11, the improvement in quality is only clear for the quantization of 100km and at bin 2. For quantizations of 1km and 10km, the best dictionaries are the ones with 1 000 and 2 000 scenes, because they present more




(a) min (b) avg




(c) max

Figure 5.10: Comparing the summary (average) of the (a) minimum, (b) average, and (c) maximum distances from a video and its 50 most activated scenes, considering the videos in the test set. In (b) and (c), the most activated scenes are coming closer to the video location as the dictionary grows. In (a), this also happens but only until the dictionary of 4 000 scenes.

small distance values than the other dictionaries. For quantization of 1km, those two above mentioned dictionaries present more scenes at a distance of up to 12km. Due to our restriction when selecting the scenes to compose the dictionary (18km distance), we could expect that, as the dictionary grows, we would have a larger number of activated scenes which are located in less than 18km distance from the video location. However, this was not observed in Figure 5.11. There are some reasons for that. First, as we have included more scenes in the dictionary considering only their geographic location, we might have included scenes with little visual information about the places. Another possible reason is the semantic gap. As the dictionary grows, there is a greater chance of

having a visually similar scene that is not from the place of the video. A third reason could be the lack of precision of the low-level description. We have used the CEDD descriptor, which is a global texture descriptor. If we have a more precise representation, based on local information, like the bags of visual words presented throughout this thesis, we could probably obtain some new matches between common monuments and places among the scenes. Another reason could be the curse of dimensionality as the smallest dictionaries (1 000 and 2 000 scenes) were less affected.

Analyzing the *average* distances in Figure 5.12, we can see more clearly the improvement in quality as the dictionary grows. As more places are comprised in the dictionary, the average distances tend to reduce, which means: the whole group of the 50 most activated scenes is becoming closer to the correct video location as the dictionary grows.

To summarize the results presented by the histograms of distances, we can point that the dictionary quality improves as more scenes (places) are added to them. However, we could not observe an improvement looking at the closest scene activated by each video. There was an improvement only when we analyze the whole set of the 50 most activated scenes, using the average distance among them. Therefore, we can say that as the dictionary grows, the visual variability of scenes increase, augmenting the chance to have a scene close to the video location with similar visual appearance.

## 5.4 Discussion

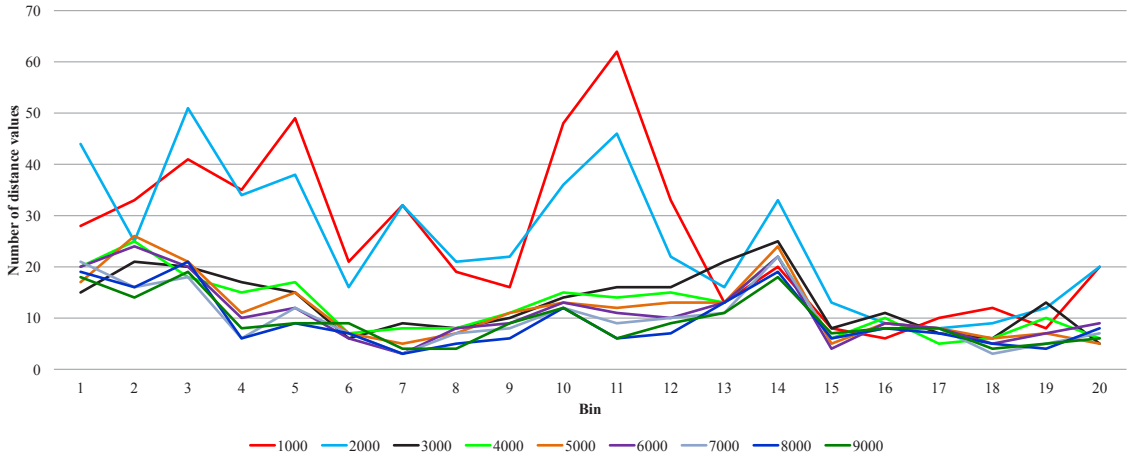
In this chapter, we presented studies over the semantic information comprised by visual dictionaries.

We have first shown that there is no semantic information in the low-level feature space, which is the space quantized for the dictionary generation. The lack of semantics makes it challenging to distinguish samples according to their semantics in that space.

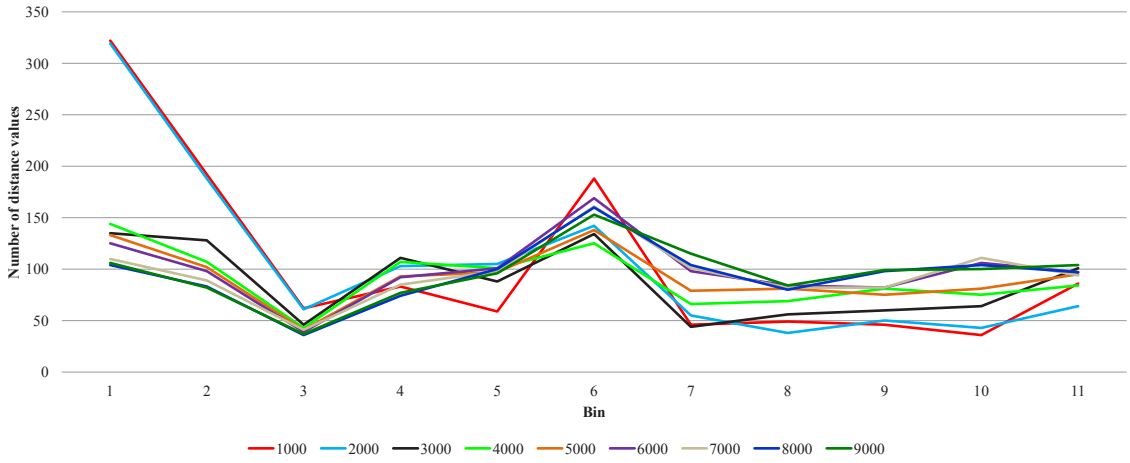
We have also analyzed the separability between distance distributions of different semantic classes of objects in the BoW (mid-level) space. Our results show that, although there is some separability for bags based on soft assignment and max pooling, the separability is very small.

This motivates the creation of a new feature space with more semantics. In this direction, we have worked on a bag-of-prototypes model, according to which the prototypes are elements containing more semantic information. This is also a step forward to reduce the semantic gap. We propose a dictionary of scenes, which could be considered a particular case of the dictionary of prototypes. Its visual words tend to have more semantics for humans than local low-level features, like SIFT, for example. Therefore, the feature space spanned by such dictionary has the property of having one dimension for each semantic concept. Due to its nature, we have performed an evaluation in a video geocoding sce-

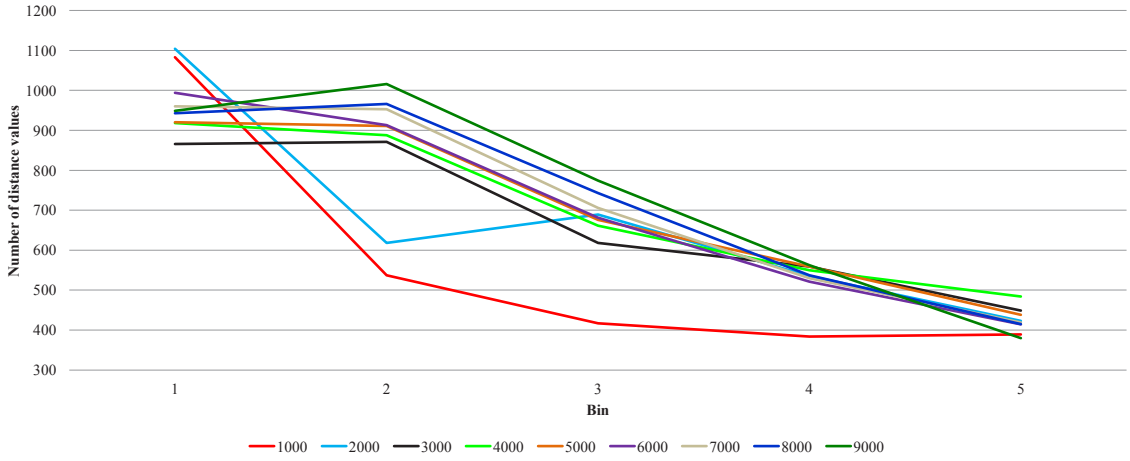
nario, in the context of the Placing Task at MediaEval 2011. Our results have shown that the proposed bag-of-scenes model is effective for video geocoding, being more precise than most of the geocoding methods presented at the Placing Task of 2011. We could evaluate the differences in creating random dictionaries and dictionaries guided by the video locations. The guided dictionaries show large improvement over the random ones. An analysis on the bag-of-scenes vector has also shown that, as more places (scenes) are included in the dictionary, the most activated scenes tend to come closer to the correct video location.




(a) quantization of 1km

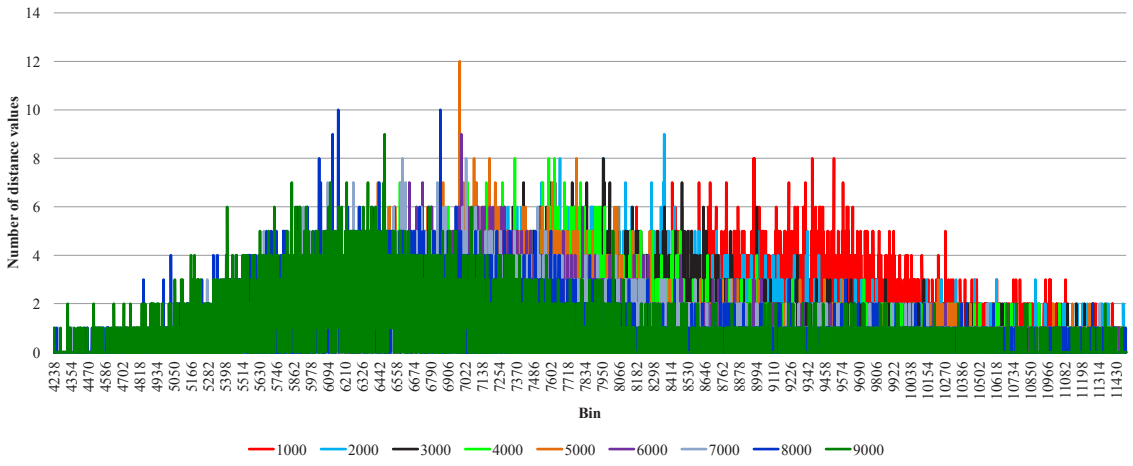



(b) quantization of 10km

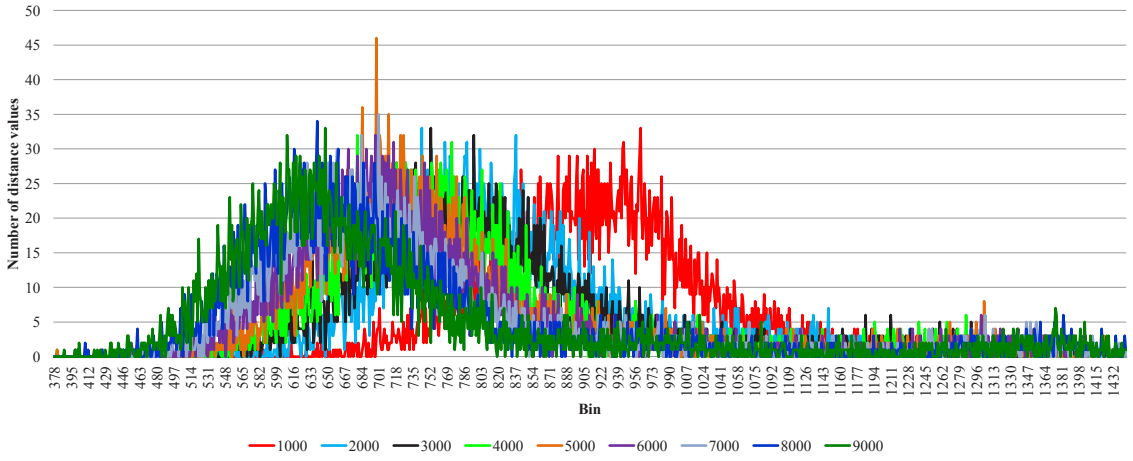



(c) quantization of 100km

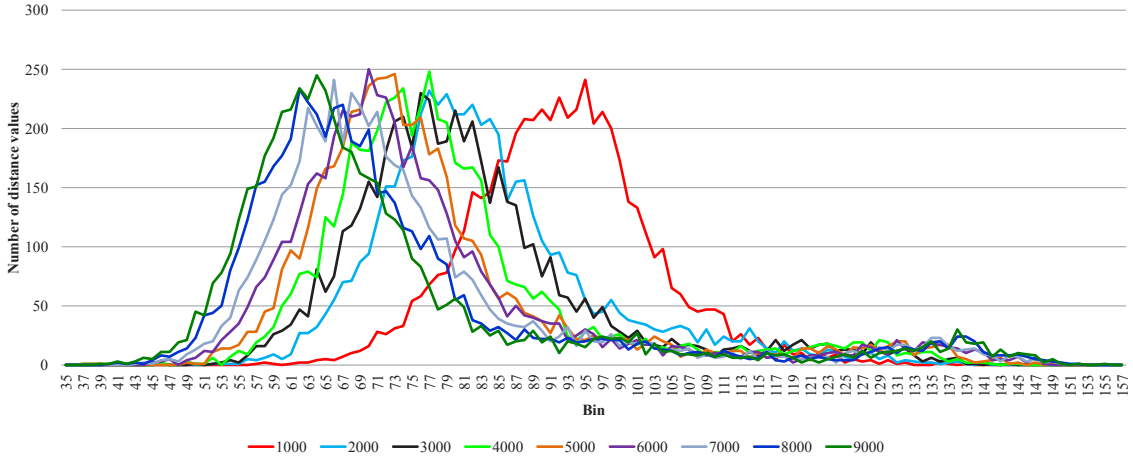
Figure 5.11: Histograms of distances considering the *minimum* distance among the 50 most activated scenes of each video. For finer quantizations (1km and 10km), the dictionaries of 1 000 and 2 000 are the best ones. Only for quantization of 100km at bin 2, the larger the dictionary, the better.




(a) quantization of 1km




(b) quantization of 10km




(c) quantization of 100km

Figure 5.12: Histograms of distances considering the *average* distance among the 50 most activated scenes of each video. In all quantization levels, the larger the dictionary, the better.

# Chapter 6

# Conclusions

Making digital visual information understandable by computers is a challenge that motivates the research described in this thesis. One of the main elements to make this possible is to represent the visual content effectively. In other words, we have to transform the raw visual information in a distinctive digital element. By raw visual information, we can have an image captured by a digital camera, for example. By distinctive digital element, we can have a feature vector, which should be representative enough to distinguish different visual concepts. Although there are several techniques for representing visual information, in this thesis, we focus on representations based on visual dictionaries. Visual dictionaries lie in the idea of describing visual content as describing text documents [70]. Therefore, a visual dictionary works as a codebook of the available elements to represent the image. This model is successful for visual recognition, however, there are challenges on how to create a suitable visual dictionary and on how to encode the spatial information of visual words, for instance. There are also questions related on how to include more semantic information into the dictionary and even on how to create a representation that is intuitive for humans [80].

In this thesis, we have presented contributions in three different topics related to the visual dictionary model.

The contributions presented in Chapter 3 state that visual dictionaries are generalizable in the sense that dictionaries generalize well from one dataset to another and from a subset to the whole dataset. We have shown through experiments that we can create a visual dictionary based on one dataset and represent effectively images from another dataset. We have also shown that we can use a very small portion of a dataset to create a good dictionary. The visual variability of a dataset is the most important characteristic to build a good dictionary. If the source image dataset is diverse enough in terms of visual appearances, the dictionary based on it may be good to represent a wider range of other datasets. Those aspects show the generality power of visual dictionaries, highlighting their

potential to be used in heterogeneous and dynamic environments, as the Web. All those conclusions also point to the direction of alleviating the cost of generating dictionaries. Many works employ efforts in creating elaborated techniques for improving the feature quantization step. However, we have shown that if the features cover a great portion of the feature space, we have enough information to use simple quantization techniques and generate a good dictionary.

In Chapter 4, we present a new pooling method for encoding the spatial arrangement of visual words, called WSA. WSA goes in the direction of solving the problem of the lack of spatial information captured by the traditional pooling approaches. Oppositely to most of the existing spatial pooling methods, WSA generates a compact feature vector and can be directly used for image retrieval and also classification. We have shown how WSA performs in experiments for image retrieval and classification. In the retrieval scenario, WSA has superior performance than the most popular approach to the spatial pooling of visual words, the Spatial Pyramids. WSA has also presented adequate performance in the classification scenario, although it was outperformed by Spatial Pyramids in very soft assignments. Considering the fact that WSA generates compact feature vectors, it is an initial step for having a spatial pooling method in Web environments, where we should be aware of storage efficiency.

Chapter 5 deals with the fact that the name *visual dictionaries* is misleading. The visual words of the most common dictionaries based on local low-level features do not have a meaning for humans. We have performed several experiments showing that there is no semantic information in the visual words of the traditional dictionaries. Although we could expect that appearances carry semantics, due to the fact that the local descriptions are very punctual and precise, we have seem that the semantic separability in the low-level feature space does not exist. We have also questioned why the BoW approach works so well if they are based on non-semantic elements. Our experiments analyzing the separability between distance distributions of different classes of objects in the BoW (mid-level) feature space have also shown that even in this space, the separability is very small. Therefore, we discuss that if we use a representation based on elements which contain more semantics, we could improve the quality of the image representations, creating a feature space with more semantic separability. In this direction, we have worked on a *bag-of-prototypes* model, according to which the prototypes are elements containing semantics. This is a step forward to reduce the semantic gap and to create a representation that is more intuitive for humans. We have presented the *bag-of-scenes* representation. It is based on a dictionary of pictures from places, thus being a representation based on elements with more semantics than local patches. The bag-of-scenes model was evaluated in the context of video geocoding and was used in the Placing Task at MediaEval 2011. Given the nature of the bag-of-scenes representation, in the geocoding task it works as a

placing activation vector, providing good insights about the video location. The results presented are promising and show an encouraging direction for the success of dictionaries based on elements having more semantic information.

## 6.1 Future work

This thesis has created opportunity for further investigations in relation to all the research challenges presented. Next, we present some of the future work envisioned in relation to each chapter.

### 6.1.1 Dictionaries generality

Considering the generality of visual dictionaries presented in Chapter 3:

- We would like to evaluate the generality of visual dictionaries in other datasets.

- We would like to explore if a feature space quantization independent of the data is also effective. For example, instead of using samples of the dataset to perform the quantization, we could use quantization schemes similar to the ones used for color spaces employed by global color descriptors [61]. In those quantization schemes, we simply select the quantization level for each channel. We know that the SIFT feature space is not uniform as the color spaces, therefore, non-uniform quantization schemes should be more promising. In case such quantization results in good dictionaries, we could perform an evaluation of the most suitable quantization levels for each kind of application.

- Considering the quantization scheme not based on the data just mentioned in the previous item, we could also be able to create a repository of codewords (visual words coordinates in the feature space) and codebooks (set of visual words) which are adequate for different types of applications. For example, we could provide a list of codewords to be used for datasets of natural images, for datasets of more heterogeneous content, for datasets of face recognition, and so on.

- In relation to our experiments performed in the Web environment presented in Section 3.4, we have used a pool of relevant images that was created initially to evaluate global descriptors [38]. Therefore, it must be biased by the global information of images and could be one reason for the low precision values presented. We plan to investigate other possibilities to evaluate the BoW representations in that scenario.

- We would like to explore if the generality of visual dictionaries is also valid on special-purpose datasets, like in applications for diagnostics in medical images or for face recognition, for example.

### 6.1.2 Spatial information of visual words

Considering the proposed WSA pooling method presented in Chapter 4 and also the challenge of encoding the spatial arrangement of visual words in general, we have identified the following possible future work:

- We would like to evaluate WSA in other datasets, considering again both the retrieval and classification scenarios.

- We would like to run more retrieval experiments in semantic-search applications. Our experiments in such applications considered image classification, therefore, we are willing to know if a comparison between images without considering the space partitioning created by SVMs would be more promising for WSA.

- We plan to investigate the use of WSA in more partial-duplicate applications. The good results of WSA in the Paris dataset are also an indication of its potential to more precise applications.

- Indexing WSA vectors is also an important aspect to assess efficiency in retrieval systems and could be addressed in future work. Considering the small vector generated per visual word (4 dimensions), using inverted files or customized trees such as in [29] could be considered as suitable solutions.

- Given the problems faced when using very soft assignments with WSA, we would like to investigate some solutions. The large improvement in accuracy when using WSA with Spatial Pyramids in very soft assignments opens opportunities for further investigations.

- The counting process of WSA depends on the points falling in one of the four quadrants. However, there are cases in which a point falls exactly in the axis. In the current version of WSA, we select only one of the quadrants to be incremented. We would like to investigate a soft counting scheme considering more than one quadrant in such cases.

- Considering the spatial information of visual words in general, we also would like to explore a scheme to work over the dense-sampling approach. Although WSA also works with dense-sampling, it is designed for sparse-sampling. A scheme inspired

on the BIC descriptor [17] was initially tried but further investigation is necessary. Using similar ideas to those employed for the Local Binary Patterns (LBP) descriptor [61] is another option.

- In relation to the spatial information of visual words in the dense-sampled image, we also plan to analyze the possibility of using graph-based approaches, like the Image Foresting Transform (IFT) [23].

### 6.1.3 Semantic information in visual dictionaries

Considering the semantic information in visual dictionaries and also the geocoding application presented in Chapter 5, we propose the following possible research opportunities:

- In the bag-of-prototypes model, we plan to create dictionaries where the prototypes are objects and use them to represent the Pascal VOC 2010 dataset. As this dataset usually has several different objects per image, a dictionary of objects would be promising to encode such information.

- Considering the dictionary of such prototypes (objects as in the case presented in the previous item), we would like to analyze the semantic separability in that space by conducting experiments similarly to the presented for the low-level and mid-level spaces.

- We plan to investigate objective measures to assess the separability between the histogram of distances presented in Section 5.2.

- The bag-of-prototypes model opens opportunities to evaluate different strategies in different applications. For example, for remote sensing applications, we could explore the use of a dictionary based on textures of interest. That dictionary could be composed of a set of textures representing the desired crop and a set of textures representing the non-crop regions. Another possibility is the use of a dictionary of face parts to be used in face recognition applications. We could also investigate the use of such model in medical applications.

- Some preliminary experiments with the bag of prototypes for image representation have shown the difficulty in selecting meaningful prototypes. An interesting possibility for the selection of prototypes is by training a classifier, like SVM, for each desired concept (e.g., object) and then use the SVM frontier as the prototype. This approach would have the advantage of being more general than using directly a representative feature vector of the desired concept. The SVM frontiers tend to better encode the intra-class differences between concepts of the same type.

- For applications of image retrieval and classification based on attributes [58, 67], we could note the effort in assigning textual attributes to images as a post-processing step, by using classification techniques like SVM. We would like to investigate if a bag-of-prototypes approach could embed this information into the image representation.

- Considering our bag-of-scenes approach, we plan to evaluate other strategies for assigning the geographic information to a test video, instead of simply copying the latitude/longitude of the closest video of the development set.

- We are also considering the use of other low-level features to represent the video frames. As we have used only global descriptions based on CEDD descriptor, we would like to try representations which encode more local information, like the bag of visual words presented throughout this thesis. A promising method would be the proposed WSA pooling approach presented in Chapter 4. Its good results in the Paris dataset are an indication of its possible success to find similar photos in a dictionary of scenes.

- We plan to evaluate the bag-of-scenes model in other applications, like video genre categorization, for instance.

## 6.2 Publications

The publications below were directly or indirectly related to the work presented in this thesis.

- *Encoding spatial arrangement of visual words* [62], O. A. B. Penatti, E. Valle, and R. da S. Torres, in the Iberoamerican Congress on Pattern Recognition (CIARP), 2011, receiving the best paper award. An extension of this work was submitted to the Pattern Recognition journal in September, 2012.

- *A Visual Approach for Video Geocoding using Bag-of-Scenes* [59], O. A. B. Penatti, L. T. Li, J. Almeida, and R. da S. Torres, in the International Conference on Multimedia Retrieval (ICMR), 2012.

- *Comparative study of global color and texture descriptors for web image retrieval* [61], O. A. B. Penatti, E. Valle, and R. da S. Torres, in the Journal of Visual Communication and Image Representation, 2012.

- *Improving Texture Description in Remote Sensing Image Multi-Scale Classification Tasks By Using Visual Words* [20], J. A. dos Santos, O. A. B. Penatti, R. da S.

Torres, P-H. Gosselin, S. Philipp-Foliguet, and A. X Falcão, in the International Conference on Pattern Recognition (ICPR), 2012.

- *Multimedia Multimodal Geocoding* [47], L. T. Li, D. C. G. Pedronette, J. Almeida, O. A. B. Penatti, R. T. Calumby, and R. da S Torres, in the International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS), 2012.

- *UNICAMP-UFMG at MediaEval 2012: Genre Tagging Task* [5], J. Almeida, T. Salles, E. F. Martins, O. A. B. Penatti, R. da S. Torres, M. A. Gonçalves, and J. M. Almeida, in the Working Notes of the MediaEval Workshop, 2012.

- *A Multimodal Approach for Video Geocoding* [46], L. T. Li, J. Almeida, D. C. G. Pedronette, O. A. B. Penatti, and R. da S. Torres, in the Working Notes of the MediaEval Workshop, 2012.

# Appendix A

# WSA: parameter evaluation of the proposed distance function for image retrieval

In this appendix, we present how the proposed distance performs with different values of its parameters. The results presented in Section 4.4 show how the proposed distance function improves the effectiveness of WSA descriptors. The MAP and precision values presented in Tables 4.2 and 4.3 are based on one of the best parameter configuration obtained when using the proposed distance function. However, we have performed an evaluation of the parameters to determine which values would be more appropriate

The parameters involved in the evaluation are: the distance function $dist_j$ used to compare a pair of WSA's (4-value set) and the threshold $\epsilon$ of those distances, which indicates if a pair of visual words is a match or not. For $dist_j$, we tested L1 and L2 and, for $\epsilon$, $\frac{1}{4}$, $\frac{1}{3}$, and $\frac{1}{2}$ of the maximum distance ($distMax$). Combined with the different assignment methods and the different window sizes tested with WSA, there is a large number of parameter combinations.

For Base-600, the results are presented in Figure A.1. We can see that the smallest window has the best performance for soft assignments. The reason is that as more visual words are assigned to each point, more counters are incremented during WSA computation. Therefore, in the case of larger windows (or no windows) too many counters will be incremented, while for a small window, only few points are considered in the counting process. We can also see that there is almost no difference when using L1 or L2 as $dist_j$. The smaller the $\epsilon$ value, the worse for harder assignments (hard, soft $\sigma$=30 and $\sigma$=60). The reason is that with a small $\epsilon$ value, fewer words are considered as common words because they do not satisfy the spatial constraint of the proposed distance function. Therefore,

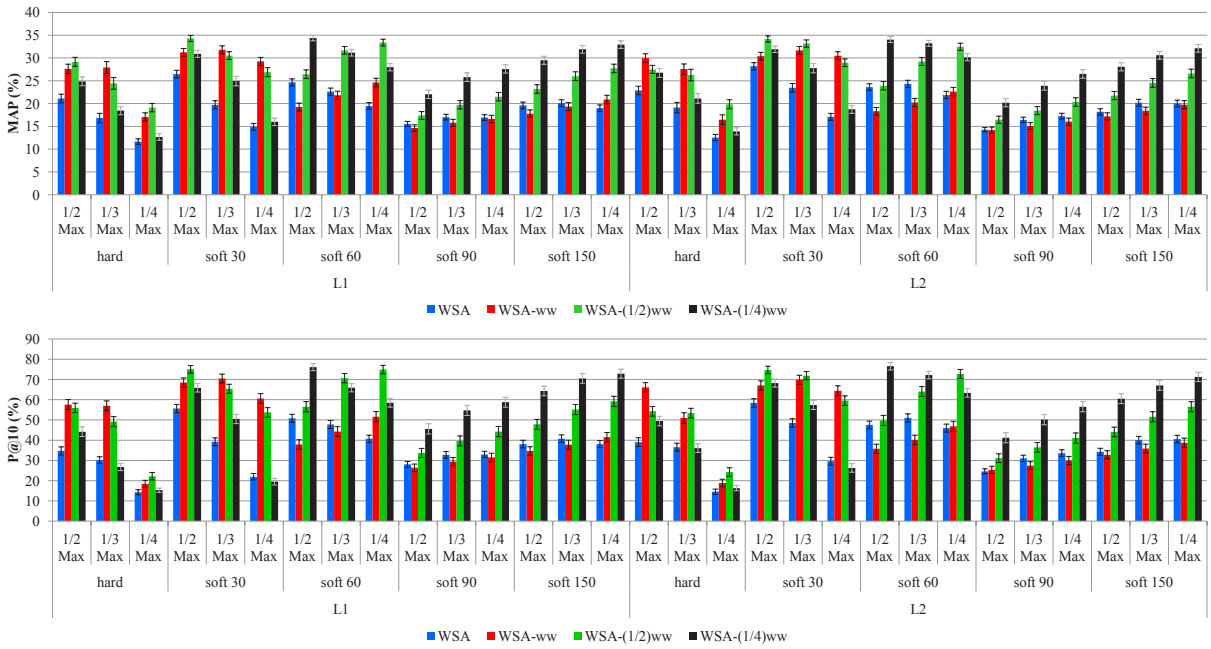



Figure A.1: Base-600: retrieval results for WSA versions varying all parameters of the proposed distance function. The first line in the graph labels is the $\epsilon$ value, while the second is the assignment type, and the last is the distance function for $dist_j$.

when increasing the soft assignment, small $\epsilon$ values become better because more words are assigned to each point, resulting in more common words.

For the Paris dataset, the results are presented in Figure A.2. We can note little difference in using L1 or L2 as $dist_j$. Some difference in favor of L2 is observed for WSA-$\frac{1}{2}$ww with $\epsilon$=$\frac{1}{3}distMax$ and for WSA-ww with $\epsilon$=$\frac{1}{4}$, and, in favor of L1, for WSA-$\frac{1}{4}$ww with $\epsilon$=$\frac{1}{4}distMax$. Considering the $\epsilon$ value, we could note that, usually, the smaller the $\epsilon$, the worse. This means that, as we increase the spatial restriction to consider a pair of visual words as a match, we end up discarding some important visual words. We have tested some even larger values for $\epsilon$ but no improvements were observed.

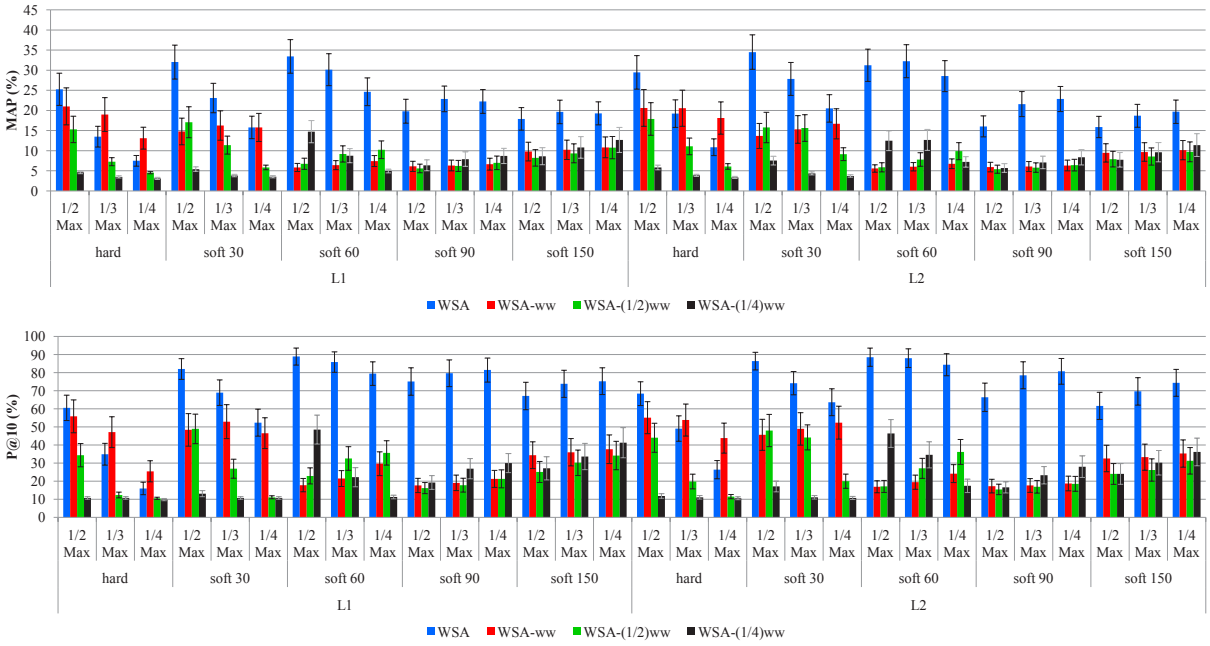



Figure A.2: Paris: retrieval results for WSA versions varying all parameters of the proposed distance function. The first line in the graph labels is the $\epsilon$ value, while the second is the assignment type, and the last is the distance function for $dist_j$.

# Bibliography


[1] A. Al-Maskari, M. Sanderson, and P. Clough. The relationship between ir effectiveness measures and user satisfaction. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 773–774, 2007.

[2] J. Almeida, N. J. Leite, and R. da S. Torres. Comparison of video sequences with histograms of motion patterns. In *International Conference on Image Processing*, pages 3673–3676, 2011.

[3] J. Almeida, N. J. Leite, and R. da S. Torres. VISON: VIdeo Summarization for ONline applications. *Pattern Recognition Letters*, 33(4):397–409, 2012.

[4] J. Almeida, N. J. Leite, and R. da S. Torres. Online video summarization on compressed domain. *Journal of Visual Communication and Image Representation*, 2012. In press.

[5] J. Almeida, T. Salles, E. F. Martins, O. A. B. Penatti, R. da S. Torres, M. A. Gonçalves, and J. M. Almeida. UNICAMP-UFMG at mediaeval 2012: Genre tagging task. In *Working Notes Proceedings MediaEval Workshop*, 2012.

[6] J. Almeida, R. da S. Torres, and N. J. Leite. Rapid video summarization on compressed video. In *International Symposium on Multimedia*, pages 113–120, 2010.

[7] F. S. P. Andrade, J. Almeida, H. Pedrini, and R. da S. Torres. Fusion of local and global descriptors for content-based image and video retrieval. In *Iberoamerican Congress on Pattern Recognition*, pages 845–853, 2012.

[8] A. Antani, R. Kasturi, and R. Jain. A Survey on the Use of Pattern Recognition Methods for Abstraction, Indexing and Retrieval of Images and Video. *Pattern Recognition*, 35(4):945–965, Apr. 2002.

[9] S. Avila, N. Thome, M. Cord, E. Valle, and A. de A. Araújo. Bossa: Extended BOW formalism for image classification. In *International Conference on Image Processing*, pages 2966–2969, 2011.

[10] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3D human pose annotations. In *International Conference on Computer Vision*, pages 1365–1372, 2009.

[11] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 2559–2566, 2010.

[12] T. Brox, L. Bourdev, S. Maji, and J. Malik. Object segmentation by alignment of poselet activations to image contours. In *Conference on Computer Vision and Pattern Recognition*, pages 2225–2232, 2011.

[13] Jr. C. Traina, A. Traina, C. Faloutsos, and B. Seeger. Fast indexing and visualization of metric data sets using slim-trees. *IEEE Transactions on Knowledge and Data Engineering*, 14(2):244–260, 2002.

[14] H. Cai, F. Yan, and K. Mikolajczyk. Learning weights for codebook in image classification and retrieval. In *Conference on Computer Vision and Pattern Recognition*, pages 2320–2327, 2010.

[15] Y. Cao, C. Wang, Z. Li, L. Zhang, and L. Zhang. Spatial–bag–of–features. In *Conference on Computer Vision and Pattern Recognition*, pages 3352–3359, 2010.

[16] J. Choi, H. Lei, and G. Friedland. The 2011 ICSI video location estimation system. In *Working Notes Proceedings MediaEval Workshop*, volume 807, 2011.

[17] R. de O. Stehling, M. A. Nascimento, and A. X. Falcão. A compact and efficient image retrieval approach based on border/interior pixel classification. In *International Conference on Information and Knowledge Management*, pages 102–109, 2002.

[18] A. del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 1999.

[19] J. A. dos Santos, F. A. Faria, R. da S. Torres, A. Rocha, P-H. Gosselin, S. Philipp-Foliguet, and A. X. Falcão. Descriptor correlation analysis for remote sensing image multi-scale classification. In *International Conference on Pattern Recognition*, pages 3078–3081, 2012.

[20] J. A. dos Santos, O. A. B. Penatti, R. da S. Torres, P-H. Gosselin, S. Philipp-Foliguet, and A. X. Falcão. Improving texture description in remote sensing image multi-scale classification tasks by using visualwords. In *International Conference on Pattern Recognition*, pages 3090–3093, 2012.

[21] R. Elwell and R. Polikar. Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, 22(10):1517–1531, Oct. 2011.

[22] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html (as of February 6th, 2013).

[23] A. X. Falcão, J. Stolfi, and R. A. Lotufo. The image foresting transform: Theory, algorithms, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):19–29, Jan. 2004.

[24] F. F. Faria, A. Veloso, H. M. Almeida, E. Valle, R. da S. Torres, M. A. Gonçalves, and W. Meira Jr. Learning to rank for content-based image retrieval. In *International Conference on Multimedia Information Retrieval*, pages 285–294, 2010.

[25] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007.

[26] J. Feng, B. Ni, Q. Tian, and S. Yan. Geometric lp-norm feature pooling for image classification. In *Conference on Computer Vision and Pattern Recognition*, pages 2609–2704, 2011.

[27] P. H. Gosselin, M. Cord, and S. Philipp-Foliguet. Combining visual dictionary, kernel-based similarity and learning strategy for image category retrieval. *Computer Vision and Image Understanding*, 110(3):403–417, 2008.

[28] C. Hauff and G.-J. Houben. WISTUD at MediaEval 2011: Placing task. In *Working Notes Proceedings MediaEval Workshop*, volume 807, 2011.

[29] N. V. Hoàng, V. Gouet-Brunet, M. Rukoz, and M. Manouvrier. Embedding spatial information into image content description for scene retrieval. *Pattern Recognition*, 43(9):3013–3024, Sep. 2010.

[30] J. Huang, S. R. Kumar, M. Mitra, W. Zhu, and R. Zabih. Image indexing using color correlograms. In *Conference on Computer Vision and Pattern Recognition*, pages 762–768, 1997.

[31] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *European Conference on Computer Vision: Part I*, volume 5302, pages 304–317, 2008.

[32] H. Jégou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *International Journal of Computer Vision*, 87(3):316–336, 2010.

[33] H. Jégou, M. Douze, C. Schmid, and P. Perez. Aggregating local descriptors into a compact image representation. In *Conference on Computer Vision and Pattern Recognition*, pages 3304–3311, 2010.

[34] R. Ji, H. Yao, X. Sun, B. Zhong, and W. Gao. Towards semantic embedding in visual vocabulary. In *Conference on Computer Vision and Pattern Recognition*, pages 918–925, 2010.

[35] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *International Conference on Computer Vision*, volume 1, pages 604–610, 2005.

[36] H. Kang, M. Hebert, and T. Kanade. Image matching with distinctive visual vocabulary. In *IEEE Workshop on Applications of Computer Vision*, pages 402–409, 2011.

[37] P. Kelm, S. Schmiedeke, and T. Sikora. Multi-modal, Multi-resource Methods for Placing Flickr Videos on the Map. In *International Conference on Multimedia Retrieval*, pages 52:1–52:8, 2011.

[38] P. A. S. Kimura, J. M. B. Cavalcanti, P. C. Saraiva, R. da S. Torres, and M. A. Gonçalves. Evaluating retrieval effectiveness of descriptors for searching in large image databases. *Journal of Information and Data Management*, 2(3):305–320, 2011.

[39] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2–3):107–123, 2005.

[40] M. Larson, M. Soleymani, P. Serdyukov, S. Rudinac, C. Wartena, V. Murdock, G. Friedland, R. Ordelman, and G. J. F. Jones. Automatic tagging and geotagging in video collections and communities. In *International Conference on Multimedia Retrieval*, pages 51:1–51:8, 2011.

[41] R. R. Larson. Geographic information retrieval and digital libraries. In *European Conference on Research and Advanced Technology for Digital Libraries*, volume 5714, pages 461–464, 2009.

[42] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, and F. Stentiford. Video copy detection: a comparative study. In *International Conference on Image and Video Retrieval*, pages 371–378, 2007.

[43] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178, 2006.

[44] L-J. Li, H. Su, E. P. Xing, and L. Fei-Fei. Object bank: A high-level image representation for scene classification and semantic feature sparsification. In *Neural Information Processing Systems*, 2010.

[45] L. T. Li, J. Almeida, and R. da S. Torres. RECOD working notes for placing task MediaEval 2011. In *Working Notes Proceedings MediaEval Workshop*, volume 807, 2011.

[46] L. T. Li, J. Almeida, D. C. G. Pedronette, O. A. B. Penatti, and R. da S. Torres. A multimodal approach for video geocoding. In *Working Notes Proceedings MediaEval Workshop*, 2012.

[47] L. T. Li, D. C. G. Pedronette, J. Almeida, O. A. B. Penatti, R. T. Calumby, and R. da S. Torres. Multimedia Multimodal Geocoding. In *ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 474–477, 2012.

[48] J. Liu, Y. Yang, and M. Shah. Learning semantic visual vocabularies using diffusion distance. In *Conference on Computer Vision and Pattern Recognition*, pages 461–468, 2009.

[49] L. Liu, L. Wang, and X. Liu. In defense of soft-assignment coding. In *International Conference on Computer Vision*, pages 2486–2493, 2011.

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

[51] J. Luo, D. Joshi, J. Yu, and A. Gallagher. Geotagging in multimedia and computer vision–a survey. *Multimedia Tools and Applications*, 51(1):187–211, 2011.

[52] S. Maji, L. Bourdev, and J. Malik. Action recognition from a distributed representation of pose and appearance. In *Conference on Computer Vision and Pattern Recognition*, pages 3177–3184, 2011.

[53] E. Mbanya, S. Gerke, and P. Ndjiki-Nya. Spatial codebooks for image categorization. In *International Conference on Multimedia Retrieval*, pages 50:1–50:7, 2011.

[54] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

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

[56] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[57] G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. In *ACM Multimedia*, pages 65–73, 1996.

[58] G. Patterson and J. Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *Conference on Computer Vision and Pattern Recognition*, pages 2751–2758, 2012.

[59] O. A. B. Penatti, L. T. Li, J. Almeida, and R. da S. Torres. A Visual Approach for Video Geocoding using Bag-of-Scenes. In *International Conference on Multimedia Retrieval*, pages 53:1–53:8, 2012.

[60] O. A. B. Penatti and R. da S. Torres. Eva - an evaluation tool for comparing descriptors in content-based image retrieval tasks. In *International Conference on Multimedia Information Retrieval*, pages 413–416, 2010.

[61] O. A. B. Penatti, E. Valle, and R. da S. Torres. Comparative study of global color and texture descriptors for web image retrieval. *Journal of Visual Communication and Image Representation*, 23(2):359–380, 2012.

[62] O. A. B. Penatti, E. Valle, and R. da S. Torres. Encoding spatial arrangement of visual words. In *Iberoamerican Congress on Pattern Recognition*, volume 7042, pages 240–247, 2011.

[63] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *European Conference on Computer Vision*, volume 6314, pages 143–156, 2010.

[64] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[65] A. Rae, V. Murdock, P. Serdyukov, and P. Kelm. Working notes for the placing task at MediaEval. In *Working Notes Proceedings MediaEval Workshop*, volume 807, 2011.

[66] S. Savarese, J. Winn, and A. Criminisi. Discriminative object class models of appearance and shape by correlatons. In *Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2033–2040, 2006.

[67] W. J. Scheirer, N. K., P. N. Belhumeur, and T. E. Boult. Multi-attribute spaces: Calibration for attribute fusion and similarity search. In *Conference on Computer Vision and Pattern Recognition*, pages 2933–2940, 2012.

[68] W. J. Scheirer, A. Rocha, R. J. Micheals, and T. E. Boult. Meta-recognition: The theory and practice of recognition score analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1689–1695, Aug. 2011.

[69] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their location in images. In *International Conference on Computer Vision*, volume 1, pages 370–377, 2005.

[70] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *International Conference on Computer Vision*, volume 2, pages 1470–1477, 2003.

[71] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[72] B. Thomee, M. J. Huiskes, E. Bakker, and M. S. Lew. Large scale image copy detection evaluation. In *International Conference on Multimedia Information Retrieval*, pages 59–66, 2008.

[73] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *Conference on Computer Vision and Pattern Recognition*, pages 1521–1528, 2011.

[74] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: a survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, Jul. 2008.

[75] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, 2010.

[76] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Empowering visual categorization with the GPU. *IEEE Transactions on Multimedia*, 13(1):60–70, 2011.

[77] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J-M Geusebroek. Visual word ambiguity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(7):1271–1283, 2010.

[78] O. van Laere, S. Schockaert, and B. Dhoedt. Ghent university at the 2011 placing task. In *Working Notes Proceedings MediaEval Workshop*, volume 807, 2011.

[79] V. Viitaniemi and J. Laaksonen. Experiments on selection of codebooks for local image feature histograms. In *International Conference on Visual Information Systems: Web-Based Visual Information Search and Management*, pages 126–137, 2008.

[80] J. Vogel and B. Schiele. Semantic modeling of natural scenes for content-based image retrieval. *International Journal of Computer Vision*, 72(2):133–157, 2007.

[81] J. Wang, J. Yang, K. Yu, F Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Conference on Computer Vision and Pattern Recognition*, pages 3360–3367, 2010.

[82] R. Weber, H. Schek, and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *International Conference on Very Large Data Bases*, pages 194–205, 1998.

[83] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *International Conference on Computer Vision*, volume 2, pages 1800–1807, 2005.

[84] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010.

[85] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Conference on Computer Vision and Pattern Recognition*, pages 1794–1801, 2009.

[86] W. Zhang and H. Deng. Understanding visual dictionaries via maximum mutual information curves. In *International Conference on Pattern Recognition*, pages 1–4, 2008.

[87] W. Zhang and T. G. Dietterich. Learning visual dictionaries and decision lists for object recognition. In *International Conference on Pattern Recognition*, pages 1–4, 2008.

[88] W. Zhou, H. Li, Y. Lu, and Q. Tian. Large scale image search with geometric coding. In *ACM Multimedia*, pages 1349–1352, 2011.

[89] W. Zhou, Y. Lu, H. Li, Y. Song, and Q. Tian. Spatial coding for large scale partial-duplicate web image search. In *International Conference on Multimedia*, pages 511–520, 2010.