ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

MINES ParisTech

École doctorale 432: SMI - Sciences des Métiers de l'Ingénieur

Doctorat ParisTech

THÈSE

pour obtenir le grade de docteur délivré par

l'École nationale supérieure des mines de Paris

Spécialité "Informatique temps réel, robotique et automatique - Paris "

présentée et soutenue publiquement par

Andrei BURSUC

21 Décembre 2012

Object-based visual content indexing and retrieval

Indexation et recherche de contenus par objet visuel

Directeur de thèse : **Françoise PRETEUX**

Co-encadrement de la thèse : **Titus ZAHARIA**

Jury

M. Nicu SEBE, Professeur, Université de Trento	Rapporteur
M. Mohammed DAOUDI, Professeur, TELECOM Lille1	Rapporteur
Mme. Jenny BENOIS-PINEAU, Professeur, Université Bordeaux	Examinateur
M. Gérard MOZELLE, Expert conseil, Alten	Examinateur
Mme. Françoise PRETEUX, Professeur, Mines ParisTech	Examinateur
M. Titus ZAHARIA, Professeur, Télécom SudParis	Examinateur

THÈSE

MINES ParisTech
Mathématiques et Systèmes, CAOR - Centre de CAO et Robotique
MINES ParisTech, 60 Boulevard Saint-Michel 75006 Paris, France

To my spouse and to my parents

Acknowledgment

My PhD time has been an exciting and rewarding experience thanks to many people that have contributed to my research and to my personal development. I am fortunate to have had remarkable mentors, colleagues, friends and family who have offered me invaluable guidance and support. I will attempt at thanking as many as I can.

I am profoundly grateful to my thesis director, Professor Françoise Prêteux for the significant role she has in my decision to become a researcher and for helping me to take my first steps in research in the ARTEMIS department at Télécom SudParis and in the CAOR department at MINES ParisTech. I would like to thank her for the trust she invested in me and for seeing in me a future researcher. With her personal example, she has taught me that good research can only be achieved with hard work, precision and rigor. I take this opportunity to express my gratitude and appreciation for her guidance, availability and constant support.

Equally, I would like to thank my thesis co-director, Professor Titus Zaharia for being an excellent advisor, mentor, collaborator and friend. Titus has had a great influence on my work and research, my interests and perspectives of life. He has taught me how to tackle on unsolved difficult and meaningful problems, how to persevere when results are less encouraging, how to envision research and results on a longer term, how to have confidence in me and in my work and how to present it to others. I would like to express my deep gratitude and admiration for his guidance, expertise, energy, enthusiasm and full investment in working with me.

I will never forget the time and care that both have invested in me. I am eternally grateful to them as this thesis would not have been possible without their participation.

I would like to express my gratitude to Professor Mohamed Daoudi for granting me the honor of being a reviewer for my thesis, for his interest in my work, for his thoughtful comments and for the perspectives that he has envisioned for the thesis.

My sincere gratitude goes to Professor Nicu Sebe for accepting to review my thesis, for his time and attention that he has dedicated to my thesis, for his precious feedback and for our discussions. I hope I will make good use of his advice and suggested perspectives in my future work as a researcher. I hope I will become as enthusiastic and passionate about research as he is.

I would like to send my special thanks to Professor Jenny Benois-Pineau, for presiding the jury, for the time, patience and expertise she invested in evaluating my work. Her work, feedback and questions have helped me in improving my work and in foreseeing new perspectives of research. I keep her patience and attention to details as guidelines for my future research.

I would like to express my appreciation to PhD Mr. Gérard Mozelle for bringing his industrial, strategic and pragmatic point of view in the evaluation of my work. His example and advice will be surely useful for my future projects.

I would like to thank Mr. Arnaud de La Fortelle, director of the CAOR department and Mr. François, Goulette, thesis responsible at CAOR, who welcomed me at the CAOR department of MINES ParisTech. I appreciate their interest showed for my work during the Doctorades and for their thoughtful and constructive feedback. I would also like to thank Mrs. Christine Vignaux, for her patience and valuable help in the administrative matters.

I am grateful and appreciative to the team from Alcatel Lucent – Bell Labs, with whom I had the pleasure to closely work within the Ubimedia Common Lab. I would like to thank in particular to Yann Gasté, Olivier Martinot, Emmanuel Marilly and Gérard Delegue for their help and support, for the fruitful meetings and for the inspiring exchange of ideas.

I am sending my special thanks to: To HDR Mihai Mitrea for facilitating my arrival in the ARTEMIS Department the first time, and for his precious advice and support during these years. To Raluca Sambra for her generosity and for her perpetual willingness to help, discuss and give advice. To Thomas Laquet for his kind and prompt help with the MPEG-7 libraries and complex programming matters.To Marius Preda and Christian Tulvan for the useful discussions and for sharing their broad vision and expertise on multimedia related and technical concepts. To Afef Chammem for her support and for optimism in facing all types of challenges and events in life. To Mrs. Evelyne Taroni for her patience, discipline and help with administrative matters, her kind understanding and her valuable support and advice in both work-related and personal matters. I owe her thanks for the informal French lessons that she gave me whenever I went to her office.

I am very happy to have had the opportunity to be a part of the ARTEMIS team. I take advantage of this occasion to thank the entire team, former and present members that I have met and from which I have learned so much.

I dedicate my thesis to my wife, to my parents and to my family. Words will never be enough to thank them for their unconditioned love, patience and support.

Table of Contents

1. Introduction

1.1 Context and objectives

The last decade has been characterized by the impressive explosion of image and video content available and shared over Internet. Since 2010, the amount of video content exchanged over Internet account for more than half of the total traffic in the United States [Wired 2010], overcoming web or peer-to-peer traffic. Such a phenomenon is closely related with the emergence of web2.0 technologies, which operated profound transformations in our manner to consume and share the video content.

As a notable example, let us first mention the YouTube platform, which reports one hour of video data uploaded every second and 4 billion videos are viewed every day, resulting in 140 views for every person on Earth in 2011 [YouTube 2012c]. In parallel, the Flickr on-line repository reports 4.5 million photos uploaded daily [Pingdom 2012]. This shows the growing importance of video/image content in the user's everyday life: video and, more generally, visual content become today the new media to be considered.

Moreover, a finer analysis brings us some interesting insights. Thus the most popular camera used in the case of the Flickr images is not a camera *stricto sensus*, but a smartphone (*i.e.*, the iPhone4) [Flickr 2012]. We can thus expect that in the near future an increasing number of uploaded videos will be recorded by ubiquitous, mobile devices. Over the last decade, such mobile devices have been undergoing a booming prosperity. A broader variety of handheld devices with audio/video playback functionalities are available in the market at an affordable price for consumers. In addition, the huge steps forward made by third generation communication networks enable telecom operators to provide enhanced mobile multimedia services, with smoother streaming, reduced video uploading time and higher video quality/resolution.

Such complex socio-economic and technological evolutions through important challenges, related to the capacity to store, search/retrieve and organize huge video/image data collections. A new approach for video content organization, taking into consideration a better understanding of the visual content is needed in order to overcome some of these issues. In addition, tools permitting a more comprehensive visualization of the video content, including advanced navigation facilities and adapted to the multiple terminals and operating systems available, would greatly help to improve the access to video content. Let us underline that the elaboration of advanced query and search strategies is today required in order to provide the user with a rapid and fine access to elements of interest that are present in the video content. Existing video description approaches are most often based on monolithic and global textual representations of the video, which do not take into account the complexity and the heterogeneity of the information usually present in the video content. Such simple strategies should be replaced by new paradigms, able to provide precise and accurate access to the elements/object of interest. Part-based visual representations and content-based retrieval methodologies become the key ingredients that can make it possible to achieve such a fine level access to the desired information.

Several potential applications are directly concerned by such video indexing, search and navigation methodologies. Among them, let us mention:

- **Web-based video search engine.** Most of the video search engines use textual annotations obtained usually from user tags or comments. Although there could be many cases demanding the use of visual cues from user for retrieving particular objects, currently no search engine allowing queries by visual content is available at the scale of the web. Such a tool would be notably useful as an alternative to the poorly organized tag-based querying process, currently employed by the majority of available image/vide search engines.
- **Object retrieval in video surveillance footage.** Video surveillance repositories usually consist of enormous amounts of video footage. Finding the various instances an object of interest can become rapidly an almost impossible task. Methods allowing to select an object of interest and to retrieve automatically the video shots where the considered object appears would save time and improve the work efficiency in this field.
- **Logo/brand use evaluation and discovery.** With the explosion of the amount of available video content and of communications channels, it becomes increasingly difficult to track and evaluate the effectiveness of a advertising campaign. Finding the occurrences of a logo or a brand throughout a video database could become essential tool for the evaluation of advertising campaigns or for measuring video popularity. In addition, a consumer could also use his mobile phone to identify and discover supplementary information about products of interest by just taking a picture or a short video of the item of interest.
- **Automatic tagging and annotation of video collections.** Most of the professional video archives (such as those of video producers or those available at the national archiving organizations) rely on expert users to annotate and appropriately index the videos. Methods permitting to retrieve the occurrences of an object in a large archive would allow the annotation of the results by tag propagation. This will reduce the annotation burden, making the database more accessible for search/retrieve purposes, and facilitating the content re-use.

The detection and recognition of objects of interest from video content is among one of the most difficult problems in the computer vision field, since it requires the use of some partial image representations that can enable efficient macthing strategies. Fast, accurate, scalable and robust object-based visual search is a highly challenging issue, because of the following difficulties:

- **Variations in visual appearance**. The appearance of an object can vary dramatically depending on a series of challenging imaging conditions. Changes in lighting, scale, pose and rotations can alter strongly the visual appearance of the object, thus increasing the difficulty of recognizing a query object.
- **Part based matching.** Video sequences are usually dynamic and objects move and interact with other objects across different sequences. Objects can be occluded and only partially visible or can be deformed.
- **Computation time and scalability.** Despite the huge amount of information included in videos, it is mandatory to ensure both retrieval accuracy and computational speed when increasing the size of the considered data set.
- **Issues related to video content specificity.** Directly related with the previous set of challenges, one of the main difficulties in video content search is the representation of the video. Typically, a reduced set of key-frames is selected and used for the search. However many of them can be blurred or miss the object of interest occurring for a very short time in

the video clip. Identifying the set of most representative key-frames is a difficult challenge to tackle.

The mentioned challenges and difficulties are illustrated in Figure 1.1, Figure 1.2 and Figure 1.3 on typical Flickr videos extracted from the TRECVID 2012 Instance Search Task corpus [Smeaton 2006].

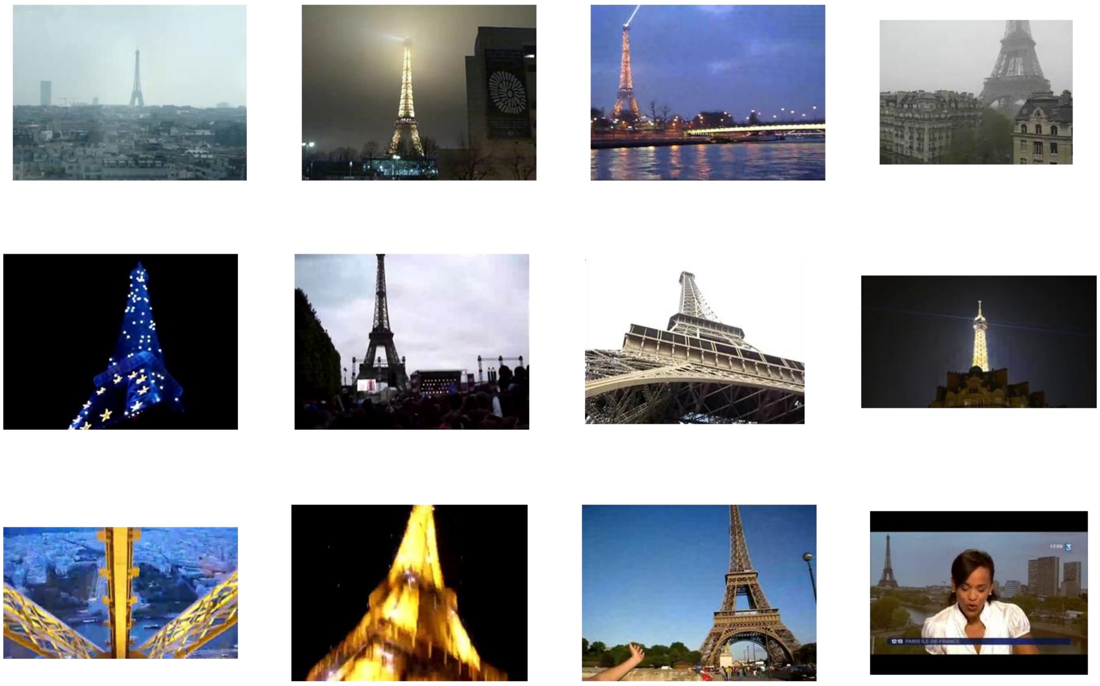


Figure 1.1. Multiple instances of the object "Eiffel tower" within different video clips.

1.1 Contributions

1.1.1 The DOOR framework

The first part of this thesis tackles the issue of retrieving different instances of an object of interest within a given video document or within a video database. Within this context, objects are standalone visual entities with-well defined boundaries, shapes and colors/textures. Examples of typical objects include buildings, cars, electronic devices, animals, logos.... Let us underline that this work concerns the detection of different instances of the same object and not the retrieval of objects belonging to the same class of objects (*e.g.*, different types of chairs, cars), as in the case of semantic image categorization approaches.

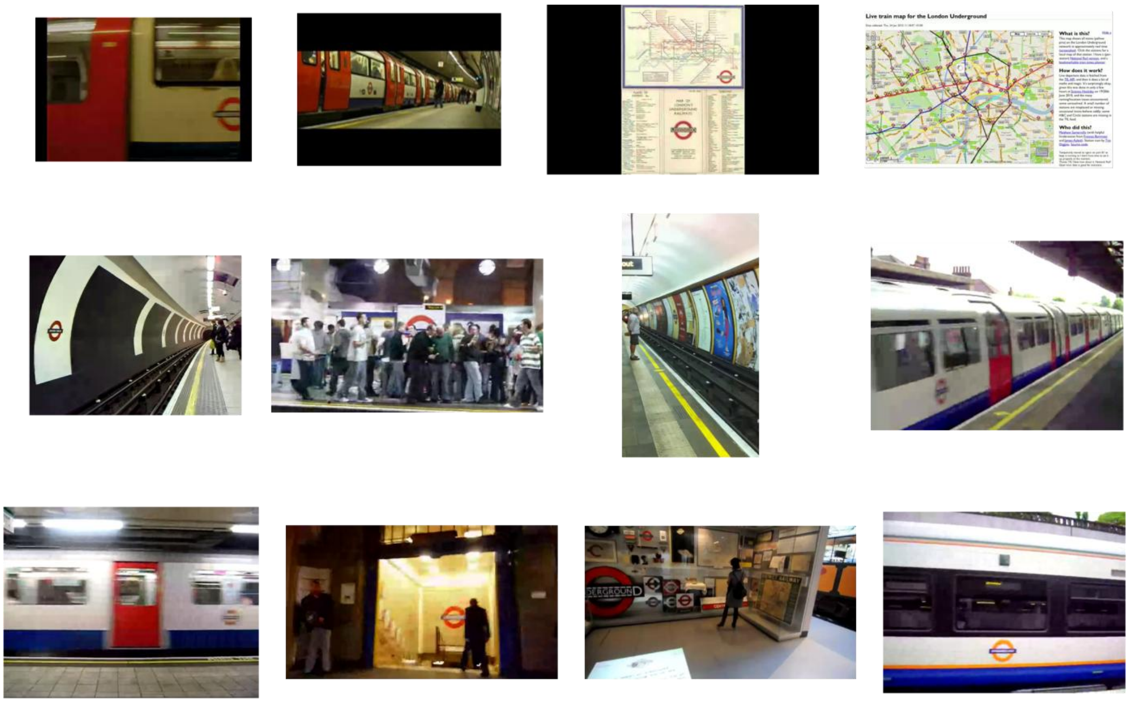


Figure 1.2. Multiple instances of the object “London Underground logo” within different video clips.

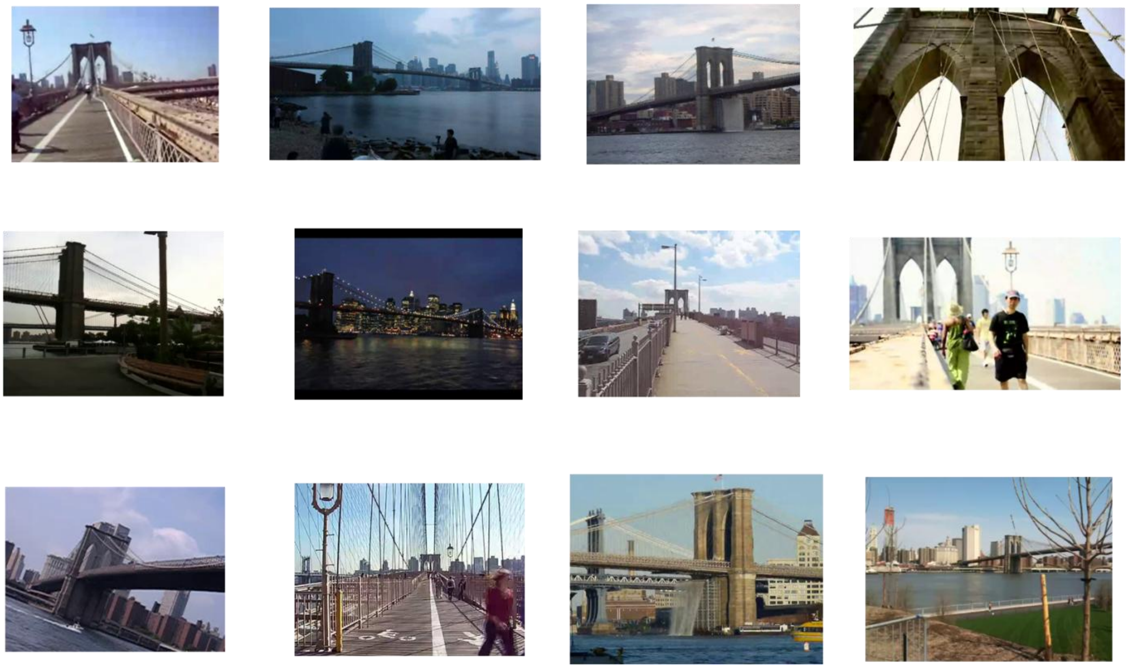


Figure 1.3. Multiple instances of the object “Brooklyn bridge tower” within different video clips.

The methodological framework proposed, so-called DOOR (*Dynamic Object Oriented Retrieval*) exploits a semi-global image representation obtained with the help of an over-segmentation of image frames. The advantage of region-based approaches comes from the possibility of directly exploiting the connectivity information (*i.e*. adjacency between regions), which can be highly useful in the matching stage. An aggregation mechanism is considered in order to group a set of sub-regions into an object similar to the query, under a global similarity criterion.

However, determining the optimum solution in such a situation proves to be an NP-Hard problem. We propose several optimization techniques for determining near-optimal solutions. The key ingredient of our approach is a dynamic region construction procedure, which makes it possible to regroup different individual regions into a candidate object. A global matching score, which measures the similarity of the candidate object to the given query, needs to be minimized. In order to determine the global optimum of the similarity function, three different types of optimization strategies are proposed, including greedy-based techniques, simulated annealing and Graph Cut methodologies. Different visual representations can be considered, including color, texture and interest points descriptors. Let us also underline that arbitrary segmentation methods can be considered, since our goal is to achieve independency with respect to the adopted segmentation procedure.

The object retrieval framework allows the integration of other different visual descriptors and segmentation methods. In particular, a hybrid region representation, integrating interest point information is also proposed. The optimization in this case is performed with a spectral graph matching method.

1.1.2 The OVIDIUS platform

The second part of this work, introduces a novel on-line video browsing and retrieval platform, so-called **OVIDIUS** (*On-line VIDeo Indexing Universal System*). In contrast with traditional and commercial video retrieval platforms, where video content is treated in a more or less monolithic manner (*i.e.*, with global descriptions associated with the whole document), the proposed approach makes it possible to browse and access video content in a finer, per-segment basis. The hierarchical metadata structure exploits the MPEG-7 approach for structural description of video content. The MPEG-7 description schemes have been here enriched with both semantic and content-based metadata. OVIDIUS integrates the DOOR framework and allows fast video object retrieval.

The developed approach shows all its pertinence within a multi-terminal context and in particular for video access from mobile devices.

1.2 Structure of the manuscript

The rest of the manuscript is organized as follows.

Chapter 2 describes the state of the art in the field of object-based recognition methods. The main families of existing approaches are here identified and described, with principle, advantages and limitations.

A first contribution, concerning a region-based visual representation obtained with the help of existing segmentation techniques is introduced in Chapter 3. The challenge here concerns the various optimization techniques that are required in order to match a visual object, defined as an image sub-part with candidate images from a considered database. A global matching score, involving a color-based quadratic error measure, which makes it possible to evaluate objects described by different numbers of regions, is considered. Four different optimization strategies are here proposed. The first one concerns the greedy and relaxed-greedy methods, which aims at providing a rapid solution. In contrast, the second method introduced adopts a simulated annealing approach, aiming at optimizing the retrieval performances. A final optimization scheme proposed involves a Graph Cut technique, which requires re-visiting the energy functional involved.

Chapter 4 tackles the issue of interest point representations. The various methods proposed in the literature, for both semantic categorization and object identifications, are here described. A detailed analysis of existing interest point detectors, interest point descriptors, and related classification methods is also proposed. Based on this analysis, a reference interest point technique is retained, which includes the most promising approaches at all of the various stages involved. This method is used as a baseline technique for our experiments, in order to compare the various approaches proposed. An additional contribution is proposed in this chapter, which concern a query extension mechanism. A multi-modal principle is here exploited: starting from a set of textual tags supposed to be available, a set of representative images, gathered from general public image repositories (*e.g.*, Flickr) is determined. Such images are then exploited to perform visual queries in order to retrieve relevant video content.

Chapter 5 introduces a novel, hybrid approach, which integrates both a region-based representation and a set of interest points. The search and retrieve functionalities are achieved with the help of a spectral graph matching techniques, under a global similarity measure, integrating both color and interest-points descriptors.

Chapter 6 presents and analyzes the various experimental results obtained. Experiments have been carried out on four different datasets, including the Raymond cartoon corpus, and the TRECVID 2010; 2011; and 2012 natural video datasets. The proposed methods are here evaluated under different experimental settings, with various color spaces and segmentation methods.

In the second part of this thesis, we detail our contribution related to the web-based indexing platform.

Chapter 7 proposes an overview of the state of the art. The existing systems are here described, including both desktop-dedicated and mobile platforms, with functionalities supported and related interaction capabilities.

Chapter 8 introduces the OVIDIUS platform. The focus is here put on the modular, distributed architecture proposed, which makes it possible to deploy the system on various terminals, independently of the exploitation systems involved. The technological choices related to video players, communication protocols, interaction capabilities and supported functionalities are described in detail. Several use case scenarios are also introduced in this chapter.

Finally, Chapter 9 concludes this manuscript and opens some perspectives of future work.

PART I. The DOOR framework: Dynamic Object Oriented Retrieval

2. Object-based retrieval: state of the art

Abstract: *In this chapter we present a general review of the state of the art in object recognition and retrieval techniques. The techniques are classified into two major families, according to their unit of description considered for representing the object. The first one is based on interest points, while the second involves a segmented region-based representation. The main principles of the most representative techniques are described along with an analysis of their advantages and limitations. We conclude this chapter with a discussion about the main challenges in object instance retrieval and identify some directions of improvement.*

Keywords: *object retrieval, region-based representation, interest point representation, multiple instance detection, object structure, geometric information.*

Résumé: *Dans ce chapitre nous présentons une revue générale de l'état de l'art sur la reconnaissance des objets et sur les techniques de recherche et indexation. Les techniques sont classées en deux grandes familles en fonction de leur unité de description considérée pour la représentation de l'objet. La première est basée sur des points d'intérêt, tandis que la seconde implique une représentation basée sur des régions segmentées. Les principes de fonctionnement des techniques les plus représentatifs sont décrits avec une analyse de leurs avantages et leurs limites. Nous concluons ce chapitre par une discussion sur les principaux défis dans la recherche des instances d'objets et nous identifions quelques directions d'amélioration.*

Mots clés: *recuperation d'objets, représentation basée sur regions, représentation basée sur points d'intérêt,structure de l'objet, informations sur la géométrie.*

2.1 Introduction

Video object retrieval is among the most challenging tasks in the field of computer vision and multimedia indexing. The fundamental problem to be addressed and solved is how to perform efficiently a *partial matching* between a visual object, defined as a sub-part of an image/video frame, and a given image. Within this context, the object modeling and description is a fundamental issue that needs to be considered appropriately. The video object retrieval methodologies borrow from both object recognition and image retrieval techniques.

In the field of object recognition, we can identify two different types of methods, each with its specific objectives. They concern: (1) semantic category identification and (2) object instance detection.

In the case of category identification methods, the objective is to determine automatically the semantic category/concept associated with different instances of objects present in an image/video scene. This process is illustrated in Figure 2.1, in the case of three semantic classes corresponding to cars, giraffes and chairs. Let us underline that, for a given category, different objects belonging to the given class can appear. In addition, a same object can be represented in various postures. For these reasons, the intra-class variability, in terms of visual appearances, can be highly important.

Existing solutions are based on a learning process, and involve various visual features, with supervised or semi-supervised classifiers. Such an approach requires the availability of a *ground truth* dataset, *i.e.*, as set of images for which the corresponding categories are known. Whatever the visual features/descriptors used for image representation and classification techniques involved, the quality of the considered ground truth is determinant for achieving successful category recognition. Notably, in order to deal with the issue of intra-class variability and to ensure good generalization capabilities, the ground truth should include, for each category considered, a rich variety of object instances, with different visual appearances.

As an example, let us mention the ground truth used in the ImageCLEF visual concept detection task [IClef 2011, IClef 2011] and TRECVID semantic indexing task [Smeaton 2006]. In the ImageClef tasks the dataset is composed of 1 million Flickr images, namely the MIR Flickr collection [MIR 2012] and the ground truth consists of 25000 manually annotated images, for a set of 90 to 100 concepts. For TRECVID's Semantic Indexing Task the test dataset consists of 600 hours of content (approximately 200 hours for training) and the certain number of concepts is selected each year from a list of 500 concepts mostly derived from the LSCOM ontology [LSCOM 2006].

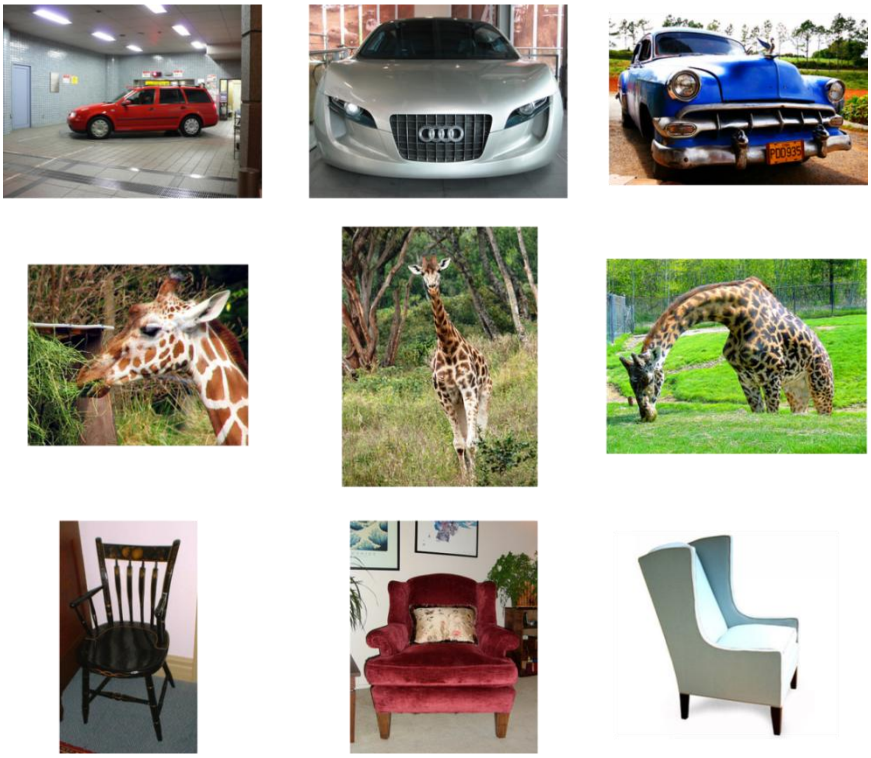


Figure 2.1. Semantic category examples: car, giraffe, chair.

In the last years an increasing number of solutions have provided a variety of interesting results for concept detection and object categorization in videos [Snoek 2008b]. For the TRECVID tasks, up to 500 concepts can be detected. Hauptmann *et al.* [Hauptmann 2007] estimate that a concept lexicon should be sized to at least 5000 items in order to obtain useful descriptions.

However, a major drawback of the learning-based category recognition methods relates to the strong dependency of the results on the considered ground truth. In addition, many of the concepts employed are rather general (*e.g.*, vegetation, outdoor) or vague (*e.g.*, city life, euphoric). Even used in combination, such concepts can hardly describe a specific object, whose instances need to be retrieved.

Let us now state the principle of the second family of approaches, which concerns the object instance detection. In contrast to the semantic concept detection, in this case, the objective is to identify various instances of a *same* object in different scenes and contexts. This principle is illustrated in Figure 2.2, where different instances corresponding to the Statue of Liberty, a van with specific painting or the swimmer Michael Phelps are presented.

As illustrated in Figure 2.2, here again we encounter the problem of variability, a given object being presented in various postures, from different angles of view, and on different backgrounds. However, the main difference with respect to semantic identification approaches is that, in this case, the objective is to detect a given, unique object instead of classes of objects. Usually, the recognition is performed starting from a single view of the target object. The recognition process involves some partial matching techniques, which require adequate, semi-global feature representations.

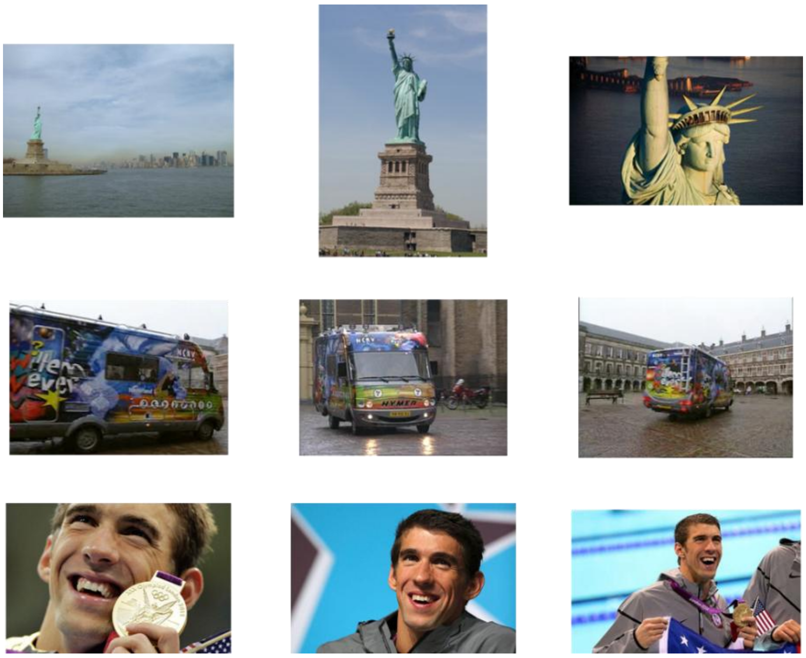


Figure 2.2. Different object instances: Statue of Liberty, painted van, Michael Phelps.

Concerning the image retrieval techniques, most of the pioneering recognition approaches [Lowe 1985, Nevatia 1977] were *model-based*. An object model, such as a 3D representation, is here considered in order to inject some strong *a priori* knowledge related to the object of interest. The task then consisted in developing some 2D/3D matching strategies, in order to perform localization and pose estimation from a single view. An outcome of such approaches concerns the various viewpoint independent representations of generic 3D shapes, as well as the formal, rule-based geometric reasoning methods that have been proposed.

While model-based vision systems represent an important conceptual milestone in the field of object recognition, their applicability in practice is severely limited by their reliance on relatively weak and uninformative image features such as line and curve segments. In addition, they lack of flexibility when modeling non-parametric deformations. For these reasons, a different recognition paradigm emerged, namely the appearance-based recognition. Here, instead of using precise 3D geometric descriptions, a statistical model of the 2D object appearance is proposed, leveraging on discriminative image features. Early work on appearance-based object recognition has mostly involved *global* image descriptions. Thus, the majority of the early methods [Niblack 1993, Schiele 2000] are globally characterizing the entire image with the help of color or texture histograms. The main drawback of such methods is their lack of robustness to clutter and occlusion. For this reason, global recognition methods have been substituted over the last decade by *part-based* methods that seek to identify statistically or structurally significant object patches that can capture salient appearance information.

Within this context, one of the first questions to be solved is the following: how can we define and describe in a consistent and reproducible manner the relevant parts of an image? Let us note that in the case of model-based approaches, a part is defined as a 3D geometric primitive. In contrast, appearance-based approaches have adopted a much more flexible notion of an image part, which can be defined in various manners and involve different visual features.

As a first example, let us mention the approach proposed in [Schneiderman 2004]. Here, authors define image parts in the wavelet transformed domain as groups of highly correlated wavelet coefficients. However, the majority of existing approaches operate directly in the initial image domain. They involve image fragments or segments obtained with the help of sampling procedures, [Ullman 2001, Mahamud 2003,Yang 2007], corner-like interest points [Agarwal 2002] or scale-invariant salient regions [Fergus 2003].

During the last decade, the field of object-based content retrieval has been considerably enriched. Consistent improvements have been brought in terms of object representation, detection and description of object parts or entities, descriptor clustering, image and object matching strategies, learning and classification. The most popular methods involve the Bag-of-Words representation [Sivic 2003] and the discriminatively trained deformable part-based models [Felzenszwalb 2008]. Such methods have been mostly used for object category identification by employing supervised learning models and algorithms such as Support Vector Machines (SVM) [Boser 1992, Cortes 1995]. While impressively effective, such methods require the use of an off-line training phase, which is dependent of the considered training set and of the pre-defined categories.

Recently, an increasing interest has been directed towards the object instance search with reduced positive example instances. This relatively recent topic of research has been considered in the TRECVID 2010 [Smeaton 2006] evaluation campaign, under the so-called *Instance Search Task*, launched for the first time in 2010 and continued since within the framework of TRECVID 2011 and 2012 editions.

Related work includes two major families of approaches depending on their unit of description considered for object representation. The first category of approaches considers representations based on sets of interest points, while and the second one relies on region-based representations. Let us begin the analysis of the state of the art with the interest point-based representations.

2.2 Interest-point-based representations

Interest points are among the most popular tools for object recognition and classification for both images and videos and are extensively used in a variety of computer vision applications, such as object tracking [Gabriel 2005], motion estimation [Torr 2000], image matching [Chum 2008b, Jégou 2008, Jégou 2010], scene classification [Snoek 2008], image understanding [Fergus 2007, Leibe 2008], stereoscopic correction [Matas 2002] and disparity field estimation [Wills 2006].

Early approaches for object retrieval using interest points have been developed by Sivic and Zisserman in their Video Google system [Sivic 2003]. Inspired by text retrieval techniques, the bag-of-words (BoW) representation is obtained by extracting and describing scale invariant patches from images, clustering them into “visual words”, quantizing them to the “visual words” and obtaining a histogram of occurrences of the visual words for each image. In this case, SIFT descriptors [Lowe 2004] are extracted from video keyframes with the help of two types of overlapping image patches: *Harris-affine* [Mikolajczyk 2002] regions and so-called *Maximally Stable Extreme Regions* (MSER)

[Matas 2002]. The BoW is used for achieving fast and efficient retrieval of objects interactively selected by the user with the help of a bounding box. However, since the BoW is practically an order-less collection of visual words and their relative frequencies of occurrence, all the spatial information describing the geometric relative position of the visual words in the images is discarded.

The lack of spatial information represents the major drawback of bag-of-words inspired methods. Numerous researches have investigated various ways for improving this aspect. In a general manner, the objective is to improve the power of the representation by injecting some spatial localization information in the visual representation. In this respect, several approaches employ the *RANdom SAmple Consensus* (RANSAC) algorithm [Fischler 1981]. RANSAC is a robust method for model fitting to noisy data with outliers. The main idea is to randomly sample a minimal set of correspondences, and compute the aligning geometric transformation. Because of the associated computational complexity, RANSAC is typically applied in a post-search process, solely on a set of top retrieved results, which are re-ranked according to their spatial consistency. Different RANSAC improvements and variants have been proposed in the last years [Chum 2003, Chum 2005, Philbin 2007, Raguram 2008, Ni 2009].

Another family of solutions introduces the spatial information directly in the BoW representation or in the matching process. Lazebnik *et al.* [Lazebnik 2006] proposed a spatial pyramid matching in order to encode and partition the image into sub-regions at different levels of detail, from coarse to fine. The image can be then represented with multiple local histograms concatenated in a single global image histogram, where each local histogram corresponds to an image sub-region. The local histograms are weighted in such a manner that matches identified in larger cells are penalized as they involve increasingly dissimilar features. Vedaldi *et al.* [Vedaldi 2009] propose the use of dense and sparse visual words at different levels of spatial organization. Wu *et al.* [Wu 2009] leverage on the high scale representation of the MSER regions to group Harris-affine regions. Harris-affine regions having the centroids inside the same MSER elliptical shape are grouped into clusters and a dedicated BoW vector is computed for each such cluster. In this manner, the BoW includes implicitly regions which are located in a relatively close neighborhood.

Other solutions act in the local feature detection phase. Instead of using scale or affine invariant region detectors, such methods perform a dense sampling of the image with a regular grid (possibly defined over a range of scales) [Fei-Fei 2005, Jurie 2005, Tuytelaars 2007, Tola 2008]. In this manner, the neighborhood of the interest points can be defined on the grid. In addition, less textured image patches that would have been omitted by the local feature detectors can be now described. Let us note that in this case, the scale invariance can be ensured by performing the sampling at different scales. In other words, multiple grids need to be defined for each image. Such approaches prove to be particularly useful for stereo matching purposes [Tola 2008]. On the downside, dense sampling cannot reach the same level of repeatability as obtained with interest points, unless sampling is performed extremely dense. However, in this case the number of features becomes unacceptably large. The principle is illustrated in Figure 2.3.

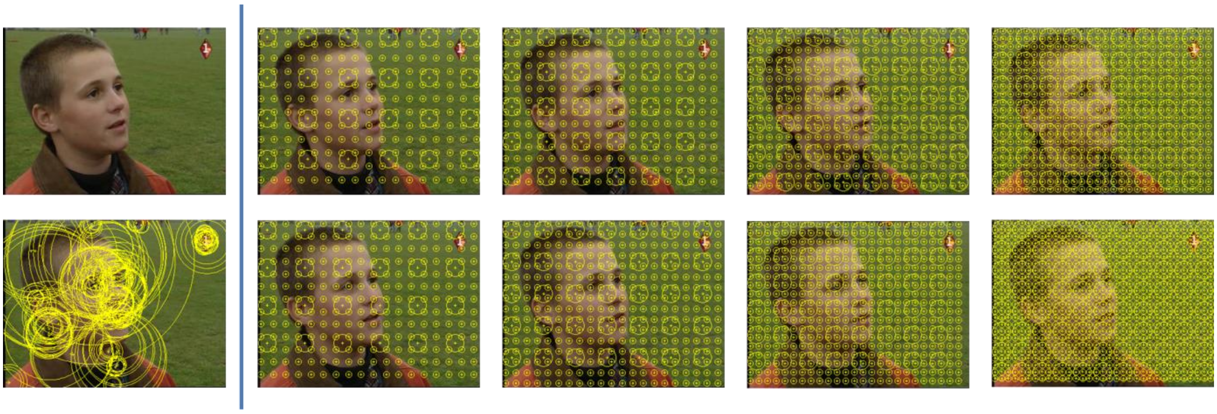


Figure 2.3. Dense sampling. In the first column we illustrate the original image and the regions detected by the Hessian-affine detector. In the rest of the columns dense sampling examples with various grid settings are illustrated. Notice how, for coarse grids many textured regions are missed, while for fine grids every patch of the image is sampled.

In order to combine the advantages of both schemes, Tuytelaars [Tuytelaars 2010] has recently introduced the so-called dense interest points method. Starting from densely sampled image patches, the author applies for each detected patch a local optimization of the position and scale within a bounded search area. The outcome of this process is a set of interest points defined on a semi-regular grid, densely covering the entire image as is the case of dense sampling, but with repeatability properties closer to those of standard interest points. Thus, the newly obtained interest points inherit from dense sampling the simple spatial relation between points. This principle is illustrated in Figure 2.4.

In a similar framework, Ferrari *et al.* [Ferrari 2006] propose a robust method for simultaneous object recognition and segmentation from two images. Starting from a set of initial matches, the method gradually explores the surrounding image areas; constructing in a recursive manner a set of additional matching regions. In this way, the object is covered with a set of matched regions. A final integration stage measures the consistency of configurations of groups of regions associated to different model views. While the method is robust to clutter, occlusion and viewpoint changes, its computational cost makes it prohibitive in an indexing, search and retrieval framework.

In a different setting and aiming to identify recurring objects in large collections of tourist photographs, Philbin *et al.* [Philbin 2011] construct a matching graph, where each image represents a node. Images presenting a large number of matched visual words are inter-connected with a weighted edge. In this manner, images containing the same objects can be identified as connected components in the graph. Objects are discovered accurately with the help of a Latent Dirichlet Association algorithm [Blei 2002] performed on the graph. The method presents interesting performances on touristic landmarks image datasets. However, the construction of the image graph is expensive as each image from the dataset is queried over the entire dataset in order to identify its most similar images. In addition, whenever a new set of images is added to the dataset, the graph needs to be reconstructed and each image is queried again.

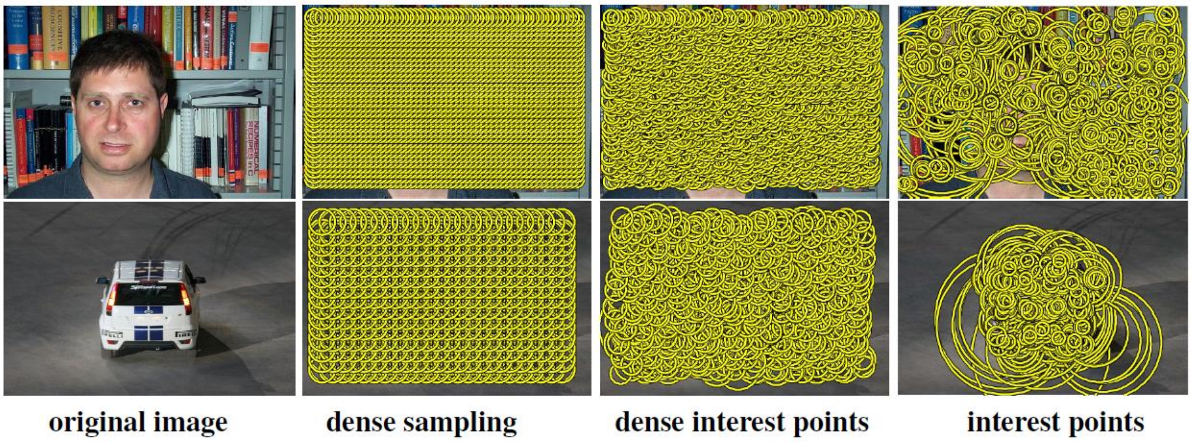


Figure 2.4. Dense interest points (middle) form a hybrid scheme between dense sampling on a regular grid (left) and interest point detection (right).

Abandoning the BoW paradigm and starting from the method proposed in [Jiang 2007], Li *et al*. [Li 2010] group points of interest in graphs by using Delaunay triangulations. They introduce different geometric constraints with the goal of characterizing the geometric properties of the neighborhood of each node. Moreover, each node is represented as an affine combination of its neighboring nodes, whose weights can be determined with a least squares fitting method. The obtained graph model is then matched at different scenes using linear programming techniques, in order to determine the object of interest.

Cho *et al.* [Cho 2009] also employ a graph matching technique. Here, the graph nodes represent pairs of points from the two images to be matched and the edges connect agreeing pairs. A hierarchical agglomerative clustering algorithm is employed to identify the strongest nodes in the graph (*i.e.*, best matches) leveraging on both photometric and consistency of local features.

Duchenne *et al*. [Duchenne 2009] use higher-order constraints instead of unary or pairwise ones between nodes, which result in a tensor representation capturing the affinity between tuples of features. The resulting energy function is optimized using the multi-dimensional power iteration method [Golub 1996], computing quickly the main eigenvector of the affinity matrix associated to the pairs of points from the two images.

Such methods are highly powerful, being in particular robust to strong deformations of the objects. Their main limitation concerns the computational complexity, which makes them inappropriate for search and retrieval applications in large repositories.

A different method, exploiting the Histogram of Oriented Gradients (HOG) [Dalal 2005] has been proposed in [Felzenszwalb 2008]. The object models consist of hierarchical structures of rectangular groups of HOGs, associated with a set of sliding windows, specified at different scales. The method achieved the top performances in the Pascal VOC challenge in the past years [Everingham 2010]. While providing extra spatial information and robustness to object deformations, a significant number of sliding windows verifications is required for retrieving object candidates. This has a strong impact on the computational time needed for the matching process. In addition, many objects in practice cannot be fully covered by the considered bounding box representation, which leads to erroneous results.

In a general manner, interest-point-based representations provide useful solutions for both instance object search and semantic categorization purposes. The main limitation to be overcome is related to the issue of spatial information, which is not straightforward to handle in pure interest point-based representations. In addition the interest-point representation decrease dramatically in performance in the cases of texture-less objects when few interest points are detected and used for recognition.

The second family of object retrieval approaches exploits a region-based representation, obtained with the help of image segmentation techniques.

2.3 Region-based representations

Inspired by the Bag-of-Words method, Gu *et al.* [Gu 2009] propose the so-called Bag-of-Regions technique. Here, a set of regions is obtained with the help of the hierarchical segmentation algorithm described in [Arbelaez 2009]. A set of weights, representing the probability of belonging to an object category and corresponding to multiple descriptors (contour shape, edge histogram, color histogram and textons) are assigned to each region.

In [Chevalier 2007] authors construct region adjacency graphs of pre-segmented objects and retrieve similar objects with the help of a new graph matching method based on an improvement of the relaxation labeling technique [Hummel 1983].

Starting from the assumption that no segmentation method can be perfect, in terms of identified objects in a given scene, in [Pantofaru 2006, Pantofaru 2008] authors exploit multiple image segmentations with different parameters. The various segmentation results obtained are then combined and the resulting regions are described with both SIFT and color descriptors, with the help of the so-called Region Context Features (RCF). RCFs rely on both segmented regions and local feature patches. For each segmented region, the visual words from its neighborhood are counted into a histogram of occurrences. The histograms of all regions are then clustered into a vocabulary of RCFs and the regions are assigned to their nearest RCF.

In [Gorisse 2010], the individual video frames are divided into rectangular cells forming a grid. Different visual descriptors, such as HSV histogram, MPEG-7 Edge Histogram, wavelet histogram, are associated to each cell. The descriptors of each cell are clustered into visual vocabularies (*i.e.*, one for each descriptor) similarly with the BoW framework [Sivic 2003]. The cells are then quantized and a BoW vector is generated for each frame. This approach is however restricted to rectangular queries. In [Vieux 2012], a Bag of Regions representation is computed from segmented regions and low level descriptors. Regions are generated with the efficient graph-based [Felzenszwalb 2004] and TurboPixels [Levinshtein 2009] segmentation methods, and described with HSV color histograms and Local Binary Patterns [Ojala 2002]. Different vocabularies are built for every combination of regions and descriptors, while the results for all runs of the same query are then aggregated in a single ranked list.

Different state-of-the-art object retrieval techniques rely on an exhaustive search over the image to determine the optimal candidate object position within a sliding window. Thus, in [Sande 2011], authors employ segmented regions to define possible object entities. This selective search reduces significantly the number of object locations to consider. The method starts from an over-segmentation obtained with the graph-based segmentation method from [Felzenszwalb 2004]. A greedy algorithm is used in order to iteratively group the two most similar regions together and compute the similarities between the new regions and their neighbors (*i.e.*, texture and size similarity). The process leads to the construction of a hierarchical structure of regions. Object locations can be then identified by considering all segments through the hierarchy. The newly defined object entities can be then described by popular object recognition techniques such as BoW with SIFT descriptors densely extracted from each pixel on a single scale.

A similar and complementary approach has been proposed in [Malisiewicz 2007] where authors study the use of multiple segmentations as spatial support to accurately define objects from the ground truth of the Pascal challenge database [Pascal 2006], as an alternative to bounding boxes.

A region-based method using graphs for discovering object instances from images of daily living has been proposed in [Kang 2011]. Regions obtained with two different segmentation methods (*i.e.*, efficient graph based segmentation [Felzenszwalb 2004] and active segmentation with fixation [Mishra 2009]) are compared with the help of color histograms, SIFT and shape descriptors. They are then grouped in a graph of regions according to their similarity. The similarity between regions is adaptive with respect to the degree of texturing of the regions. Thus, textured regions are represented with quantized SIFT features, while for less textured regions a color-based representation is proposed. The pairs of regions are further verified in a shape similarity step. Regions belonging to the instances of the same objects are thus determined as connected components in a graph containing all the regions in the dataset and connecting the similar regions (Figure 2.5). Co-occurring segments are used for composing object models consisting of multiple object parts.

Let us also mention the approach introduced in [Kim 2011]. Here, a different region-based representation is proposed. The image is represented as a dense map of (overlapping) regions. Starting from multiple overlapping segmentations [Arbelaez 2009], a distance transform is computed with respect to the boundary of each segment. Then, each segment is divided into a set of cells distributed over a regular grid. For each cell, an "element" is sampled at the location of maximal distance transform value within the cell. The radius of the element is set to the corresponding maximal distance value. The elements are then linked together under spatial (Euclidean distance) and similarity (contour strength [Maire 2008]) constraints. An element and its linked elements constitute a boundary preserving local region, which is described with Pyramids of Histograms of Oriented Gradients [Bosch 2007]. The determined regions can then be matched individually according to this descriptor.

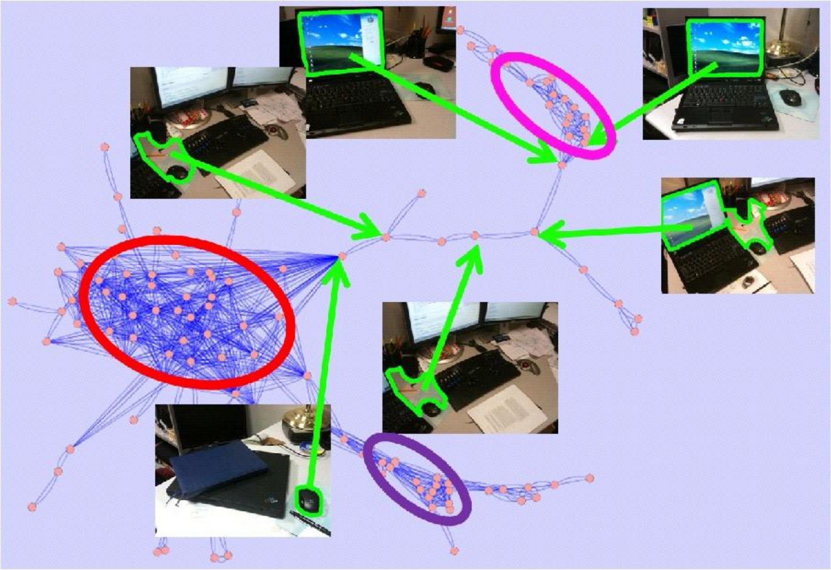


Figure 2.5. Graph of regions and discovery of regions belonging to the same object as connected components [Kang 2011].

Vijayanarasimhan and Grauman [Vijayanarasimhan 2011] build a region adjacency graph using superpixels and describe each region with SURF points [Bay 2008] and shape descriptors [Gu 2009]. Objects or object parts are identified as connected subgraphs of the adjacency graph with an improved branch and bound scheme [Lampert 2008], which maximizes the score of a classifier. The similarity scores are computed by summing the responses of the classifier for each identified subgraph. Recently, Duchenne *et al.* [Duchenne 2011] proposed a different, graph of regions representation. Here, authors formulate the object recognition problem as a multi-label MRF optimization. A set of regions is extracted from a coarse grid and represented with the help of a graph structure. A MRF-like energy optimization is considered in order to take into account the similarity of pairs of regions. The energy functional involved is optimized with an extension [Ishikawa 2003] of the GraphCut technique [Boykov 2001], able to solve multi-labeling problems. As the graphs to be matched play asymmetric roles in this energy function, the similarity between two graphs is computed with a kernel defined over the energies between the two graphs. This kernel is then employed for training a support vector machine classifier to retrieve different object instances. The approach is mostly suitable for object retrieval in images containing solely a unique object. Another limitation is related to the scale invariance issues, which are treated in a more or less heuristic manner.

One of the major advantages of the region-based approaches comes from the possibility of taking into account the spatial information in a natural way, by exploiting the adjacency relations between the considered regions. However, the challenge that needs to be solved is related to the optimization procedures involved, which need to exploit such information in an effective and computationally efficient manner.

2.4 Discussion

The analysis of the state of the art shows that the task of retrieving different instances of the same object is still a challenging issue of research. Most of the existing approaches are currently focused on identifying generic categories of objects. The choice of descriptors and segmentations specific to each category is then performed during the off-line learning stage. While solving the issue of how to match two images or two object entities, such approaches transfer the problem to the off-line learning stage. However, the problem of retrieving different instances of specific objects becomes even more difficult, as distinctive ground truth sets need to be generated for each such object in order to train the classifier.

Concerning the identification of the location of the object candidates, many approaches rely on an exhaustive sliding window search, making it increasingly difficult to retrieve new objects in an unsupervised manner. We have seen that segmented regions can provide good spatial support and that regions and group of regions can identify reliably objects or objects parts [Sande 2011], thus greatly reducing the number of sliding windows to consider. However, the number of possible object configurations to check is still elevated.

Large scale search systems can be effectively developed with interest points by employing the BoW framework. The main difficulty is related with the lack of relative geometric information between interest points. RANSAC based approaches solve this problem partially, by identifying consistent groups of points from two matched images. However, such consistency verifications compute homographies only by using the coordinates of the matched points and usually identify correctly the consistent matches for limited viewpoint variations. The insertion of spatial information by dividing the image at different scales [Lazebnik 2006] is difficult to extend to large datasets and vocabularies (e.*g.*, a spatial pyramid on 2 layers with 4 cells in the second layer extends the size of the BoW vector 5 times).

Many researchers have concluded that the inclusion of spatial information has proven to be essential in improving the retrieval results for interest point approaches. The most performing approaches formulate the problem of interest point matching of two images or objects as a graph matching problem. Such methods show superior robustness to many object transformations and deformations. The price to pay is related to the expensive computational cost. For this reason, no such technique has been yet proposed for large scale search yet. However, many of the findings in this field could inspire a more light-weight object representation robust to multiple variations of object pose, scale and small deformations.

An important drawback of the interest points approaches is that less textured images with a low number of interest points become practically non-retrievable by the search engine or non-matchable by interest point graph matching techniques. Usually such problems are tackled with region-based representations relying on segmented regions and their color information. Most of the approaches do not define a consistent object model, but represent it as a pool of regions collected from different images where the object of interest occurs. The pertinence of each region for the object class is computed by a classifier which extracts useful information from a set of manually annotated images. The matching between images and objects is performed only at the level of individual regions and the matching of groups of regions it still in its early days [Vijayanarasimhan 2011].

Only a few approaches [Chevalier 2007, Duchenne 2011, Vijayanarasimhan 2011] consider objects as consistent groups of regions; usually formalized as graph of regions over a grid or region adjacency graphs. Graph-based structures provide increased flexibility to different object poses, viewpoints, occlusions and deformations and leverage on multiple studies on combinatorial optimization to identify groups of relevant regions from each object. Yet, graph-matching is among the most computationally expensive approaches and new light weight variants should be proposed in order to use them for large scale search.

Finally, recent researches have shown that the recall of retrieval methods can be improved by leveraging on the top retrieved results. The query can be enriched/expanded with the descriptors of the top results and a new query is issued in order to retrieve more positive results [Chum 2007]. In a similar manner the top results can be used to train a linear classifier which is then for finding new object instances [Arandjelovic 2012]. The drawback of such methods is that they rely on the results retrieved from a first query and if the top results are false the query expansion fails to improve the recall. An alternative mechanism that would ensure that the expanded query consists of positive object instances should be developed for improvements in recall.

In our work, we propose an object retrieval framework relying on a region-based representation. Frames are represented as region adjacency graphs and objects are identified are sub-graphs according to a global similarity criterion with the query object. We propose multiple optimization methods to identify the most relevant region configurations for an object given a query model. In addition, we propose a hybrid object representation relying on both interest points (*i.e.*, affine co-variant regions) and segmented regions. For each representation we propose a matching technique and an optimization strategy. Our aim is to include pertinent relative geometric information of groups of regions in the object representations and to localize the object without the use of exhaustive sliding windows.

Query expansion techniques improve significantly the recall of the search engines but are limited by the quality of the first retrieved results. We propose a technique that overcomes this limitation by leveraging on general public search engines for retrieving positive instances of query objects to be used for building a novel query. The most representative instances are chosen after a verification stage and a new query is generated with the description of these images. We show how a visual query can be generated by using textual descriptions of the object to be retrieved in a video repository.

3. Region-based representation

Abstract: *In this chapter, we propose a region-based representation for objects. Multiple segmentation methods (MeanShift, Superpixels, Efficient Graph Based Segmentation) that are suitable for such an approach are first briefly recalled. The segmented regions are described with the help of an extended version of the MPEG-7 dominant color descriptor, able to support an arbitrary number of colors. The segments adjacency information is also retained and stored with the help of an adjacency graph. The identification of groups of regions corresponding to an object instance that is similar with an object model is an NP-hard problem. We propose a global similarity score for measuring the correspondences between two groups of regions and employ it as an energy functional to be minimized over multiple region configurations. In order to identify the global minimum of the energy, we propose and describe four different strategies, namely Greedy, Relaxed Greedy, Simulated Annealing and GraphCut.*

Keywords: *energy minimization, region based representation, Greedy, Simulated Annealing, GraphCut, MPEG-7 DCD, segmentation, MeanShift, SuperPixels, efficient graph based segmentation.*

Résumé: *Dans ce chapitre nous proposons une représentation une représentation d'objet basée sur des régions segmentées. Plusieurs méthodes de segmentation (e.g., MeanShift, Superpixels, Segmentation Efficace Basée sur Graphes) qui se prêtent à une telle approche sont d'abord brièvement rappelés. Les régions sont décrites par une version étendue du descripteur MPEG-7 Couleur Dominante, capable de supporter un nombre arbitraire de couleurs. L'information d'adjacence des segments est également exploitée et stocké à l'aide d'un graphe d'adjacence. L'identification des groupes de régions correspondant à une instance d'objet qui est semblable à un modèle d'objet est un problème NP-difficile. Nous proposons un score de similarité globale pour mesurer les correspondances entre deux groupes de régions et nous l'en utilisons comme une énergie fonctionnelle à minimiser par rapport aux multiples configurations des régions. Afin d'identifier le minimum global de l'énergie, nous proposons et décrivons quatre stratégies différentes, à savoir Greedy, Greedy Détendu, Recuit Simulé et GraphCut.*

Mots clés: *minimization d'énergie, representation basée sur régions, Greedy, recuit simulé, GraphCut, MPEG-7 DCD, segmentation, MeanShift ; SuperPixels, segmentations basée sur graphe.*

The principle of the proposed Dynamic Object-Oriented Retrieval (DOOR) approach consists of representing an image as a set of regions/segments, obtained with an arbitrary segmentation technique. As we cannot expect to dispose of an “ideal” segmentation technique, able to provide a precise region of support for each object present in a video scene, the goal here is to exploit an over-segmentation of each image into segments. The problem then can be formulated as follows: given a query object *Q*, represented as a set of regions homogeneous with respect to a given criterion, together with their adjacency information, and a candidate image *I*, also decomposed into a set of segments on the same basis, how can we determine the sub-set of regions in image *I* that optimally fits (with respect to a visual similarity measure) the query object Q?

Within this framework, each set of segments under consideration is supposed to be described by some visual descriptors. In our case, we have adopted a color-based description, based on the MPEG-7 Dominant Color Descriptor (DCD) [MPEG 2002].

Concerning the construction of the candidate object, the following strategy has been adopted. In a first stage, we perform a pre-filtering of each candidate image, which aims at eliminating individual regions with far-off colors, based on a color similarity criterion. Let us underline that the objective of this stage is not to directly (and precisely) determine the candidate object, but to roughly restrict the number of candidate regions, by eliminating colors that are highly improbable to belong to the query object. Thus, a rather permissive threshold has to be used. The resulting regions are labeled into connected components. Each connected component is then considered as an initial candidate object to be matched with the query.

Next, each candidate object is iteratively refined by successively removing and/or adding regions until the global matching distance is minimized. An overview of the proposed approach is illustrated in Figure 3.1.

Let us note that an exhaustive search approach which would test all possible configurations is computationally intractable (non-polynomial complexity with respect to the number of regions). In order to obtain a method of reasonable complexity, we have considered the following four optimization approaches:

- a recursive, greedy optimization strategy,
- a relaxed greedy method,
- a simulated annealing-based approach [Kirkpatrick 1983],
- a Graph Cut method [Boykov 2004].

Our approach relies on a region-based image representation, obtained with the help of a segmentation method. The next section recalls the three segmentation methods adopted in our work, with principle and choices of the main parameters involved.

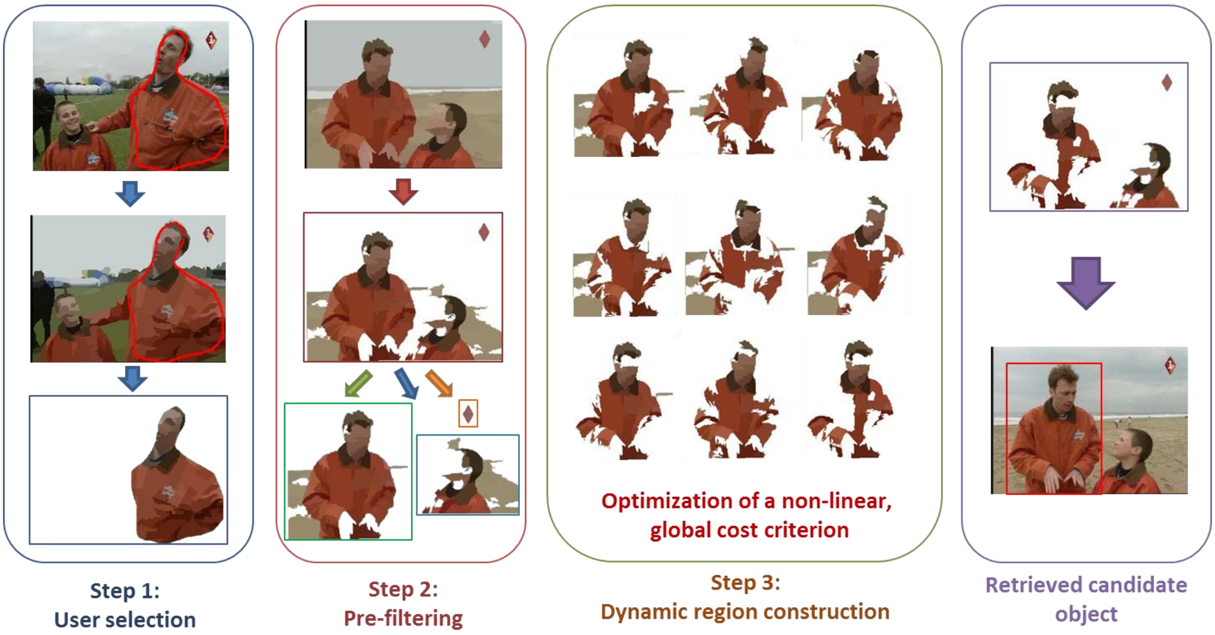

Optimization of a non-linear, global cost criterion
Step 1: User selection
Step 2: Pre-filtering
Step 3: Dynamic region construction
Retrieved candidate object

Figure 3.1. Overview of the DOOR approach: Step1: The user selects an object and the corresponding group of segmented regions is extracted; Step.2: Regions with colors highly different from the query are filtered out. Remaining regions are separated into connected components and each component is considered as a candidate object; Step 3: Different configurations of candidate objects are generated by adding and removing color segments; Step 4: The object with the minimal distance from the query is selected and displayed in its bounding box.

3.1 Adopted segmentation approaches

A first segmentation approach is the well-known MeanShift approach.

3.1.1 MeanShift segmentation

The MeanShift segmentation technique [Comaniciu 2002] relies on a feature space analysis. The technique includes two basic steps: a mean shift filtering of the original image data (in the feature space) and a subsequent clustering of the filtered data points.

The filtering step of the mean shift segmentation algorithm consists of analyzing the probability density function underlying the image data in feature space. The feature space consists of the (*x*, *y*) coordinates each pixel in the image and the (smoothed) pixel color in L*u*v* space (L*, u*, v*). The modes of the probability density function underlying the data in this feature space will correspond to the locations with highest data density, and data points close to these modes can be clustered together to form a segmentation. The mean shift filtering step consists of determining these modes through an iterative kernel density estimation of the gradient of the probability density function. Edge information is also considered, in order to better guide the clustering process. Among the various variants of the Meanshift algorithm, we have adopted the method integrated in the publicly available EDISON system, described in [Christoudias 2002]. Let us note that the algorithm can be parallelized and implemented on GPU, boosting its computational efficiency.

As we can observe, the regions generated by the Mean Shift segmentation accurately follow the edges of the image. The number of regions is not specified, but is instead determined by the bandwidth parameters involved and by the image content. Since our objective is to use the segmented regions as primitives for object construction, we require a relatively high number of segmented regions. Thus, the segmentation parameters (*e.g.*, color range, spatial range, minimum region size) are adjusted in order to perform an over-segmentation of the images. This results in up to 200-300 regions per image, with an average of 160 regions per image (*cf.* Section 6.1.2).

Figure 3.2 and Figure 3.3 illustrates some examples of MeanShift segmentation, for both synthetic (cartoon) and natural images.

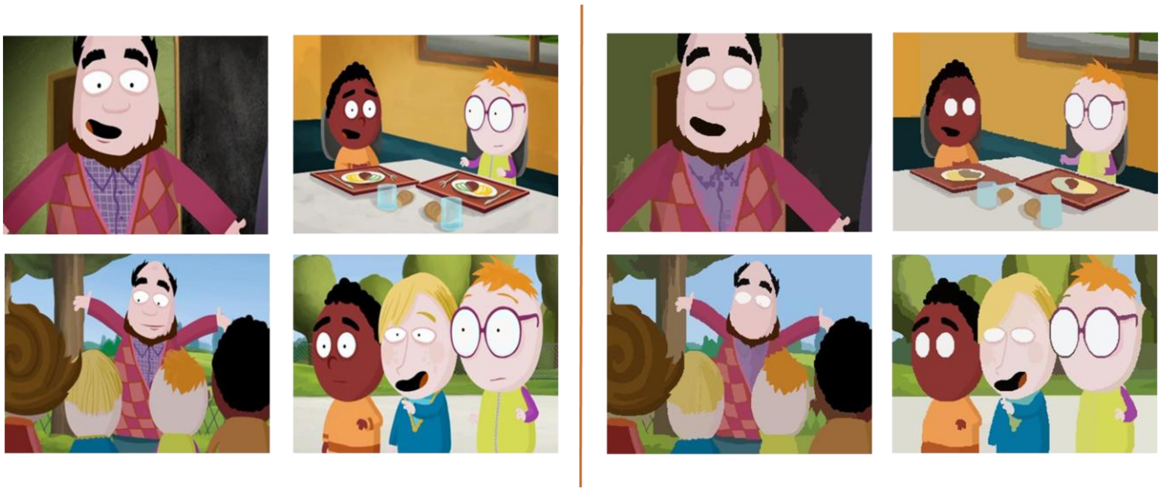


Figure 3.2. Cartoon video frames (left) and their MeanShift segmentations (right). A number of 80-150 segmented regions provide in this case a “recognizable” image. The regions are displayed with their corresponding Dominant Color.

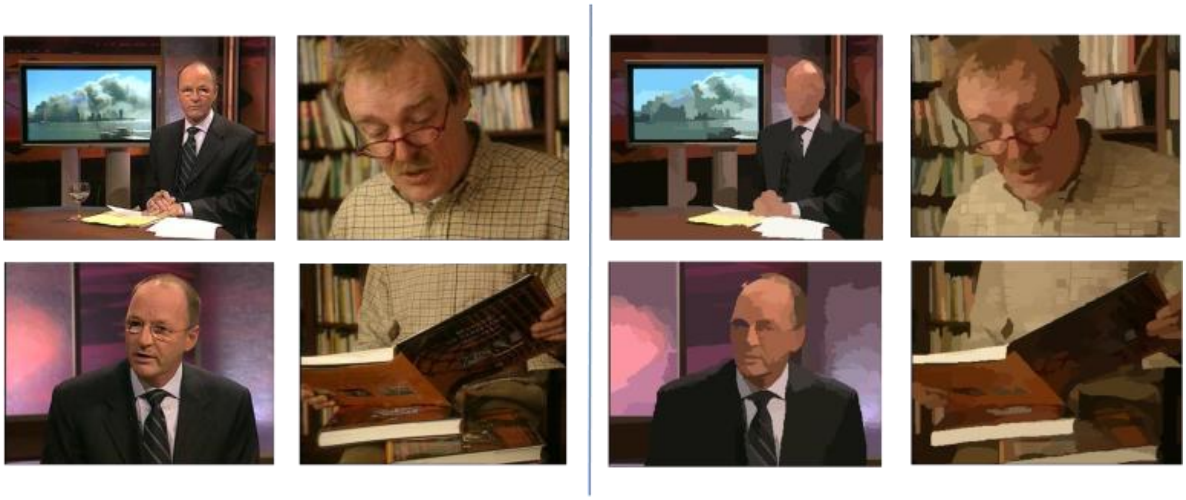


Figure 3.3. Video frames (left) and their MeanShift segmentations (right). A number of 200-300 segmented regions provides in this case a “recognizable” image. The regions are displayed with their corresponding Dominant Color.

In the segmented images the color of each region corresponds to the average value of the colors of all the pixels included in the considered region. We can observe that despite the inherent loss in accuracy, the image content can still be visually recognized from the segmented images, which offers good premises in terms of object recognition capabilities.

Let us finally note that due to its advantageous properties, the MeanShift algorithm is a frequent choice in the field of object-based retrieval [Pantofaru 2006, Malisiewicz 2007, Yang 2007, Pantofaru 2008].

A second segmentation technique retained is presented in the following section.

3.1.2 Efficient graph-based segmentation

The efficient graph-based segmentation (EGBS), introduced by Felzenswalb and Huttenlocher in [Felzenszwalb 2004] uses a variation of a single linkage clustering [Comaniciu 1999] based on dilating points in a parameter space. The authors employ a minimum spanning tree of the data points (pixels) from which any edges with length greater than a given hard threshold are removed. The method eliminates the need for a hard threshold, which is here replaced with a data-dependent term.

Thus, the merging of two components is performed with a variable threshold, unlike MeanShift where the pixels are assigned region labels under fixed thresholds. This adaptive thresholding mechanism allows two components to be merged effectively if the minimum edge connecting them does not have a length greater than the maximum edge in either of the components' minimum spanning trees. The EGBS technique is thus adaptively sensitive to edges in areas of low variability, and less sensitive to them in areas of high variability.

The efficient graph-based segmentation has been successfully employed in vision applications on object recognition [Pantofaru 2008, Kang 2011, Sande 2011, Vieux 2012] for its simplicity and for its segmentation pertinence to object entities, as in many cases objects are segmented into a single region. While a fixed number of regions per image cannot be set, we have tuned the algorithm to return an average of 170 regions with a maximum value of 1000 regions

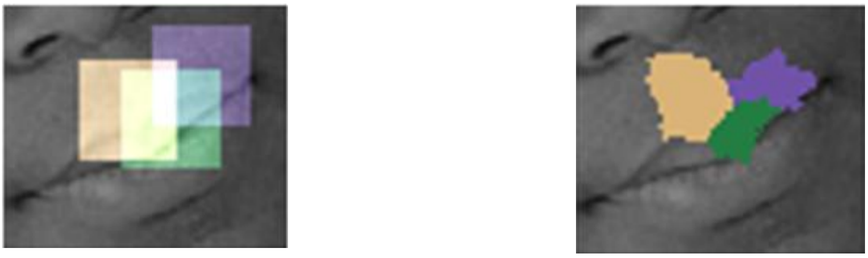
3.1.3 Superpixels

Many vision applications benefit from representing an image as a collection of superpixels [Ren 2003, Malisiewicz 2007, Pantofaru 2008, Fulkerson 2009]. Intuitively, a superpixel is regarded as a perceptually meaningful atomic region of the image. Thus, a superpixel should contain pixels that are similar in a certain feature (*e.g.* color, texture…) and therefore are likely to belong to the same physical world object. Moreover, the size and shape of the superpixels of a given image should be highly similar.

Let us mention that the "superpixel" denomination is often used for regions resulted from popular segmentation methods (*e.g.*, normalized cuts [Shi], MeanShift [Comaniciu 2002], graph based [Felzenszwalb 2004]) when such techniques are tuned such that they lead to an over segmentation of the image. However, when running in superpixel mode, such methods do not follow the boundaries of the objects and the resulting superpixels have irregular shapes and sizes. These drawbacks have been approached by specialized superpixels segmentation algorithms such as normalized cuts superpixels [Ren 2003], TurboPixels [Levinshtein 2009], energy-based superpixels [Veksler 2010].

We have adopted the energy-based superpixels, recently introduced by Veksler *et al.* [Veksler 2010] for their superior performances over Turbopixels in both boundary precision and computation time. The superpixels are obtained with a graph-cut optimization approach [Boykov 2001, Boykov 2004, Kolmogorov 2004] which finds an optimum distribution of a collection of overlapping square patches covering the whole image. Each pixel is covered by several patches, and the task is to assign a pixel to one of them. If two neighboring pixels are assigned to the same patch, there is no penalty. If they belong to different patches, then there is a stitching penalty that is inversely proportional to the intensity difference between the pixels. Intuitively, patches are stitched in such a manner that the seams are encouraged to become aligned with the image edges. The principle is illustrated in Figure 3.4, where three patches are considered (Figure 3.4a). In this example, there is a strong intensity gradient at the level of the lip boundary. Therefore, the cut between patches should be aligned with the lip boundaries (Figure 3.4b). This process of boundary regularization is due to the stitching energy function. A superpixel cannot be too large, *i.e.*, not larger than the considered size of the patch. Small superpixels are also discouraged because they contribute with a higher cost to the stitching energy.

An important advantage of the superpixels is their regularity in size and shape. This makes it possible to specify the approximate number of superpixels for each image. In our approach, superpixels can also be regarded as a quasi-regular grid that provides information about the positioning of the superpixels and groups of superpixels relative to each other and does not cross the object boundaries. In order to test the influence of the number of regions in the object retrieval process, we have considered three levels of granularity for the superpixels segmentation with an average number of 250, 500 and 750 superpixels.

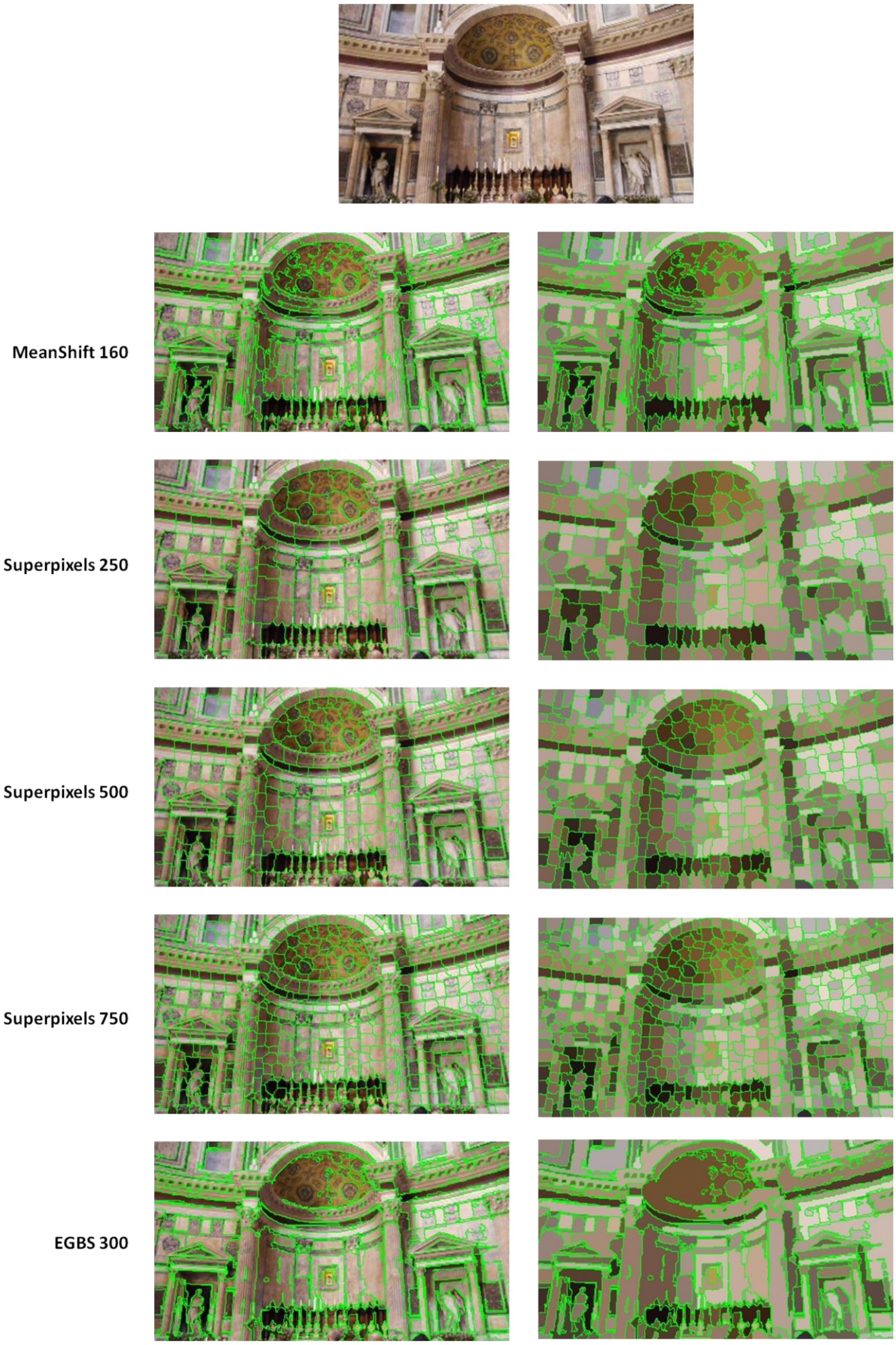


(a)Three patches.

(b) Obtained superpixels.

Figure 3.4. Patch stitching for superpixels. Left: three patches (orange, green, purple). Right: the resulting superpixels.

Figure 3.5 illustrates the results of the three segmentation techniques retained for a same image. We can observe the high regularity of the regions obtained with the help of the Superpixels method.


MeanShift 160
Superpixels 250
Superpixels 500
Superpixels 750
EGBS 300

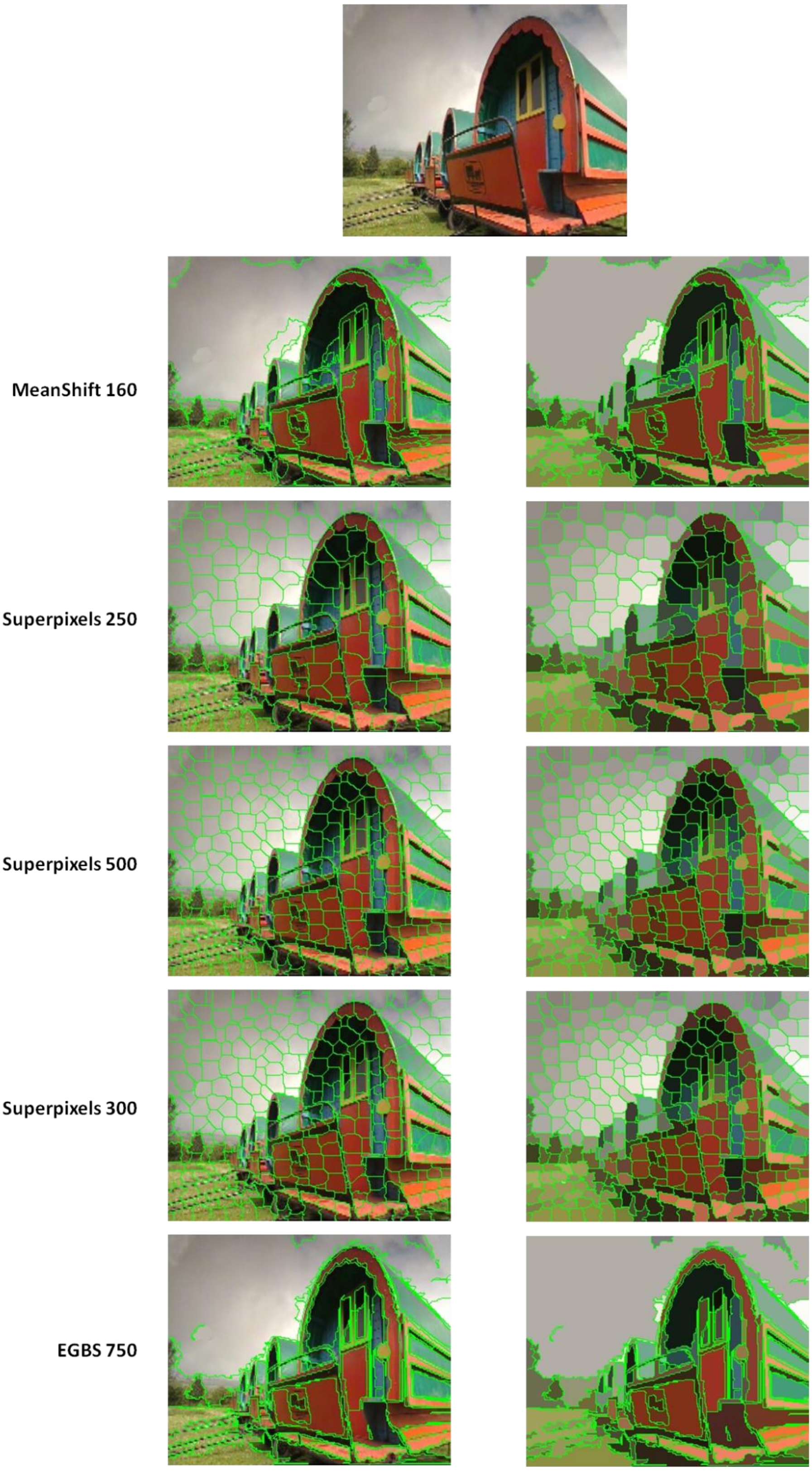


Figure 3.5. Examples of segmentation results for the considered techniques. In the right column, the regions are displayed with their corresponding average color.

Whatever the segmentation technique, the resulting segmented image should be described appropriately in order to enable object-based retrieval. In our case, we have adopted a color-based representation, based on the MPEG-7 dominant color descriptor, presented in the following section.

3.2 DCD description

Each region (or segment) determined needs to be described individually in such a manner that its descriptors can be also integrated in the description of a group of regions. In our approach, each segment is described by a unique, homogeneous color, defined as the average value of the pixels of the given region. The set of colors, together with their percentage of occupation in the image (*i.e.*, the associated color histogram) are regrouped into a visual representation, which is similar to the MPEG-7 DCD [MPEG 2002].

More precisely, let $C_I = \{c_1^I,\ c_2^I,\ \dots\ c_{N_I}^I\}$ be the set of N_I colors obtained for image *I*, and $H_I = (p_1^I,\ p_2^I,\ \dots\ p_{N_I}^I)$ the associated color histogram vector. The visual image representation is defined as the couple (C_I, H_I). In contrast with the MPEG-7 DCD, which supports a maximal number of eight colors, in our case an arbitrary number of dominant colors is supported. Note that other descriptors such as color histograms, contour shape or local gradient descriptor for each region can be employed in this framework.

The query is by definition an object of arbitrary shape and is processed in the same manner in order to derive its visual representation. Let us also note that more sophisticated DCD-based approaches, such as those introduced in [Yang 2008, Zin 2009], can also be considered.

The advantage of the DCD representation comes from the fact that objects with arbitrary numbers of colors can be efficiently compared by using, for example, the Quadratic Form Distance Measure introduced in [Hafner 1995], which can be re-written for arbitrary length representations as described by the following equation:

$$D_h^2(H_Q, H_I) = \sum_{i=1}^{N_Q}\sum_{k=1}^{N_Q} a(c_i^Q,\ c_k^Q)\, p_i^Q p_k^Q + \sum_{j=1}^{N_I}\sum_{l=1}^{N_I} a(c_j^I,\ c_l^I)\, p_j^I p_l^I - \sum_{i=1}^{N_Q}\sum_{j=1}^{N_I} a(c_i^Q,\ c_j^I)\, p_i^Q p_j^I \quad , \qquad \text{(3-1)}$$

where $H_Q = (p_1^Q,\ p_2^Q,\ \dots\ p_{N_Q}^Q)$ and $H_I = (p_1^I,\ p_2^I,\ \dots\ p_{N_I}^I)$ respectively denote the DCD histogram vectors of length N_Q, and N_I respectively associated to the query (*Q*) and candidate (*I*) images. The function *a*, describe the similarity between two colors c_i and c_j and is defined as*:*

$$a(c_i,\ c_j) = 1 - \frac{d(c_i, c_j)}{d_{max}} \quad , \qquad \text{(3-2)}$$

where *d* is the Euclidean distance between colors c_i and c_j and d_{max} is the maximum Euclidean distance between any 2 colors in the considered color space .

Let us note that each color region in a candidate image has a specific contribution to the global distance. Thus, the contribution of color c_j^I in an image I to the global distance between image I and query Q is expressed as:

$$C(c_j^I, Q) = \sum_{l=1}^{N_I} a(c_j^I,\, c_l^I)\, p_j^I p_l^I - \sum_{i=1}^{N_Q} a(c_i^Q,\, c_j^I)\, p_i^Q p_j^I \qquad (3\text{-}3)$$

Let us emphasize the flexibility provided by this similarity measure. As groups of regions evolve and regions are added or removed, the contributions of each region to the similarity measure can be easily updated according to the current region configuration.

The above-defined distance is used as a global criterion in the matching stage. Here, the objective is to determine, in each key-frame of the considered video sequence, candidate regions which are visually similar with the query.

In order to capture the internal structure of the objects as clusters of neighbor regions, we employ a graph-based representation. The advantage of graph based representation lies in its flexibility, the integration of spatial information and in the possibility of defining relevant parts of the object as sub-graphs or connected components.

Let us consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with n vertices and m edges. Each vertex represents a segmented region, hence n is the number or regions in the given image. Vertices are inter-connected with an edge if their corresponding regions are adjacent. In our case, the weights of the vertices are their corresponding dominant colors. For the edges, we have explored different weighting mechanisms, related to the neighborhood structure (*i.e.*, Potts-like potential) or to the similarity of the two corresponding regions (*cf*. Section 5.2).

An important characteristic of the graph is the possibility identifying and using sub-graphs and connected components. More precisely, the objective is to determine, for each candidate image, a sub-graph that minimizes the dissimilarity measure defined in Equation (3-1).

Within this framework, we have considered several optimization strategies. The first one, described in the next section, concerns a greedy region construction mechanism.

3.3 Greedy region construction

The algorithm starts from the initial set of regions obtained after the filtering stage. At each stage, we consider the current candidate object in image I and attempt to improve the current similarity measure between query and candidate objects. More precisely, we recursively eliminate the color segment which provides the highest contribution to the global distance (Equation (3-3)). We then check if the global distance is decreasing or not. If yes, we eliminate the corresponding region, update the color frequency vector H_I, and re-iterate the algorithm on the new candidate object obtained. If not, the region is maintained and the algorithm successively tries to eliminate the following regions (sorted by decreasing order of their contribution to the global distance).

Let us note that each time an attempt to eliminate a segment is performed, the region connectivity needs to be re-calculated in order to determine the eventually newly created connected components. Each connected component is then treated separately.

The algorithm stops when no improvement to the current global distance is obtained, whatever the region under investigation. We then return to the previously obtained best score configuration and stop the algorithm.

The strategy of recursively eliminating the highest contributor to the global score increases the speed of the algorithm, by pruning the search space. However, there is a risk to remain blocked in a local minimum, because whenever the distance is increasing, the algorithm stops. For this reason we investigated a different approach, based on a modification of the current greedy scheme.

3.4 Relaxed Greedy Scheme

Here, the principle consists of relaxing the exit condition in the previously described greedy scheme, in order to allow the algorithm to test additional region configurations, which are not necessarily lowering the global score at the current step. Thus, we constrain the algorithm to stop generating configurations when the current distance becomes "considerably" higher than the previous one. We consider that if the current distance is δ % higher than the previous obtained one, the candidate object has a low probability of reaching a configuration with a better score. In this case, the algorithm should stop and return the current best distance. Otherwise, it should continue removing the regions with the highest contributions to the score as it could find another minimum after this "uphill" configuration. In our experiments, we have used values of δ between 5% and 20%, which provide a good trade-off between the number of generated configurations and the computational time.

Let us note that the Relaxed Greedy (RG) scheme, although simple, becomes more useful in the case of images described by a relatively high number of segments since in this case the probability of getting stuck in a local minimum is significantly increasing. The main limitation of the greedy approaches is that they do not ensure the retrieval of an optimal solution. In order to achieve asymptotic optimality, we have adopted a simulated annealing matching strategy, described in the next section.

3.5 Simulated Annealing Optimization

The Simulated Annealing (SA) algorithm is a well-known stochastic optimization technique inspired from the behavior of condensed matter at low temperatures. The procedure employs methods that originated from statistical mechanics to find global minima of systems with large numbers of degrees of freedom. The correspondence between combinatorial optimization problems and the way natural systems search for the ground state (lowest energy state) was first realized by Kirkpatrick *et al.* [Kirkpatrick 1983] who applied Monte-Carlo methods in order to determine the solution of global optimization problems. Furthermore, authors generalized the Metropolis algorithm [Metropolis 1953] by using an approach with successively decreasing temperatures. At each stage, the system is simulated by the Metropolis procedure until the system reaches equilibrium.

The system starts with a high temperature *T*. Then, a cooling (or annealing) scheme is applied by slowly decreasing the temperature *T* according to some given procedure. At each temperature *T*, a series of random new states are generated and the states that improve the cost function are accepted. Instead of always rejecting states that do not improve the cost function, such states can be accepted with some finite probability depending both on the amount of energy increase and of the current temperature *T*. This process randomizes the iterative improvement phase and also allows occasional uphill moves (*i.e.*, moves that do not improve the solution) in an attempt to reduce the probability of blocking the algorithm into a local minimum. As temperature *T* decreases, configurations that increase the cost function are more likely to be rejected. It has been demonstrated that the SA procedure is asymptotically optimal, *i.e*. leads to a solution that is arbitrary close to the global minimum [Lundy 1986].

We employ the SA algorithm in order to take advantage of the higher number of possible region configurations and, thus, determine the global optimum score for all candidate regions. The energy *E* considered here is defined as the global matching score between the query and the candidate objects (Equation (3-1)). A binary state S = *S*(*c*) is associated to each color segment *c* indicating whether the segment is considered as part of the current object or not.

During each step, the algorithm attempts to change the state of the current color segment, by investigating the variation of the global energy $\Delta E = E(S') - E(S)$ when the state S is set to its complementary value *S*'. If this variation $\Delta E \leq 0$, then the current state S is replaced by its complementary value S'. If the energy variation ΔE has a positive value indicating an augmentation of the energy, then a random variable α, $0 \leq \alpha \leq 1$ is generated. The current state S is replaced byS' if the following condition is satisfied:

$$\alpha \leq e^{(-\Delta E/T)} \text{ and } E(S') > E(S) \quad , \tag{3-4}$$

where *T* denotes the value of the temperature at the current state.

A multiple number of iterations, denoted by n_{it}, are performed for a given temperature. The temperature of the system is then iteratively lowered, according to a given freezing scheme (or annealing schedule). In our work, we have adopted the following temperature variation:

$$T_{n+1} = \tau T_n \tag{3-5}$$

where τ is the (constant) cooling rate with value between 0 and 1, and T_n is the temperature at the n^{th} iteration. Let us note that as the temperature is decreased byτT_n, the probability of accepting a large decrease decays exponentially towards zero. The algorithm starts at an initial temperature T_0 and stops when a freezing temperature T_f is reached.

In addition, the connectivity information is used to guide the SA process. Thus, in order to ensure a smooth variation of the energy functional, a given segment is allowed to change state only if this process does not modify the topology of the candidate object (*i.e.*, it does not create holes or new, isolated connected components).

Let us also observe that, in contrast with the greedy-based approaches (where the only operation supported is the removal a given segment), here a color segment can be both removed and added to the current candidate object.

A particular attention has to be paid to the parameters involved in the considered freezing scheme. Keeping the same temperature for a long period of time will guarantee finding the best solutions since the SA algorithm is asymptotically optimal. This means that the longer the algorithm runs, the better is the quality of the solution obtained. However, from a practical point of view, this is not acceptable because of computational issues. Concerning the other parameter values, we consider the initial temperature T_0 between 0.5 and 0.9 and the freezing temperature T_f in the range $10^{-3} – 10^{-4}$. Finally, the typical values of the cooling factor τ are in the range from 0.9 to 0.99.

The considered SA approach is illustrated in Figure 3.6. At step 1, we randomly select a segment from the current region configuration and add it or remove it from the current regions configuration depending on its previous state. At step 2, we check if the operation made to the current region changes the consistency of the current region configuration. If it generates two new connected components or creates a whole within the current region configuration, it will not be accepted and marked as processed and we return to step 1 to select another segment from the non-processed segments. If the region passes the consistency check, at step 3 we compute the color similarly between the query model and the current region configuration using Equation (3-1). This similarity represents the energy of the current object state *E(S')* and we compare with the energy of the previous state *E(S)*. If there is an improvement in the energy score from the previous step (*i.e.*, $\Delta E = E(S') - E(S) < 0$) we accept the current state and replace the previous state with it at step 5 and then return to step 1 to select another segment to add or to remove. If there is no improvement in the energy score (*i.e.*, $\Delta E > 0$), we perform the test from Equation (3-4) to decide whether to accept the current state or not. If the test is passed, the current state is accepted and replaces the previous one at step 5. We return then to step 1. If the current state does not pass the test from step *4*, the current state is rejected and the algorithm goes directly to step 1 to test another segment. When all segments from the current region configuration have been processed, the current iteration is completed. We then mark as non-processed all the segments and start a new iteration. We perform n_{it} = *5-10* such iterations for each temperature. After n_{it} iterations the temperature decreases as indicated in Equation (3-5). The temperature is then successively decreased until it reaches the freezing temperature T_f.

The SA algorithm offers the advantage of generality, being able to optimize arbitrary energy functions. However, approaching the global optimum requires a very slow cooling rate (*i.e.*, parameter τ close to 1) and as a consequence, the optimization process becomes very slow and intractable in practice. For this reason, it is necessary to decrease the algorithm's temperature parameter faster than required by the theoretically optimal schedule. The price to pay is of course the sub-optimality of the resulting solution, since in this case the algorithm will converge to a local minimum: Greig *et al*. [Greig 1989] demonstrate that practical implementations of simulated annealing lead to results that are far from the global optimum.

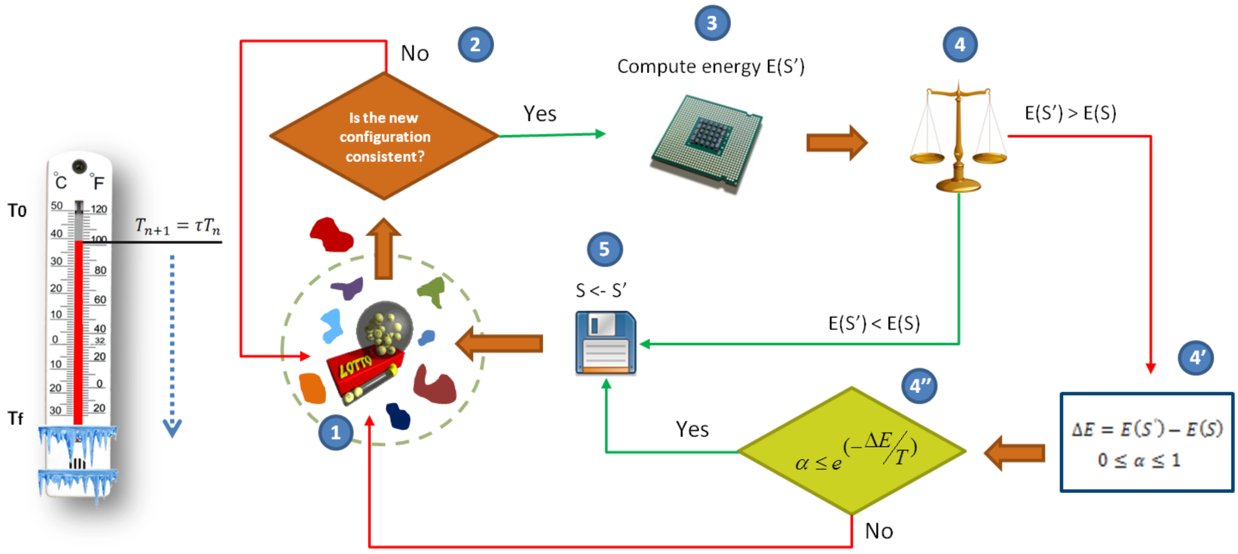


Figure 3.6. Overview on the Simulated Annealing implementation.

Moreover, all the above described optimization strategies integrate solely a minimal spatial information, related to the connectivity between adjacent regions. Spatial information could help in identifying pairs or groups of matched regions which co-occur in both query and test images. This information could be then used to privilege or penalize groups of regions in the region construction process.

In order to overcome such limitations, the graph-cut optimization method, presented in the following section, adopts a different approach, which considers no longer individual state modifications, but operations associated to groups of regions, so-called moves.

3.6 Graph-Cut Optimization

Let us consider a set of sites $\mathcal{P}$ (which can be sets of pixels in an image, or sets of nodes in a graph representation) and a set of labels $\mathcal{L}$. The objective is to determine a labeling function f (*i.e.* a mapping from $\mathcal{P}$ to $\mathcal{L}$) which minimizes a given energy function.

The Graph-Cut optimization method has been developed within a Markovian framework and is thus dedicated to the minimization of Gibbs functionals, which can be expressed as described by the following equation:

$$E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{p,q \in \mathcal{N}} V_{p,q}(f_p, f_q), \qquad (3\text{-}6)$$

where $\mathcal{N} \subset \mathcal{P} \times \mathcal{P}$ is a neighborhood system supposed to be available.

The term $D_p(f_p)$ is a function derived from the observed data that measure the cost of assigning the label f_p to the site p (*i.e.*, it measures how well the label f_p fits the site p given the observed data). It can also be referred to as the *data term*, or the *unary term*. The term $V_{p,q}(f_p, f_q)$ measures the cost of assigning the labels f_p, f_q to the adjacent sites p, q and is used to impose spatial smoothness. It is also referred to as the *smoothness term* or the *pairwise term*.

Let us note that at the borders of objects, adjacent sites will often have different labels. In this case, it is important that the energy functional E does not add extra penalization to such labeling. This requires that the term $V_{p,q}$ should be a non-convex function of $|f_p - f_q|$. Such an energy function is called *discontinuity-preserving* [Geman 1984, Kolmogorov 2004].

Since functions like E are non-convex functions in a space with thousands of dimensions, their minimization is a highly challenging issue. They have been traditionally minimized with general-purpose optimization techniques, such as simulated annealing [Kirkpatrick 1983], that can minimize arbitrary energy function. The Graph-Cut technique has been developed as an alternative to such high complexity methods. Let us recall the basic principle.

3.6.1 Graph-Cut basics

Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ be a graph which consists of a set of nodes $\mathcal{V}$ and a set of directed edges $\mathcal{E}$ that connect them. The set of nodes $\mathcal{V} = \{s, t\} \cup \mathcal{P}$ contains two specific *terminal nodes*, so-called *source* s and *sink* t, as well as a set of non-terminal nodes $\mathcal{P}$.

Each graph edge is assigned a non-negative weight (or cost) denoted by $c(p, q)$. A cost of a directed edge (p, q) may differ from the cost of the reverse edge (q, p). An edge is called a *t-link* if it connects a non-terminal node in $\mathcal{P}$ with a terminal. An edge is called *n-link* if it connects two non-terminal nodes. Let us denote by $\mathcal{N}$ the set of all n-links. The set of all graph edges $\mathcal{E}$ consists of n-links in $\mathcal{N}$ and t-links $\{(s, p), (p, t)\}$ for non-terminal nodes $p \in \mathcal{P}$. Such a graph is illustrated in Figure 3.7, where t-links are shown in red (for the source node s) and blue (for the sink node t), while n-links are shown in yellow.

An *s/t* cut C is by definition a partition of the graph nodes into two disjoint subsets S and T such that the source node s is in S and the sink node t is in T. An example of of a cut is illustrated in Figure 3.7.

To each cut $C = \{S, T\}$, we can associate by a binary labeling f defined from the set of the vertices $\mathcal{V} - \{s, t\}$ to $\{0,1\}$, with $f(p) = 0$ for $p \in S$ and $f(p) = 1$ for $p \in T$.

Thus, the minimization of the energy functional in Equation (3-6) can then be formulated as the determination of an optimal cut, as described in the following section.

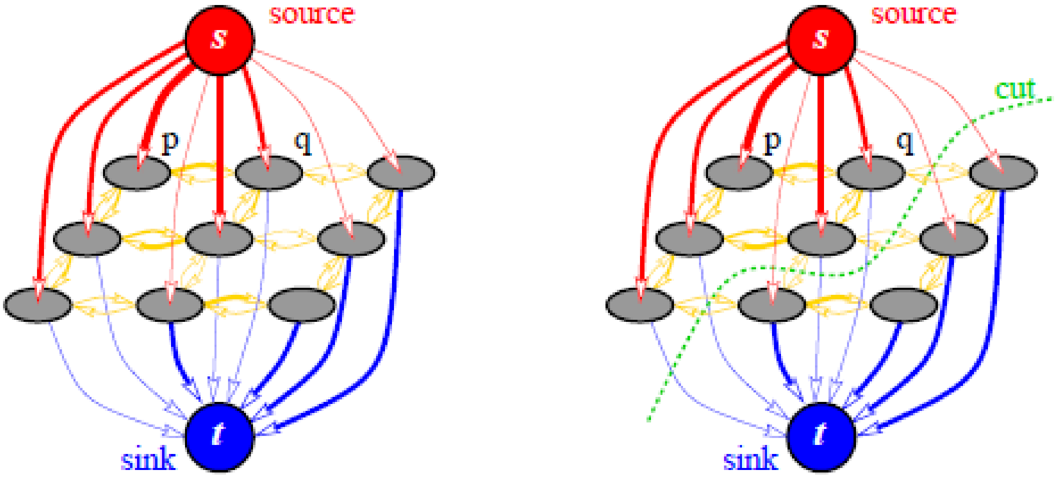


Figure 3.7. Graph construction in Greig *et. al*. Edge construction reflected by thickness. A graph $\mathcal{G}$ (left) and a cut on $\mathcal{G}$ (right).

3.6.2 The Min-Cut and Max-Flow problem

To each cut, a cost measure is associated with. By definition, the cost of a cut $C = \{S, T\}$ is the sum of costs of all *boundary* edges (p, q) such that $p \in S$ and $q \in T$. If (p, q) is a boundary edge, then we say that cut C severs the edge (p, q). Intuitively, the cost of the cut is the sum of costs of all edges that go from S to T:

$$C(S,T) = \sum_{p\in S, q\in T, (p,q)\in \mathcal{E}} c(p,q) \quad . \qquad \text{(3-7)}$$

The *minimum cut* problem is to determine a cut that has the minimum cost among all the possible cuts. One fundamental result in combinatorial optimization states that the minimum *s/t* cut problem can be solved by finding a *maximum flow* from the source *s* to the sink *t*. Intuitively, this can be interpreted as follows. If we consider the source node as a source of water, the sink node as a tank and the graph edges as directed “pipes” with capacities equal to the edge weights, the maximum flow is the maximum amount of “water” that can be transported from the source to the tank. The theorem of Ford and Fulkerson [Ford 1962] states that a maximum flow from *s* to *t* corresponds to a set of edges in the graph partitioning the nodes into two disjoint parts $\{S, T\}$ corresponding to a minimum cut. Thus, min-cut and max-flow problems are equivalent and the maximum flow value is equal to the cost of the minimum cut.

There are many standard polynomial time algorithms for min-cut/max-flow [Cook 1998]. Such algorithms can be divided into two main groups: “push-relabel" methods [Goldberg 1988] and algorithms based on augmenting paths [Ford 1962]. In practice, the push-relabel algorithms perform better for general graphs. In the field of computer vision, more specialized algorithms are used, in particular the method of Boykov and Kolmogorov [Boykov 2004] and its more recent extensions [Kohli 2007, Goldberg 2011], described in the next section.

3.6.3 Energy minimization using graph cuts

In order to minimize E using graph cuts, a specialized graph is created such that the minimum cut on the graph also minimizes E (either globally or locally). The form of the graph depends on the exact form of $V_{p,q}$ and on the number of labels. In certain restricted situations, it is possible to efficiently compute directly the global minimum. Thus, in the case of binary labeling Greig *et. al*. [Greig 1989] proposed a method to compute the global minimum of E. This is also possible for an arbitrary number of labels as long as the labels are consecutive integers and $V_{p,q}$ is the L_1 distance. The construction is due to [Ishikawa 1998] and is a modified version of [Roy 1998]. This construction has been further generalized to handle an arbitrary convex potentials $V_{p,q}$ [Ishikawa 2003].

However, a convex $V_{p,q}$ is not discontinuity preserving and optimizing an energy function with such a $V_{p,q}$ leads to over-smoothing at the borders of objects. The ability to determine the global minimum efficiently, while theoretically of great value, does not overcome this drawback.

Moreover, efficient global energy minimization algorithms even for the simplest discontinuity-preserving energy functions do not exist. Let us consider the following form for the pairwise potential:

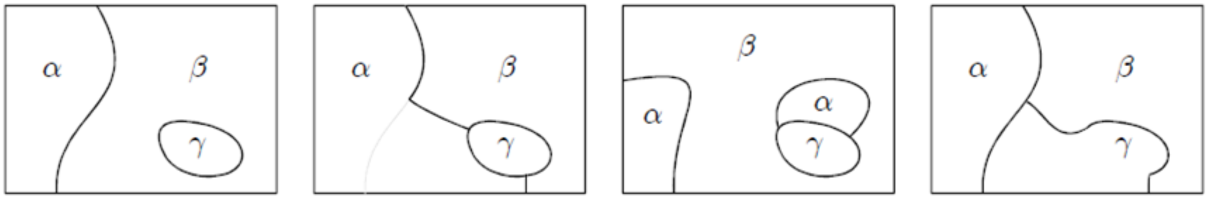
$$V_{p,q}(f_p, f_q) = T[f_p \neq f_q] \ . \tag{3-8}$$

, where the indication function $T[\cdot]$ is 1 if its argument is true and zero otherwise.

This smoothness term corresponds to a Potts model [Potts 1952], and is discontinuity preserving. Yet, it is known that its minimization is a NP-hard problem [Boykov 2001].

However, graph cut algorithms have been developed that compute a local minimum in a strong sense [Boykov 2001]. Such methods minimize an energy function with non-binary variables by iteratively minimizing an energy function with binary variables. Boykov *et al*. [Boykov 2001] introduce two algorithms based on graph cuts that can find efficiently a global minimum with respect to two types of large moves, namely *expansion moves* and *swap moves*, which simultaneously change the label of multiple sites. This is in contrast with popular optimization algorithms previously proposed, such as Iterated Conditional Modes (ICM) [Besag 1986] and Simulated Annealing [Geman 1984], which allow only one site at a time to change its label.

Any labeling f can be uniquely represented by a partition of image sites $\mathcal{P} = \{\mathcal{P}_l | l\epsilon\mathcal{L}\}$ where $\mathcal{P}_l = \{p\epsilon\mathcal{P} | f_p = l\}$ is a subset of sites assigned with the label l. In the case of the swap moves, given a pair of labels α, β, a move from a labeling f or a partition $\mathcal{P}$ to a new labeling f' or partition $\mathcal{P}'$ is called $\alpha - \beta$ swap is $\mathcal{P}_l = \mathcal{P}'_l$ for any label $l \neq \alpha, \beta$. More precisely, this means that the only difference between $\mathcal{P}$ and $\mathcal{P}'$ is that some pixels that were labeled α in $\mathcal{P}$ are now labeled β in $\mathcal{P}'$, and some pixels that were labeled β in $\mathcal{P}$ are now labeled α in $\mathcal{P}'$. Let us note that a special case of a $\alpha - \beta$ swap is a move that gives the label α to a set of pixels previously labeled β. Figure 3.8 illustrates examples of α-β-swap moves.

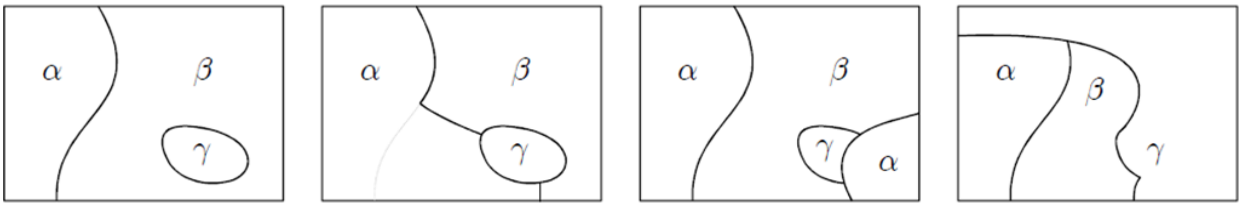


current labeling *f* a α-β-swap another α-β-swap a γ- β-swap

Figure 3.8. Examples of swap moves with respect to the current labeling *f* is shown at left. A α-β-swap move is made from binary choices: each f_p involved can choose either α or β (illustration from [Delong 2011]).

For the expansion moves, given a label α, a move from a partition $\mathcal{P}$ (or labeling *f*) to a new partition $\mathcal{P}'$ (or labeling f') is called an α-expansion if $\mathcal{P}_\alpha \subset \mathcal{P}'_\alpha$ and $\mathcal{P}_l \subset \mathcal{P}'_l$ for any label $l \neq \alpha$. This means that an α-expansion move allows any set of image pixels to change their label to α. Examples of expansion moves are illustrated in Figure 3.9.

Summarizing the two methods, for the swap moves, given an input labeling *f* (partition $\mathcal{P}$) and a pair of labels α, β, the objective is to determine a labeling $\hat{f}$ that minimizes E over all labelings within one $\alpha - \beta$ swap of *f*. In the case of the expansion moves, given an input labeling *f* (partition $\mathcal{P}$) and a label α, a labeling $\hat{f}$minimizing E over all labeling within one α-expansion of f has to be determined.



Current labeling *f* A α-expansion Another α-expansion A γ-expansion

Figure 3.9. Examples of expansion moves with respect to the current labeling *f* is shown at left. An α-expansion move is made from binary choices: α can either expand to pixel *p*, of leave f_p as it is (illustration from [Delong 2011]).

In our work we have adopted the expansion move algorithm proposed in [Boykov 2004] which is one of the most effective algorithms for minimizing discontinuity-preserving energy. This algorithm can be used whenever $V_{p,q}$ is a metric on the space of labels; which includes several important discontinuity preserving energy functions.

The expansion move algorithm cycles through the labels α in a fixed or random order and determines the lowest energy α-expansion move from the current labeling. If this expansion move has lower energy than the current labeling, then it becomes the current labeling. The algorithm terminates with a labeling that is a local minimum of the energy with respect to expansion moves. More precisely, there is no α –expansion move, for any label α, with lower energy.

3.6.4 DOOR Graph-Cut integration

Let us first note that the direct minimization of the global quadratic similarity measure defined in Equation (3-1) is not possible with the Graph Cut technique, since such a measure is not reducible to a Gibbs energy functional (Equation (3-6)) [Kolmogorov 2004, Freedman 2005]. Instead, we guide our Graph Cut process with a dedicated Gibbs energy, which is defined as described in the following.

We formulate the problem of region based object recognition as a binary labeling graph cut problem. Similarly with the segmentation approaches, the values of the binary labels can be *foreground* and *background* [Boykov 2001b, Li 2004, Boykov 2006], indicating whether the considered segment is inside or outside the object of interest. The foreground label is assigned to the segments similar with the regions of the query object, while the background label is assigned to non-similar regions from the current image.

We have considered the graph-based representation described in Section 3.2. Thus, the set of nodes $\mathcal{P}$ represents the set of segments obtained with an arbitrary segmentation procedure. The adjacency relations between them are also supposed to be available.

Let $f = \left(f_1, f_2, \dots, f_p, \dots, f_{|\mathcal{P}|}\right)$ be the binary vector of labels associated with the set of segments $\mathcal{P}$. For each segment p, the corresponding label can be either “foreground” ($f_p = 1$) or “background” ($f_p = 0$). In this manner, each vector f defines an object candidate, constructed from with the regions labeled as “foreground”. In order to obtain the vector f, we define a set of soft constraints on region and boundary properties of f with the help of the following cost function:

$$E(f) = \lambda \cdot D(f) + V(f) \qquad \text{(3-9)}$$

, where $D(f)$ is the regional, data or unary term and represents the likelihood energy, encoding the cost when the label of p is f_p:

$$D(f) = \sum_{p\in\mathcal{P}} D_p\left(f_p\right). \qquad \text{(3-10)}$$

The function $V(f)$ is the smoothness or pairwise term and represents the internal energy, specifying the cost when the labels of adjacent nodes p and q are f and f_q respectively:

$$V(f) = \sum_{(p,q)\in\mathcal{N}} V_{p,q} \cdot \delta_{f_p \neq f} \qquad \text{(3-11)}$$

with

$$\delta_{f_p \neq f} = \begin{cases} 1 & if\ f_p \neq f_q \\ 0 & if\ f_p = f_q \end{cases}. \qquad \text{(3-12)}$$

The real, positive parameter λ ($\lambda \geq 0$) from Equation (3-9) weights the relative importance of the data term *D(f)* with respect to the boundary properties term *V(f)*. Let us note that such an approach has been extensively used within the framework of interactive image segmentation. In such cases, the user marks the image with so-called *scribbles* indicating regions that belong to the foreground or objects of interest and regions that belong to the background [Boykov 2001b, Li 2004, Boykov 2006, Bai 2007]. The segmentation algorithm employs these scribbles as seeds for building regions and segmenting out the objects of interest. Such scribble information can be integrated naturally in the GraphCut optimization in the data term as source/foreground and sink/background nodes. The scribbles typically boost the segmentation accuracy and speed, and can be deployed in general public video editing tools (*e.g.*, Adobe After Effects [Bai 2009, Adobe 2012b]).

The principle is illustrated in Figure 3.10. Users draw on a given image a set of scribbles hinting the object of interest (foreground) with blue curved lines and the non-interesting regions (background) with green curved lines. The segmentation algorithm performs the segmentation starting from these user hints. The result is then displayed to the user and if the segmentation is not accurate, he can then add other helping scribbles to be considered for improving the segmentation results.

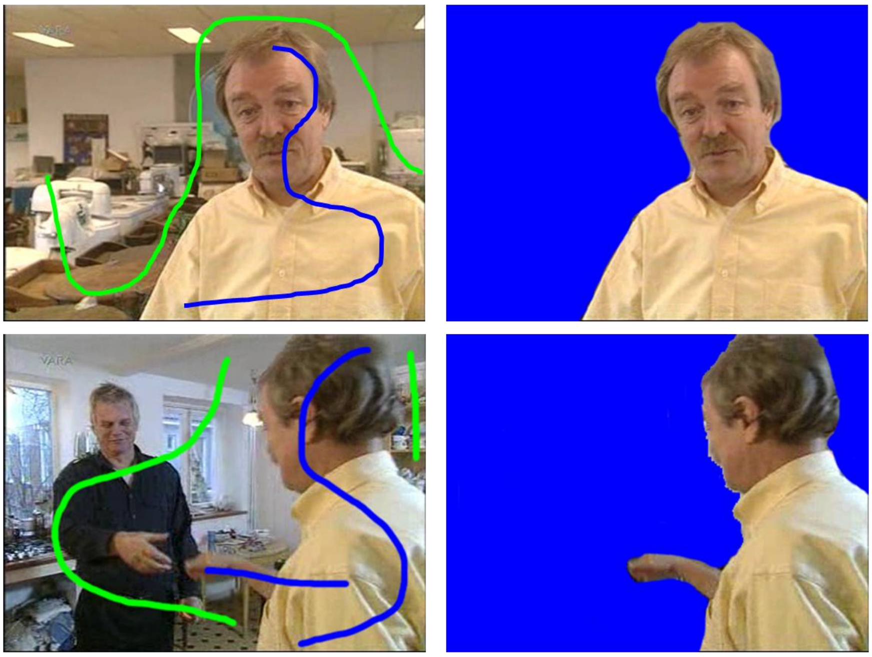


Figure 3.10. Principle of scribble-based interactive segmentation. In the left column, images with user drawn scribbles (foreground - blue curves, background – green curves). The segmentation results are presented in the right column.

It would be of great help to manage to introduce such a scheme within our object retrieval scheme in an automatic manner. This requires constructing automatically the scribbles, as described in the following section.

3.6.4.1 *Pseudo scribble construction*

Since we dispose of a query model indicated by the user, we introduce a pseudo-scribble approach for enhancing the object detection. The regions that compose the query object are considered as foreground scribbles.

For each candidate image under consideration, the corresponding set of regions is filtered, according to their color similarity with the query regions (*cf*. Section 3.3). Here; we adopt a relaxed threshold in order to avoid eliminating potentially useful regions that might present low similarity scores because of lightening variations or noise. The filtered-out regions are considered as background regions and are annotated as background pseudo-scribbles. The rest of the regions are considered as detected foreground regions.

Let us note that in our case, and in contrast with traditional segmentation approaches, the foreground scribbles do not belong to the candidate image, but to the query model.

This principle is illustrated in Figure 3.11, Figure 3.12 and Figure 3.13. Figure 3.11 illustrates the query model, which is interactively specified by the user. Its constitutive segments will define the set of foreground scribbles.

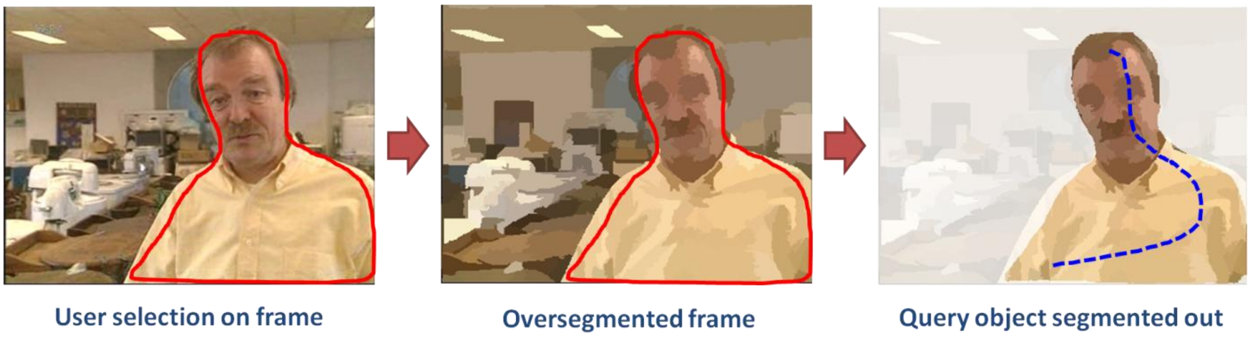


Figure 3.11. Query object extraction from user selection. After the regions composing the query model have been extracted, all these regions become foreground pseudo-scribbles (dashed blue-curve).

Figure 3.12 illustrates the case of a candidate image which is visually similar with the query frame (*i.e.*, the pose of the character of interest is almost the same, the color tones of the background are highly similar).

Despite the similarity, some mislabeled regions are here detected (*e.g.*, pieces of furniture assigned to foreground, parts of face and shirt assigned to background).

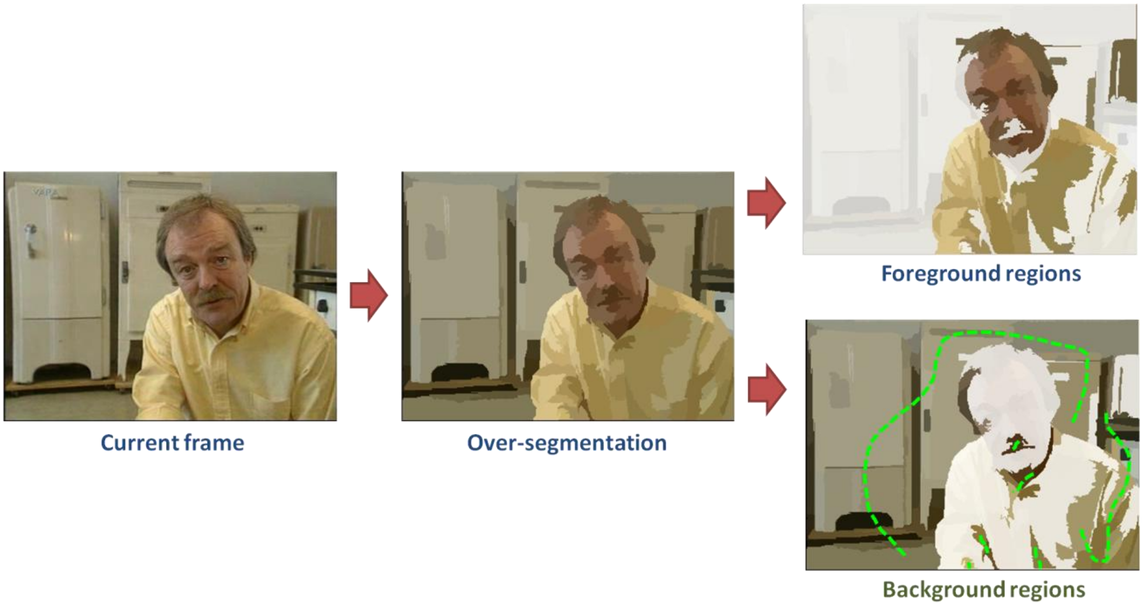


Figure 3.12. Background pseudo-scribbles extraction from a frame similar with the query frame illustrated in Figure 3.12. Each image is firstly passed through color filtering that removes all regions with colors far-off from the ones in the query model. The filtered out colors become background pseudo-scribbles (dashed green line).

In Figure 3.13 we illustrate a different case, where the object of interest is significantly different in the query and the candidate image.

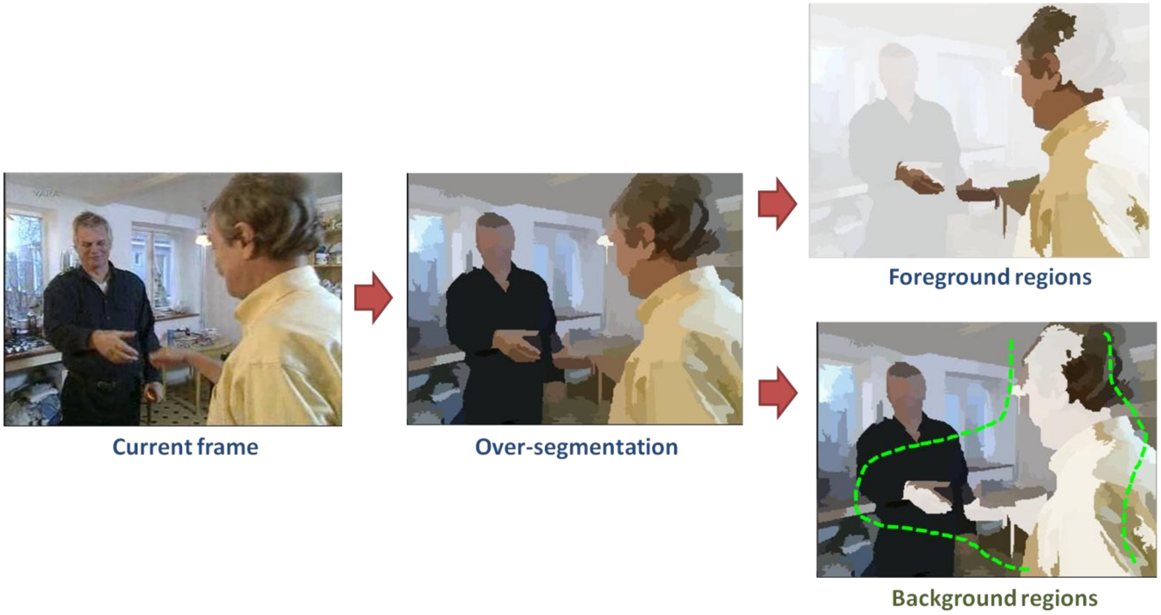


Figure 3.13. Background pseudo-scribbles extraction from a frame non-similar with the query frame illustrated in Figure 3.7. In this case we can notice that some of the filtered out regions are part of the queried object and they should be integrated back in the candidate object configuration. The color similarity of the border regions makes it possible to recover such regions and to reject other non-similar ones.

Here again, the filtered-out regions are labeled as background scribbles. We can notice that due to the light intensity variation, a part of the object of interest has been rejected by the color filter. In addition, a noisy region has been retained in the current object configuration. We also observe that these mislabeled regions are located at the boundary between the candidate object and the background. We should then foresee such cases and define them in the energy function considered (Equation (3-9)). Regions at the boundary should be given the possibility to change state (*i.e.* foreground/background) according to their visual similarity and spatial relationship with other regions from both foreground and background. The definition of the energy functional considered is presented in the following section.

3.6.4.2 Energy definition

For the data term *D(f)*, we consider as nodes the regions obtained from the image segmentation. For each node *p*, we have to define the weights of the t-links with the source (foreground) $D_p(f_p = 1)$ and the sink (background) $D_p(f_p = 0)$. These weights are defined with the help of the color similarity between a given segment and the sets of foreground scribbles, within the query model, and the set of background regions from the considered candidate image.

We first compute the minimum distance from its color *C(p)* to the foreground scribbles, as expressed by the following equation:

$$d_p^{\mathcal{F}} = \min_{m \in \{1, \ldots, F\}} \left\| C(i) - C^{\mathcal{F}}(m) \right\|_{L_2} , \tag{3-13}$$

where *F* is the number of foreground scribbles (*i.e.*, the number of regions composing the query model).

Similarly, we compute the minimum distance from the color *C(p)*of the segment *p* to the set of background regions:

$$d_p^{\mathcal{B}} = \min_{n \in \{1, \ldots, B\}} \left\| C(i) - C^{\mathcal{B}}(n) \right\|_{L_2}, \tag{3-14}$$

, where *B* is the number of background regions.

Then, the weights associated to the t-links for each node *p* are defined as follows:

$$D_p(f_p = 1) = \frac{d_p^{\mathcal{F}}}{d_p^{\mathcal{F}} + d_p^{\mathcal{B}}} \qquad D_p(f_p = 0) = \frac{d_p^{\mathcal{B}}}{d_p^{\mathcal{F}} + d_p^{\mathcal{B}}} . \tag{3-15}$$

Concerning the pairwise energy term $V_{p,q}(f)$ we consider only the regions on the boundary between the foreground and background. Similarly with [Li 2004] we employ a pairwise potential $V(f)$ based on the color gradient along the object boundary, as described in the following equation:

$$V_{p,q} = \frac{1}{1+\|C(p)-C(q)\|_{L_2}}, \tag{3-16}$$

where $\|C(p) - C(q)\|_{L_2}$ is the color distance between the colors $C(p)$ and $C(q)$ of regions p and q. This distance can be chosen according to the choice of the color space (*e.g.*, Euclidean for RGB, CIE76 or CIE94 for the CIE-L*a*b* color space).

As mentioned in Section 3.6.4, the potential $V_{p,q}$ is weighted with the term $\delta_{f_p \neq f_q}$ which allows us to capture the gradient information only along the current object boundary. We notice that the more similar the colors of the two nodes, the higher $V_{p,q}$ becomes, meaning that the less likely the edge is on the object boundary. In practice, the term $V_{p,q}(f)$ penalizes adjacent nodes which are assigned with different labels.

The GraphCut object construction process is described in the following section.

3.6.4.3 GraphCut-based object retrieval

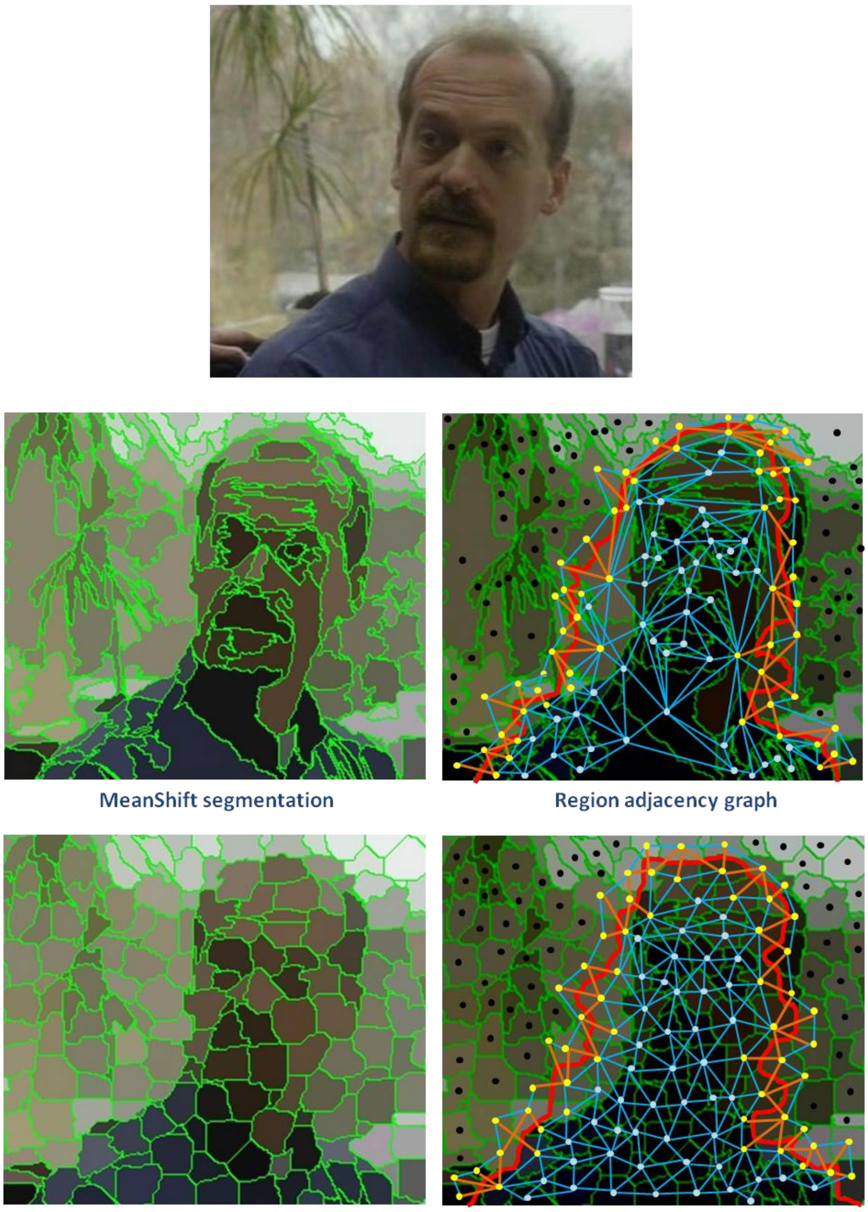
Starting from the initial segmentation obtained after the color filtering process, the various connected components of the detected foreground regions are considered as object candidates and optimized with the GraphCut algorithm.

The corresponding graph considered includes the set of foreground regions, together with their neighboring background regions, as illustrated in Figure 3.14. The rest of the background regions (illustrated with black dots in Figure 3.14) are ignored at this stage.

The minimum cut configuration is then determined as presented in Section 3.6.2. As discussed in the previous section, the foreground/background pseudo scribbles determined after the filtering stage can include some erroneous items. In order to deal with such cases, we propose to recursively apply the GraphCut algorithm, for a pre-defined number of steps. At each step, the result obtained from the previous one is considered as initialization of the foreground/background region specification. The regions adjacency graph and its corresponding weights are updated at each step. The foreground pseudo scribbles are fixed, as they correspond to the query model. Solely the background pseudo scribbles are updated.

In our experiments, we have observed that in a majority of cases the configuration yielding the optimal similarity score is reached within the first 10-15 steps.

The proposed workflow is illustrated in Figure 3.15 and Figure 3.16. In Figure 3.15 we present the case of a frame visually similar with the query frame (*i.e.*, the object of interest displays a similar pose and background). Here, the best candidate has been identified at the fourth step (illustrated with a green bounding box). Let us emphasize that between consecutive steps, groups of regions are added or removed from the current candidate object configuration. This makes is possible to accurately retrieve the object of interest. In the final steps the algorithm switches between the same two states, meaning that no better configuration can be identified.



MeanShift segmentation

Region adjacency graph

SuperPixels segmentation

Region adjacency graph

Figure 3.14. Structure of graph to be optimized with the GraphCut algorithm. The nodes are regions from typical segmentation methods (MeanShift - second row, SuperPixels - third row). The blue dots indicate the foreground regions for the current configuration and the black dots show the background regions. The yellow dots show the regions from the object boundary illustrated with a thick red curve, with the yellow dots inside the curve belonging to the current object configuration (foreground). The orange edges connect adjacent regions and are weighted with the pairwise energy term. The blue edges connect non-boundary non-boundary adjacent regions are weighted with zero.

In Figure 3.16 we illustrate the case of a frame where the object of interest is in a different pose. The query object is the same one as in Figure 3.15 (*i.e.*, man with blue shirt). Here, the character of interest is displayed in a zoomed view and the background is slightly different. Most of the background segments are rejected in the first steps and further the configuration of the object of interest is refined by adding and removing regions. The best configuration is obtained at the fifth step.

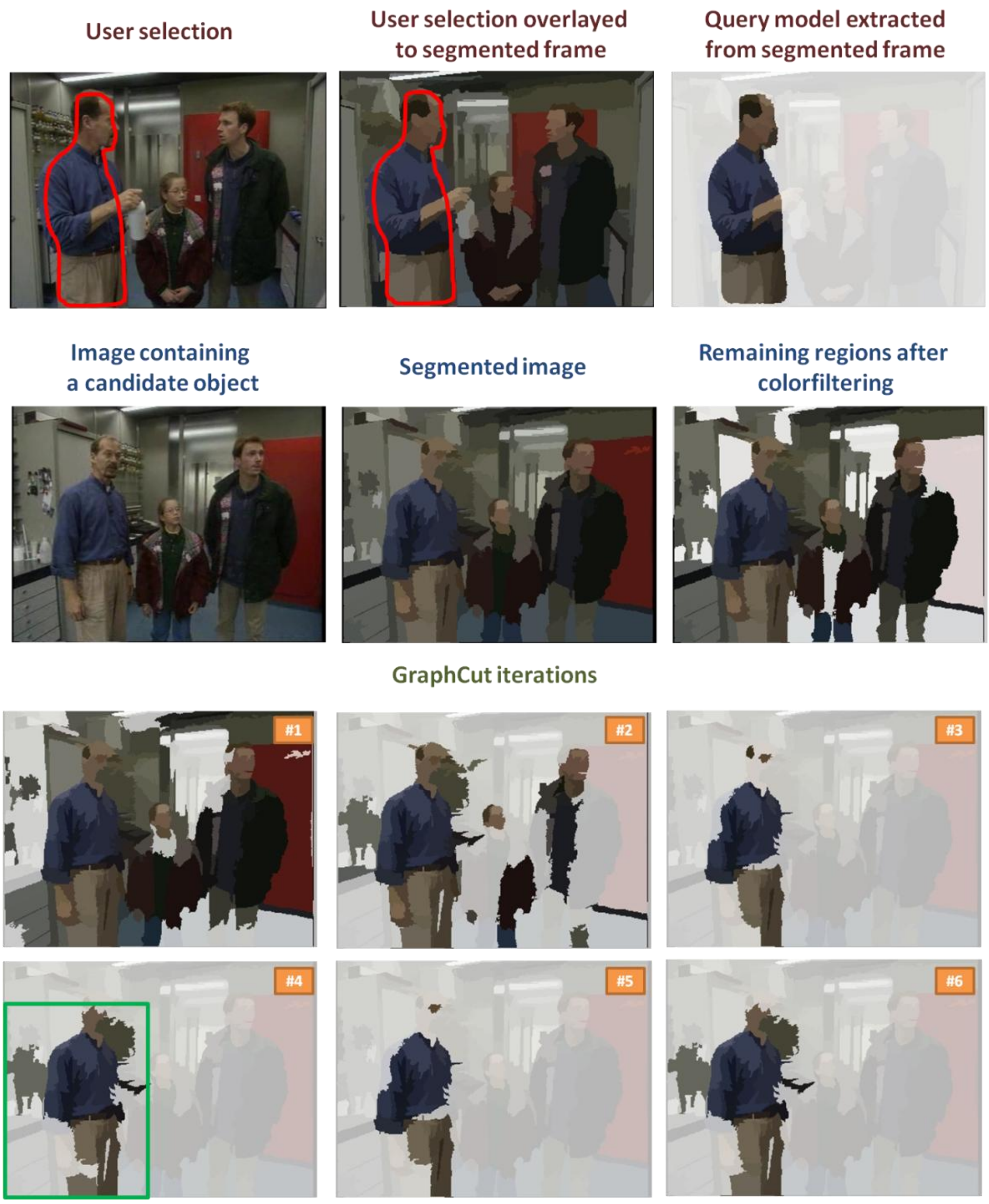

User selection
User selection overlayed to segmented frame
Query model extracted from segmented frame
Image containing a candidate object
Segmented image
Remaining regions after colorfiltering
GraphCut iterations
#1
#2
#3
#4
#5
#6

Figure 3.15. GraphCut based object retrieval: the case of a frame highly similar with the query.

3.7 Conclusion

In this chapter, we have introduced the region-based representation proposed. The principle consists of obtaining an over-segmentation of the image into a set of segments which are relatively homogeneous with respect to a colorimetric criterion. Within this framework, three different segmentation methods have been adopted, namely MeanShift, EGBS and SuperPixels. The resulting regions are described with the help of an extended version of the MPEG-7 dominant color descriptor, able to support an arbitrary number of colors. The segments adjacency information is also retained and stored with the help of an adjacency graph.

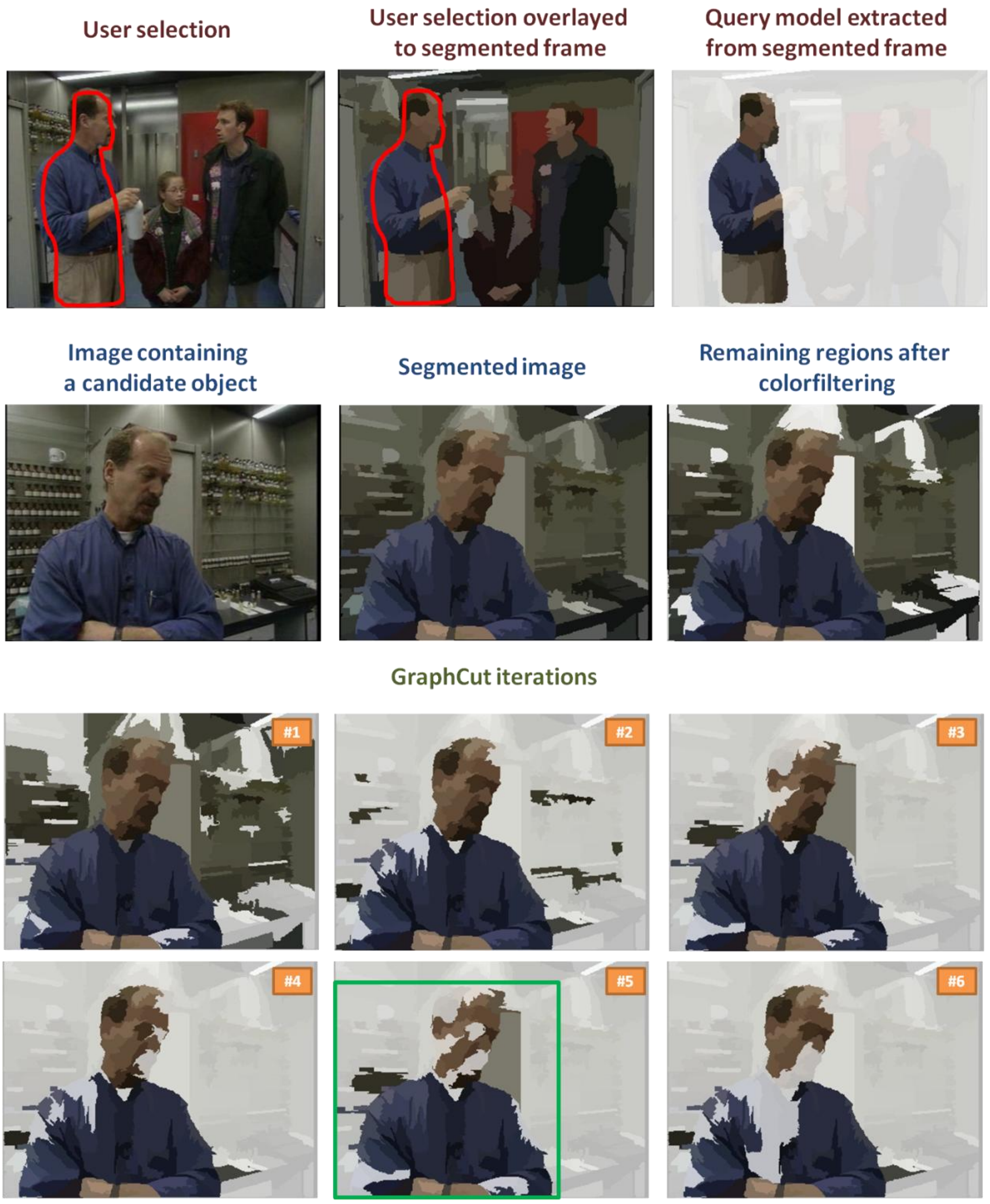

User selection
User selection overlayed to segmented frame
Query model extracted from segmented frame
Image containing a candidate object
Segmented image
Remaining regions after colorfiltering
GraphCut iterations
#1
#2
#3
#4
#5
#6

Figure 3.16. GraphCut based object retrieval in the case of a frame non-similar with the query frame.

Due to the quadratic similarity measure, such descriptions can be matched for arbitrary numbers of dominant colors. However, optimizing the object’s configuration is a combinatorial optimization problem. This requires a search strategy, able to determine an optimal sub-graph, with respect to the considered similarity measure. We have employed the quadratic similarity measure as energy to be minimized over multiple region configurations corresponding to a relevant object instance of the query model. Four different search strategies have been proposed and described, namely Greedy, Relaxed Greedy, Simulated Annealing and GraphCut.

4. Bag-of-Words representation

Abstract. *The Bag-of-Words framework is among the most popular technique employed in multimedia content retrieval. In this section, we present an in-depth analysis of the rich literature dedicated to BoW approaches and describe the main techniques involved at each stage, with related advantages and limitations. Such techniques include interest point detectors, interest point descriptors/representations, classification schemes, matching approaches and geometry consistency checking/re-ranking methods. The analysis is performed from the perspective of large scale object retrieval applications, in order to adopt an optimized solution that can serve as reference for benchmarking purposes. Finally, we propose a query expansion technique that makes it possible to retrieve objects of interest in videos in a multi-modal manner, starting from a textual query performed upon existing image repositories.*

Keywords: *Bag-of-Words, state of the art, local feature detectors, local feature descriptors, scalable clustering, descriptor quantization, query expansion.*

Résumé: *La méthode Sac-de-Mots (Bag-of-Words) est l'une des techniques les plus populaires utilisés dans la recherche de contenu textuel et multimédia. Dans, cette section, nous présentons une analyse approfondie de la littérature riche consacrée à cette méthode et nous décrivons les principales techniques impliqués à chaque étape en illustrant leurs avantages et leurs limites. Ces techniques comprennent les détecteurs des points/régions d'intérêt et leur descripteurs et représentations, les systèmes de classification et les méthodes de calcul de similarité et de vérification de la cohérence géométrique. L'analyse est effectuée dans la perspective d'applications de recherche des objets à grande échelle afin d'adopter une solution optimale qui peut servir en tant que référence pour une analyse comparative. Enfin, nous proposons une technique d'expansion de requête qui permet de récupérer des objets d'intérêt dans des vidéos d'une manière multimodale, à partir d'une requête textuelle effectués sur des collections d'images grand publique.*

Mots clés: *Sac-de-Mots, état de l'art, détecteurs de caractéristiques locales, descripteurs de caractéristiques locales, regroupement évolutif, quantification des descripteurs, extension de requête.*

Recent work in object-based content retrieval has adopted the paradigms of textual-based retrieval systems, by creating an analogy between the so-called "visual words" and keywords. A visual word is a prototype visual descriptor, represented as a vector in a multidimensional space and obtained with the help of a clustering/vector quantization process. It captures the local characteristics of an image part, which is most often a salient region defined within the neighborhood of an interest point.

This principle can be summarized as follows. For each image within a considered a set of training images, a set of salient regions/interest points is determined. Each item of interest detected is described with the help of a visual descriptor. The set of visual descriptors obtained for the whole training set is then clustered into a vocabulary of prototype vectors, which represents the visual words.

Once the visual words available, the description of an arbitrary image becomes straightforward: first, the set of interest points/salient regions together with the corresponding descriptors are determined. Then, each descriptor is quantized to the closest prototype in the vocabulary. Finally, a histogram of the visual words determined in the image is created and used further for image representation/retrieval purposes. In practice, an object is represented as a subset of the local features associated with an image. This model has been popularized as the Bag-of-Words/Bags-of-Visual Words/Bags-of-Features model.

Despite the analogy between "visual words" and words in text documents, the trade-offs in ranking images and web pages are somewhat different. The query image contains significantly more words (usually a few hundred) than textual queries (usually less than 10). In addition, the spatial structure of an image is a highly important feature, while a textual query can only impose an ordering of the query words at most. Thus, the spatial relationships between different visual words of the query image can provide essential information for enhanced retrieval. Usually such information is discarded in the standard BoW models, but it is reinserted at the end of the search process, where the top retrieved results are refined and re-ranked according to their spatial consistency with the query model.

The first BoW approaches in computer vision used *textons* [Julesz 1981] as visual words. The textons are defined as the centers of clusters obtained after clustering the responses of linear texture filters [Leung 2001]. This early framework has evolved and different local descriptors and clustering algorithms have been proposed. The most popular choice for local descriptors are the interest points [Sivic 2003, Sivic 2009], but other types of descriptors have been equally tested and proposed: dense sampled regions [Jurie 2005], segmented regions [Vieux 2012] or segmented regions and neighboring interest points [Pantofaru 2006].

The BoW model is today among the most popular and effective object representation and retrieval frameworks, and notably largely exploited in the field of semantic image classification. The main steps of the BoW framework are the following:

I. Offline pre-processing
   a. Detect interest points or blobs as affine covariant regions in each key-frame of a given video (or sets of videos).
   b. Extract interest point descriptors from the detected affine covariant regions.
   c. Sample evenly a subset of the descriptors from each frame and construct a visual vocabulary by clustering the sampled descriptors.

d. Map each region descriptor to its nearest cluster word or words from the visual vocabulary.
e. Compute term frequency (*tf*) vectors.
f. Build the inverted index structure.
g. Compute *tf-idf* weighted vectors and perform L2-normalization

II. Run-time querying

a. Detect affine covariant regions from query and extract local descriptors.
b. Map each query descriptor to the visual words from the vocabulary.
c. Compute *tf-idf* vector for query image and perform L2-normalization
d. Use query visual words to compile a list of plausible frame candidates from the inverted index.
e. Compute matching between the query histogram and the histograms corresponding to the key-frames from the list of candidates.
f. Perform re-ranking of first *N* retrieved key-frames using geometric consistency verification.

Let us now detail on the main components of this framework.

4.1 Local Feature Detectors

A huge amount of research work has been dedicated to the specification and automatic detection of salient visual features that can be effectively exploited within an image retrieval framework. Let us first review the most representative approaches of the state of the art; with main principles and associated properties.

4.1.1 State of the art review

Features describing the local appearance of the content are at the base of some of the most successful and popular object recognition systems. Such features typically describe the texture of a given image patch. Objects are then represented by the appearance of a collection of such local patches.

The desired characteristics of such features are the following:

- Repeatability, *i.e.,* the propriety of local region of being re-detected in other image under different camera viewpoints, illumination conditions and noise,
- Discriminative power, *i.e.*, local features with different appearances can be discriminated in the descriptor space
- Redundancy of the object description, which makes it possible to cope with missing or mismatched regions in the case of partial object occlusions.

In the literature, several entities such as points, edge segments or small image patches are considered as local features. Additionally the *local feature* denomination is not unique in the literature and other term such as interest point, key point, patch or region can be encountered.

Local features have proved to be well-suited for a large variety of computer vision application such as wide baseline matching for stereo pairs [Tuytelaars 2000, Matas 2002], object retrieval in video content [Sivic 2003], image stitching in panoramas [Brown 2003], symmetry detection [Cho 2009b], texture recognition [Lazebnik 2003], image retrieval [Sande 2010, Philbin 2007].

Early work on the extraction of local features in natural images includes the interest operators of Moravec [Moravec 1977] and Harris [Harris 1988]. Garding and Lindeberg [Garding 1996] have shown how to design an affine blob detector using an affine adaptation process based on the second moment matrix. Lindeberg [Lindeberg 1998] extends a multi-scale blob detector based on maxima of the Laplacian in the framework of automatic scale selection. Here a "blob" is defined by a scale-space location where a normalized Laplacian measure attains a local maximum. In an informal manner, the spatial coordinates of the maximum become the coordinates of the center of the blob, and the scale at which the maximum is identified becomes its characteristic scale.

Lowe proposed an efficient algorithm [Lowe 1999] for scale invariant interest point detection based on local maxima in the scale-space pyramid built with the approximation of Difference of Gaussian (DoG) filters. The input image is iteratively smoothed with a Gaussian kernel and sub-sampled. The local maxima in the pyramid representation are extracted by comparing each point with its 8 neighbor from the same scale and with nine neighbors from the scales above and below, considering a 26 point neighborhood. This makes it possible to effectively compute the localization and the scale of the interest points. The DoG operator is actually an approximate of the Laplacian of Gaussians function with significantly better computational performances. The main drawback of the representation proposed by Lowe [Lowe 1999] is that local maxima can be detected also in the neighborhood of contours or straight edges, where the signal changes in only one direction.

A Hessian fast detector is used by Bay *et al.* [Bay 2008] for the Speeded-Up Robust Features (SURF) scale-invariant feature detector. Here, the Hessian matrix is approximated by convolving the image with Haar wavelets, which are approximated, at their turn, by Gaussian second derivates. The locations and scales of the interest points are determined by maximizing the determinant of the scale normalized Hessian matrix. The SURF detection method has been reported to be more than five times faster than the DoG approach of Lowe [Lowe 1999].

Mikolajczyk and Schmid [Mikolajczyk 2002] proposed an extension of the Harris detector [Harris 1988] that re-estimate the interest point's position and scale by employing the trace and the determinant of the Hessian matrix $\mathcal{H}$. The trace of the Hessian matrix $\mathcal{H}$ is equal to that of the LoG, but detecting simultaneously the maxima of the determinant penalizes points for which the second derivatives detect signal changes in only one direction. The method provides a scale invariant set of interest points, each described by its own scale. This scale is used for estimating an affine shape of a point neighborhood; ensuring the affine invariance. Such features are known in literature as *affine invariant* or *affine covariant* features as they are robust to viewpoint changes described by an affine transformation. In other, more recent approaches [Mikolajczyk 2004, Mikolajczyk 2005] for affine covariant feature detection, the Harris detector is replaced by a Hessian detector and the interest points are detected in a similar manner.

The Maximally Stable Extremal Regions (MSER) [Matas 2002] detector uses a watershed segmentation that starts from local intensity extreme, identifies connected regions and grows them over pixel intensity variations. The most stable regions over the intensity variations are considered as local features. A comprehensive review of local feature detectors can be found in [Tuytelaars 2008] and a comparison of their performance can be found in [Mikolajczyk 2005].

Affine covariant regions can be exploited to cope with the geometric and photometric deformation of images. Typically such regions have an elliptical shape while other approaches such as DoG [Lowe 1999] use fixed, circular support regions. The advantage of elliptical shaped regions over circular ones is illustrated in Figure 4.1. The key principle of the affine covariant region detectors is that the shape of the region is automatically adapted to underlying image intensities in a single image in such a way that regions detected independently in each image correspond to the same 3D surface patch. The size and shape of such regions transform in a covariant manner with a 2D image transformation. In most of the cases, an affine transformation is a reasonably good local approximation to transformations arising from viewpoint changes of locally planar surfaces. In the case of video content, typically most of the objects and characters are in motion and suffer different transformations. In the context of video object retrieval, this specific issue of the video content is a major problem that needs to be addressed. Affine covariant regions are a more suitable solution to this problem comparing to its DoG correspondent. Therefore, in our work we adopt the affine covariant regions.

Concerning the detector, the detailed study and comparison of affine covariant region detectors from [Mikolajczyk 2005] concludes that no detector outperformed the other ones in all experiments. Still, MSER and Hessian-Affine had consistently high scores. MSER perform well on images containing homogenous regions with distinctive boundaries but the number of detected regions is rather reduced comparing to Hessian-affine. Hessian-affine and Harris-affine provides more interest points/regions than other detectors, making them more suitable for identifying cluttered or occluded objects. As shown by Lindeberg [Lindeberg 1998], the Hessian operation has similar selection properties as the Laplace operator. Additionally, the Hessian operator allows simultaneous localization of scale-space maxima both in location and space [Perdoch 2009]. We have thus adopted the Hessian-affine region detector, detailed in the following section.

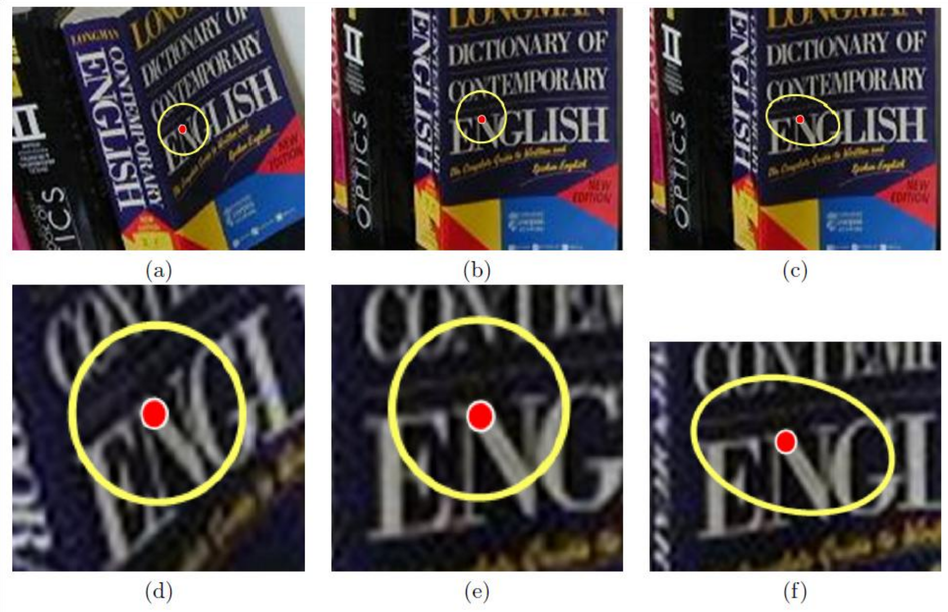


Figure 4.1. Limitation of circular support regions under large viewpoint changes [Mikolajczyk 2005]: (a) First viewpoint. (b)-(c) second viewpoint. The circular region in (b) does not cover the same object surface patch as the circular region in (a). What is needed is a deformation of the circular region in (b) by an anisotropic scaling to the ellipse shown in (c). Note that regions in (a) and (c) cover approximately the same surface patch on the book. (d)-(f) close-ups of (a)-(c) (illustration from [Mikolajczyk 2005]).

4.1.2 Hessian-affine detector

Interest points can be detected equally with the Harris detector or with a detector based on the Hessian matrix. For both cases, the scale-selection is based on the Laplacian and the elliptical region is determined with the second moment matrix of the intensity gradient.

The interest points and their scales are computed and selected with the help of the Hessian matrix. The algorithm searches over a fixed number of predefined integration scales $\sigma_1 \dots \sigma_n$, with $\sigma_i = k^i \sigma_0$ and $k = 1.4$ [Mikolajczyk 2004]. For each integration scale σ_I, an appropriate local differentiation scale σ_D is set with the help of a constant multiplicative factor s, as a function of the integration scale: $\sigma_D = s\sigma_I$, where $s = 0.7$.

Given image I, the interest points p are detected in the Gaussian scale space with the Hessian matrix $\mathcal{H}$. For each integration scale σ_I, the Hessian matrix is defined as:

$$\mathcal{H} = \mathcal{H}(p, \sigma_D) = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} = \begin{bmatrix} I_{xx}(p,\sigma_D) & I_{xy}(p,\sigma_D) \\ I_{xy}(p,\sigma_D) & I_{yy}(p,\sigma_D) \end{bmatrix}, \tag{4-1}$$

where $I(p)$ denotes the image intensity function and subscripts x and y indicate the partial derivatives with respect to the spatial coordinates.

The derivatives are computed at the current integration scale and are thus the derivatives of an image smoothed by a Gaussian kernel $g(\sigma_I)$.

$$I(p, \sigma_D) = g(\sigma_I) * I(p) \ . \tag{4-2}$$

The determinant and the trace of the Hessian matrix are computed as described in Equation (4-3) and (4-4).

$$\det\big(\mathcal{H}(p,\sigma_D)\big) = \sigma_I^2 \left(I_{xx}(p,\sigma_D) I_{yy}(p,\sigma_D) - I_{xy}^2(p,\sigma_D)\right). \tag{4-3}$$

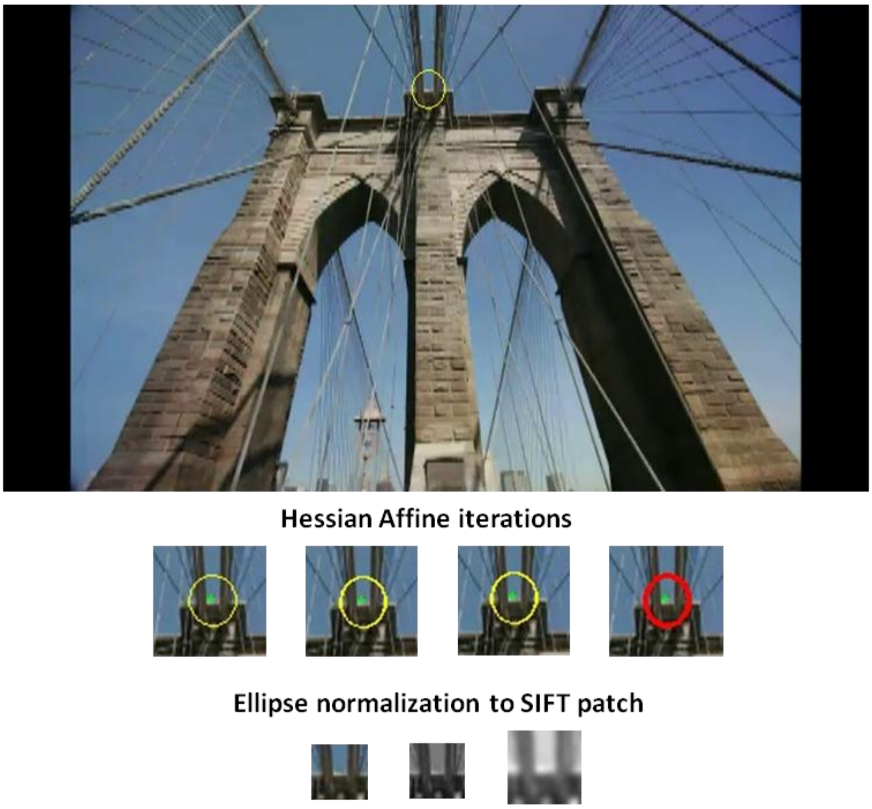
$$\operatorname{trace}\big(\mathcal{H}(p,\sigma_D)\big) = \sigma_D\big(I_{xx}(p,\sigma_D) + I_{yy}(p,\sigma_D)\big) \ . \tag{4-4}$$

The trace of this matrix is often referred to as the *Laplacian*. At each scale, the interest points are identified as the points that are simultaneously local extreme of both the determinant and trace of the Hessian matrix (*i.e.* a local maximum of the determinant decides if it is a blob/interest point, a local maximum in the trace decides it this is its characteristic scale). The determinant of the Hessian matrix penalizes very long structures for which the second order derivative in one particular direction is very small.

Furthermore, the Hessian Affine detector uses the detected points and their characteristic scale in an iterative shape adaptation algorithm to compute the local affine transformation for each interest point (*i.e.* elliptical affine region). The second moment matrix is employed for estimating the affine shapes. This matrix is also called auto-correlation matrix and describes the gradient distribution in a local neighborhood of a point. The matrix must be adapted to scale changes to make it independent of the image resolution. The eigenvalues of the second moment matrix are used to measure the affine shape of the point neighborhood, by computing the transformation that projects the intensity pattern of this neighborhood to one with equal eigenvalues. The affine region is skewed or stretched to a normalized circular region where the second moment matrix is isotropic. A new location and scale are detected in the normalized region. If the eigenvalues of the second moment matrix for the new point are equal, the estimation is correct. Otherwise, a new affine shape is estimated with the second moment matrix and tested.

The resulting shapes of the regions are adapted to the underlying intensity patterns and ensure in this manner that the same parts of different instances of the same region are covered despite the deformations caused by viewpoint changes or image rotations.

The Hessian-affine elliptical patches are warped around the detected blobs into a circular patch of 41x41 pixels and rotated based on the dominant gradient orientation to compensate for the affine geometric deformations. The principle is illustrated in Figure 4.2 for two images containing instances of the Brooklyn Tower Bridge.


Hessian Affine iterations
Ellipse normalization to SIFT patch

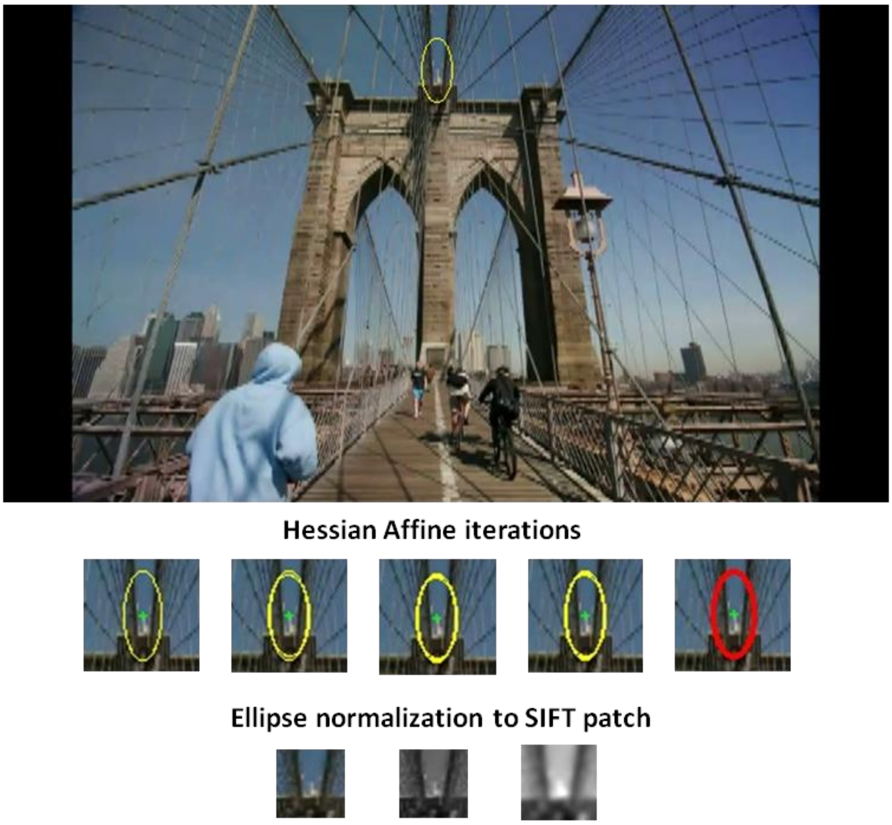

Hessian Affine iterations
Ellipse normalization to SIFT patch

Figure 4.2. Detection and description of Hessian-affine regions. The same region from the two instances of the Brooklyn Tower Bridge are detected. Note the change in the elliptical shape for the two instances. The ellipses are than warped to a squared patch which is then normalized to a 41x41 patch.

Once the interest points together with their associated local regions determined, the next step concern the description/representation of such image structures, which is based on the specification of local feature descriptors.

4.2 Local Feature Descriptors

Most of the local descriptors exploit the underlying information driven by the detectors and integrate it in the final description. Other local descriptors are completely independent from the considered interest point detectors. A multitude of local feature descriptors have been proposed over the recent years. The choice of the descriptor is dependent of the considered application, as each technique provides different levels of robustness and distinctiveness. Fairly comprehensive reviews of the local region descriptors and experimental evaluations and comparisons can be found in [Mikolajczyk 2004, Sande 2010].

Let us start our analysis with the most traditional and popular approaches which are today extensively used in a wide variety of applications.

4.2.1 Traditional approaches

Let us first mention the popular SIFT (*Scale Invariant Feature Transform*) descriptor, introduced by Lowe in [Lowe 2004], which consists of a gradient orientation histogram obtained from DoG points. The principle is illustrated in Figure 4.3. In its most popular form, the SIFT descriptor is a 128-histogram storing in each bin the magnitude of a local gradient in a certain direction, with every bin representing a direction. The SIFT descriptor is constructed from a 4x4 grid centered on the considered interest point. Each cell of the grid quantizes the gradient direction into 8 bins. A smoothing Gaussian function is added in order to emphasize the information in the close neighborhood of the interest points. In order to achieve the rotation invariance, all gradients within the patch are computed relative to a dominant gradient orientation, which is obtained as the highest peak in a histogram of all gradient orientations within the patch.

The SIFT descriptor has been experimentally shown [Mikolajczyk 2004] to outperform other descriptors like steerable filters [Mikolajczyk 2001], complex filters [Schaffalitzky 2002], and cross-correlation on raw pixel intensities, in the context of matching affine covariant regions. Such nice properties are due to its high dimensionality (when compared to filter banks representations) and to its tolerance to small localization errors, which often occur.

The main drawback of SIFT is however its relatively high dimensionality which increases the computation time during the matching step. In order to overcome this problem, Ke and Sukthankar [Ke 2004] applied PCA on the gradient image resulting in a 36-dimensional descriptor called PCA-SIFT. PCA-SIFT provides significantly faster matching, but is slower to compute and less distinctive than the original SIFT. Very recently, Jégou *et al.* [Jégou 2012] applied PCA on the SIFT descriptors to reduce them from 128 to 64 dimensions, with PCA rotation matrices learned from an independent dataset. While this method reduces the computational burden during the matching stage, in the case of the BoW-based object recognition, the performances are degraded.

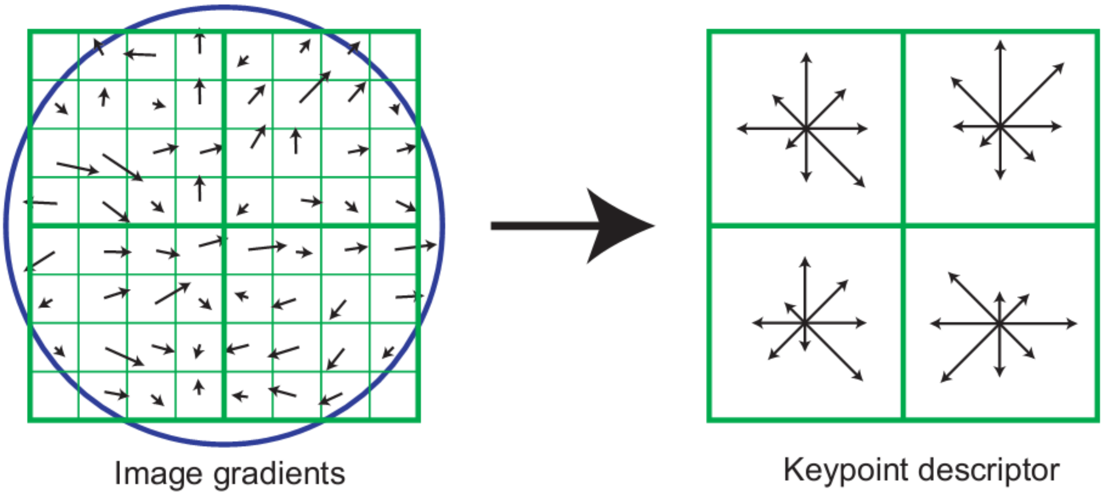


Figure 4.3. SIFT descriptor [Lowe 2004]. Image gradients within a patch (left) are accumulated into a coarse 4x4 spatial grid (right). In this example we show only a 2x2 grid. A histogram of gradient orientations is constructed in each grid cell. 8 orientation bins are used in each grid cell giving a descriptor with the dimension 128) 4x4x8 (illustration from [Lowe 2004]).

The Speeded-Up Robust Features (SURF) [Bay 2006, Bay 2008] have become one of the first alternatives to the SIFT descriptors. Its main advantage is the processing speed achieved by the use of the approximation of the Hessian matrix obtained by convolving the image with Haar wavelets, which at their turn are approximated by Gaussian second order derivatives. SURF makes an efficient use of integral images. The extraction of interest point locations and scales is performed simultaneously by maximizing the determinant of the scale-normalized Hessian matrix. The SURF representation is more compact than the SIFT descriptor and features a 64-dimensional descriptor. Similarly with SIFT, the SURF descriptor splits the support window into 4x4 square sub-regions. Experimental evaluation has demonstrated that the detection of SURF points is more than five times faster than SIFT's DoG detector, while the descriptor computation is three times faster than SIFT's. In the experiments of [Bay 2006], the SURF descriptor yielded higher performances in terms of matching comparing to its SIFT counterpart.

In recent years, the rapid technological advances of mobile devices, have lead to a variety of dedicated light-weight interest point descriptors, adapted to the computational capacities of such devices. Such descriptors are described in the following section.

4.2.2 Light-weight approaches

Typically, such descriptors take binary values. Among the most representative descriptors, let us mention the BRISK (*Binary Robust Invariant Scalable Keypoints*) [Leutenegger 2011], BRIEF (*Binary Robust Independent Elementary Features*) [Calonder 2010], ORB (*Oriented FAST and Rotated BRIEF*) [Rublee 2011], and FREAK (*Fast REtinA Keypoint*) [Alahi 2012] descriptors. Apart from their fast extraction procedures, such descriptors also offer very fast matching procedures with the help of distance measure adapted for binary values, such as the Hamming distance. This actually represents the biggest advantage of binary descriptors, as it replaces the more costly Euclidean distance.

The BRIEF [Calonder 2010] and ORB [Rublee 2011], descriptors build on the FAST keypoint detector [Rosten 2006] and compute the descriptor quickly as binary strings extracted by comparing the intensities of pairs of point along the same lines. While resistant to noise and rotation invariant (only for the ORB descriptor) these descriptors are not scale invariant, which is a major drawback for object instance recognition.

Similarly, Leutennegger et al. [Leutenegger 2011] build the BRISK descriptor bit-stream by considering a limited number of points in a specific sampling pattern, with each point scoring to multiple pairs. The pairs of points are divided in short-distance and long-distance subsets. The long-distance subset is used to estimate the direction of the keypoint while the short-distance subset is used to build the binary descriptor after rotating the sampling pattern.

Alahi *et. el* [Alahi 2012] propose the FREAK descriptor inspired by the retina of the human eye and compute a cascade of binary strings by comparing pairs of image intensities over a retinal sampling pattern. This makes it possible to sample points from a circular grid with a high density of points near the center and with exponential decrease, for selecting the most relevant Difference of Gaussians (Figure 4.4). The FREAK descriptor performs better than the rest of the binary interest point descriptors in terms of both extraction/matching time and accuracy. Yet, its matching performances are significantly lower than those of the SIFT descriptor, in particular when the scale variations become significant (less than 0.5 or greater than 2.5).

Meanwhile, within the context of large scale image and video search engines, several enhancements and enrichments of the SIFT descriptor have been studied. In particular, such extensions concern the inclusion of color information in the considered representations.

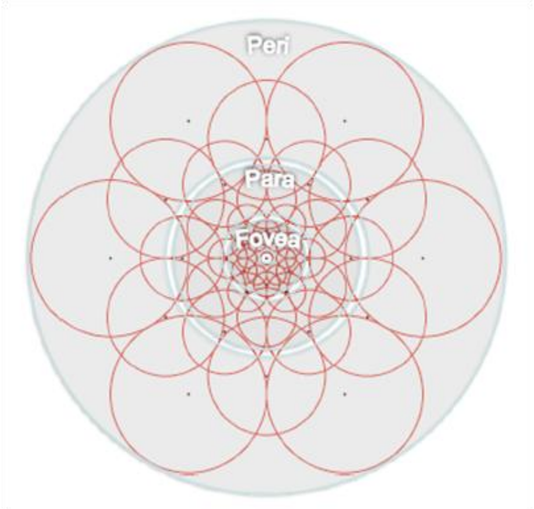


Figure 4.4. Illustration of the FREAK sampling pattern similar to the retinal ganglion cells distribution with their corresponding receptive fields. Each circle represent a receptive field where the image is smoothed with is corresponding Gaussian kernel (illustration from [Alahi 2012]).

4.2.3 Color-based extensions

Typically, local features and their descriptors are extracted by performing pixel intensity analysis. Researchers have explored the possibility of developing new SIFT-like descriptors employing the color information. In this respect, van de Sande *et al.* [Sande 2010] introduce the Opponent SIFT descriptor. OpponentSIFT describes all of the channels in the so-called opponent colors space with the help of SIFT descriptors. The opponent colors are defined as described in Equation (4-5):

$$\begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} \dfrac{R-G}{2} \\ \dfrac{R+G-2B}{\sqrt{6}} \\ \dfrac{R+G+B}{\sqrt{2}} \end{pmatrix} \tag{4-5}$$

Here, the considered color channels describe the color information in the image, while being invariant to changes in light intensity. Let us mention that other color-based SIFT descriptor have been equally proposed (*e.g.* Hue-SIFT [Weijer 2006], HSV-SIFT [Bosch 2008], C-SIFT [Burghouts 2009], RGB-SIFT [Sande 2010]). The benchmarking on local descriptors on object and scene recognition presented in [Sande 2010] demonstrates that color information improves the original SIFT descriptor with up to 8%. However, the study does not identify a descriptor that scores best in all the considered tasks. The choice of the descriptor still depends on the specificity of the task and of the available dataset. Authors recommend the use of OpponentSIFT when no prior knowledge about the data set is available and when the target object is also available.

A very recent extension of the SIFT descriptor, so-called RootSIFT, has been proposed by Arandjelovic and Zisserman in [Arandjelovic 2012]. The RootSIFT descriptor is described in the following section.

4.2.4 RootSIFT descriptor

In [Arandjelovic 2012], authors introduce a new distance measure for the SIFT descriptors, embedded in a new descriptor called RootSIFT. SIFT was originally designed to be used with Euclidean distance, but since it can be viewed as histogram with 128 bins, authors propose to measure the distance between SIFT descriptors with other, higher performance histogram measures such as the χ^2 or the Hellinger kernel. Given two n-dimensional vectors p and q with Euclidean norm, the Euclidean distance $d_{Euclid(p,q)}$ between them is related to their similarity/kernel $S_{Euclid(p,q)}$ as:

$$d_{Euclid}(p,q)^2 = \|p-q\|_{L2}^2 = \|p\|_{L2}^2 + \|q\|_{L2}^2 - 2p^Tq = 2 - 2\,S_{Euclid}(p,q)\,, \tag{4-6}$$

where $\|p\|_{L2}^2 = \|q\|_{L2}^2 = 1$ (p and q are L$_2$-normalized) and $S_{Euclid}(p,q) = p^Tq$ .The Hellinger kernel for two L$_1$ normalized histograms, p and q, is defined as:

$$S_{Hellinger}(p,q) = \sum_{i=1}^{n} \sqrt{p_i q_i}\ , \tag{4-7}$$

where $p_i \geq 0$ and $q_i \geq 0$, with $\sum_{i=1}^{n} p_i = 1$ and $\sum_{i=1}^{n} q_i = 1$. The Euclidean similarity is then replaced by the Hellinger kernel for the comparison of SIFT vectors as follows:

1. L$_1$ normalize the SIFT vector;
2. Square root each element. This means that $S_{Euclid}\left(\sqrt{p},\sqrt{q}\right) = \sqrt{p}^T\sqrt{q} = S_{Hellinger}(p,q)$, and the resulting vectors are L$_2$ normalized since $S_{Euclid}\left(\sqrt{p},\sqrt{p}\right) = \sum_{i=1}^{n} p_i = 1$.

Thus, the new RootSIFT descriptor is defined as an element-wise square root of the L$_1$ normalized SIFT vectors.

Arandjelovic and Zisserman [Arandjelovic 2012], have shown that comparing RootSIFT descriptors using Euclidean distance is equivalent to using the Hellinger kernel to compare the original SIFT descriptors since the following relation holds:

$$d_{Euclid}(\sqrt{p},\sqrt{q})^2 = 2 - 2S_{Hellinger}(p,q)\ . \tag{4-8}$$

The performance improvements in terms of object retrieval on the Oxford Buildings 5k and 105k [Philbin 2007] and Paris 6k [Philbin 2008] datasets brought by the RootSIFT descriptor comparing to SIFT are up to 7% in Mean Average Precision [Arandjelovic 2012], in all the experiments performed.

An illustration of the performances of SIFT and RootSIFT matching between 2 images is illustrated in Figure 4.5. The query image and region are shown in the left of each pair and the matching result with the estimated corresponding region of interest is shown on the right. RootSIFT identifies more matches and the object localization is improved.

As remarkable results, let us note that the RootSIFT without spatial re-ranking (*cf.* Section 4.4.4) outperformed SIFT with spatial re-ranking. Because of such highly promising recognition performances, we have adopted the RootSIFT descriptor in our work.

Once the RootSIFT description is computed, the following stage concerns the vector quantization/clustering of the obtained descriptors.

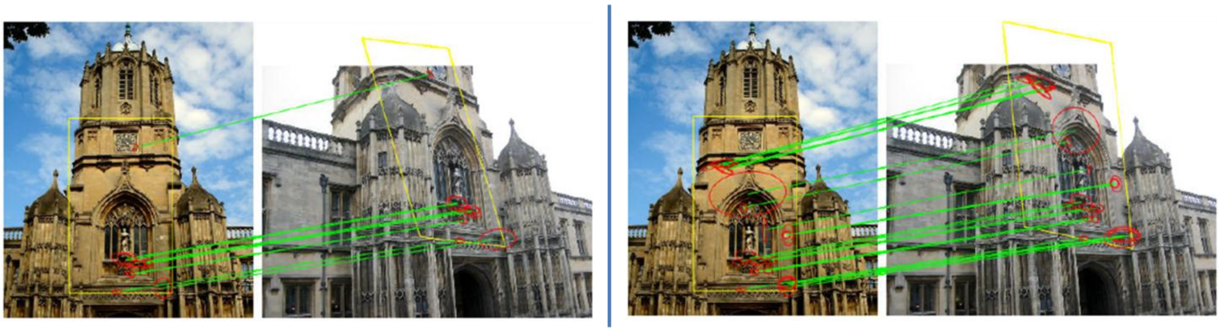


Figure 4.5. Comparison between SIFT and RootSIFT matching performances. On the left: SIFT with L2 distance - 10 matches. On the right: RooSIFT - 26 matches (illustration from [Arandjelovic 2012]).

4.3 Clustering visual descriptors

In order to reduce the number of considered features and optimize the search time, large scale image or video retrieval systems cluster the high-dimensional descriptors, keeping only a limited set of descriptors that form a vocabulary of visual words. Additionally, the role of the vocabulary is to establish potential correspondences between local image regions. By analogy with text retrieval the visual words are then used in a similar manner as text words and the standard text-retrieval methods are directly used in this vision case. Once the vocabulary is generated and all the regions from the dataset are labeled with their corresponding visual word, all regions having the same label are considered matched. A major advantage of this approach is the efficiency boosting. Different terms are used in literature for naming this structure: (visual) vocabulary, codebook, or dictionary.

While enhancing the efficiency in the run-time phase, the clustering is typically the most time computationally demanding part of the offline phase. Clustering very large collections of high dimensional features (*e.g.*, N = 245M descriptors of 128 dimensions for the TRECVID 2012 INS dataset) is a major challenge for any clustering algorithm. Even sub-sampling 10% of such dataset, would still require the clustering of 24M 128-dimensional descriptors.

Early systems used a flat k-means clustering [Sivic 2003] that was effective but impossible to scale to large vocabularies. A large amount of research has been targeting the scaling of the clustering algorithms. The biggest step towards this direction was made by Nister and Stewenius [Nister 2006] which introduced the *vocabulary tree* obtained from hierarchical k-means (HKM) clustering and brought significant improvements in the retrieval accuracy. Here; the problem of clustering a large pool of descriptors in multiple clusters is divided in smaller clustering problems organized in a tree hierarchy. On the first level of the tree, all data points are clustered using k-means to a small number (*e.g.* $K = 10$) of cluster centers. On the next level, k-means with the same $K = 10$ is run again within each of the previously determined partitions independently and so on. The result consists of K^n clusters at the n^{th} level of hierarchy. For example, using a branching factor of 10 with 6 levels will results in 10^6 leaf nodes. This principle is illustrated in Figure 4.6.

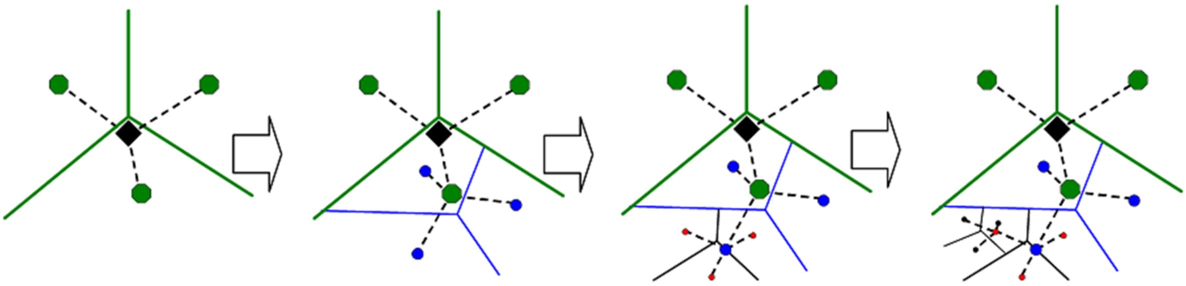


Figure 4.6. Building a vocabulary tree using HKM. The hierarchical quantization is defined at each level by *k* centers (here *k* = 3) and their Voronoi regions (illustration from [Nister 2006]).

An advantage of the vocabulary tree approach is that it allows more complex assignments for the visual words labeling. For example, instead of assigning each data point to a single leaf node at the bottom of the tree, the points can be assigned to some intermediate nodes included in the path from the root to the leaf node. Thus, the effects of quantization errors can be diminished in the case when the data points lies close to the Voronoi region boundary for each cluster center. Due to its reduced complexity, the method can scale to very large numbers of cluster and feature points (*i.e.*, more than 1 million visual words). Very recently, Zu and Satoh [Zhu 2012] updated the hierarchical k-means algorithm to be used on even larger dataset (*e.g.*, N=2.3x10^9 descriptors). For each sub-tree of the vocabulary, they sample a fraction of the features for clustering, while propagating the rest down to the sub-tree. Since the propagation of the non-sampled descriptors to their corresponding sub-trees is faster than clustering, the computational burden is largely reduced. An advantage of the HKM and vocabulary tree is the hierarchical scoring. By considering a set of nodes from several levels in the similarity score, and by weighting the contribution of each level to the score with an entropy factor, the quantization error can be reduced. Similarly, the vocabulary tree can be used jointly with Pyramid Matching [Grauman 2005] which can compute a similarity score over multiple layers in the vocabulary tree as proposed in [Zhu 2012]. The number of levels to be taken in consideration for the scoring depends of the application and the on the size of the vocabulary. For object instance retrieval and large vocabularies (over 1 million), solely two levels are usually selected.

Philbin *et al.* [Philbin 2007] introduce the Approximate k-means method (AKM), which is a scalable version of the k-means algorithm. While in the exact k-means, most of the computation time is spent on retrieving nearest neighbors between feature points and cluster centers, the AKM optimizes this step with a so-called approximate nearest neighbor method. Authors employ a forest of several randomized k-d trees, as proposed in [Lepetit 2005, Muja 2009], which are built over the cluster centers at the beginning of each iteration in order to increase the processing speed. Unlike regular k-d trees [Friedman 1977], where each node splits the dataset using the dimension with the highest variance, in the case of randomized k-d trees the splitting dimension is chosen randomly from the set of dimensions with the highest variance. The union of such trees creates an overlapping partition of the feature space and helps to diminish the quantization effects, where features which fall close to a partition boundary are assigned to an incorrect nearest neighbor.

Philbin demonstrates in [Philbin 2010] that AKM exactly minimizes the k-means cost function. In addition, for moderate values of K, the percentage of points assigned to different cluster centers differs from the exact version by less than 1%. Moreover, when compared to the HKM approach, as demonstrated experimentally in [Philbin 2007, Philbin 2010], the AKM method consistently outperform HKM on multiple experiments, even when using hierarchical scoring.

In our work, we have adopted the AKM approach because of its nearly optimal performances. The AKM is easily parallelizable and its distributed memory computed can be achieved with the open source MPI library [MPI 2012]. For a dataset of N = 24 million descriptors, the clustering into K = 1 million clusters on a 4 core machine took less than one day. A publicly available implementation of AKM is available from [Philbin 2012].

Another issue to be addressed is the size of the vocabulary compared with the size of the dataset, as no clear recommendation exists. In general, the greater the size of the vocabulary, the better the object detection/recognition results obtained. Philbin [Philbin 2010] tested vocabulary sized to 2 million visual words, while Nister and Stewenius [Nister 2006] reached 16 million points for a vocabulary size. For a dataset of 16 million points, Philbin obtained a peak in performance around 500,000 to 1,000,000 clusters. Using more data for clustering seems to help, but the performance increase is less significant than for varying K. In his case the retrieval performance lowered as the vocabulary size passed 1 million prototype descriptors. Nister and Stewenius [Nister 2006] experimented with several vocabulary sizes and concluded that for a large range of vocabulary sized (up to 1 and 16 million leaf nodes) the retrieval performance increases with the number of leaf node. Authors recommend large vocabularies for improving retrieval provided that the weak weights are given to the inner nodes of the vocabulary tree, as the leaf nodes are much more powerful than the inner nodes. The issue is then posed as choosing between prioritizing the retrieval accuracy or the computational performance (which can vary dramatically when shifting from 1 million to 16 million).

Selecting the size of the vocabulary is today still an open issue. In general, the size of the vocabulary can be tuned to the required applications: smaller vocabularies can capture intra-class variations for recognition of object categories, while larger vocabularies are more suitable for object instance recognition. In order to ensure an accurate recognition, within reasonable computational times we have limited the size of the vocabulary to 1 million nodes.

Let us note that the management of such large vocabularies represents a computational challenge. Usually the high dimensional vocabularies (1M visual words of 128 coefficients) are stored in binary files and data is fetched by reading the file by small chunks of data. In our implementation we employ the HDF5 (Hierarchical Data Format) file format [HDF5 2012] adapted for the organization of large amounts of numerical data. Practically, the data is stored as compressed table files and its elements can be accessed easier offering multiple other uses in the same time. HDF5 displays a C++ API and the size on disk of a vocabulary containing 1 million RootSIFT descriptors is approximately 420 MB.

4.4 Quantization and matching

Once the vocabulary generated, the local features from all frames in the dataset are mapped onto the vocabulary and BoW representations are computed for each frame. Typically, each local feature is assigned to its best match from the vocabulary (*i.e.*, the visual word yielding the highest similarity from all vocabulary matches). Two image features are considered identical if they are assigned to the same visual word, while two features assigned to different visual words are considered as completely different. These mappings of each point to the vocabulary are used to build a vector of visual word relative frequencies for each image.

However, instead of directly using such a histogram-based representation, the components of the visual word vector are weighted in order to overcome biases related to the uneven number of visual words per image, or to the influence of frequently encountered visual words, with poor discriminative power. This weighting mechanism, described in the next section, adopts the well-known *tf-idf* scheme, largely exploited within the framework of textual-based retrieval.

4.4.1 The *tf-idf* matching scheme

The standard weighting scheme for the BoW representation is adapted from text retrieval and is known as “term-frequency – inverse document frequency” (*tf-idf*) [Baeza 1999]. Given a vocabulary of *K* words, each image *f* is represented by a vector of weighted visual word frequencies:

$$V_f = (t_1, \dots, t_i, \dots, t_K) \ . \tag{4-9}$$

The t_i components are defined as described in the following equation:

$$t_i = tf \times idf = \frac{n_{if}}{n_f} \log \frac{N}{N_i} \ , \tag{4-10}$$

where n_{if} is the number of occurrences of the visual word i in image f, n_f is the total number of words in the frame f, N_i is the number of images containing word i and N is the number of documents in the whole database.

The weighting is a product of two terms: the *word/term frequency (tf)* expressed as:

$$tf = n_{if}/n_f \tag{4-11}$$

and of the *inverse document frequency* (*idf*) defined as:

$$idf = \log N/N_i \tag{4-12}$$

The role of the word frequency term is to assign a higher weight to the words occurring more often in a given image. Usually, the frequency term is normalized to the total number of interest points, in order to avoid biases in the case of images with higher numbers of interest points. The inverse document frequency term gives a lower weight to the words that appear highly frequently in all the images of the dataset and do not help in discriminating between different documents.

The similarity between two images is then expressed with the help of a histogram similarity measure. The *tf-idf* BoW representations of the images to be compared are considered as histograms with equal number of bins (*i.e.*, the size of the vocabulary). The most popular measures used to compare frames are the cosine measures (Equation (4-14)) as well as the L_p distances (which are, most often, L_1 and L_2 distances) (Equation (4-13)). Thus, the similarity of two images in the data set described by *tf-idf* vectors *p* and *q* can be computed as follows:

$$d_{Lp}(p,q) = (\sum_i |p_i - q_i|^p)^{\frac{1}{p}} \tag{4-13}$$

$$d_{cos}(p,q) = \frac{\sum_i p_i \cdot q_i}{\sqrt{\sum_i p_i^2} \cdot \sqrt{\sum_i q_i^2}} \tag{4-14}$$

Note that when the vectors are normalized using the L_2 norm, the terms $\sqrt{\sum_i p_i^2}$ and $\sqrt{\sum_i q_i^2}$ from Equation (4-14) are actually the L_2 norms of vectors *p* and *q* and are equal to 1. The L_2 distance from Equation (4-13) can then be then rewritten as a function of the cosine distance as follows:

$$d_{L2}(p,q) = 2(1 - d_{cos}(p,q)) \tag{4-15}$$

Therefore, if vectors *p* and *q* are pre-normalized with respect to the L_2 norm, ranking in increasing order of $d_{L2}(p,q)$ is equivalent to raking in decreasing order of dot products $p \cdot q$. In practice, vectors *p* and *q* are very sparse, notably in the case where large vocabularies are used, and this dot product can be computed very quickly by only considering the non-zero from the vector with the lowest number of non-zeros term.

Other variants of the typical *tf-idf* scheme and histogram matching, leveraging on popular text retrieval techniques have been also proposed in [Yang 2007b, Sivic 2009, Tirilly 2010]. Yang *et al.* [Yang 2007b] propose and test different normalization and weightings for the *tf* and *idf* terms, while Tirilly *et al.* [Tirilly 2010] perform an in-depth study of the probabilistic weighting models such as BM25 [Robertson 1977], which weights the *tf* term by considering that word occurrences are distributed following two Poisson distributions and the *idf* term according to the probability ranking principle which states that results should be ranked according to their probabilities of relevance with the query. In the latter, different variants of the L_p histogram distances from Eq. (4-13) have been tested as well. Authors concluded that the choice of the technique depends on the dataset and its size, on the size of the vocabulary and on the use case.

Sivic and Zisserman [Sivic 2009] propose for testing multiple weighting schemes (including different normalizations of the *tf* and *tf-idf* vectors) with corresponding similarity measures, such as L_1, L_2, χ^2 [Leung 2001], Kullback–Leibler (KL) divergence [Varma 200], Bhattacharyya distance [Aherne 1998]. Their experiments demonstrated that the standard *tf-idf* and Bhattacharyya ranking lead to the best scores, followed closely by the KL divergence.

Since such studies show that no technique clearly overpasses the other ones, in our work we adopt the standard *tf-idf* weighting. We perform a L_2 pre-normalization of the frame vectors in order to optimize the matching times.

When computing the *tf-idf* vectors, each point is assigned to its best match from the vocabulary. In some cases, this hard assignment leads to errors because of variability in the feature descriptor. Even though SIFT like descriptors are invariant or robust to several image alterations, in some cases image noise, illumination changes or other non-affine changes of the images can strongly affect the descriptor value. As a result the same patch can be assigned to different visual words in different images.

4.4.2 Hard versus soft assignment

In order to overcome this drawback, several methods [Philbin 2008, Gemert 2010] propose to improve the quantization process with so-called *soft assignments*. The principle consists of considering for each point multiple matches from the vocabulary. In this case, the *tf* weights assigned depend of the corresponding similarity scores. A second approach consists of inserting in the representation additional information about the relative position of the points from the cluster center in the Voronoi cells [Jégou 2008]. Here, the position of the current point in the Voronoi cell with respect to its visual word correspondent is mapped onto a Hamming space and added to the BoW histogram as a binary signature.

Such methods enrich the BoW representation and improve the search and retrieval performances. The price to pay is the significant increase in terms of storage requirements as the BoW vectors become denser. Additionally, the dense BoW vectors lead to higher matching times. However, recent experimental studies [Jégou 2010, Arandjelovic 2012] have shown that if solely the query descriptors are softly assigned the results can be significantly improved. In this case, the database images continue to be hard assigned and their storage requirements are unchanged.

In our work, we have considered both hard and soft assignments of the query descriptors. For the soft assignment, for each query point descriptor we consider the first five nearest neighbors from the visual vocabulary. The bins corresponding to the five visual words are weighted with the similarity between the query descriptor an each visual word considered (*i.e.*, $1 - d_{Euclid(p,q)}$ as described in

Equation (4-6)). Hard quantization errors are thus overcome for most of the points as multiple correspondences from the vocabulary are considered for each point. The correspondences between each descriptor and the descriptors from the vocabulary are computed with the fast approximate nearest neighbors matching method (FLANN) proposed by Muja and Lowe [Lowe 2004].

In order to speed up the retrieval process in the case of large databases, we have adopted an inverted index structure, described in the following section.

4.4.3 Inverted index

An inverted index file is a commonly used indexing structure borrowed from text retrieval applications, which is similar with a book index. The inverted index has an entry for each word in the corpus, followed by a list of all occurrences in the items (images/videos) of the considered dataset. For each word, the number of occurrences and the identifiers of the items where the word is occurring are stored. Depending on the computations required at run-time, the inverted index can be more complex and additionally store for each occurrence the position in the corresponding frame and the distance to the 15th nearest neighbor in the image [Sivic 2009] (for fast spatial verification), the position and the coordinates of the elliptical shape of each point [Philbin 2010] (for fast geometric verification). Jégou *et al.* [Jégou 2007] employ a tree-based inverted index for fast search, while Nister and Stewenius [Nister 2006] build virtual inverted files for higher levels of the vocabulary in order to efficiently parse their vocabulary tree.

The utility of the inverted index is emphasized the most at run-time, when the user selects a query region which contains a set of visual words. The query visual words are searched in the inverted index and are used to generate a list of plausible candidates, by selecting only images that contain at least an occurrence of one query visual word. In this manner, the number of images to be compared with is significantly reduced.

A final issue involved in the matching process concerns the re-ranking of the top retrieved candidates, which exploits geometric/spatial localization information.

4.4.4 Geometric consistency check and re-ranking

The standard BoW representation discards all spatial information relative to the geometric relative position of the visual words in the images. In order to inject spatial information in the retrieval process, numerous approaches employ the RANdom SAmple Consensus (RANSAC) algorithm [Fischler 1981]. RANSAC is a robust method for model fitting to noisy data with outliers. The main idea is to randomly sample a minimal set of correspondences, and compute the aligning transformation.

The data consists of "inliers", *i.e.*, data whose distribution can be explained by some set of model parameters, and "outliers" which are data that do not fit the model. Transformations are scored by the agreement with the remaining correspondences. A correct transformation should have a large support from other correct correspondences, whereas an incorrect transformation would be consistent with only a small number of correspondences. RANSAC is employed for checking the top results of a query and to re-rank them according to their spatial consistency with the query. Starting from a list of correspondences (at least four) between the two frames, the algorithm generates transformation hypotheses and then evaluates each hypothesis based on the number of "inliers" among all features among that hypothesis. In this way, the RANSAC algorithm estimates a transformation of the query image and each target image, based on how well its feature locations are predicted by the estimated transformation. In general, RANSAC manages to find mismatches provided that the initial list of inliers contains at least four good matches.

The original RANSAC has been extended in [Chum 2005], which proposed the PROSAC (Progressive Sample Consensus) method that weights correspondences by employing an external measure of confidence, which is used as a priority for guiding the search process towards optimal solutions. This method has been further improved by integrating its sampling strategy into a real-time robust estimation framework called ARRSAC (Adaptive Real-Time Random Sample Consensus) [Raguram 2008]. Another RANSAC variant is GroupSAC [Ni 2009], which partitions points into groups based on similarity information. In LO-RANSAC [Chum 2003], a re-estimation step was introduced into RANSAC, which was shown to generate improved hypothesis with more support. In their large scale object retrieval framework, Philbin *et al.* [Philbin 2007] discard RANSAC as the estimation of a full 3D fundamental matrix or 2D projective homography between two images is too general and it runs very slowly. Instead they employ LO-RANSAC [Chum 2003]. The approximate model considered here is built iteratively from single pairs of correspondences verified through a class of transformations of the affine-invariant regions corresponding to the matched points.

Another type of approaches encode the spatial information in the BoW representation or in the matching process. Lazebnik *et al.* [Lazebnik 2006] proposed a spatial pyramid matching in order to encode and partition the image into sub-regions at different levels of detail, from coarse to fine. The image can be then represented with multiple local histograms concatenated in a single global image histogram and each local histogram corresponds to an image sub-region. The local histograms are weighted in such a manner that matches identified in larger cells are penalized as they involve increasingly dissimilar features. The principle is similar with the pyramid matching kernel introduced by Grauman and Darrell [Grauman 2005]. Vedaldi *et al.* [Vedaldi 2009] propose the use of dense and sparse visual words at different levels of spatial organization.

The approach proposed in [Philbin 2007] is the most suitable for our framework as it is fast, effective and can generate transformation hypotheses even with a single pair of corresponding features, which is of great use when searching for object instances and partial matching is quite often. The number of possible hypotheses to be considered is thus reduced significantly and the matching procedure is hence speeded-up. The randomness in the generation of the hypothesis specific to RANSAC is removed and the procedure becomes deterministic. For the evaluation of the consistency of single pairs of matched points (correspondences), we consider the elliptical representations of the local regions, which are transformed with the help of a 5 degrees of freedom transform introduced in [Philbin 2007], which takes into account for translation, anisotropic scaling and vertical-preserving shear.

A loose threshold on the inlier distance (an inlier is a spatially verified matched point) is also exploited. Such a loose threshold allows the matching of frames with significant perspective distortions. The transformation between the points of the single correspondences is used for generating a hypothesis and applied to the rest of the correspondences. The correspondences within the threshold are considered to be verified for this hypothesis and added to a list of verified inliers. The inliers from the list are then used to re-estimate a full affine homography with the least-squares method. In practice this step can be discarded as accurate results can be obtained also by simply counting the verified inliers for each hypothesis and selecting the one with the highest number of inliers.

The elliptical regions are constrained to be oriented "up" as it is a good assumption for videos to be filmed in without significant rotations in the viewpoint. The transform is computed from a single correspondence of two elliptical regions C_p and C_q from the test image and query region. The region centroids are used to compute the translation, while the affine 2 x 2 sub-matrix H_{qp} is computed as $H_{qp} = H_p^{-1} H_q$ (Figure 4.7), where H_p and H_q project the ellipses to a unit circle such that the orientation of the unit vector in the y direction (*i.e.* "up") is maintained. The transformation considered in this case is modeled with the following functional matrix:

$$A = \begin{bmatrix} a & 0 & t_x \\ b & c & t_y \\ 0 & 0 & 1 \end{bmatrix} \qquad (4\text{-}16)$$

This geometric consistency check technique is used in the following manner. After the first part of the search stage, where the distances between the BoW representations of the query frame and the image candidates have been computed, a ranked list of frames containing plausible instances of the queried object is generated.

The geometric consistency check and re-ranking is performed on the top N frames. First, we perform a matching between the interest points of the query frame and the test frame. In order to avoid the inconvenience of the quantization errors, we perform the matching directly between the raw descriptors of the image. The frame matching is speeded-up with the fast approximate nearest neighbors algorithm (FLANN) proposed in [Muja 2009]. The non-discriminative matches are filtered out with the ratio test proposed by Lowe [Lowe 2004]. Once the list of candidate matches computed, we check them iteratively by employing the 5 degrees of freedom transform. This transformation is considered as a hypothesis (as in RANSAC) and applied to all matched points from one frame, projecting them into the other frame. The points whose projections are localized close to their corresponding points are considered as inliers and added to the list of verified matches. The configuration yielding the highest number of inliers is returned and the N considered frames are re-ranked in decreasing order of the number of verified inliers.

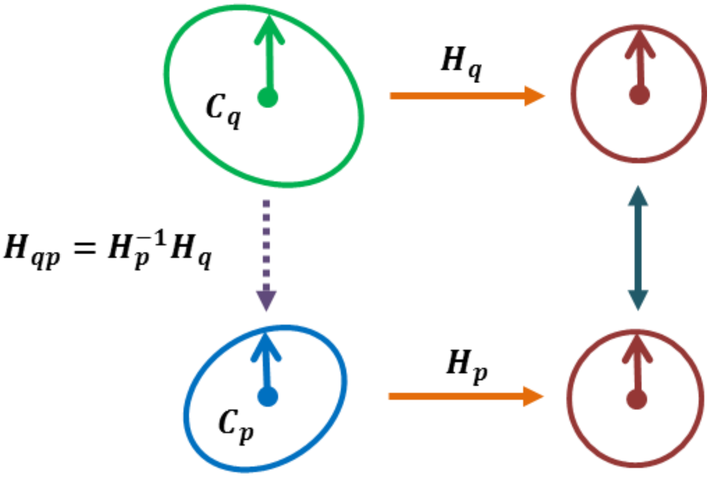


Figure 4.7. Computing $H_{qp} = H_p^{-1} H_q$ by projecting ellipses C_p and C_q to a unit circle and preserving the "up" orientation

The BoW has been proven to be a very effective solution for large scale image retrieval, with successful results of datasets reaching even 100 million images [Jégou 2012]. Its effectiveness has determined scientists to apply the BoW framework to video content.

4.5 Video framework: shot based BoW representation

When considering the issue of object search and retrieval in video databases, a set of specific questions need to be addressed and solved: how to divide the video content into basic units? How many video frames need to be considered? What is the optimal video resolution to use? How to avoid blurry and noisy frames, such as those corresponding to gradual transition effects? How to size the descriptor according to the length of the videos and the size of the dataset?

In the last 2 years, such problems have been addressed within the TRECVID evaluation campaign [Smeaton 2006], namely in the Instance Search Task. This task brings in the spotlight the challenge of retrieving different instances of the same object in a large scale dataset (up to 75,000 video clips) starting from a couple of query images. While there have been some encouraging results in the task [Zhu 2012], the problem remains open.

The problem of blurry and noisy frames has been tackled by Sivic and Zissermann [Sivic 2003] where frames are sampled densely from the video and the information of the detected regions is aggregated over a sequence of frames. The regions are tracked over consecutive frames and any region which does not survive for more than three frames is rejected. The estimate of the descriptor for each region is then computed by averaging the descriptors throughout the sequence. Such methods are computationally expensive and the dense sampling of the frames adds up significantly to the number of BoW vectors to compare. The principle is illustrated in Figure 4.8. The descriptors from each sampled frame are here quantized.

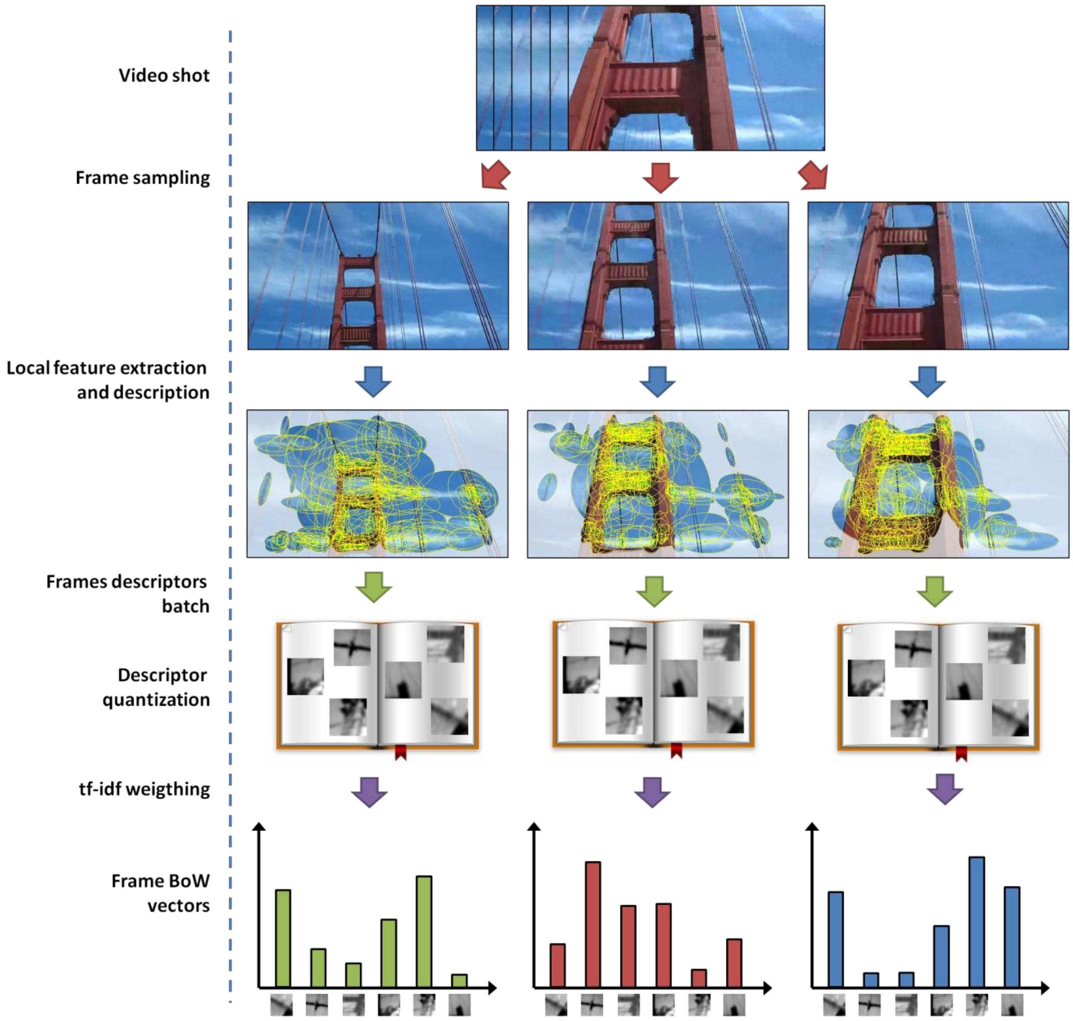


Figure 4.8. Frame-based BoW quantization.

The most popular choice for the selection of the representative frames for a given video is based on a shot boundary detector, as recommended in [Snoek 2008b]. Here, videos are segmented automatically in video shots by comparing consecutive video frames with the help of color-based descriptors. Each video shot determined can be then represented by a key-frame, which is typically the central frame of the shot. While the problem of noisy frames is not solved in this case, the reduction of dimensionality given by such methods has made them the most popular video representation techniques in the TRECVID tasks [Smeaton 2006].

Recently, Zhu *et al.* [Zhu 2012] proposed a very dense sampling of the video frames (3 frames per second) and a BoW vector computation for each video shot. This approach has recorded the best scores for the TRECVID Instance Search Task of 2011. The high number of sampled frames per video overcomes the noisy frames drawback, as the chances of fetching inlier, noise-free frames are increasing. Yet, the computational cost of the method is significant, since such a high number of sample video frames is not always necessary, as in the case of shots with poor motion activity.

We propose an approach which is similar with [Zhu 2012]. Given a video shot, we perform a uniform sampling of 1 frame per second. The sampled frames are refined by rejecting the near-duplicate frames. This is highly useful for static video sequence, where successive frames are very similar and do not bring additional information. The near-duplicate frames can be computed quickly with global descriptors such as MPEG-7 ColorStructure descriptor [MPEG 2003] and color histograms [Bursuc 2011] or with more advanced techniques employed for shot boundary detectors [Tapu 2011].

The video dimensionality is thus drastically reduced. The regions and descriptors detected in the sampled frames are then quantized together into a single BoW vector, associated to the entire video shot. This step reduces significantly the number of BoW vectors to be considers and hence boosts the computational speed in the search stage. For the TRECVID INS 2012 dataset the number of BoW vectors was thus reduced from 683,433 (the total number of extracted frames) to 74,958 (the total number of video shots). The principle is illustrated in Figure 4.9. The descriptors from all video frames are projected into a pool of descriptors along with the rest of the descriptors and a single BoW vector is computed after the quantization. Note that in this case, three views of the same object (*i.e.*, Golden Gate Bridge) are integrated in the same representation. This provides additional robustness to viewpoint changes.

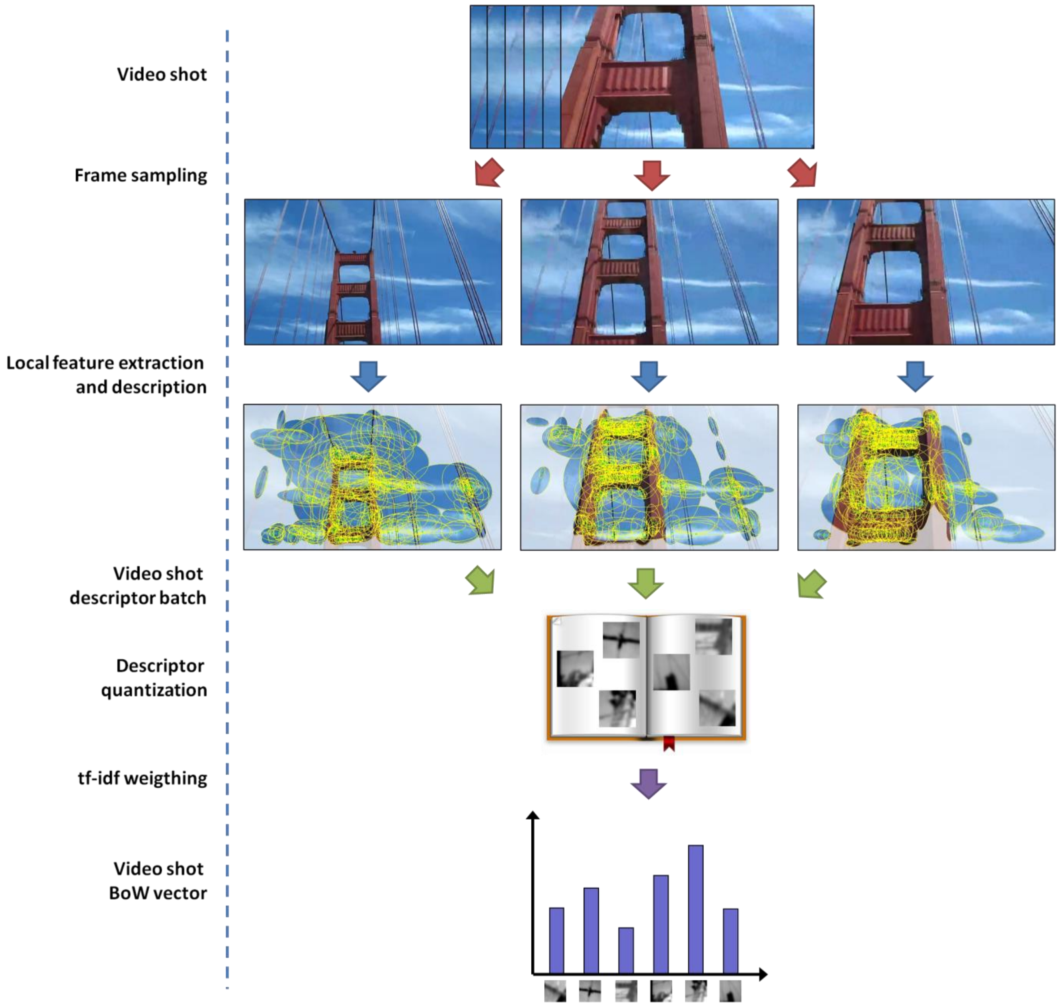


Figure 4.9. Shot based BoW quantization.

In order to improve the accuracy of the retrieval process, a variety of methods adopt the query expansion paradigm, specific to the domain of text retrieval [Chum 2007]. In the following section, we introduce a novel query extension technique.

4.6 Query expansion

The underlying principle of query expansion methods can be stated as follows. An initial user-selected query is formulated, returning an initial list of results. Then, the first ranked documents are selected from this list and used to construct a new, richer query, which contains additional terms, relevant to the intended query. The enriched query is then re-issued and the results returned to the user. If properly used (*i.e.*, if noisy inliers and false positives are not added to the new query), the technique can boost the recall of the retrieval, and as a collateral effect, its precision.

In [Chum 2007] the query expansion paradigm has been for the first time exploited within the context of object-based image retrieval. Here, the query is enriched with geometrically verified points from the top ranked results obtained by performing a RANSAC consistency check [Fischler 1981, Sivic 2003]. A set of new BoW vectors are generated from the verified matches and the new query vector is obtained by averaging the initial query vector with the verified BOW vectors. A new enriched query, embedding information that might have been missed in the initial query (*e.g.*, different object pose), is then presented to the system. The obtained results display a significantly higher recall rate. Up to 50 top results are considered during the expansion and the technique can be iterated several times.

An improved version of this method has been recently proposed in [Chum 2011]. Here, the geometric verification and the re-ranking build a statistical model of the query object. In addition, the relevant spatial context is learned in order to further improve the retrieval performance.

Arandjelovic and Zisserman [Arandjelovic 2012] introduce another improvement of the query expansion from [Chum 2007], so-called discriminative query expansion. Once the expanded BoW vectors computed, they are used as positive training data to train a linear SVM classifier [Boser 1992, Cortes 1995]. Images with low *tf-idf* similarity scores are used as negative examples in the learning process. The weights learned by the classifier from the small training dataset are then used to re-query the database and to rank images according to their distance from the decision boundary.

The main drawback of such methods is related to their dependency on the quality of the first query considered and of the first retrieved results. If no good matches are found in the first result set, the query can be altered and the recall rate will be degraded.

We propose a novel approach for query expansion that does not rely on the first retrieved results, but instead leverages on the vast amount of manually annotated content from image portals and search engines such as Flickr [Flickr 2012] and Google Images [Google 2012c]. Relevant content can be retrieved with text-based queries and the retrieval results and data from these portals can be accessed automatically through their APIs. In our approach, we employ such data in order to enrich the existing query object or to generate a set of visual queries starting from a textual description of the query object.

Our approach is based on the assumption that an elementary textual description is available for each query and is inspired by a use-case considered in the TRECVID 2012 Instance Search Task. Here, a collection of queries consisting of multiple example frames containing the object of interest is provided. In addition, each query includes a simple textual description of the object to be retrieved (*e.g.*, “Brooklyn bridge tower”, “Hagia Sophia interior”). We propose to leverage on these textual descriptions (which contains keywords commonly used for a textual search) in order to retrieve additional content that can be exploited for enriching the given queries. Let us note that, in a more general case, the given visual queries can be completely discarded and a visual query can be constructed solely from a set of textual descriptions. Let us also underline the multimodal character of this query technique: text descriptions are used for retrieving images, which are used at their turn for retrieving video content.

The approach consists of the following successive steps, also illustrated in Figure 4.10:

1. Issue a text based query on a web image search engine and download the first $N_{download}$ results.
2. Extract local features from the retrieved images and perform a one-to-one, exhaustive interest point matching, together with the geometric consistency verification between all images. Build then a query graph using the retrieved images as nodes. The graph edges are used here to connect images consistently matched. Identify different instances of the query object as connected components in the graph.
3. Determine the most representative image from each connected component,
4. Build query descriptors using the representative images together with their best matches,
5. Aggregate query descriptors, and formulate queries on the video dataset in order to retrieve relevant videos.

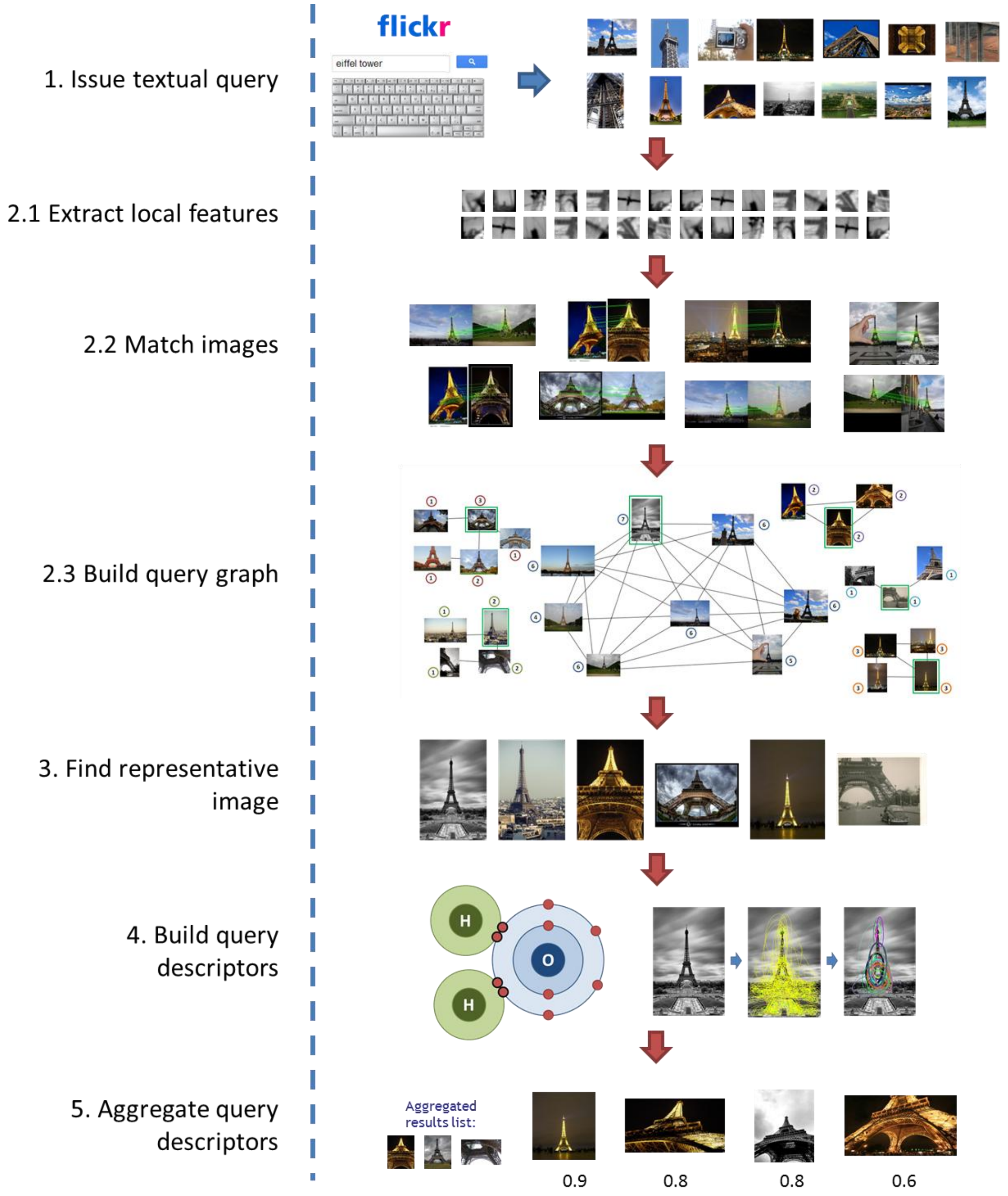


Figure 4.10. Multi-modal query expansion work flow.

The various steps involved are described in details in the following sections.

4.6.1 Text based queries on web search engines

Most of the general public image search engines provide public APIs which allow automatic querying and download of results. In our case, we employed the Flickr API which is the most popular choice for many datasets [Philbin 2007, Philbin 2008, Jégou 2008, Kalogerakis 2009, Kang 2011].

Concerning the settings related to the search parameters, we have experimented with both tag-based search (the query keywords are matched to the tags given by the owner of the image) and text-based search (the query keywords are matched with the words from the textual description/title of the image given by the owner). In addition, for the tag-based search we have tested different constraints on the number of query keywords to be matched (to match all keywords or to match a part of the keywords). We have noticed that the tag-based search provides less relevant results and it is more suitable for class or category recognition datasets. For example, a search for the keywords “Brooklyn”, “bridge”, “tower” on the tags, returns few instances of the Brooklyn bridge tower and the rest of the results consist of different high buildings and towers and photos from the Brooklyn neighborhood. Such results can be useful for building a dataset for different concepts or classes (*e.g.*, bridge, tower, New York, Brooklyn neighborhood), but less useful in our object instance retrieval framework. For this reasons, we have retained the text-based search performed on the image titles.

Figure 4.11 illustrates the results of a query performed on Flickr for retrieving the Eiffel tower. Notice that different instances of the object of interest are retrieved. Such results contribute to building a rich model of the Eiffel Tower visual query.

Concerning the number of images $N_{download}$ to consider and download from the result list, we experiment with values between 25 and 100. Such aspects are discussed in details in the Experiments and Results section (Chapter 6).

In the same time, a certain number of outliers are also retrieved due to annotations error. Such outliers are filtered out in the following step.

4.6.2 Construction of the query graph

We first detect the local features and extract their descriptors by employing the Hessian-affine covariant region detector [Perdoch 2009] and the RootSIFT descriptor [Arandjelovic 2012]. All images from this set are matched one by one among themselves using the RootSIFT descriptor and Lowe’s ratio test [Lowe 2004] for selecting the reliable matches. The matched points are checked for geometric consistency as described in Section 4.4.4. We consider that two images contain the same object if they have at least N_{min} geometrically verified interest points successfully matched.

The role of this exhaustive matching procedure of all retrieved images is twofold. First, it makes it possible to reject the false positive images that have been retrieved (Figure 4.11). Usually such false positives are quite different from the rest of the true positive matches and they will be cleared out when matched with the rest of true positives. Second, as we can observe in Figure 4.11, the web search has retrieved different instances of the object of interest which are less likely to be matched using interest point matching. In this case, the role of the one to one matching is to identify groups of similar instances of the same object (*e.g.*, Eiffel Tower seen from distance, Eiffel Tower photographed from one of it pillars.) in order to construct multiple query examples and descriptors.

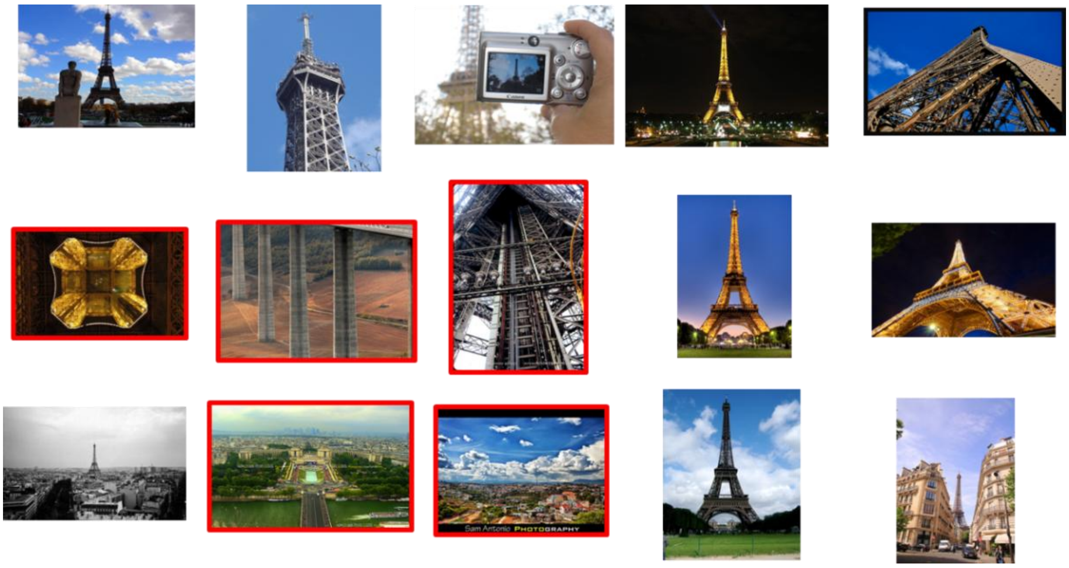


Figure 4.11. Flickr search results for "Eiffel tower". Notice that different instances of the Eiffel tower are retrieved along with a number of false positives highlighted with a red rectangle.

In order to identify the different instances of the query object, we employ the matching results from the previous step and construct a query graph similarly with the image graph introduced in [Philbin 2011]. The nodes of this graph are images and the edges connect images which have at least N_{min} geometrically verified matches. Note that unlike [Philbin 2011], which constructed a large scale image graph over an entire dataset, our query graph is built only over the candidate query images. Thus, such a graph is more lightweight and can be constructed and exploited efficiently.

An example of such a graph is illustrated in Figure 4.12. The graph is computed from the first 50 images returned by Flickr's search engine. Note how the false positives from Figure 4.11 have been discarded, as such images have been identified as isolated nodes, with no edge to any other images in the data set. In addition, the number of images to be considered has been significantly reduced (half of the initial number of retrieved images).

In Figure 4.12 we can notice that images containing similar instances of the object of interest are strongly inter-connected. In addition, the clusters of inter-connected regions can be easily identified as connected components in the query graph.

In the case of the query graph in Figure 4.12 we can extract 6 connected components, each consisting of images with similar instances of the Eiffel tower (*e.g.*, tower viewed from distance, pillar view, night view,…). The less representative instances are either completely rejected in the matching sequence or compose small connected components with poor interconnectivity.

In the following stage, the goal is to identify the most representative instances of a given object in an unsupervised manner.

4.6.3 Identification of representative images from each connected component

While the number of images to consider for building a query visual descriptor has been reduced in the matching stage, the number of images is still high. In order to further reduce this number, we select only the most representative images for an object for each connected component determined.

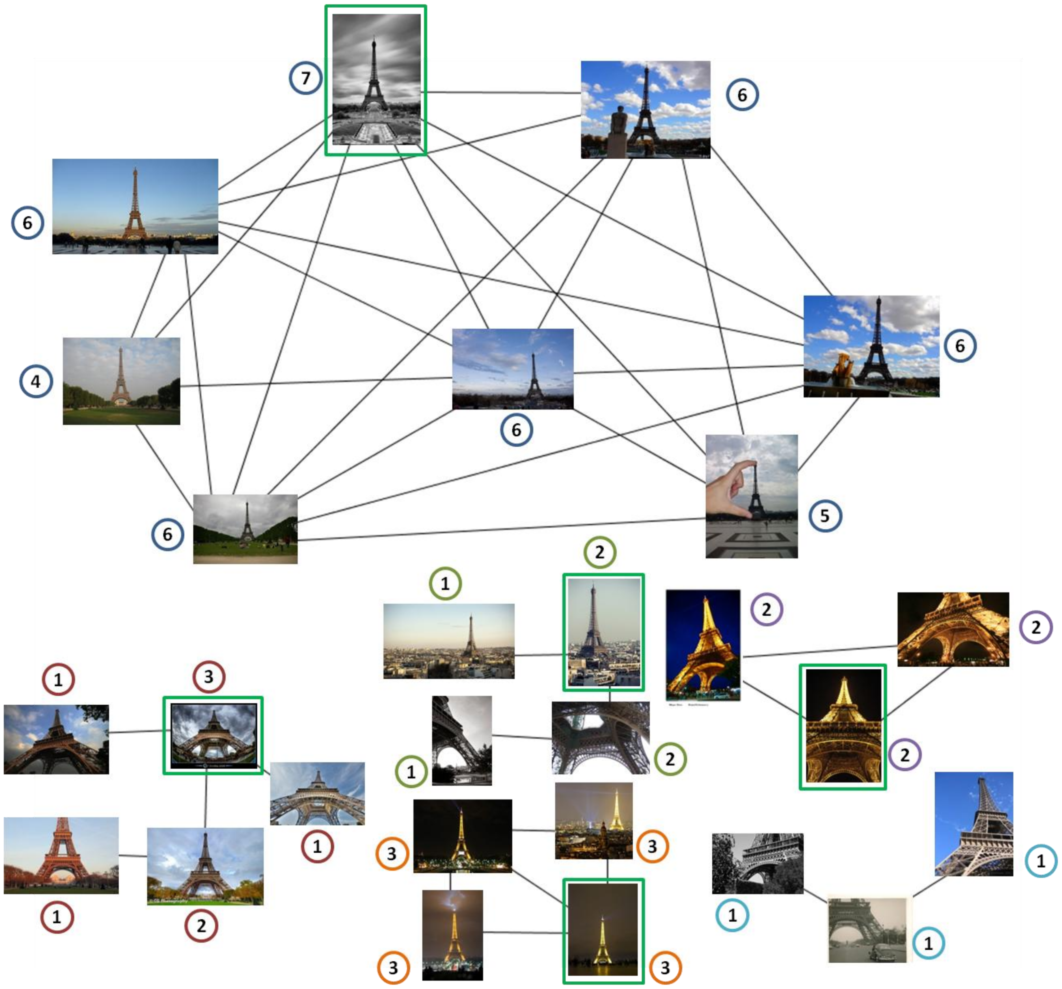


Figure 4.12. Query graph obtained for "Eiffel tower". Each node has marked its degree in a circle. The colors of the circle indicate the connected component which the current node is a part of. The iconic images of from each component are highlighted with a green bounding box.

The images from each connected component are ranked according to the number of matched images (*i.e.*, edges in the graph). This measure corresponds to the degree (or valence) of each node within the graph. For images having the same degree, the ranking order is made according to the total number of geometrically verified matches. This is the case of the purple, green and orange components in Figure 4.12, where several images have the same degree and the iconic image is selected by comparing the number of spatially verified matches cumulated over the all the matchings of the images. Thus, the retained representative image is the one yielding the highest number of verified matches.

In Figure 4.12 the iconic images are highlighted with a green bounding box. We can notice that the iconic images contain the most common views for the given object of interest. In addition, in order to avoid less representative images, we constrain them to have at least two verified matched images. The iconic images obtained for our example are illustrated Figure 4.13.

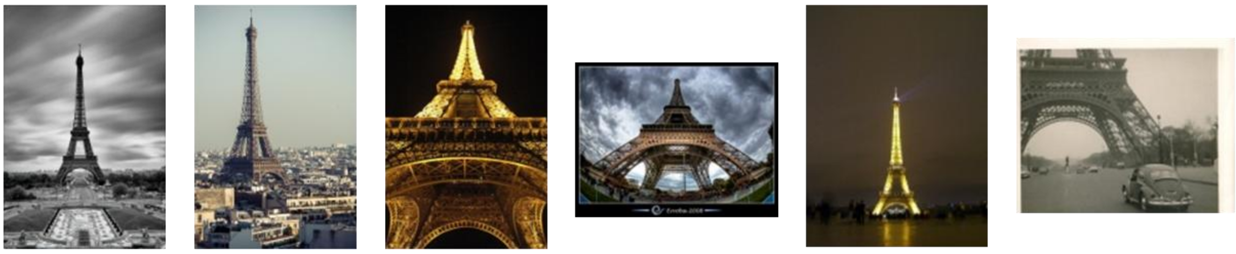


Figure 4.13. Iconic images for the "Eiffel tower".

Once the iconic images identified, we proceed with the computation of their visual descriptors.

4.6.4 Visual descriptor construction from representative images

At this stage, the problem is similar with the query expansion framework defined [Chum 2007]. We dispose of a collection of geometrically verified and representative images that can be used for generating a new expanded query. In [Chum 2007], a BoW vector is computed for each representative image individually. The obtained vectors are then averaged to obtain a single BoW representation for the whole set of queries.

In our case, we propose instead to exploit the information from the images that have been matched with the representative images obtained.

The proposed approach is inspired by the covalent bonding process [Campbell 2006] from chemistry. A chemical bond is an attraction between atoms that allows the formation of molecular structures that contain multiple atoms. The bond is caused by the electrostatic force of attraction between opposite charges, either between electrons and nuclei or as the results of a dipole attraction. A covalent bond is a strong chemical bond that involves the sharing of pairs of electrons between atoms. For many molecules, the sharing of electrons allows each atom to acquire a stable electronic configuration. The covalence measure takes important values in the case of atoms with similar electro-negativities. Atoms share uncoupled electrons to create new pairs with the uncoupled electrons of other atoms in order to reach a stable molecular configuration consisting of multiple atoms (Figure 4.14). An atom can simultaneously share different electrons with multiple atoms [Campbell 2006].

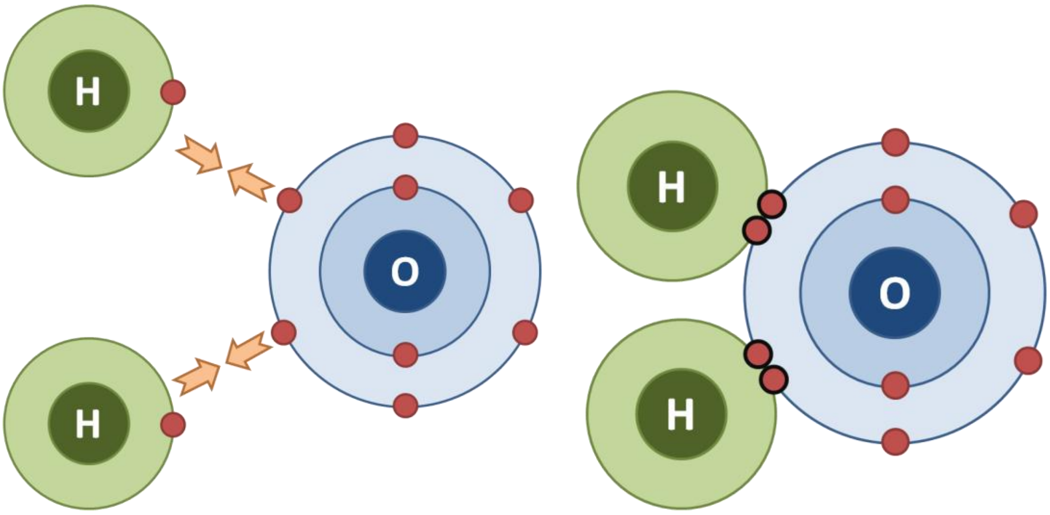


Figure 4.14. Two Hidrogen atoms and an Oxygen atom (left) share electrons in a covalent bond in order to reach a stable H_20 molecule (right).

In our case, in analogy with this bonding process, atoms represent images and electrons are local features from each image. Two images have a covalent bonding if they have at least N_{min} geometrically verified matches. A representative image represents a stable molecule, with most of its electrons shared with other similar atoms. The molecule with the highest number of bonded atoms and shared electrons is the most representative one for a given element (*i.e.*, object). We can now define each molecule through its electrons and its shared electrons. A representative image can be thus described by its own features and by the features that have been matched with other similar images. The matched features complement the existing features and are used to enrich the current image. Practically, for each matched feature, when computing the BoW vectors, we assign a weight proportional to the number of verified matches. For example, for a feature that has been matched three times, its non-normalized *tf* weight is updated from 1 to 4. We will refer to this new descriptor as *centered representative query descriptor*. Alternatively, the representative image descriptor can be computed by considering the descriptors from the images matched with the representative image. These descriptors are collected in a pool of descriptors and quantized in a single BoW vector. We refer to this representation as the *distributed representative query descriptor*.

In Figure 4.15, we illustrate the feature matching between a representative image and its similar images from the same connected component of the query graph. Figure 4.16.c illustrates the weighted centered representative image. The thickness of the elliptical shapes is proportional with the number of verified matches of the respective region. Complementary, the distributed representative image descriptor is composed from the point descriptors from all the images matched with the representative image. This weighting mechanism enriches the query descriptors and emphasizes the most representative features of the current object, increasing the chances of retrieving it accurately.

The final step of our technique concerns the aggregation of the various query descriptors associated with the set of representative images.

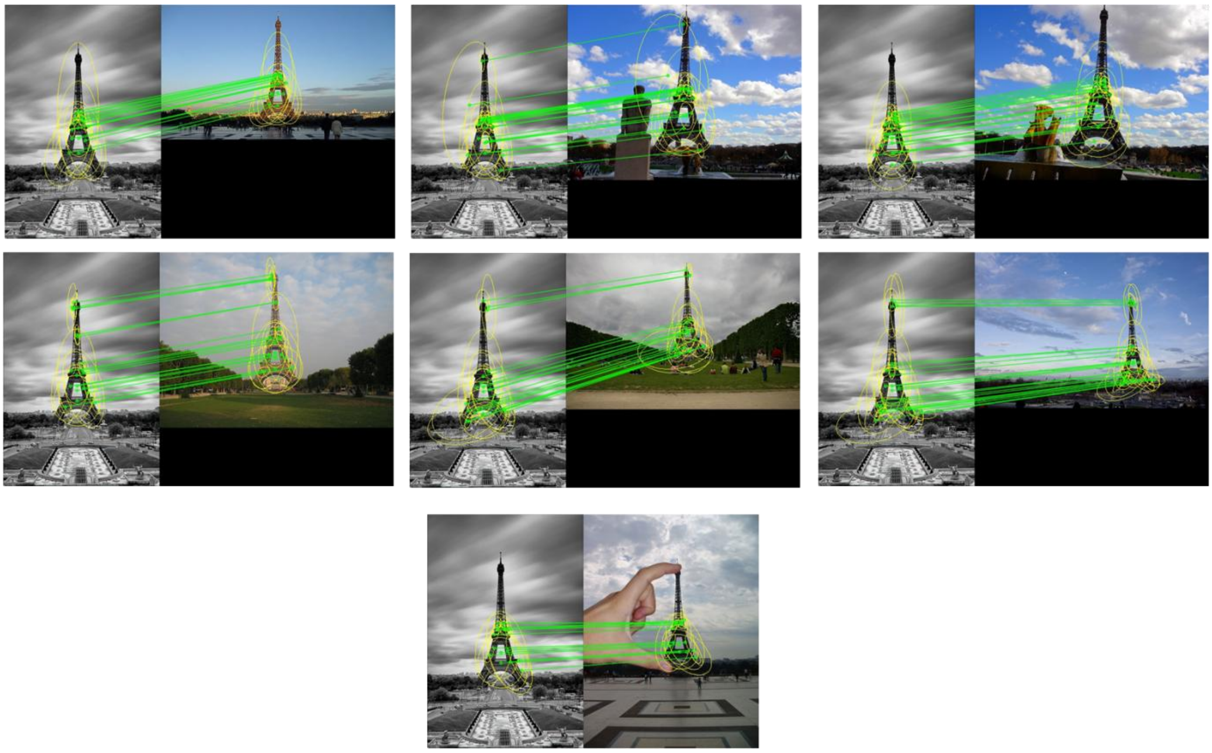
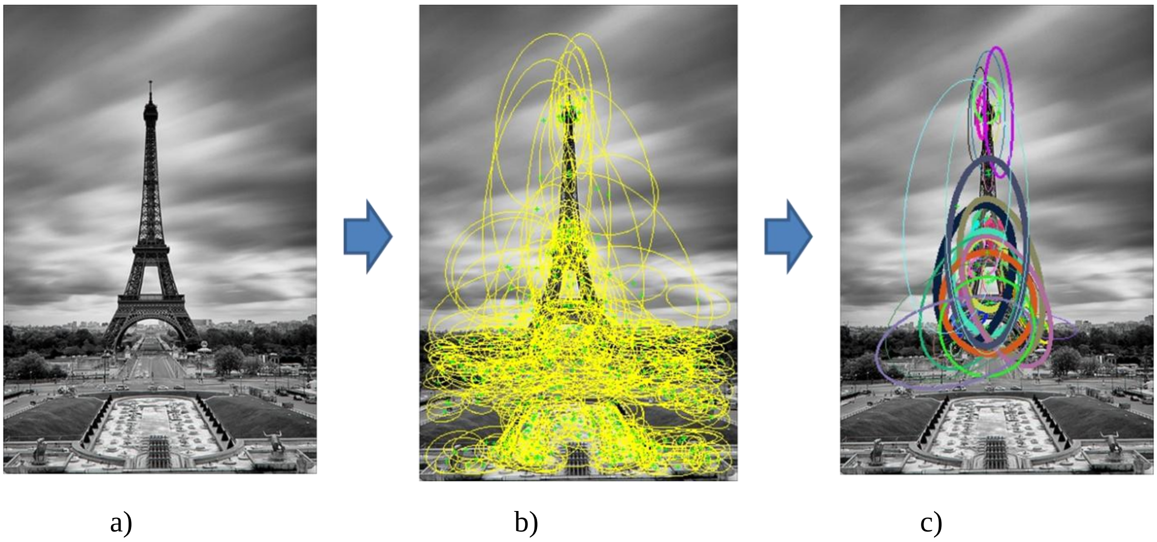


Figure 4.15. Representative image and its geometrically verified matches.



a) b) c)

Figure 4.16. Enriched representative image with geometrically verified regions weighted proportionally with the number of matches from Figure 4.15. a) Original image, b) Detected Hessian-affine regions, c) Verified regions and their weights (the ellipse thickness is proportional with the weighting).

4.6.5 Aggregation of the query descriptors and retrieval process

In order to formulate queries, the set of enriched representative images can be considered either as individual queries, or be composed into a single query.

The advantage of composing a single query descriptor is related to the computational cost of the retrieval process, which is considerably reduced, notably in cases where the number of representative images is relatively important (higher than 10). On the downside, in this case, the queries are individual images which are globally less similar with each other than a set of video frames from a given video shot, where the final BoW vectors is very similar with the individual BoW vectors of the frames. When composing a single query from multiple images, relevant information that is found in only one of the images will be weighted lower in the final BoW vector. This is the case of patches that are visible only for certain poses or view points of the object of interest. On the other hand, the grouping process assigns higher weights to features that appear in multiple images and which are thus more representative for the query object. For these reasons, we decided to build a single query descriptor out of all enriched representative images using the procedure proposed in Section 4.5 for video shot BoW descriptors. Features from all iconic images are projected in the same pool of features and a BoW vector is computed by quantizing each feature as if they belong to the same image.

Alternatively, when the number of iconic images is more reduced (*i.e.*, less than 10) individual BoW vectors can be computed for each representative image and used for querying the video dataset. In this case, we obtain a number of query results equal to the number of representative images. Such results need then to be aggregated into a single ranked list.

In order to achieve this purpose, we employ the best score selection ranking for all runs as proposed in [Bursuc 2011]. For each video, the best score from all runs is selected and all videos are then re-ranked according to their best score. This principle is illustrated in Figure 4.17. Here, we consider three representative images for querying (column 1) and their corresponding lists of results (columns 2-5) ranked in decreasing order of similarity to the query. For each video frame, we check its scores in each of the three runs and consider the optimal one (*i.e.*, the one yielding the highest similarity score). In Figure 4.17, for the video highlighted with the blue bounding box, the best score has been obtained in the first run, marked with a green bounding box. The aggregated list of results is obtained by re-ranking the videos in decreasing order of similarity according to their aggregated similarity scores.

The experimental evaluation of the proposed query expansion technique is presented and discussed in details in Chapter 6, for different sizes of the representative image dataset and for various query descriptors.

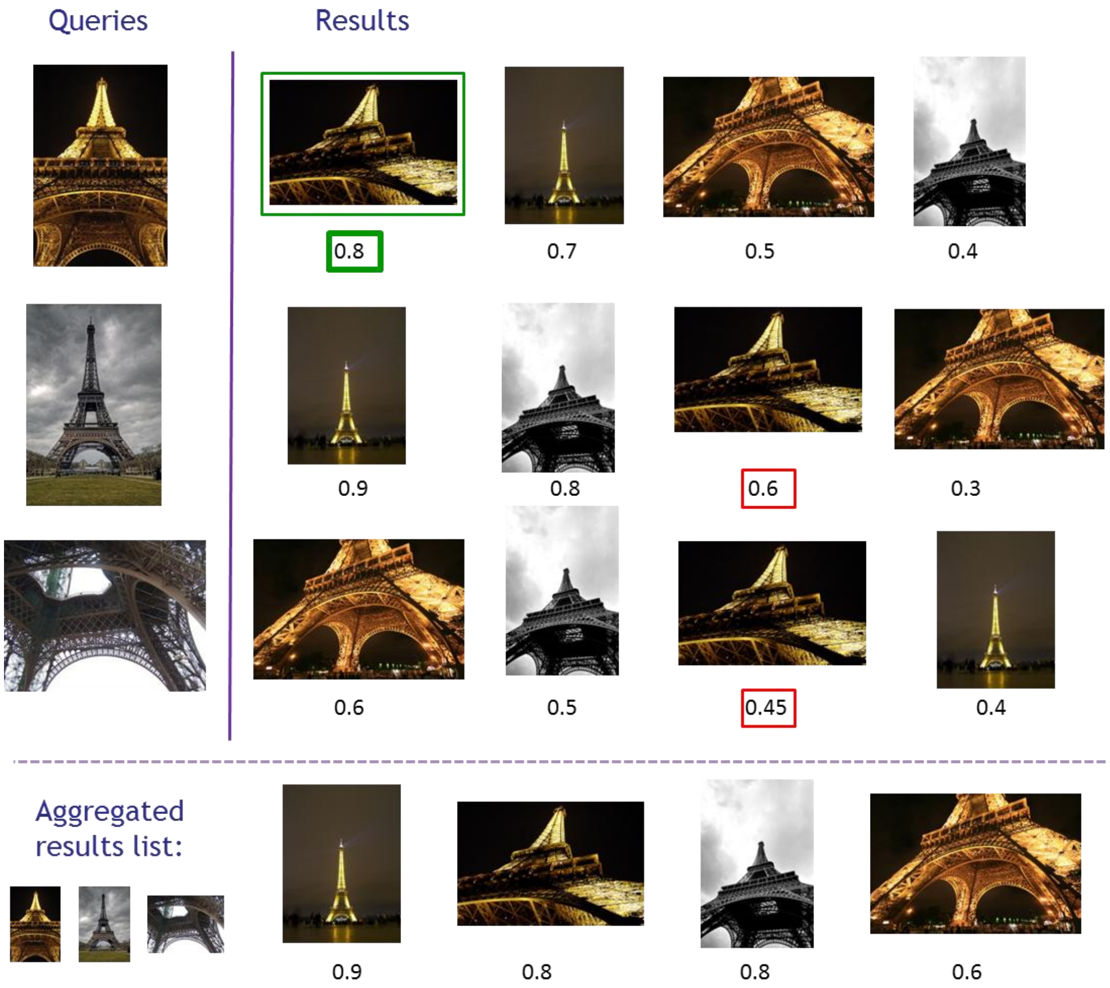


Figure 4.17. Best score selection ranking. Three queries are issued on the same dataset(column 1) and the results are returned in decreasing order of similarity (columns 2-5). The results are aggregated by choosing the best score from all runs.

4.7 Conclusion

In this chapter, we have tackled the issue of Bag-of-Words (BoW), interest points-based representations for object detection and retrieval in video sequences in large video repositories.

First, we have presented an in-depth analysis of the rich literature dedicated to this issue, including both interest point detection methods, associated visual representations, clustering procedures and matching strategies/similarity measures. The most representative methods are here detailed with principle, advantages and limitations from the perspective of large scale object retrieval. For each block of the BoW framework we specify a set of recommendations for its implementation as various choices can be made of each such block.

The analysis of the state of the art led us to retain a Hessian-affine interest point detector. Concerning the visual descriptors, we have adopted the recently proposed RootSIFT representation, which offer the most promising performances in terms of object retrieval rates. For the clustering stage, we have considered the Approximated k-means method (AKM), a scalable version of the k-means algorithm approach, which offers a good trade-off between accuracy of representation and computational complexity.

We have then proposed a meaningful and effective video representation for reducing the size of the descriptor set by computing BoW vectors on the shot level. Finally, we proposed a query definition and expansion technique that makes it possible to retrieve in a multi-modal manner objects of interest in videos, starting from a textual query performed upon existing images repositories.

5 Hybrid object representation

Abstract. *In this chapter, we introduce a novel hybrid representation technique which integrates the advantages of segmented region-based representations and interest-point-based approaches. The proposed technique makes it possible to extend the adjacency information naturally included in the region-based representations to the interest points. In this case, the segmented regions are used as spatial support for identifying adjacent interest points which are then used to construct an interest point-based image graph. The matching between the query model and a candidate image is then formulated as a graph matching problem, which is solved with the help of a greedy spectral matching technique. Such an approach makes it possible to integrate not only the unary (individual) similarity between nodes in the graph, but also the pairwise similarity associated with pairs of matched interest points. Finally, we propose a merged similarity measure, integrating both interest point and color similarities and exploit it for object-based retrieval purposes.*

Keywords: *graph matching, object representation, interest point adjacency, pairwise constraints.*

Résumé: *Dans ce chapitre, nous présentons une nouvelle technique de représentation hybride d'objets qui comprend également les avantages des représentations basée sur régions segmentées et des approches utilisant des points d'intérêt. La technique proposée permet d'étendre l'information d'adjacence inclus naturellement dans les représentations basées sur régions vers les points d'intérêt. Dans ce cas, les régions segmentées sont utilisés comme support spatial pour identifier les points d'intérêt adjacents qui sont ensuite utilisés pour construire un graph de points pour représenter une image. La correspondance entre le modèle de requête et une image candidate est alors formulée comme un problème d'appariement de graphes, qui est résolu à l'aide d'une technique glouton d'appariement spectral. Une telle approche permet d'intégrer non seulement la similitude entre nœud unaires (individuels) dans le graphe, mais aussi la similitude deux à deux associées à des point d'intérêt appariés. Enfin, nous proposons une mesure de similarité fusionnée, qui intègre à la fois la similitude des points d'intérêts et la similitude de couleurs et nous l'en exploitons pour la recherche des objets.*

Mots clés: *appariement de graphes, représentation d'objet, contiguïté des points d'intérêt, contraintes par paires.*

In the case of interest points, the BoW framework discards completely the spatial information related to the relative position of the considered points in the image. This information is used solely in the re-ranking stage with the help of a RANSAC estimation of image transformations. However, the specificity of the internal structure of the object is still neglected in the re-ranking stage.

Grouping interest points into meaningful and consistent entities is a challenging problem. Existing methods identify pairs of interest points in candidate images according to pairs of interest points in a query image. Empirical thresholds are here used for the distances between pairs of points [Leordeanu 2005] or for quantizing the deformations of the elliptical regions associated to such point pairs [Cho 2009, Cho 2012]. Such thresholds can be determined with the help of a learning stage where the analysis of the positive samples emphasizes specific distances and structures [Leordeanu 2005]. Another solution is to partition the image with a regular grid and to express the spatial proximity between the pairs of points by using the grids as units of measure [Duchenne 2011]. The spatial relationship between features can also be determined by employing a Delaunay triangulation for generating a graph of interest points connected by edges [Li 2010, Li 2011].

In our work, we propose a hybrid representation which jointly exploits the Hessian-affine interest point detector (*cf.* Section 4.1.2) and the region-based representation introduced in Chapter 3.

The Hessian-affine interest point detector makes it possible to take advantage of the elliptical regions associated to the detected interest points. Such ellipses define a support region for each interest point considered. This support region is superposed over the available region-based representation. Thus, for each interest point, we determine the set of image segments which are intersecting with its representative ellipse. This set is called *the set of support segments* associated to the considered interest point.

The set of support segments naturally gathers the connectivity information of the considered segments and makes it possible to transpose the adjacency relationships to the set of interest points. Thus, two interest points are declared to be neighbors if they share at least one segment in the corresponding sets of support segments. Some groups of neighbor interest points are illustrated in Figure 5.1.

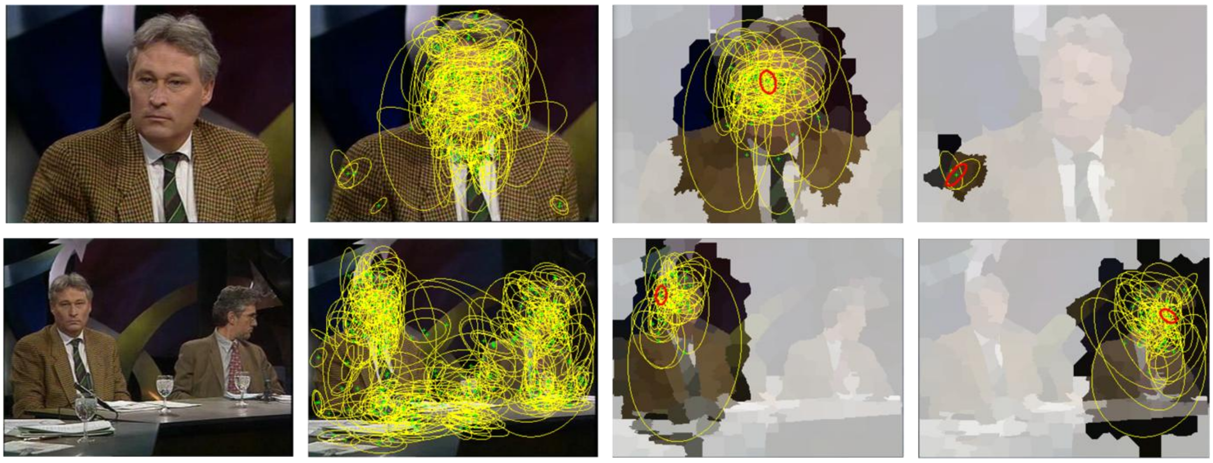
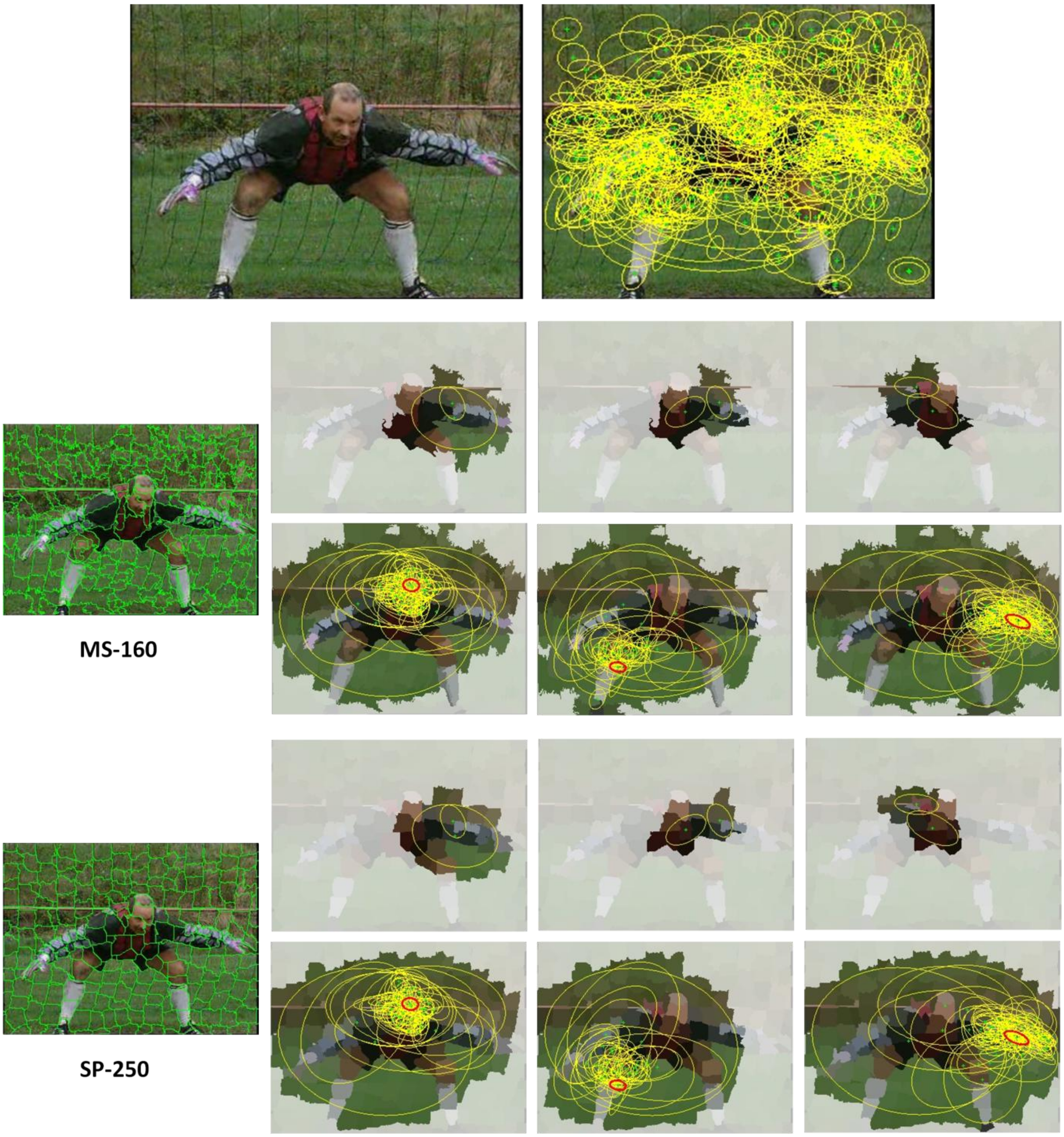


Figure 5.1. Groups of neighboring regions. The yellow ellipses represent the neighbor ellipses of the red ellipse. Segments for Superpixels segmentations with 250 segments per image are here considered.

The proposed principle is illustrated in Figure 5.2. Three different segmentation procedures have been here considered MeanShift (with 160 segments) and Superpixels (with 250 and 500 segments). Let us note that each interest point and its neighbors can compose meaningful clusters of points representing objects or parts of objects. In addition, whatever the segmentation method employed, the resulting clusters are highly similar.

Let us also note that the ellipses of some affine covariant regions detected at higher scales can cover large surfaces in the image. This can lead to a loss of distinctiveness at the level of the corresponding set of support segments, as such a set can cover the majority of segments, and can affect the proposed adjacency construction strategy.



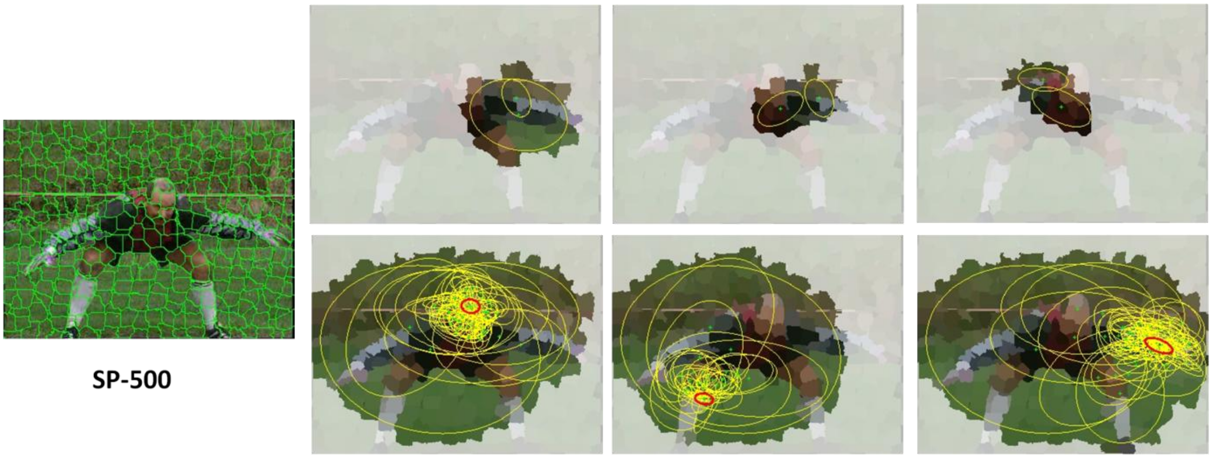


Figure 5.2. Interest points' adjacency with segmented regions spatial support. Results with 3 types of segmentations (MeanShift 160, Superpixels 250, Superpixels 500) are illustrated. In the first row of each group pairs of adjacent interest points are displayed along with their underlying segmented regions. In the second row of each group we illustrate all adjacent points (marked with yellow ellipses) of a single interest point (marked with red ellipse). Note how an interest point and its adjacents define a meaningful cluster in the frame.

In order to overcome this drawback, we integrate a scale test for the regions that have been found to be connected. Intuitively, the neighborhood of two affine covariant regions with significantly different scales should not be allowed as they do not provide relevant spatial information. We compute the surface areas of both elliptical regions and compute their ratio. If this ratio between the smaller and larger region is found to be below a threshold τ, the connection between the two regions is discarded. For parameter τ we recommend values ranging from 0.3 to 0.75. This thresholding mechanisms permits to connect solely regions with similar scales. Note that other measures (*e.g.* scale of detected affine covariant region) can be used for this purpose. The improvements brought by this scale thresholding are illustrated in Figure 5.3. Interest points with similar scale sizes are now grouped together and each point and its immediate neighbors define more meaningful entities (*e.g.*, head, leg, arm).

The connectivity information of the interest points can be employed to define an interest point adjacency graph for the whole image, as illustrated in Figure 5.4 and Figure 5.5. In Figure 5.4 we illustrated the influence of the threshold τ on the configuration of the adjacency graph built over a Superpixels 250 segmentation. For values higher than 0.5, the graph becomes more sparse leaving multiple interest points outside the graph structure. These are usually isolated and low scale points that cannot be considered adjacent to other regions because of the more restrictive scale test. Higher values of the threshold generate less complex graphs but provide a reduced number of pairs of points that can be exploited in a graph matching context. We are interested in identifying representative pairs of interest points and higher number of edges in the graph would facilitates this. In our experiment a threshold of $\tau = 0.5$ provided the best balance between accuracy and computational time.

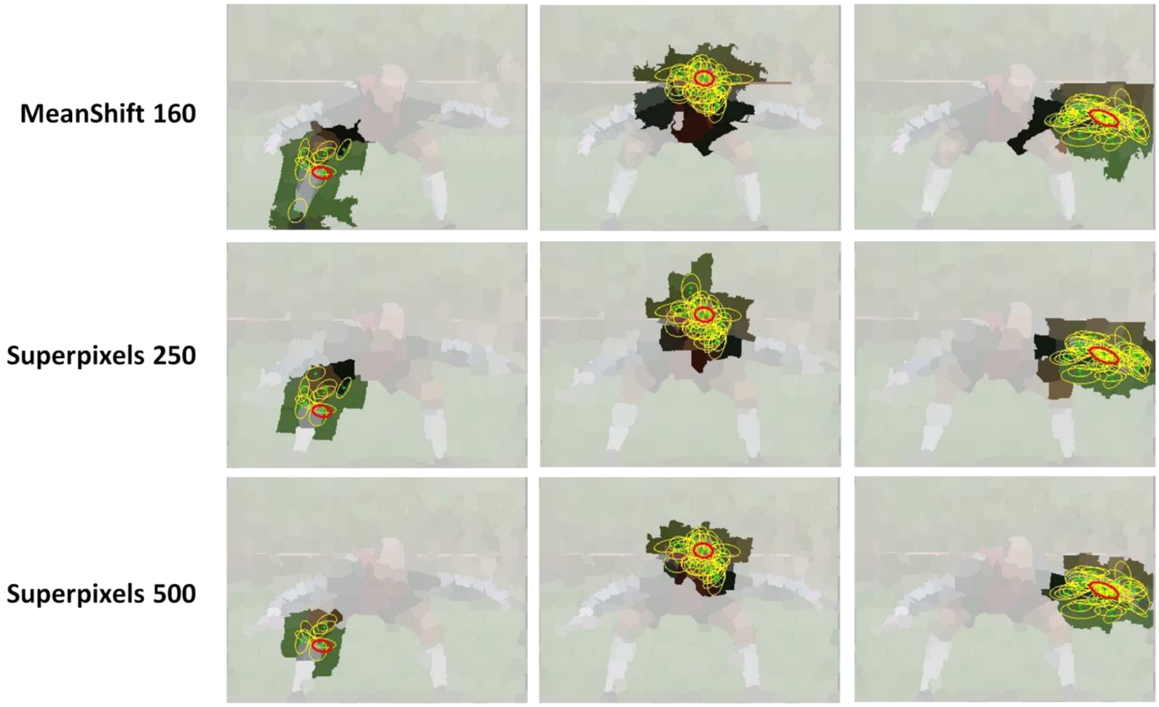
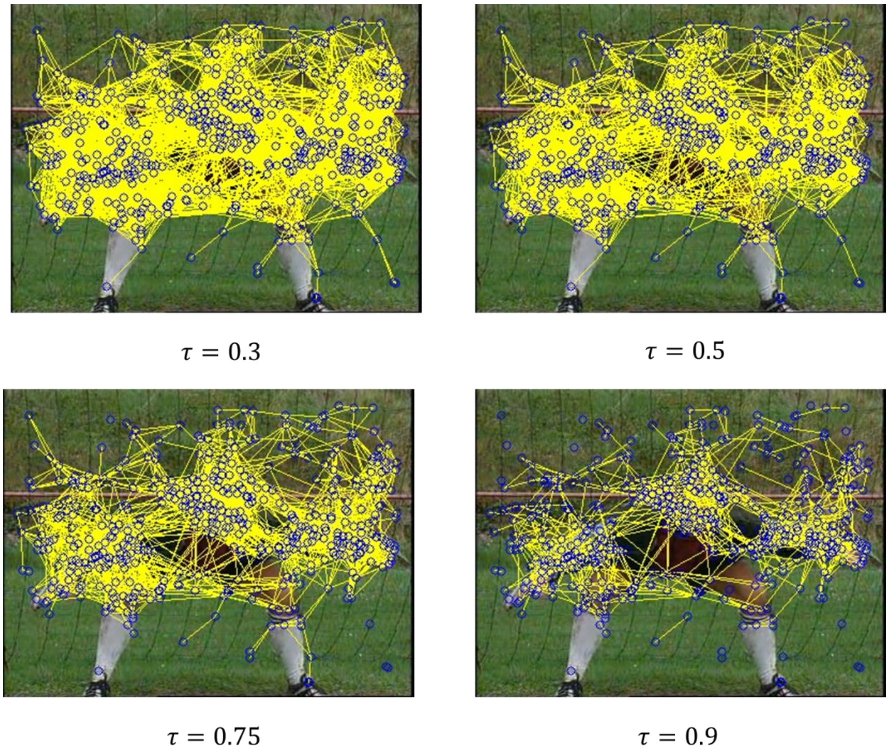


Figure 5.3. Interest points and their adjacent points after the scale thresholding with $\tau = 0.5$.


$\tau = 0.3$

$\tau = 0.5$

$\tau = 0.75$

$\tau = 0.9$

Figure 5.4. Interest point adjacency graph obtained for Superpixels 250 segmentations with different values of the threshold τ. For higher values of τ the graph becomes more sparse and some points. The position of the interest points is indicated with yellow circles and two adjacent points are connected with a blue edge. As the number of regions per frame decreases, the graph becomes more sparse.

In Figure 5.5 we illustrate the graphs constructed from different segmentations with $\tau = 0.5$. We can notice that the point adjacency graphs are quite dense and their density is inversely proportional with the number of regions per frame, with the MeanShift segmentations generating the highest density graphs. In cases of poorly textured images with few interest points and large uniform regions, the MeanShift graphs are more advantageous. Here, large segments allow the distant and sparse interest points to be considered adjacent. In addition, a meaningful graph can be constructed from the image.

Once the interest point adjacency graph constructed, the considered image is represented as a two layer graph with the point adjacency graph on the first level and segment adjacency graph on the second level. The connection between the two levels is achieved through the nodes of the interest point graph, which represent the parents of the regions covered by its elliptical representation.

Once the graph-based representations constructed for both query model and candidate image, the object retrieval/detection is formulated as a sub-graph matching problem and solved with the help of a spectral graph matching technique, as described in the following section.

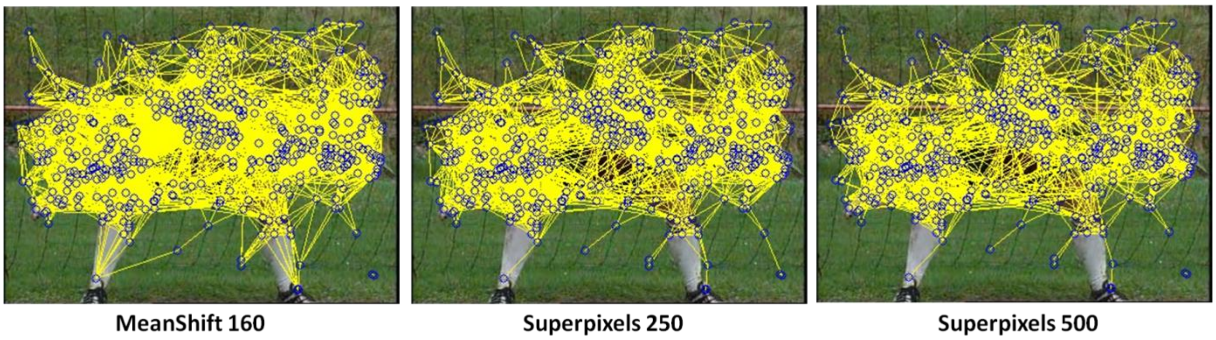


Figure 5.5. Interest point adjacency graph obtained for 3 segmentations with $\tau = 0.5$.

5.1 Spectral graph matching optimization

Graph matching is a powerful and robust technique widely used in computer vision, pattern recognition and machine learning. The objective of graph matching techniques is to determine a mapping between the nodes of the two graphs to be compared which preserves the relationships of the nodes as much as possible. In object recognition, a model and a test image are represented as graphs using salient visual features. The graph matching technique determines the object by determining the optimal fit sub-graphs, which are minimizing the distances between associated visual features.

The main limitation of graph matching methods is related to the computational complexity, since determining an exact solution is proved to be a NP-hard problem [Karp 1972, Garey 1979]. In order to overcome such a strong limitation, recent researches have proposed various approximations. Yet, the computational costs in terms of time and memory are still limiting the sizes of the graphs to be matched. For this reason, most of the graph matching methods deal with problems where a sparse set of discriminative features can be selected or pre-defined.

Berg *et al.* [Berg 2005] modeled the problem as a quadratic integer programming model, where affine terms and quadratic terms of the objective function represent node-to-node and edge-to-edge similarities between the two graphs, respectively. The model was relaxed into a continuous domain, solved, and mapped back onto the original solution space. Leordeanu and Hebert [Leordeanu 2005] proposed a spectral method working on a matrix where the diagonal elements represent one-to-one assignment costs, and other elements represent pairwise agreements between potential correspondences. The correspondences are then obtained by mapping the principal eigenvector of the matrix to the discrete solution space using a greedy algorithm. In order to automatically set the weights of the terms belonging to the affinity matrix, several learning methods that optimize the weights in both supervised [Caetano 2009] and unsupervised [Leordeanu 2009] manners have been proposed.

Cour *et al.* [Cour 2006] propose a spectral relaxation method for the graph matching problem that incorporates one-to-one or one-to-many mapping constraints, and present a normalization procedure for existing graph matching scoring functions that can dramatically improve the matching accuracy.

Cho *et. al.* [Cho 2009] formulate the graph matching problem as an unsupervised multi-class clustering problem on a set of candidate feature correspondences. Starting from a set of relaxed matches between the graph nodes, a hierarchical agglomerative clustering is employed for determining multiple object-level clusters of correspondences in an unsupervised manner. Object-based image matching is then performed by taking into consideration both how well the descriptors of the features match and how well their pairwise geometric constraints are satisfied within each cluster.

Finally, let us mention the approach introduced in [Duchenne 2011], where authors introduce a graph matching algorithm using a grid-based representation of the images, which is significantly robust to object deformations.

In our work, we have retained the spectral graph matching technique proposed by Leordeanu and Hebert [Leordeanu 2005]. This method can identify consistent correspondences between two sets of features by taking into consideration both how well the descriptors of these features match and how well their pairwise constraints are fulfilled. Our objective is to identify pairs of correspondences that can provide additional and useful information about the internal structure of an object.

Let us first recall the basic principle of the spectral graph matching method.

5.1.1 Spectral Graph Matching

In a general form, the graph matching problem can be formulated as follows. . Let us consider two attributed graphs to be compared, $\mathcal{G}^Q = \langle \mathcal{V}^Q, \mathcal{E}^Q, \mathcal{A}^Q \rangle$ describing the query model and $\mathcal{G}^P = \langle \mathcal{V}^P, \mathcal{E}^P, \mathcal{A}^P \rangle$ describing the candidate image. The number of nodes in graph $\mathcal{G}^P$ (resp. $\mathcal{G}^Q$) is denoted by N_P (resp. N_Q). The set of attributes $\mathcal{A}^P$ includes a collection of feature vectors A_i^P associated with each node $i \in \mathcal{V}^P$ as well as a set of features A_{ij}^P associated with each edge $(i, j) \in \mathcal{E}^P$. The A_i^P attributes describe the appearance of node i and the attributes A_{ij}^P describe a pairwise affinity between the nodes i and j. The attributes $\mathcal{A}^Q$ of the graph $\mathcal{G}^Q$ are defined in a similar manner. For each pair of nodes $i \in \mathcal{V}^P$ and $i' \in \mathcal{V}^Q$ a score function $M_{ii';ii'} = f\left(A_i^P, A_{i'}^Q\right)$ is defined, which measures the agreement or similarity between the attributes A_i^P and $A_{i'}^Q$ describing the nodes i and i'. Similarly, for each pair of edges $(i, j) \in \mathcal{E}^P$and $(i', j') \in \mathcal{E}^Q$ a pairwise affinity function $M_{ii';jj'} = f\left(A_i^P, A_{i'}^Q, A_{ij}^P, A_{i'j'}^Q\right)$ is defined, which measures the relationship/agreement between the unary attributes described by A_i^P and $A_{i'}^Q$, and the pairwise agreements defined by A_{ij}^P and $A_{i'j'}^Q$.

The set of affinity functions can be regrouped in a matrix M of size $N = N_P \times N_Q$, where the $M_{ii';ii'}$ values are stored on the diagonal of M. The graph matching problem can be formulated as an integer quadratic program which consists of determining the indicator vector X^* that respects a set of specific mapping constraints (*e.g.*, one- to-one, many-to-one) while maximizing the following quadratic score function:

$$X^* = \text{argmax}_n(X^T M X). \qquad \textbf{(5-1)}$$

Here, $X \in \{0,1\}^N$, is an *indicator vector* such that $x_{ii'} = 1$ if node i is matched to the node i' from the other graph and zero otherwise. Typically, one-to-one constraints are imposed on X such that one feature from an image can be matched to at most one feature from the other image.

Let also observe that the matrix M can be interpreted as a weighted adjacency matrix of a super-graph structure, which includes as nodes the union of $\mathcal{V}^P$ and $\mathcal{V}^Q$. Its spectral properties can be used to avoid the combinatorial explosion of the correspondence problem when matching two sets of features.

The requirements for the formulation of the M matrix are that the unary $M_{ii';ii'}$ and pairwise $M_{ii';jj'}$ scores to be non-negative and to increase with the quality of the match. When the two pairs of features put in correspondence by two potential assignments agree in terms of their pairwise geometry, there will be an agreement link (positive edge) formed between the two assignments. Otherwise, there will be no link between the two assignments (edge of zero weight). These scores can capture any type of deformations/changes/errors at the levels of both appearance as well as geometric relationships.

Let us also observe that the match matrix M has a very specific structure: a strong block with large values formed by the correct assignments, and mostly zero values everywhere else. Such block structures can be effectively tackled with the help of a spectral analysis, carried out in terms of eigenvectors and associated eigenvalues. Thus, if we drop the binary condition on vector X and search for a solution in the space of unitary real vectors, the solution to Equation (5-1) is straightforward: the vector X^* can be simply determined as the unitary eigenvector of highest eigenvalue (also called *principal eigenvector*). A binarization procedure can then be applied in order to determine the principal eigenvector and to provide an approximate solution to the matching problem.

This principle, exploited by Leordeanu and Hebert in [Leordeanu 2005], is described in the following section.

5.1.2 Greedy spectral optimization

Leordeanu and Hebert [Leordeanu 2005] propose a greedy algorithm for determining the solution to the correspondence problem. The eigenvector value corresponding to a particular correspondence (i, i') is interpreted as a confidence measure associated to the hypothesis that (i, i') represents a correct assignment. The algorithm starts with the extraction of the principal eigenvector of M and then accepts as correct the correspondence (i, i') for which the eigenvector value $X^*_{ii'}$ is maximal. The match (i, i') is actually the correspondence with the highest confidence of being correct. Next, all the other correspondences that are in conflict with (i, i') with respect to the one-to-one mapping constraint are rejected. The next correct correspondence accepted is the one with the second highest chance of being correct that has not yet been rejected and that is not in conflict with the one already accepted. New highest confidence correspondences that are not in conflict with the previous accepted ones are accepted iteratively until all correspondences are either accepted or rejected. Finally, the set of candidate correspondences is split into two sets: the set of correct assignments $C^*$and the set of rejected assignments R. The two sets have the property that every correspondence from R will be in conflict with some assignments from C^* of higher confidence. Therefore, no element from R can be included in C^* without having to remove from C^* an element of higher confidence.

Let us note that the proposed greedy spectral optimization procedure has been proved to be several orders of magnitude faster than linear programming approximation to the quadratic problem.

The following section describes how the adopted spectral graph matching approach is exploited of object retrieval purposes, by considering the proposed hybrid representation.

5.2 Object Retrieval with Spectral Matching

The above-presented Spectral Matching technique can integrate spatial information of region configurations through its pairwise affinity terms. Such information could be of great help in validating pertinent object candidates and rejecting outlier regions. Let us note that the hybrid graph improves the original Spectral matching technique with the meaningful adjacency of the interest points based on their common region spatial support and on the scale similarity. Most of the graph matching approaches do not consider such aspects when identifying pairs of correspondences.

Yet, the construction of the affinity matrix M and the computation of its eigenvectors can easily become a computational burden when the number of interest points increases (*e.g.*, more than a few hundred). In such cases, a pre-processing stage becomes necessary, in order to filter out points that highly improbable candidates for matching.

To obtain initial candidate feature matches we use the Hessian-Affine feature detector and the RootSIFT descriptor. Using the Euclidean distance for matching the RootSIFT descriptor (*cf.* Section 4.2.4), we identify all the possible candidate matches yielding a score under a relaxed threshold $\delta = 0.5 - 0.55$. Multiple correspondences for each interest point are here allowed. In addition, we restrict the maximum number of interest points put into correspondence to 1000. The pairwise terms are then determined using the adjacency relation construction presented in Section 5.1.1.

Concerning the similarity of the pairwise correspondences, we measure the agreement of the two pairs of matches by jointly analyzing their visual, geometric and spatial similarities. More precisely, the pairwise term for a pair of correspondences $(i,j) \in \mathcal{E}^P$ and $(i',j') \in \mathcal{E}^Q$, where feature i matches feature i' and feature j matches feature j', is defined as follows:

$$sim_{ii';jj'} = \alpha\left(1 - d_{geom}(ii',jj')\right) + \beta(1 - d_{spatial}(ii',jj')) + \gamma(1 - \min\left(d_{vis}(i,i'), d_{vis}(j,j')\right).. \tag{5-2}$$

where $d_{geom}(ii',jj')$ is the geometric dissimilarity between the two feature correspondences as introduced in [Cho 2009], $d_{spatial}(ii',jj')$ describes the Euclidean distances between the center of the features (*i.e.*, interest points) and $d_{vis}(i,i')$ and $d_{vis}(j,j')$ refer to the distances between the corresponding RootSIFT descriptors of features i and i' and, respectively of features j and j'.

For the geometrical distance, we use the elliptical representations of the interest points described by $(x_i, x_{i'}, H_{ii'})$, where x_i denotes the center of i, $x_{i'}$ denotes the center of i' and $H_{ii'}$ is the homography from i to i'[Mikolajczyk 2004]. Similarly, features j and j' can be described by $\left(x_j, x_{j'}, H_{jj'}\right)$. The geometrical distance is then defined as:

$$d_{geom}(ii',jj') = \frac{1}{4}\left(d_{geom}(ii'|jj') + d_{geom}(jj'|ii')\right) \; , . \tag{5-3}$$

where

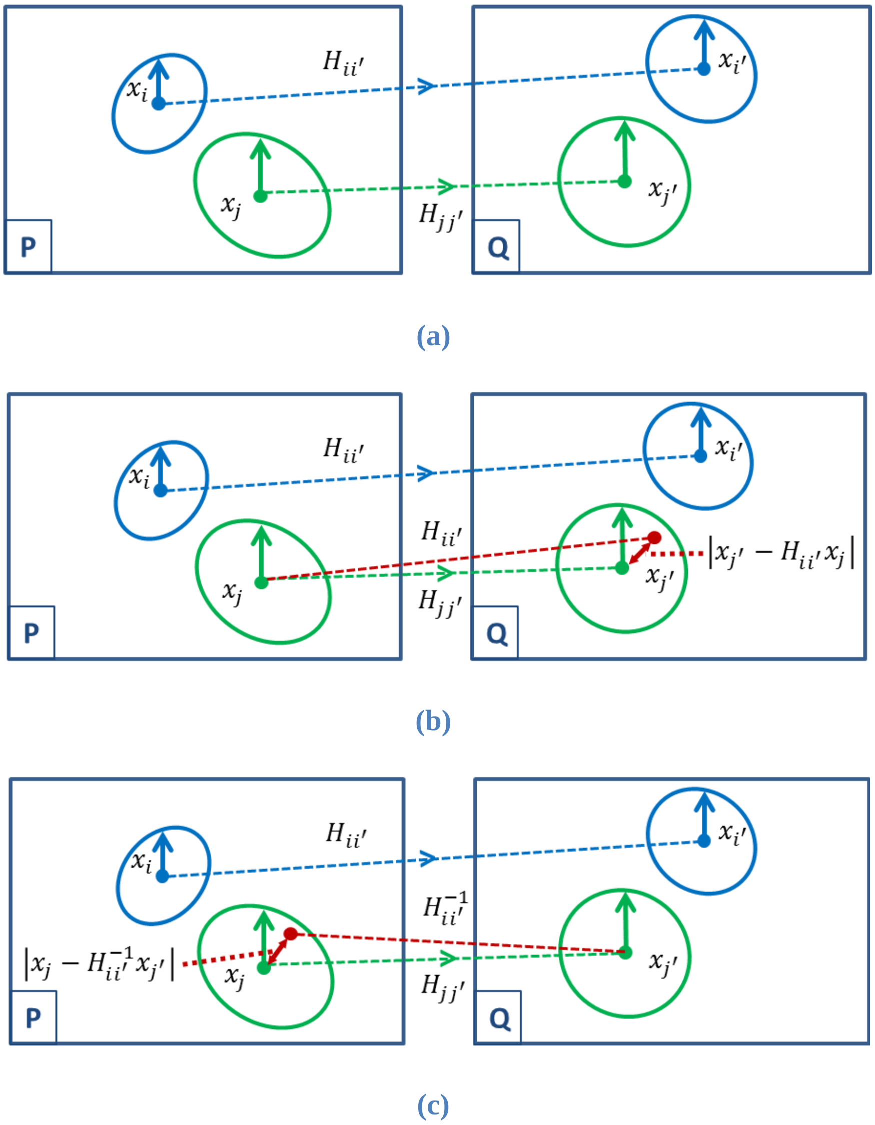
$$d_{geom}(ii'|jj') = \left\|x_{i'} - H_{jj'}x_i\right\|_{L_2} + \left\|x_i - H_{jj'}^{-1}x_{i'}\right\|_{L_2} \; . \tag{5-4}$$

and

$$d_{geom}(jj'|ii') = \left\|x_{j'} - H_{ii'}x_j\right\|_{L_2} + \left\|x_j - H_{ii'}^{-1}x_{j'}\right\|_{L_2}. \; . \tag{5-5}$$

with $\|\cdot\|_{L_2}$ denoting the Euclidean norm.

The geometric distance considered corresponds to a projection error, when matching two interest points and it should be small if the transformations $H_{ii'}$ and $H_{jj'}$ are similar to each other. When computing the homographies between the elliptical shapes, we introduce an “upness” constraint for the orientation of the ellipses (i.e., we assume that both frames have the same orientation). Philbin *et al.* [Philbin 2007] illustrated that this is often a good assumption for photos on the web as they are rarely uploaded in the wrong orientation. We extend this assumption to video content since wrong oriented video clips are seldom. Let us note that the distance $d_{geom}(ii',jj')$ allows for translation, anisotropic scaling and vertical-preserving shear and provides a robust measure for taking into account the deformation of the corresponding Hessian-affine regions. An illustration of this geometrical distance measure is presented in Figure 5.6. The distance is here normalized to the length of the diagonal of the largest image.



(a)

(b)

(c)

Figure 5.6. Affine features matching and mutual projection error. For two matches (i, i') and (j, j') (a)) the mutual projection error of (j, j') with respect to (i, i') is computed are the sum of the length of the red arrow based on the homography transformation $H_{ii'}$ (b)) and the length of the red arrow based on the homography transformation $H_{ii'}^{-1}$ (c)

The role of the spatial distance $d_{spatial}(ii', jj')$ is to privilege pairs of regions which are situated at similar distances. The inclusion of this distance aims at retrieving very similar instances of the queried object in the top results. On the contrary, objects with strong deformations and changes in the relative position of the interest points to each other are penalized by this term. The spatial distance is computed as follows:

$$d_{spatial}(ii', jj') = \left| \|x_i - x_{i'}\|_{L_2} - \left\|x_j - x_{j'}\right\|_{L_2} \right|, \qquad (5\text{-}6)$$

where x_i, $x_{i'}$,x_j and $x_{j'}$ are the position coordinates of features i, i', j and j'. The terms $\|x_i - x_{i'}\|_{L_2}$ and $\left\|x_j - x_{j'}\right\|_{L_2}$ are normalized to the diagonal of their corresponding images. For similar objects a large spatial difference due to a scale change, is compensated by the geometrical distance.

The specification of the distance in Equation (5-2) makes it possible to complete the construction of the affinity matrix associated with the query and candidate image graphs (*cf.* Section 5.1.1). The optimal indicator vector *X** (*cf.* Equation (5-1)) is then determined with the help of the greedy spectral matching approach presented in Section 0. The indicator vector specifies a candidate object $\mathcal{G}^P$that optimally matches (with respect to the considered greedy spectral technique), a sub-graph $\mathcal{G}^Q$ of the query model. Let us underline that, in contrast with the previously considered approaches such as region-based and BoW, the method allows to match not the entire query model, but solely a sub-part of it, which optimally fits a sub-set of the candidate image.

In order to enable object-based retrieval and similarity ranking, an adequate similarity measure needs then to be associated to the obtained candidate object. In our case, we have adopted a fusioned similarity measure which includes both interest points matching cost and color similarity.

The interest point matching measure takes advantage of the eigenvector decomposition of the affinity matrix *M* (*cf.* Section 5.2). The values of the principal eigenvector represent the confidence/strength of each match as it was computed from the affinity matrix containing the similarities between corresponding points and pairs of corresponding points. Once the best correspondences have been identified with the greedy algorithm proposed by Leordeanu and Hebert [Leordeanu 2005], the confidences of these correspondences can be summed in order to have a global measure of the similarity between the query object and the current image containing a candidate object instances. We will refer to this measure as spectral score and denote it as $SM_{score}(\mathcal{G}^P,\mathcal{G}^Q)$. Notice that it value grows with the number of matches and their quality, hence it is a pertinent evaluation measure of the similarity of two images.

The color similarity measure, denoted by $d_{color}\left(\tilde{\mathcal{G}}^P,\tilde{\mathcal{G}}^Q\right)$ is the quadratic color measure introduced in Section 3.2, Equation (3-1), and applied to the optimal sub-graph configurations determined by the spectral matching technique.

Finally, the fusioned similarity score is defined as:

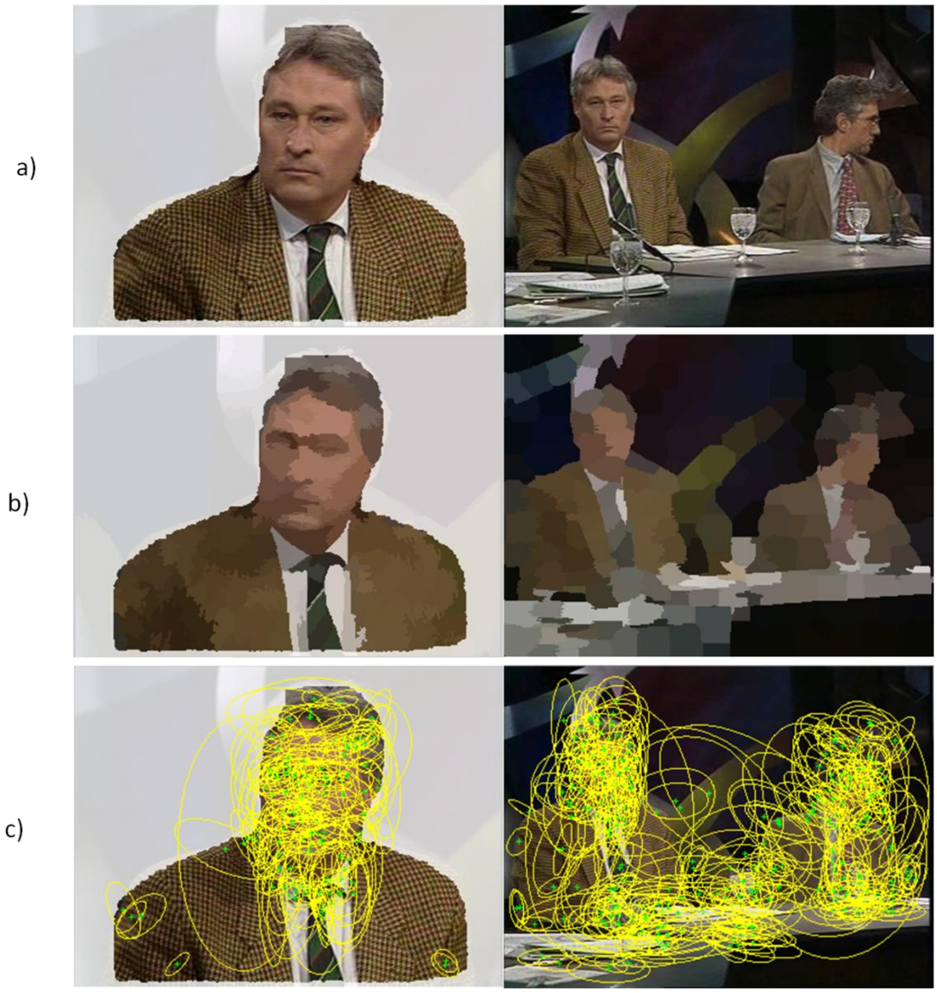
$$d_{fusioned}(\mathcal{G}^P,\mathcal{G}^Q) = \lambda \times \left(1 - d_{color}\left(\tilde{\mathcal{G}}^P,\tilde{\mathcal{G}}^Q\right)\right) \times SM_{score}(\mathcal{G}^P,\mathcal{G}^Q).\,, \qquad (5\text{-}7)$$

where parameter λ weights the importance of the color similarity measure and $\tilde{\mathcal{G}}^P$ and $\tilde{\mathcal{G}}^Q$ represent the sub-group of regions from image *P* and *Q* corresponding to the points that have been matched by the spectral matching technique.

Let us mention that, depending on the use case, other descriptors can be embedded in this score as weighting factors (*e.g.*, ratio of query interest points that have been matched, ratio of query regions that have been matched).

Figure 5.7 illustrates the spectral graph matching object retrieval mechanism. In Figure 5.7a we illustrate the query model on the left and a candidate image on the right, which contains an instance of the object of interest. Their corresponding superpixels segmentation is shown in Figure 5.7b. In this case the superpixels are tuned to yield approx. 250 segments per image. Figure 5.7c illustrates the Hessian affine covariant regions detected in the video frames. The initial matches obtained with the relaxed threshold are presented at Figure 5.7d. Figure 5.7e displays the pairs of matches from the two frames. Here, corresponding point pairs are connected with an edge illustrated with the same color. Figure 5.7f shows the optimal correspondences obtained with the spectral graph matching technique, while Figure 5.7g illustrates the corresponding regions.

The color similarity term in $d_{fusioned}$ is obtained by performing the quadratic color distance between the regions emphasized in both frames. Let us mention that the percentages of the regions from both query and test image are updated before the score computation according to the current region configuration.



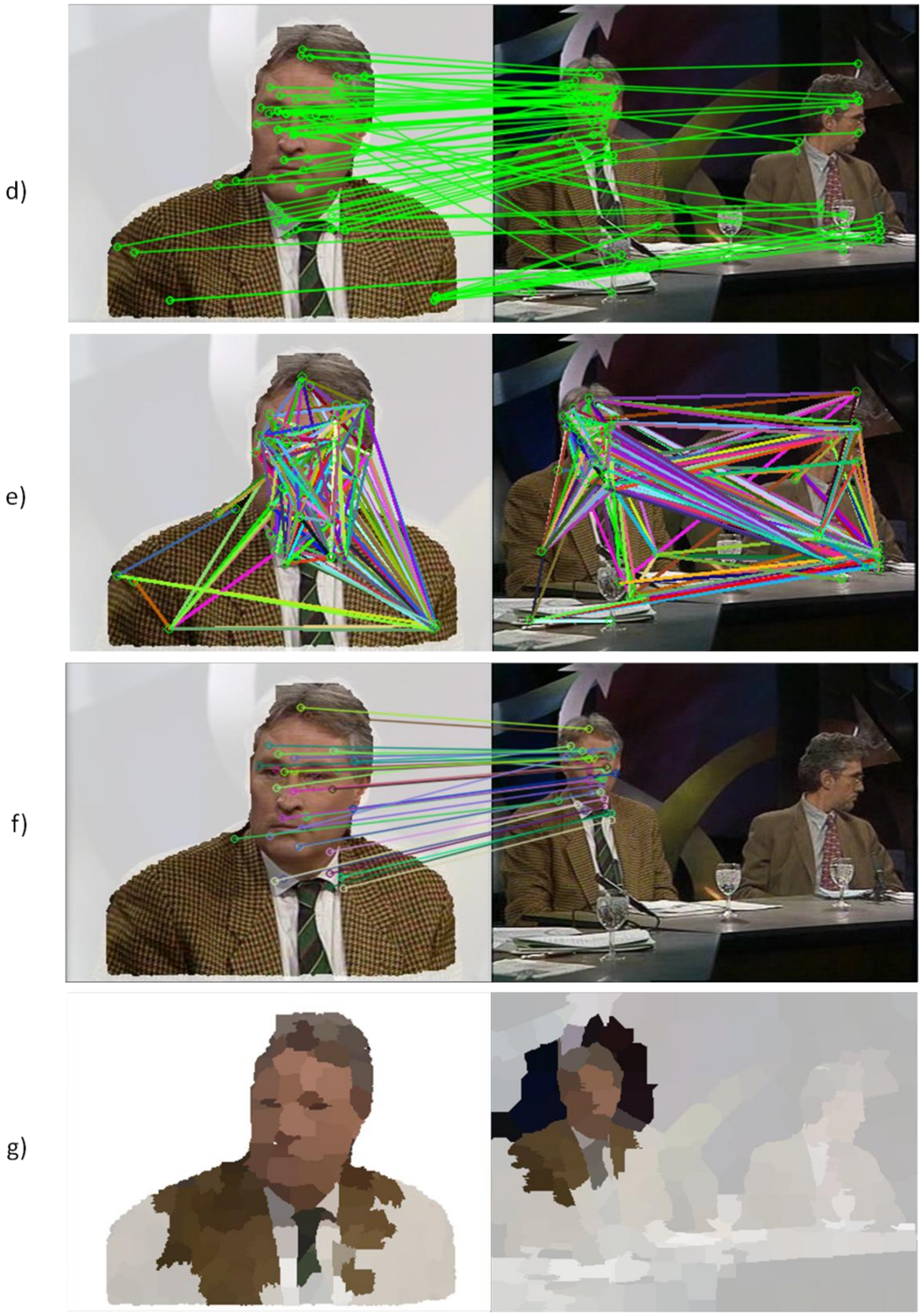


Figure 5.7. Workflow for object retrieval using Spectral Matching.

5.3 Conclusion and perspectives

In this chapter, we have introduced a novel hybrid representation technique which includes both region-based representations and interest-point approaches. The proposed technique makes it possible to extend the adjacency information naturally included in the region-based representations to the interest points.

The matching between a query model and image candidate is formulated as a graph matching problem, which is solved with the help of a greedy spectral matching technique. Such an approach makes it possible to integrate not only the unary similarity between nodes in the graph, but also the pairwise similarity associated with pairs of matched interest points. A merged similarity measure, integrating both interest point and color similarities has been proposed and exploited for object-based retrieval objectives.

This framework can be further extended by identifying the most representative clusters of interest points and their neighbor points for a given image. A graph can be constructed from the interest points and their neighbors. The most influential/representative vertices can be then identified by employing algorithms such as the Google Page Rank algorithm [Page 1999]. These clusters would represent meaningful objects or parts of objects that could be considered as object candidates for object recognition algorithms. Alternatively; separate BoW representations can be computed for each such influential node along with its neighbors resulting in multiple BoW vectors per frame similarly with [Wu 2009]; each representing, at least partially, an object.

6. Experimental results

Abstract. *In this chapter we evaluate the performances of the proposed methods. In this respect we employ four different datasets corresponding to different use cases (e.g., search in cartoon datasets, search in moderate size datasets using a single object query, search in large scale databases using multiple object queries for the same object). We evaluate the influence of the segmentation methods and of the color space on the region-based representation. We then analyze and compare the performances of all the proposed methods. Concerning the large scale search, we test the influence of the resolution of the videos to the search results and compare the performances of the frame and shot descriptors in the context of a query with multiple example instances. Finally, we describe different settings for the query expansion and evaluate their performances on the Flickr dataset.*

Keywords: *evaluation, experiments, TRECVID, MAP, Bag-of-Words, query expansion.*

Résumé: *Dans ce chapitre, nous évaluons les performances des méthodes proposés. A cet égard, nous comptons quatre corpus différents, correspondant aux différents cas d'utilisations (e.g., recherche dans des collections de dessins animés, recherche dans de collections de vidéos de taille modérée en utilisant une seule requête pour l'objet, recherche dans des bases de données à grande échelle en utilisant plusieurs requêtes pour le même objet). Nous évaluons l'influence des méthodes de segmentation et de l'espace colorimétrique sur les performances de la représentation basées sur régions. En ce qui concerne la recherche à grande échelle en utilisant plusieurs requêtes, nous testons l'influence de la résolution des vidéos sur les résultats de la recherche et nous comparons les performances obtenus avec les descripteurs des plans et des trames dans le contexte d'une requête avec plusieurs exemples des instances. Enfin, nous décrivons des configurations différentes pour l'expansion des requêtes et nous évaluons leurs performances sur le corpus Flickr.*

Mots clés: *evaluation, experiences, TRECVID, MAP, Sac-de-Mots, extension de requête.*

In order to experimentally validate the various approaches proposed, we have considered different test data sets, which are described in the following section.

6.1 Test datasets

6.1.1 Raymond dataset

The Raymond corpus has been established by the "2minutes" company [2minutes 2012] company within the framework of the HD3D[2] French research project. The corpus consists of 13 cartoon videos, each with the duration of about 7 minutes. This corpus has been proposed in order to test and develop different use cases for content editors and producers that usually deal with this type of content.

Such videos have of course the particularity of presenting relatively flat colors, which facilitates the segmentation task. However, they present an impressive number of characters displaying the same palette of colors. Moreover these characters are often located in crowded scenes (up to 20 characters in certain scenes), which is quite a challenge for object detection purposes. The videos have been first segmented in shots and for each shot, a representative key-frame has been determined with the help of the methods recently proposed in [Tapu 2011]. We thus constructed a database of 1630 video shots with 1630 corresponding key-frames. We have retained 12 query objects (Figure 6.2), interactively selected by different users. The queries correspond to different characters or objects present in the videos, with different postures.

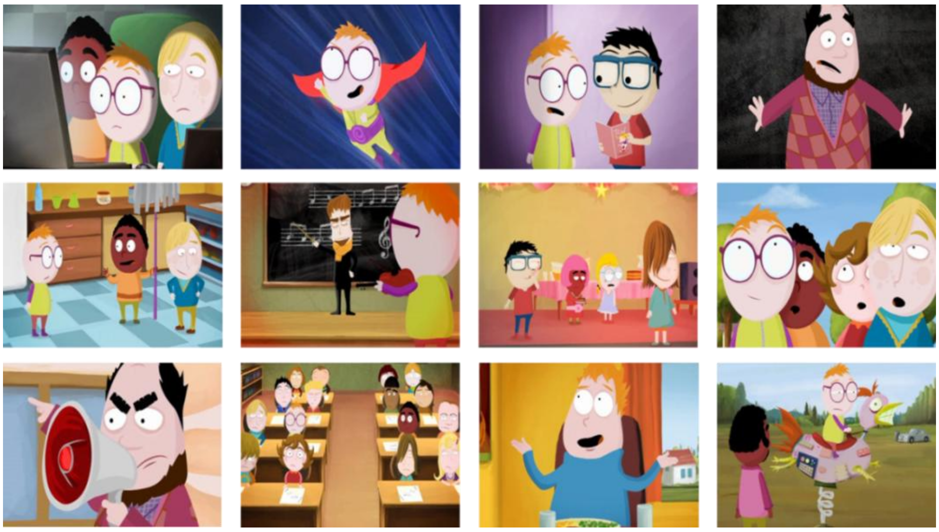


Figure 6.1. Samples from the Raymond dataset.

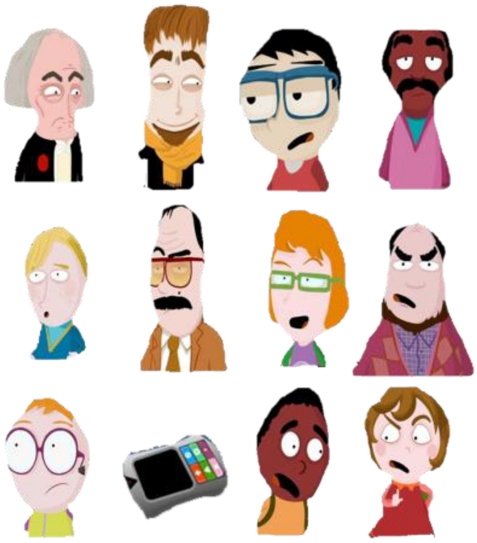


Figure 6.2. The query objects considered from the Raymond corpus for our evaluation.

The rest of the data sets retained include natural videos and have been considered during the last three years within the framework of TRECVID campaigns.

6.1.2 TRECVID 2010 Sound and Vision dataset

The *Sound and Vision* dataset has been extensively used in the TRECVID evaluation campaigns between 2007 and 2010 [Smeaton 2006]. The dataset consists of videos from news magazines, science news, news reports, documentaries, educational programming, and archival video. This dataset contains recurring people (*e.g.*, presenters, magazine hosts, characters in comic skits), objects and locations (Figure 6.3). This dataset has been proposed for testing for the pilot edition of the Instance Search Task within TRECVID 2010. All videos are provided by the Netherlands Institute for Sound and Vision from the Dutch multimedia archives.

The length of the videos ranges between 20 and 120 minutes. The videos are compressed at a relatively reduced resolution 384 x 288 in MPEG-1.

In this respect, we have used 34 videos from the *Sound and Vision* dataset summing up to approximately 15 hours of video content. A total number of 5580 key-frames corresponding to the same number of shots has been obtained with the method described in [Tapu 2011]. In order to objectively evaluate the performances of our algorithms, we have retained 12 query images (Figure 6.4), interactively selected by different users. The queries correspond to different characters present in the videos, with different postures. The query specification is not precise, the user being requested to select approximately the desired objects of interest.

Figure 6.3. Samples from Sound and Vision TRECVID 2010 dataset.

Let us underline the difficulty of the considered database, where a same object can appear in various conditions, with variations in lightning conditions (indoor/outdoor scenes), pose and partial appearances (Figure 6.5). We have then established a ground truth set of relevant key-frames, by exhaustively selecting the set of all key-frames where the query object appears.

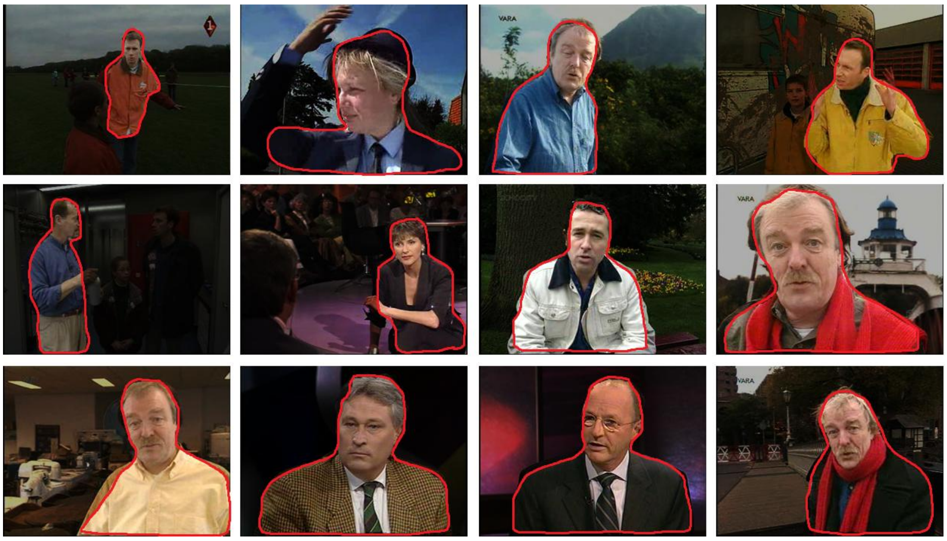


Figure 6.4. The query objects considered in the Sound and Vision ground truth data set and retained for our evaluation.

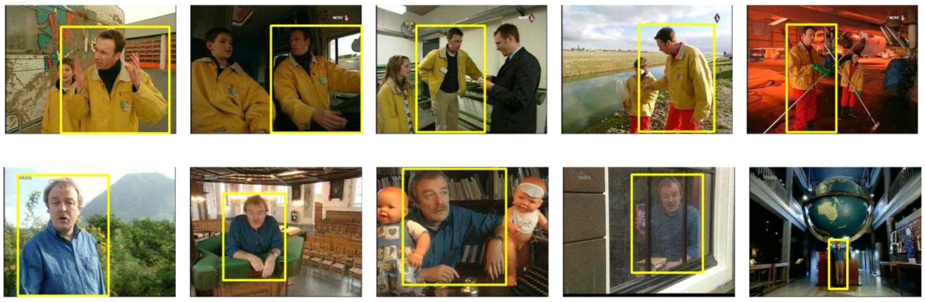


Figure 6.5. Various appearances of a same object in the Sound and Vision dataset.

6.1.3 TRECVID 2011 BBC Rushes dataset

The *BBC Rushes* dataset has been proposed for testing for the second edition of the INS task of TRECVID 2011. Rushes are the raw material from which programs/films are made in the editing room. The rushes from this dataset include those for several dramatic series as well as for travel programming and contain recurring objects, people and locations.

For this dataset, the videos have been divided in short clips with lengths between 1 and 30 seconds. This results in 10491 short video clips (Figure 6.6). In order to simulate differences that might occur if a clip came from a different camera or was taken under different lighting conditions, different transformations, including change of gamma, contrast, brightness, hue or aspect ratio (Figure 6.7) have been considered. Thus, each of the above mentioned clips is applied such transformations and the final dataset consists of 20982 video clips. In Figure 6.7 we can notice that these transforms affect significantly the quality of the video and enforce extreme conditions for feature detectors and descriptors.

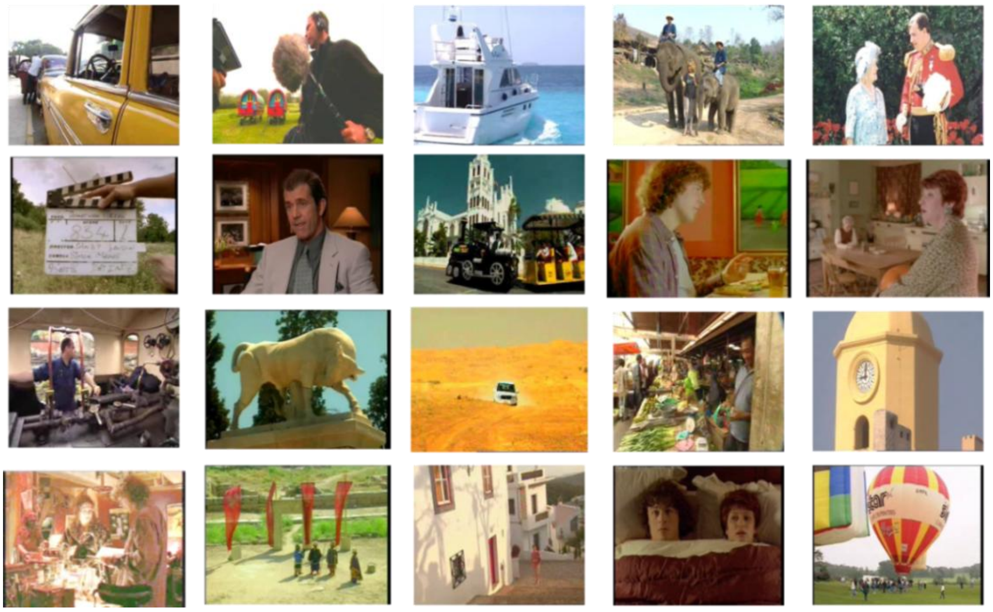


Figure 6.6. Samples from BBC Rushes 2011 dataset.

The encoding parameters are similar with the Sound and Vision dataset: resolution of 352 x 288 and MPEG-1 compression.

A number 25 search topics have been proposed for this dataset: 17 objects, 6 persons and 2 location entities (Figure 6.8). Each search topic is provided with 2 to 5 example images containing instances of the object of interest and a binary mask illustrating its position in the image. The average number of positive topics is around 73 video clips.

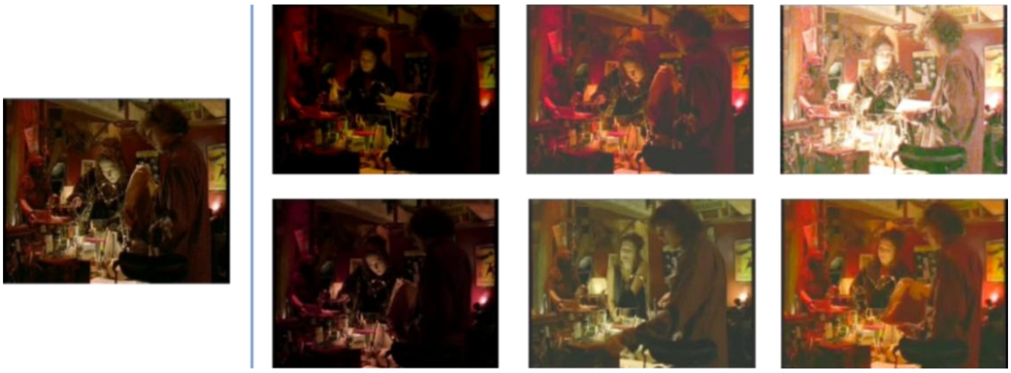


Figure 6.7. Example of content transformation in the BBC Rushes dataset for increased difficulty.

6.1.4 TRECVID 2012 Flickr dataset

The *Flickr* dataset has been introduced in 2012 for the INS task in TRECVID. The dataset consists of 74958 short video clips downloaded from Flickr and summing up to 188 hours of content. Unlike the previous datasets, this dataset contains general public videos with most of the content generated by non-professional users. The variety of content is high and many different types of clips can be retrieved in the dataset (*e.g.*, sports events indoor or outdoor, concerts, travel videos, family events, pets, personal blogs, news magazines, advertisements …). Figure 6.9 illustrates samples of the content of this dataset. A challenging aspect of this dataset is the multitude of devices used for recording, which results in a multitude of video quality levels and different video resolutions (52 different resolutions). The videos are encoded in webM format [WebM 2012].

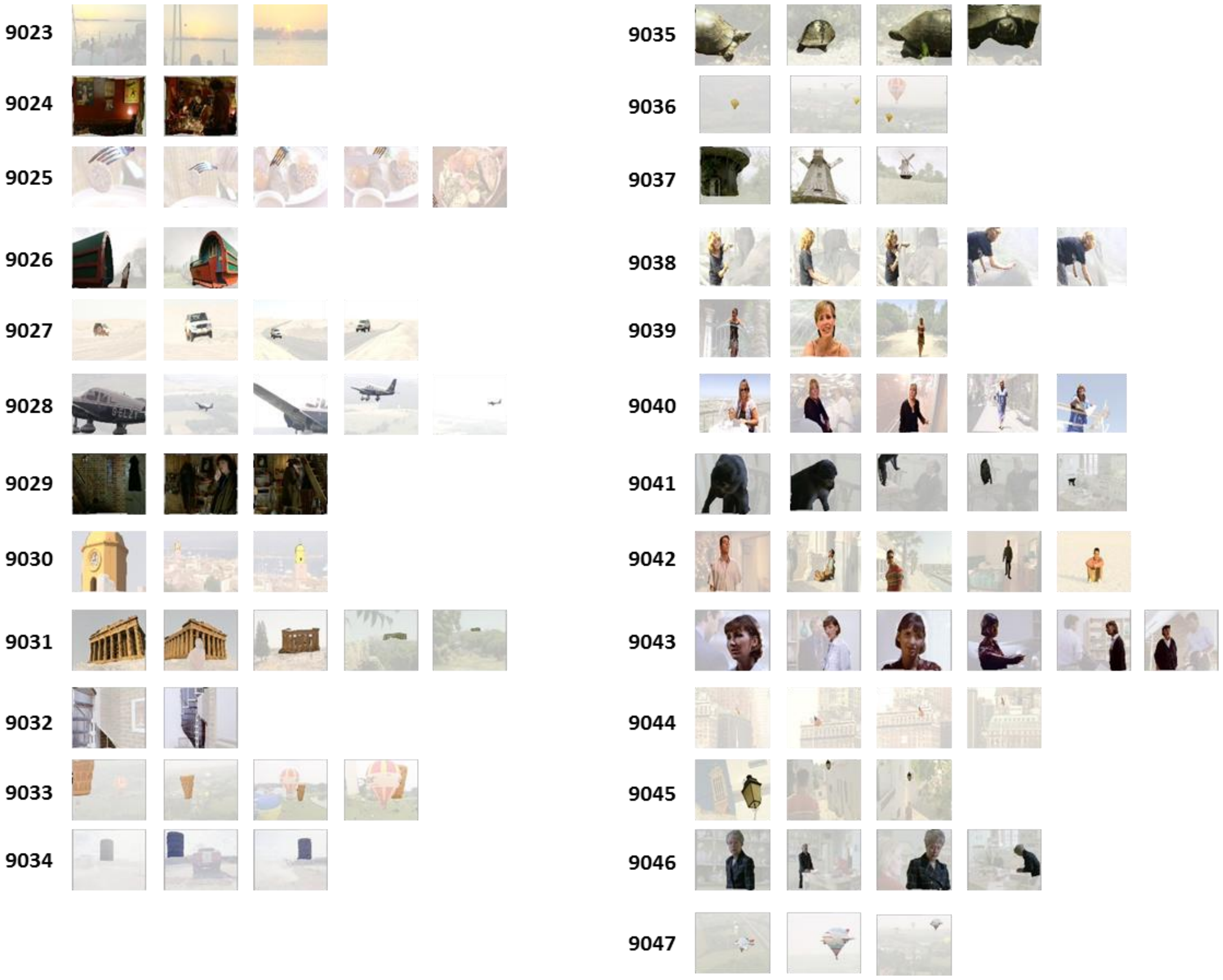


Figure 6.8. Queries and example images for the BBC Rushes dataset. The query objects are emphasized in each image.

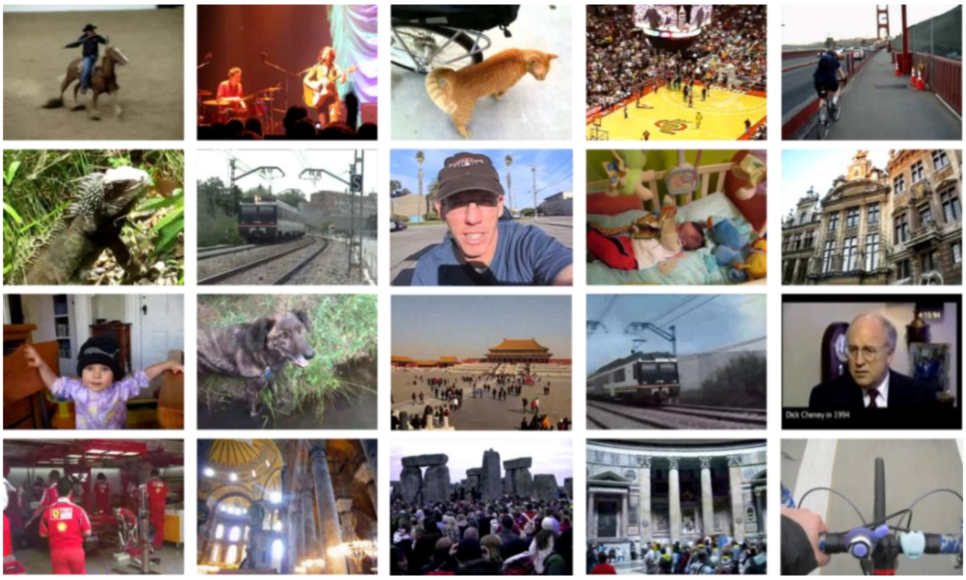


Figure 6.9. Samples from Flickr Trecvid 2012 dataset.

Another key property of this dataset is represented by the queries. In this case, the queries are popular objects, landmarks and persons that are usually the subject of search for a large number of users. The queries proposed for this dataset within the INS task are illustrated in Figure 6.10 and their textual descriptions are displayed in Table 6-1. Each query topic features up to 9 example images with different instances of the respective topic, to be used to construct a visual representation of the topic and query the dataset. The main challenges to address for this dataset are how to extract discriminative descriptors from the relatively small dimension of the query objects (*e.g.*, Mercedes logo- 9048, Sears/Willis Tower - 9055) and how to construct a robust query model from example images containing very different instances of the same object (*e.g.*, Pantheon interior – 9056, Prague castle – 9063).

Table 6-1. Textual descriptors for topics proposed for the Flickr dataset.

Topic	Textual description	Topic	Textual description
9048	Mercedes star	9059	Baldachin in Saint Peter’s Basilica
9049	Brooklyn bridge tower	9060	Stephen Colbert
9050	Eiffel tower	9061	Pepsi logo - circle
9051	Golden Gate Bridge	9062	One World Trade Center building
9052	London Underground logo	9063	Prague castle
9053	Coca-cola logo - letters	9064	Empire State Building
9054	Stonehenge	9065	Hagia Sophia interior
9055	Sears/Willis Tower	9066	Hoover Dam exterior
9056	Pantheon interior	9067	MacDonald’s arches
9057	Leshan Giant Buddha	9068	PUMA logo animal
9058	US Capitol Exterior		

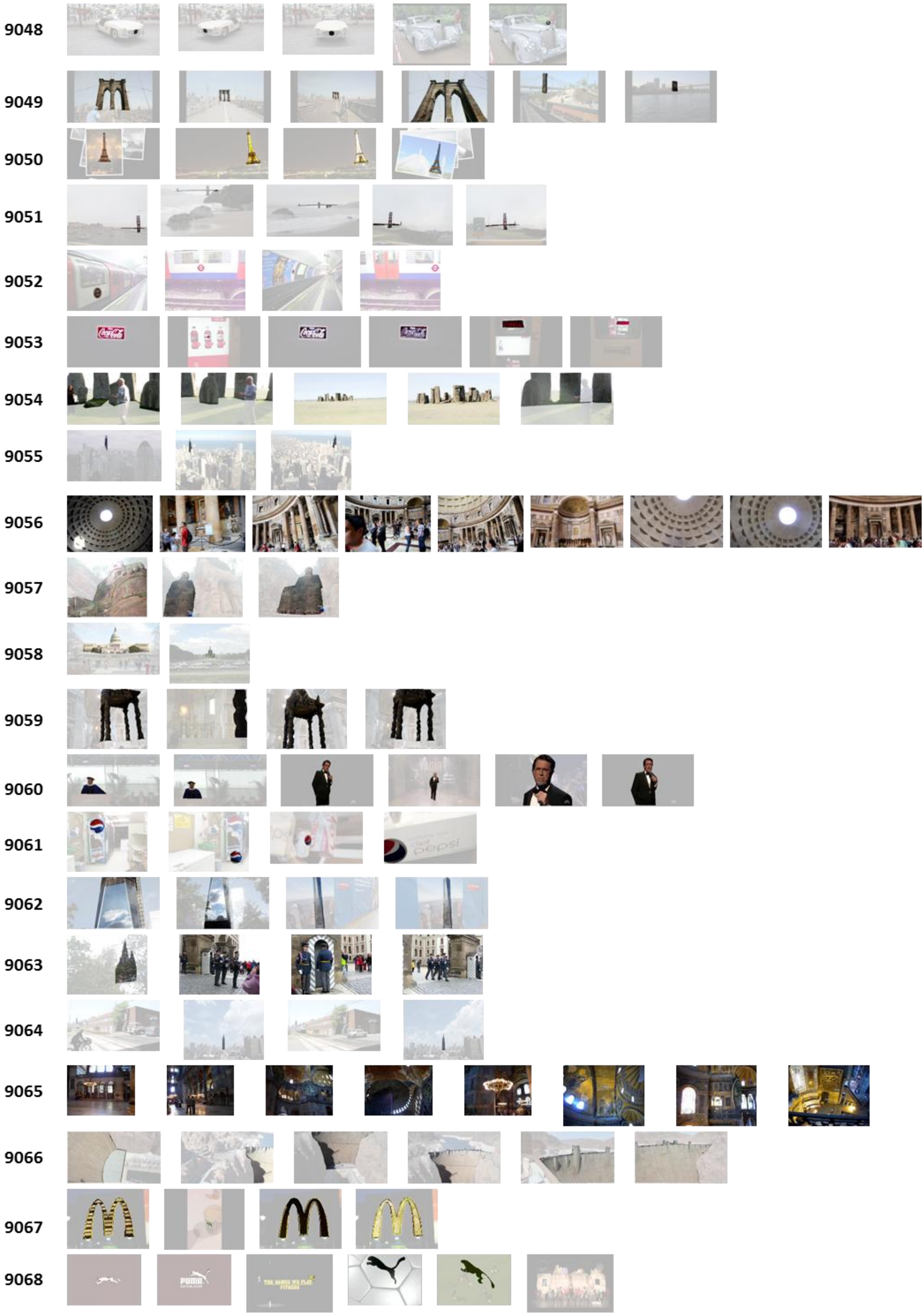


Figure 6.10. Queries and example images for the Flickr dataset. The query objects are emphasized in each image.

6.2 Evaluation measures

A first evaluation measure, largely adopted by TRECVID and PascalVOC campaigns is the average precision, recalled in the next section.

6.2.1 Average precision

The average precision is a metric for estimating the accuracy of ranked query results, which is commonly used in both information retrieval [Baeza 1999] and computer vision, especially in the TRECVID tasks [Smeaton 2006].The average precision takes into consideration both precision (recording the fraction of retrieved videos which are relevant to the query) and recall (which records the fraction of relevant videos which are successfully retrieved). The average precision is a single-valued measure that is proportional to the area under the precision-recall curve. This value is the average of the precision over all shots judged relevant. Let $\rho^k = \{r_1, r_2, \dots, r_k\}$ be the ranked list of top k retrieved items from test set T for an arbitrary query. At any given rank k, $\left|G \cap \rho^k\right|$ represents the number of relevant shots in the top k retrieved results, where G is the set of all relevant shots for the considered query. The average precision, *AP*, is then defined as:

$$AP(\rho) = \frac{1}{|G|} \sum_{k=1}^{|T|} \frac{\left|G \cap \rho^k\right|}{k} \delta(r_k) \qquad \textbf{(6-1)}$$

where $\delta(r_k) = 1$ if $r_k \in G$ and zero otherwise. $|T|$ is the number of answers in the returned ranked list. When performing experiments over multiple queries, the average precisions of individual queries can be aggregated for an overall evaluation of the performance. This aggregation is called mean average precision (MAP). MAP is obtained by computing the mean of the average precisions of the considered queries. Let us mention that in general MAP depends on the dataset and the queries used and scores of different datasets are difficult to compare.

The AP metric has been successfully employed in information retrieval and computer vision. Yet its global measure does not accurately cover some cases where one might want to measure the precision in the first ranked results. In addition, when the size of the searchable dataset is reduced (*e.g.*, a film with a few hundred shots at most) the AP might get biased as it measures the precision over the whole list of results, which is in this case quite short. This would introduce a false boost in precision, as some results might be considered among the first results, only because of the small sized of the list. In order to overcome such limitations, we have adopted a *precision@ k* score.

6.2.2 Precision@ k

For such cases we propose the *Precision@ k* measure as performance criteria. Given the set of relevant shots G for an object and the list of retrieved results T, the *Precision@ k* for a given rank *k* measures the percentage of relevant shots retrieved within the top *k* results from the list T. We compute the *Precision@ k* for two values of *k*, for $k = card\ |G|$ and for $k = 2 \times card\ |G|$. We will refer to the former as *First Tier* (FT) and for the latter as *Second Tier* (ST) or *Bull Eye* (BE). These criteria measure the accuracy of the first retrieved results, where the user usually looks in. If the relevant content is not displayed among the first results, the user usually refines his textual query or issues a new query-by-example starting from one of the retrieved results. Let us not that for reduced sizes of G (*e.g.*, 5 instances), this measure is rather severe as it only looks in the first 2 × 5 results.

6.3 Compared evaluation

We perform a first set of experiments on the Raymond dataset in order to evaluate and compare the proposed optimization techniques related to the region-based representation introduced in Section 3.

6.3.1 Raymond cartoon dataset

In the case of such cartoon images, which are rather flat, the number of segments for each frame is rather reduced with an average of 77 color regions per frame, obtained with the MeanShift segmentation technique (*cf*. Section 3.1.1).

We test the methods in the RGB color space and consider the Euclidean distance as similarity measure between region colors. We test three of the proposed methods: Greedy, Relaxed Greedy with a tolerance of 5%, 10% and 20% and the Simulated Annealing implementation with different values of the parameters T_0 (0.5 - 0.9), T_{fin} (10^{-5}-10^{-4}) and n_{it}.(5-10) leading to 66-109 temperature changes. For each considered query and method, we have performed 2 sets of runs. Firstly, we have launched the query only inside the video of the query object in order to test a typical use case for a regular user. This process is called *intra-video search*. Further on we have launched the query in the whole dataset in order to test the scalability of the proposed methods (*inter-video search*). Experiments have been run on a PC with Intel Xeon CPU W3530 at 2.80 GHz and 12 Gb RAM. Computational times are display in Table 6-2. Let us not that these times correspond to a raw implementation of the algorithms with no parallelization or significant memory access optimizations.

Table 6-2. Computational times for RGB color space on Raymond dataset.

Run	**Greedy**	**Relaxed Greedy (δ=10%)**	**SA** (T_0 = 0.5, n_{it}=5 T_{fin} = 10^{-3})
Intra-Video time	1.5s	1.7s	20s
Inter-Video time	20s	22s	175s

Next, we have computed the *Precision@ k* scores (First Tier and Second Tier) for each query. The results are summarized in Table 6-3. The average FT score obtained when performing queries inside a given video with the greedy algorithm is of 77%. Concerning the SA method, we obtained a slightly better recognition rate of 79%.

In order to investigate the scalability issues, we have also performed the queries within the entire data set of 1630 key-frames. In this case, the performances naturally degrade with detection rates of 61% for the greedy approach and of 71% for the SA technique. By providing a more global solution, the SA approach shows a more scalable behavior. However, the price to pay is the time of response to the queries: about 160-180 seconds for the SA approach, with respect to less than 20 seconds for the greedy algorithm.

Table 6-3. Experimental results on Raymond dataset.

Run	Intra video search		Inter video search	
	FT	ST	FT	ST
Greedy	77.8	87.1	61	69.1
Relaxed Greedy (δ=10%)	78.8	88.5	71	71.5
SA	83.5	90.1	73.2	80.0

We also noticed that there are some differences between the two methods in the order of similarity of the retrieved frames, with the SA better returning nearly identical objects (Figure 6.11).

Moreover, when performing a finer analysis, we observe that the SA method improves the value of the score for the frames containing objects similar with the query. In order to quantify this refinement we have measured the distances obtained between a query object and the image out of which it has been selected. In average, the SA distances prove to be 20.28% inferior to those provided by the greedy algorithm. This result proves the SA method obtains more global optima, which offers interesting perspectives in terms of scalability, in the case where larger datasets have to be considered.

In order to validate our methods in more realistic conditions we have extended our experiments on natural videos from the TRECVID INS datasets. We structure our experiments according to two instance search use cases. The first concerns the cases where the user disposes of a single object instance and wishes to retrieve other instance of the same object. This is typically the case of most query-by-example searchers, where the user wants to exploit the visual information from a frame or part of a frame to retrieve similar content. In the second use case the user disposes of a couple of frames containing different instances of the same object and wants to retrieve other ones resembling the small ground truth that he has. At limit, this problem can be posed as a classification technique with very limited ground truth sets. This case is covered by the TRECVID INS task.

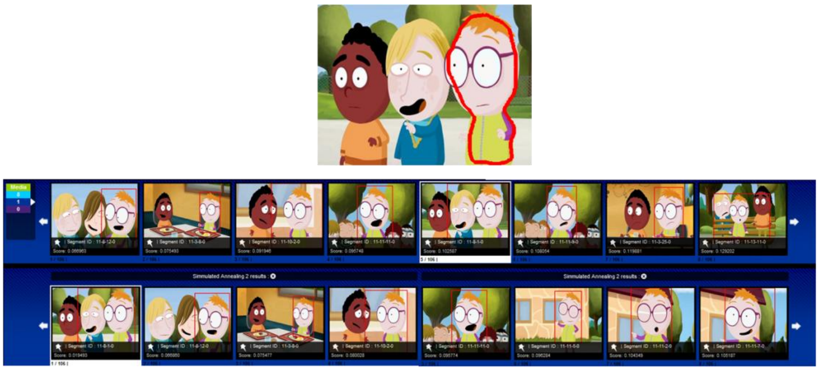


Figure 6.11. First retrieved results with both the greedy region construction (upper row) and the SA (lower row) methods. The order is improved with the SA method and the obtained global scores are lower.

We will start with the case of the single queries that we have tested on videos from the Sound and Vision dataset.

6.3.2 Single queries

6.3.2.1 Experimental setup

In our experiments we consider the 34 videos from the Sound and Vision dataset of natural videos used for the TRECVID 2010 Instance Search Task. We have divided the 15 hours of video content scenes and video shots, resulting in 5580 shots, each represented by a key-frame.

In order to test the influence of the choice of color space over the results, we have tested our methods in the RGB, HSV and CIE-L*a*b* color spaces. For the RGB color space we have considered the Euclidean distance as a similarity measure between region colors, while for HSV color we used the cylindrical color similarity introduced in [Smith 1996]. For the CIE-L*a*b* color space we have retained the two distance similarity measures so-called CIE76 and CIE94 [Brainard 2003].

In order to test the applicability of our framework to different segmentation methods, we have considered the MeanShift method [Comaniciu 2002], the SuperPixels proposed by Veksler *et. al* [Veksler 2010] and the efficient graph-based segmentation algorithm of Felzenszwalb and Huttenlocher [Felzenszwalb 2004]. SuperPixels decompose the frames in regions of similar sizes and all frames present equivalent numbers of superpixels. The efficient graph-based segmentation of [Felzenszwalb 2004] is fast and has been found well-suited for generating an over-segmentation for object recognition [Pantofaru 2008, Kang 2011, Sande 2011]. We tuned the segmentation algorithms in order to obtain recognizable object entities. MeanShift yield an average of 160 regions/frame, with the number of regions varying between 1 and 350 regions. For the SuperPixels technique, we considered three different versions, corresponding to average numbers of 250, 500, respectively 750 segments/frame. The efficient graph-based segmentation was tuned to generate an average of 292 region/frame (*i.e.*, for image with surface areas near 384 x 288 the parameter setting is: sigma = 0.5, K = 200, min = 20 [Felzenszwalb 2007]), but it displayed a strong variation in the number of regions per image (between 30 and 1030 regions/frame). The average number of regions per frame obtained for this segmentation is approximately 300. We will refer to these segmentation methods as MeanShift 160, SuperPixels 250, SuperPixels 500, SuperPixels 750, and EGBS 300. Figure 6.12 illustrates typical segmentation results using the considered segmentation techniques.

The 12 query images described Section 6.1.2 have been considered for both intra and inter-video search scenarios.

In a first experiment we tested the influence of the chosen color space over the results.

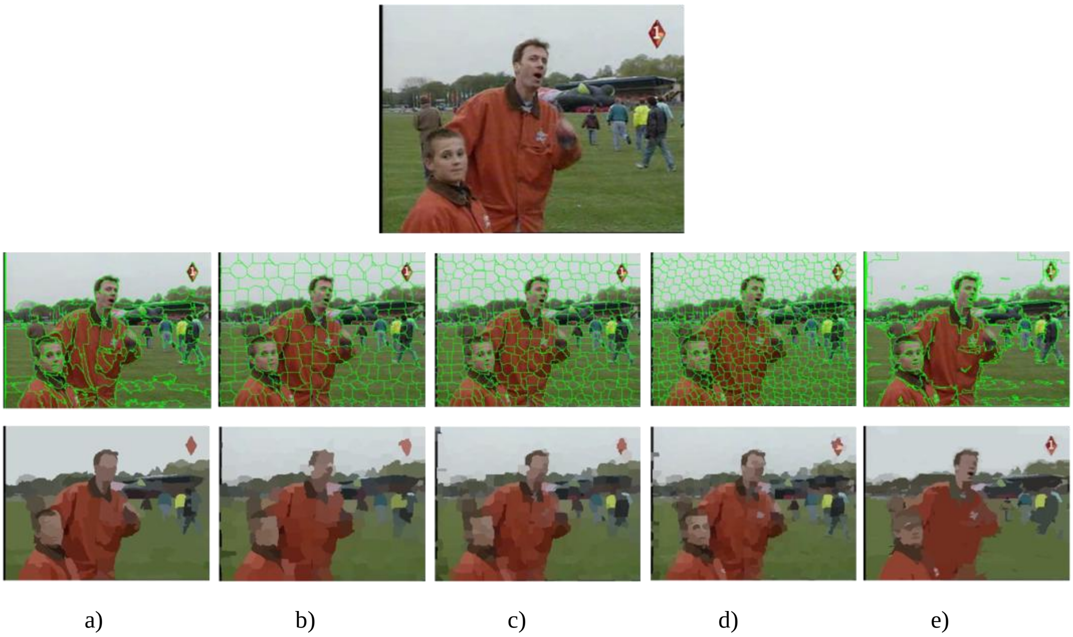


Figure 6.12. Frame segmentations using different segmentations algorithms: a)MeanShift 160; b) SuperPixels 250; c) SuperPixels 500; d) SuperPixels 750; e) EGBS 300.

6.3.2.2 Color space influence

The obtained results are summarized in Table 6-4. The analysis of the results shows that the CIE76 runs offers the highest performances for both intra and inter video search in both cases, while HSV has the weakest performance. The improvement in the case of the greedy methods is more significant: +12% on both FT and ST for the greedy method and +24% and +23% for FT and ST, respectively, comparing to the HSV scores. The basic greedy method is much faster, yet the associated retrieval performances are much inferior, with FT and ST scores of 51.7% and 69.1%, respectively for the RGB run. This drawback is partially eliminated by the relaxed greedy scheme, which achieves higher retrieval scores (FT = 55% and ST = 77%) while offering interesting performances in terms of computational complexity. Let us note that the influence of parameter δ is quite reduced, similar results (±2% in FT scores) being obtained for values of δ between 5% and 20%. For our next experiments we will consider δ = 10% for the Relaxed Greedy runs.

The analysis of the results in the case of inter-video querying shows that the CIE76 runs still perform best, while HSV has again the weakest performance. In the first case the CIE94 and RGB performances were similar, but in this situation the performance of CIE 94 is considerably lower. We have thus retained for further analysis in our experiments solely the RGB and L*a*b* CIE76/94 color spaces and distance measures.

Table 6-4. Experimental results for Greedy and Relaxed Greedy 10% using MeanShift160 when performing intra and inter-video querying on Sound and Vision dataset.

Run	Intra video search		Inter video search	
	FT	ST	FT	ST
Greedy				
RGB	51.7	69.1	19.9	28.6
HSV	39.5	58.8	13.4	16
L*a*b* CIE94	49.1	67.4	12.0	19.8
L*a*b* CIE76	63.6	80.8	29.0	33.4
Relaxed Greedy (δ=10%)				
RGB	55.0	77.2	**32.7**	35.2
HSV	48.9	68.3	15.2	22.1
L*a*b* CIE94	59.3	76.4	16.6	26.4
L*a*b* CIE76	**65.4**	**85.3**	**32. 7**	**44.4**

6.3.2.3 Influence of the segmentation method

Next, we have investigated the influence of the segmentation methods considered on the retrieval results. We want to test the applicability of the DOOR method to other popular segmentation methods such as Superpixels or the Efficient Graph-based segmentation (EGBS). The results obtained for the Greedy runs are presented in Table 6-5. Due to the high number of regions and their small contribution to the global energy, for the SuperPixels750 dataset, we chose a δ of 5% for the Relaxed Greedy run.

We can notice that globally there are no significant variations in terms of retrieval performances, whatever the segmentation technique considered. In the case where the number of segments is increased (*e.g.* above 750 regions) the performance is starting to fade slightly. In such cases, the color information provided by the high number of small regions in object configurations loses distinctiveness. Performance in these situations could be fixed, by adding more spatial information and clustering groups of small neighbor regions and add/remove them at once during the dynamic region construction.

Table 6-5. Experimental results for Greedy and Relaxed Greedy on different segmentation methods performing intra- and inter- video search.

Segmentation method	Run	Intra video search		Inter video search	
		FT	ST	FT	ST
SuperPixels 250	Greedy				
	RGB	58.0	73.4	16.0	25.2
	L*a*b* CIE76	61.3	78.0	20.0	28.6
	Relaxed Greedy 10				
	RGB	59.7	84.3	29.2	35.2
	L*a*b* CIE76	63.8	84.9	**32.7**	44.4
SuperPixels 500	Greedy				
	RGB	50.3	71.8	14.6	23.4
	L*a*b* CIE76	64.5	79.1	19.3	26.3
	Relaxed Greedy 10				
	RGB	58.9	75.6	24.0	34.0
	L*a*b* CIE76	62.8	83.8	31.2	**45.4**
SuperPixels 750	Greedy				
	RGB	50.9	67.5	14.2	23.0
	L*a*b* CIE76	64.5	79.1	19.0	24.9
	Relaxed Greedy 05				
	RGB	58.4	81.3	18.6	25.4
	L*a*b* CIE76	**67.7**	**85.5**	30.2	40.6
EGBS 300	Greedy				
	RGB	50.9	67.5	18.4	24.3
	L*a*b* CIE76	64.5	79.1	20.1	30.6
	Relaxed Greedy 10				
	RGB	54.1	73.6	22.4	30.9
	L*a*b* CIE76	62.2	78.1	25.6	35.3

Another issue that we have considered for experimentation concerns the influence of context on the retrieval performances.

6.3.2.4 *Influence of context*

The current framework employs an interactive query definition system which enables irregularly shaped objects or region entities to be drawn by the user and issued as query. In the previous experiments we have considered only the region configuration inside the contour designed by the user. Yet, in some cases the context information can provide significant information to be used for identifying instances of the same object. In the same time the context information can be noisy and influence negatively the search results.

In order to test the eventual extensions of the framework that would take advantage of the context information, we enlarge the objects defined by the user. In this respect we allow the regions that have been cut by the scribble of the user, to be added to the query configuration. In this way, the visual information within the immediate neighborhood of the object is considered for the search.

In the same time, objects that might have been improperly defined by the user can be partially restored in this case. The principle is illustrated in the Figure 6.13. The man in the query has not been fully selected and some parts are left outside the selection. After the extension of the query, the query object is more complete and some background information has been included in the same time. The superpixels segmentation are more appropriate for this technique as the pixels have similar size the amount of noisy regions that can be considered is reduced. This can be noticed more easily in the case of the MS160 segmentation from Figure 6.14. The sizes of the regions vary and bigger noisy regions can be added to the object. In order to overcome such inconvenient, the regions are added up to a limit defined by an expanded bounding box computed around the query object. We set this limit to a bounding box 15% larger than the original bounding box.

The new queries are tested on the Sound and Vision dataset with the Relaxed Greedy 10 method and the results are presented in Table 6-6. The performance on the MeanShift is lower due to the varying size of the regions. In most of the cases, the extension outside the user scribbles leads to large noisy regions added to the object configuration, affecting the results. The SuperPixels 250 display better results when searching in the same video, while the inter-video results are worse. The expansion advantages instances of the object from the same video as it can be found in a similar background throughout the video or under the same light conditions. In the case of the SuperPixels 500, the gain can be noticed when searching among different videos. Due to the fineness of the SuperPixels500 regions, the expansions bring a fair amount of extra information, while no noise is added.

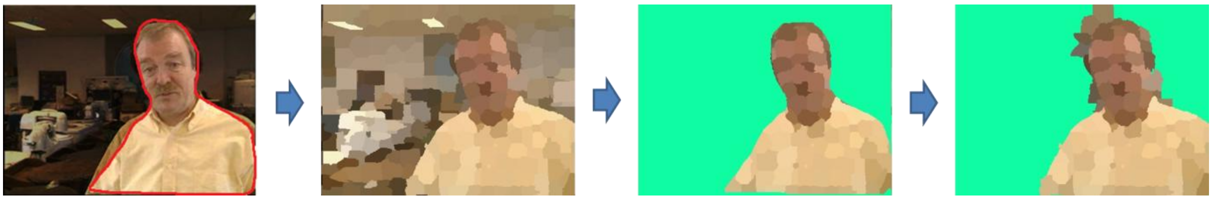


Figure 6.13. Query extension on SuperPixels 250. Regions under the scribble are added as a whole to the query object.

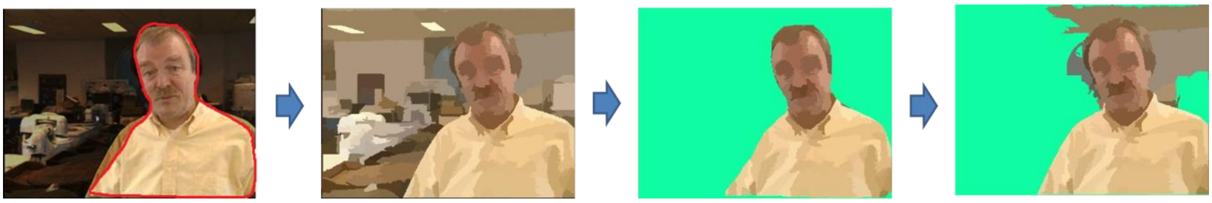


Figure 6.14. Query extension on MeanShift 160. Regions under the scribble are added as a whole to the query object. In order to avoid noisy large regions, the neighborhood is limited to 15% of the surface area of the bounding box around the object.

Table 6-6. Evaluation of extended queries using Relaxed Greedy 10.

Segmentation method	Relaxed Greedy 10	Intra video search		Inter video search	
		FT	ST	FT	ST
MeanShift 160	**Extended Query Bounding Box**				
	RGB	56.1	72.3	26.9	33.4
	L*a*b* CIE76	55.2	75.8	27	34.2
SuperPixels 250	**Extended Query**				
	RGB	54.9	81.3	22.1	32.1
	L*a*b* CIE76	**68.5**	**85.6**	30.7	42.5
SuperPixels 500	**Extended Query**				
	RGB	60.8	80.7	27.4	32.8
	L*a*b* CIE76	66.4	85.8	**33.5**	**46.2**

6.3.2.5 Simulated Annealing optimization

We test the SA method on the same dataset with on a range of different parameters. The SA optimization also presents a stable behavior, with quasi equivalent results for different values of the parameters T_0, T_{fin} and n_{it}. Here, we illustrate the results for the following parameter settings: starting temperature $T_0 = 0.5$, freezing temperature $T_{fin} = 10^{-3}$ and number of iterations to perform for each temperature $n_{it} = 5$.

Table 6-7 summarizes the results of the SA method using the MeanShift 160 segmentation. Overall, the SA-based method yields better retrieval performances than the Greedy methods, with FT and ST scores of 70% and 88%, respectively for CIE76. The gap between results on different color spaces is less significant for the SA runs, the algorithm approaching the optimal solution whatever the considered color space (also for CIE 94, not displayed here). When looking at the results on the whole dataset queries, the decrease in performance is less dramatic for the SA. This shows that the robustness of the SA approach is preserved even for larger datasets. Overall, the improvements of the SA methods are stronger, excepting the CIE76 runs where the Relaxed Greedy scores are still close to SA ones.

Table 6-7. Experimental results for Simulated Annealing using MS160 segmentation when performing intra- and inter- video search.

Segmentation method	Run	Intra video search		Inter video search	
		FT	ST	FT	ST
MeanShift 160	**SA**				
	RGB	66.0	84.1	43.3	47.0
	L*a*b* CIE76	70.2	88.4	45.0	52.2

Table 6-8 presents the results for the SA runs obtained with the SuperPixels 250 and EGBS-300 segmentation methods. The performance of the former is rather similar with the one using MeanShift as region generator. We notice that the results for the EGBS-300 are significantly lower. Due to the large variation of number of regions available in a frame, from 30 to 1030, the SA approach could not reach in this case its optimum. For such particular cases, the SA method should be tuned to perform a higher number of iterations and temperature shifts.

In Figure 6.15 we illustrate some results of the SA method using MeanShift segments in the Lab color space when performing intra video querying.

Table 6-8. Experimental results for Simulated Annealing using Superpixels 250 and EGBS-300 segmentation when performing intra- video search.

Segmentation method	**Run**	Intra video search	
		FT	ST
SuperPixels 250	**SA**		
	RGB	69.5	85.6
	L*a*b* CIE76	72.4	86.8
EGBS 300	**SA**		
	RGB	56.1	70.1
	L*a*b* CIE76	60.6	73.7

Let us now present the results obtained for the GraphCut optimization method.

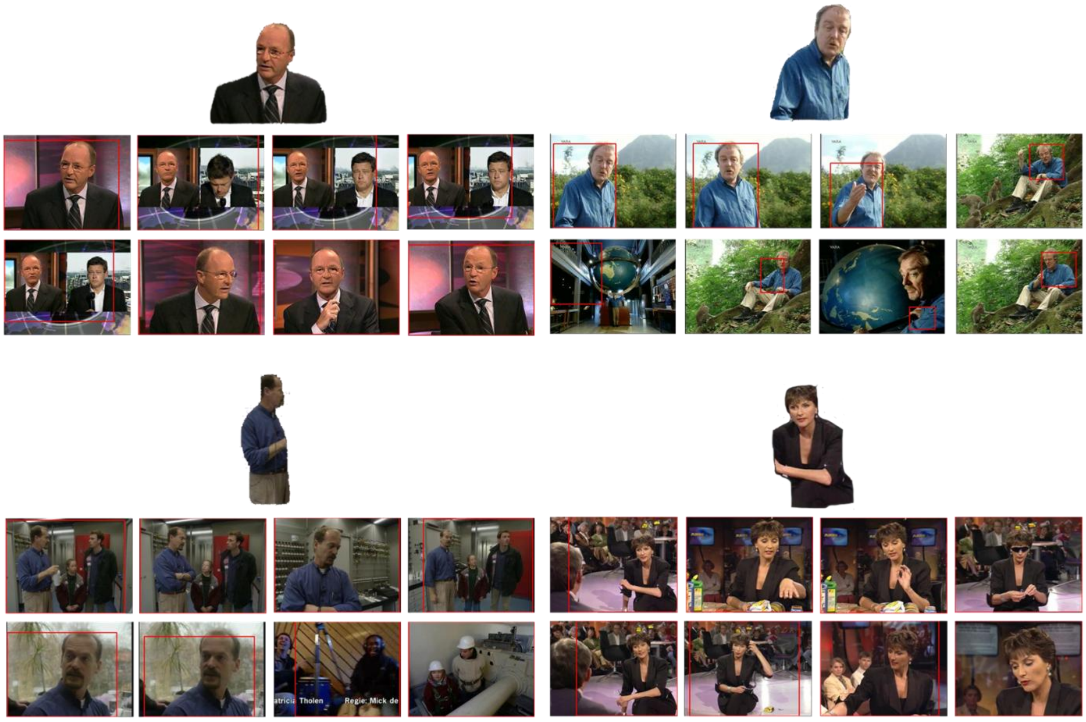


Figure 6.15. Results for SA method on MeanShift 160 when performing intra-video queries.

6.3.2.6 GraphCut optimization

Here, we have considered the MeanShift 160 and SuperPixels 250 segmentation methods; which have provided so far a fair balance between performance and computational cost and the RGB and L*a*b* color spaces. The results are presented in Table 6-9.

The results are clearly superior to the Greedy-based and methods and fairly close to the SA ones (even outperforming it for some queries). Moreover, in the case of the Lab color space, the GraphCut achieves almost the same results as the SA technique.

Table 6-9. Experimental results for the GraphCut-based approach using MeanShift 160 and SuperPixelsP250 segmentations when performing intra- video search.

Segmentation method	**GraphCut**	Intra video search		Inter video search	
		FT	ST	FT	ST
MeanShift 160	**RGB**	61.6	83.1	32.3	44.9
	L*a*b* CIE76	**68.2**	**89.8**	**41.6**	**50.8**
SuperPixels 250	**RGB**	64.3	82.9	39.8	46.0
	L*a*b* CIE76	66.6	86.5	40.9	49.0

6.3.2.7 Computational complexity

The computational times for the Greedy-based, SA and GraphCut methods are presented in Table 6-10. Here, thy experiments have been carried out in the RGB color space and with the MeanShift-160 segmentation method.

Table 6-10. Computational times for RGB color space on Sound and Vision dataset.

Run	**Greedy**	**Relaxed Greedy (δ=10%)**	**SA** (T_0 = 0.5, n_{it}=5 T_{fin} = 10^{-3})	**GraphCut**
Intra-Video time	6s	10s	816s	15s
Inter-Video time	133s	212s	19388s	257s

In terms of computational complexity, the GraphCut method is slightly slower than the Greedy methods, depending on the number of regions in the image: in average, the GraphCut is about 25% slower than the Greedy methods. However, when compared to the SA method, the GraphCut technique is of an order of magnitude faster.

This shows the interest of the GraphCut approach, which offers nearly optimal performances at a reduced computational cost.

Let us now analyze the performances of the spectral matching approach introduced in Section 5.2.

6.3.2.8 Spectral Matching

We have first applied to spectral matching optimization approach to the region-based representation. Here, we have used an experimental setting similar to the one proposed in [Leordeanu 2005] for interest point matching. The matching of the regions is performed by color distance and an experimentally determined threshold is used to filter out the non-similar colors. For the pairwise score, we have considered the Euclidean distance between the position coordinates of the regions from the same pair. The pairwise score is the computed as the difference of such distances. Only adjacent regions are considered for computing a pair of assignments. In this manner the number of possible pairwise terms is significantly reduced. In addition, in order to reduce the number of regions to be considered for generating the affinity matrix, we employ the color pre-filtering to discard regions visually different from the query regions. We then identify the connected components and perform the Spectral Matching on each such connected component.

We use as similarity scores:

(1) the sum of the values collected from the principal eigenvector (the SM_{score} from Section 5.2,), as proposed by [Leordeanu 2005], and
(2) the quadratic color distance (Chapter 3, Equation (3-1)).

The obtained results are presented in Table 6-11.

Let us note that the quadratic color distance provides significantly better results. However, the obtained scores are lower (of 40%) than the GraphCut related performances, while the computational time is one order of magnitude higher than GraphCut.

The main problem here is the low discriminative characteristic of the color distances between the regions. Thus, rather dissimilar regions are considered when constructing the affinity matrix. On one hand, this leads to an increase of the number of elements to be processed. On the other hand, most of the regions are dissimilar and the algorithm fails from identifying correct correspondences. Reducing the threshold for the individual matching, makes it possible to reduce the computational costs, but does not improve the retrieval performances.

We have then extended our experiments to the hybrid object representation, which makes it possible to integrate both color and RootSIFT interest point descriptors.

Concerning the construction of the affinity matrix, we have considered different values of the weighting factors related to the visual, spatial and geometric components (Chapter 5.2, Equation (5-2)), with α between 0.2 and 0.5, for β between 0.5 and 0.7 and for γ between 0.3 and 0.5. The results are quite stable for the different configurations considered. In the following, we will present the results obtained with parameters $\alpha = 0.2$, $\beta = 0.5$, $\gamma = 0.3$.

Table 6-11. Experimental results for the region-based Spectral Matching using MeanShift 160 and SuperPixels 250 segmentations when performing intra- video search.

Segmentation method	Spectral Matching Regions	Intra video search		Inter video search	
		FT	ST	FT	ST
MeanShift 160	**Quadratic color distance**	61.0	72.8	33.5	38.8
	Spectral Matching score	18.01	33.86	1.17	2.71
SuperPixels 250	**Quadratic color distance**	52.2	67.9	18.4	26.9
	Spectral Matching score	12.94	24.3	0.65	0.01

While the evaluation of the global similarity of two images has different solutions in literature, the evaluation of the similarity of two object instances is still a challenging problem. We propose for testing five similarity measure computed from the object entities identified by the Spectral Matching (SM). First, we employ as similarity measure the sum of the cumulated eigenvalues for generating the final object configuration with the greedy Spectral Matching [Leordeanu 2005]. This term encapsulates the similarities between the correspondences and the pairs of correspondences from the different images that have been selected for the final object configuration. We will refer to this measure as SM_{score}.

Second, we employ the quadratic color distance employed for the MPEG-7 DCD representation. We use it to compute intermediate color similarities during the SM process as the most powerful correspondences are identified and added to the region configuration. We compute the color similarity between the different region configurations; both in the query and test image, and retain the best score. We will refer to this score as $d_{color}(\tilde{\mathcal{G}}^P, \tilde{\mathcal{G}}^Q)$.

Further we propose the following three similarity measures, leveraging on the first two measures:

$$FusionScore_1 = \left(1 - d_{color}(\tilde{\mathcal{G}}^P, \tilde{\mathcal{G}}^Q)\right) \times SM_{score} \tag{6-2}$$

$$FusionScore_2 = SM_{score} \times \frac{|SM_{matches}|}{Nb_{query\ points}} \tag{6-3}$$

$$FusionScore_3 = \left(1 - d_{color}(\tilde{\mathcal{G}}^P, \tilde{\mathcal{G}}^Q)\right) \times \frac{|SM_{matches}|}{Nb_{query\ points}} \tag{6-4}$$

where $|SM_{matches}|$ is the number of points that have been matched by SM and $Nb_{query\ points}$ is the number of interest points in the query object.

$FusionScore_1$ leverages on both color information from region color similarity and intensity information of interest points. We use the sub-unitary color similarity score as a weight for SM_{score}, giving it a confidence measure if it scores high on both spectral matching and color similarity or penalizing a cluster of matched interest points with low color similarity.

$FusionScore_2$ and $FusionScore_3$ compute the ratio of matched points from the query object and weight it with SM_{score} and respectively with $d_{color}(\tilde{\mathcal{G}}^P, \tilde{\mathcal{G}}^Q)$.

The results are summarized in Table 6-12. Overall, the performances are lower than the ones of GraphCut and the search time 2-3 times longer (the algorithm was parallelized on 4 threads). Let us not that in this case the affinity matrices were computed at query time. In a large scale framework, the matrices can be computed offline for each pair of images and at query time the data corresponding to the points involved can be read directly and hence improve significantly the search time. Most of the positive results are missed due to very low number of interest points for some frames. This makes it difficult to construct meaningful groups of interest points; even though the segmented regions were used as spatial support.

Let us note that the best performances for the SM have been obtained by two of the fusioned scores, namely $FusionScore_1$ and $FusionScore_3$ which integrate the color information. This illustrates that the color information can successfully compensate the lack of intensity information from less textured images.

Table 6-12. Experimental results for Spectral Matching on hybrid object representation, using MeanShift 160, SuperPixels 250 and SuperPixels 500 segmentations.

Segmentation method	Spectral Matching	Intra video search		Inter video search	
		FT	ST	FT	ST
MeanShift 160	**SM Score**	51.0	63.4	28.2	30.6
	Quadratic Score	42.6	62.6	19.6	24.0
	Fusion Score 1	50.1	64.2	28.6	33.8
	Fusion Score 2	48.9	63.7	27.1	30.8
	Fusion Score 3	47.7	63.6	25.7	31.6
SuperPixels 250	**SM Score**	51.3	62.0	28.3	32.2
	Quadratic Score	45.4	59.9	24.9	29.7
	Fusion Score 1	**53.8**	66.0	**30.8**	33.3
	Fusion Score 2	48.1	63.7	28.8	31.4
	Fusion Score 3	52.0	64.6	30.2	33.7
SuperPixels 500	**SM Score**	51.2	64.9	27.5	30.9
	Quadratic Score	44.7	59.4	23.5	28.3
	Fusion Score 1	52.1	65.2	29.9	32.5
	Fusion Score 2	49.4	64.5	28.9	31.0
	Fusion Score 3	52.9	**66.7**	29.2	**34.2**

6.3.2.9 Methods benchmarking

Finally we evaluate the methods proposed in this work by comparing its results with a state-of-the-art approach for object retrieval in videos. We employ the popular Bag of Words technique which has been successfully used in many efficient video retrieval systems [Sivic 2003, Snoek 2008, Zhu 2012].

From the set of 5580 key-frames we detect Hessian-affine regions [Perdoch 2009] and describe them with RootSIFT descriptor [Arandjelovic 2012]. We obtain approximately 2 million descriptors and cluster them in different vocabularies using approximate k-means [Philbin 2007]. We evenly sample 10% of the descriptors and build two lightweight visual vocabularies of 10,000 and 20,000 visual words. We build two other visual vocabularies using all descriptors from the dataset and obtain a vocabulary of size 50,000 and one of 200,000. After the matching sequence the top 50 results are passed through the geometric consistency check and re-ranked according to the number of verified matches. The performances of the BoW approach (FT, ST and AP) are presented in Table 6-13, in comparison with our proposed methods.

For an easier evaluation of the performances we compute the average precision score which is typically used for evaluating BoW performances. The results illustrate the difficulty of the dataset. Out of all methods the top scorers are obtained by the SA and GraphCut methods, with equivalent performances. The Relaxed Greedy 10 method is ranked on the third position. Out of the interest point–based runs, we notice that the SM with $FusionScore_1$ achieves the best results, outperforming the BoW.

Table 6-13. Benchmark to BoW.

Segmentation method	Run	Inter video search		
		FT	ST	MAP
MeanShift 160	**Relaxed Greedy 10**	32. 7	44.4	0.3461
	SA CIE76	**45.0**	**52.2**	**0.4348**
	GraphCut CIE76	41.6	50.8	0.4299
	SM Fusion Score 1	28.6	33.8	0.2964
SuperPixels 250	**Relaxed Greedy 10**	32.7	44.4	0.3678
	GraphCut CIE76	40.9	49.0	0.4090
	SM Fusion Score 1	30.8	33.3	0.3078
SuperPixels 500	**Relaxed Greedy 10**	31.2	45.4	0.3382
	SM Fusion Score 1	30.8	33.3	0.3003
Hessian-affine RootSIFT	**BOW 10k**	21.0	24.8	0.2209
	BOW 20k	19.8	23.9	0.2102
	BOW 50k	23.6	24.6	0.2352
	BOW 200k	23.7	26.3	0.2456

All BoW runs display low scores. This might be due to the fact that most of the considered queries are poor in texture and hence in corners to be detected and described. In addition the variation in the appearance and pose of the queried objects is strong making it difficult for the interest point descriptors to capture accurately the objects. This corpus is rather difficult for the BoW approach and emphasizes the situations were region-based approaches can be more suitable. Nevertheless the hybrid region representation displayed better performances, showing its usefulness in such type of datasets.

6.3.2.10 Discussion

In this section we have tested the performances of the DOOR methods on natural videos.

We have seen that when dealing with color descriptors on such type of video content, the Lab color space and the CIE 76 color distance measure return the best results. Experiments performed over multiple segmentations displayed similar performances, leading us to conclude that our framework can be generalized to other segmentation methods.

We have evaluated and compared the performance of all methods proposed for identifying the configuration of regions (*i.e.*, sub-graph of the region adjacency graph) yielding the highest similarity score. The SA and GraphCut optimization methods provided the best results, followed by the Relaxed Greedy 10 method. The comparison with the BoW technique has illustrated that our region based representation is competitive on difficult medium sized datasets. The hybrid object representation and the Spectral Matching technique have provided less competitive results, but outperformed the BoW runs.

While the computational times of the SA method make it prohibitive for any object retrieval engine, the Graph Cut and the Relaxed Greedy could be employed for such tasks on medium sized dataset provided that they are optimized (*i.e.*, better memory utilization, parallelization). The main drawback of the proposed DOOR methods is the computational cost, unlike the BoW which returns results in less than 1 second. This shows that, in the case of inter-video search and retrieval from large databases, the most suitable use case of our methods is a re-ranking phase, applied after a high speed retrieval method such as BoW.

An advantage of our DOOR methods is the short indexation time. The indexation consists in decomposing the video into shots and over segmenting the key-frames. Practically, videos can be indexed as they are uploaded to a storage server. In a BoW framework, whenever a new video is added to a collection, the vocabulary of the entire dataset needs to be updated with the descriptors for the current video or a new dedicated vocabulary is generated for the video. This process takes longer (approximately 1 hour) than the segmentation of the frame. This feature of the DOOR methods can be very handy for general public applications where users are interested in annotating quickly their videos.

6.3.3 Multiple queries

In the second part of our experiments we tackle the issue of retrieving instances of an object starting from a small set of two to nine images containing the object of interest. Since this is also the use case of the TRECVID Instance Search Task we employ the BBC Rushes dataset used in INS TRECVID 2011 and the Flickr video dataset from INS TRECVID 2012.

6.3.3.1 BBC Rushes

Here, we have considered a limited number of key-frames per shot (up to 4). We have then over-segmented each such keyframes in order to obtain a semi-global image representation with the MeanShift 160 method (Figure 6.16).

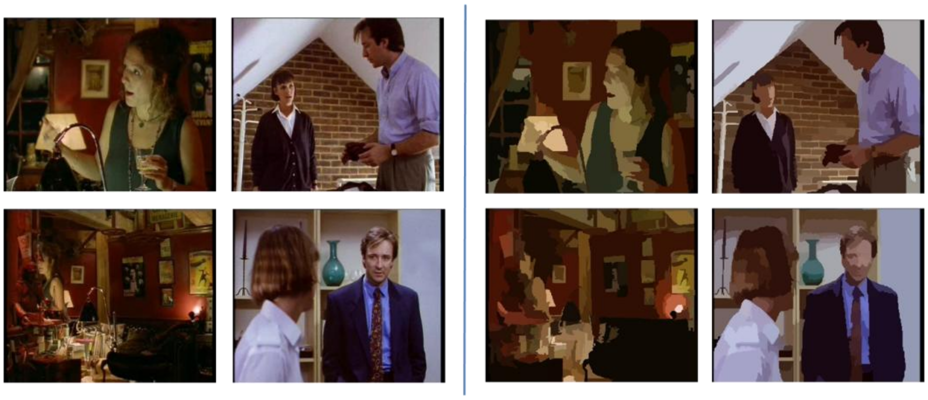


Figure 6.16. Video frames (left) and their segmentations (right). An average of 160 segmented regions provides in this case a “recognizable” image.

Experimental setup

The workflow of the offline processing of the videos is illustrated in Figure 6.17.

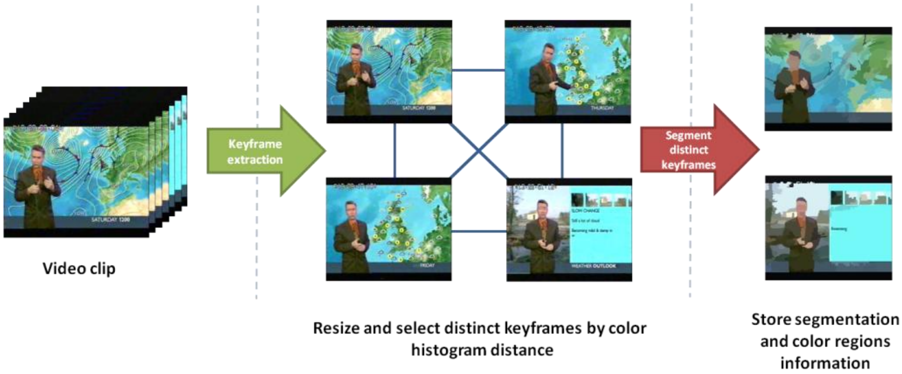

Video clip
Keyframe extraction
Resize and select distinct keyframes by color histogram distance
Segment distinct keyframes
Store segmentation and color regions information

Figure 6.17. Offline processing flow.

Most of the clips from the BBC Rushes dataset consisted of sequences containing usually one or two shots. We have extracted from these clips up to 4 keyframes at equal intervals of time. Furthermore, we have filtered out the near-duplicate frames by using a color histogram distance computed between all the frames extracted from a video clip. A total of 34614 distinct key-frames were retained and then segmented. The videos have been resized to 352×288 pixels, which corresponds to the resolution of a majority of movies among the video clips in the data set. We have left unchanged the frames with lower resolutions and modified the segmentation parameters to generate a proper number of color regions.

We have then performed searches using the Relaxed Greedy 10 scheme and the results from different examples of the same query topic were ranked according to their best score among all runs.

In order to test the influence of the number of key-frames considered for search we have performed another run on the unfiltered key-frames (approx. 72k) and on set obtained with a shot boundary detector tuned to detect light changes in the scenes (approx. 120k) [Tapu 2011]. The average computational time for the 34k dataset is 29 minutes and for the 72k dataset is 50 minutes. We have also tested the SA algorithm on the 72k dataset in order to compare its performances with the Greedy-based method. The results are presented in Figure 6.18.

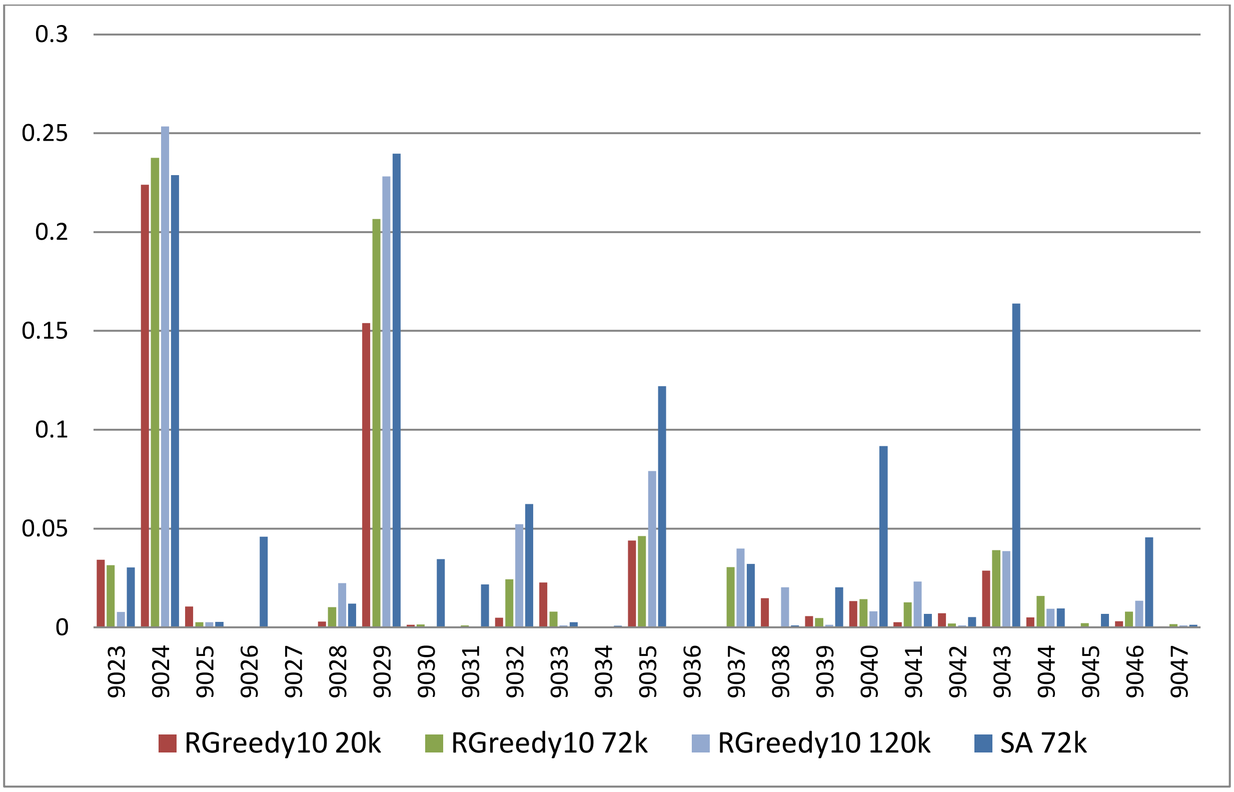


Figure 6.18. Experimental results on BBC Rushes dataset using Relaxed Greedy 10 and SA.

Results

We can notice that the performance increases with the number of considered frames and that overall the SA outperforms the Relaxed Greedy runs. The best performances are obtained for queries corresponding to location/background items. In this case the query image is larger and this feeds more region information in our methods. We notice that for some queries (*e.g.*, 9024, 9028, 9037, 9041) the

Relaxed Greedy outperforms by little the SA runs. This is a hint that in some cases the SA does not reach the global optimum, as suggested in [Boykov 2001].

Recently, Zhu *et al.* [Zhu 2012] have performed extensive experiments on this dataset and concluded that the dataset is a background oriented one. In fact when comparing the results of a query using only the content under the binary mask (*i.e.* foreground) with the results of a query using only the content outside the binary mask (*i.e.* background), the latter provided a higher performance. In our experiments we have considered only the information under the binary mask. An improvement in the performance could be then made by considering the contextual information of the queries.

6.3.3.2 Flickr dataset

We extend our experiments on the Flickr dataset which contains mainly general public videos.

Experimental setup

We have seen in the previous experiments that the number of considered key-frames is a significant factor for improving performance. In this respect, for this dataset we consider a higher number of frames per video and evenly sample 1 frame per second. This results in approx. 683,000 frames to process. In order to lower the computational cost we resize these frames to a surface area near the one of 384 x 288 resolutions. We then extract the Hessian-affine regions and the RootSIFT descriptors. We finally obtain 245,575,000 regions and their descriptors. We sample 10% of the descriptors and cluster them in a vocabulary of 1 million visual words using the approximate k-means [Philbin 2007]. In order to test the influence of the frame resolutions to the search we compute a different vocabulary from the descriptors extracted from the original frames. Thus we extract 482,440,000 regions and descriptors, sample 5% of them and obtain a vocabulary of 1 million visual words.

We propose for testing two quantization strategies: frame-based and video shot-based (*cf.* Section 4.4 and 4.5). For the former, the frames are quantized individually resulting in 683,000 BoW vectors. For the latter, descriptors from the frames of the same shots are grouped and the quantization is performed at the shot level, resulting in 74958 BoW vectors. This reduces significantly the computational cost at query time.

Concerning the contextual information, we use the binary mask provided within the TRECVID 2012 INS task. When the number of interest points is less than 20, we compute a bounding box around the object defined by the binary mask and expand it gradually until at least 20 interest points are considered. This makes the current object “searchable” by leveraging on its contextual information, but not including the whole image content in the search, which might be noisy.

Influence of video resolution

First we compare the influence of the resolution of the indexed videos in the search performances. For all runs we consider the original sized queries in order to capture as much information as possible from the query images. We propose for testing the shot based BoW description and we summarize the results in Figure 6.19.

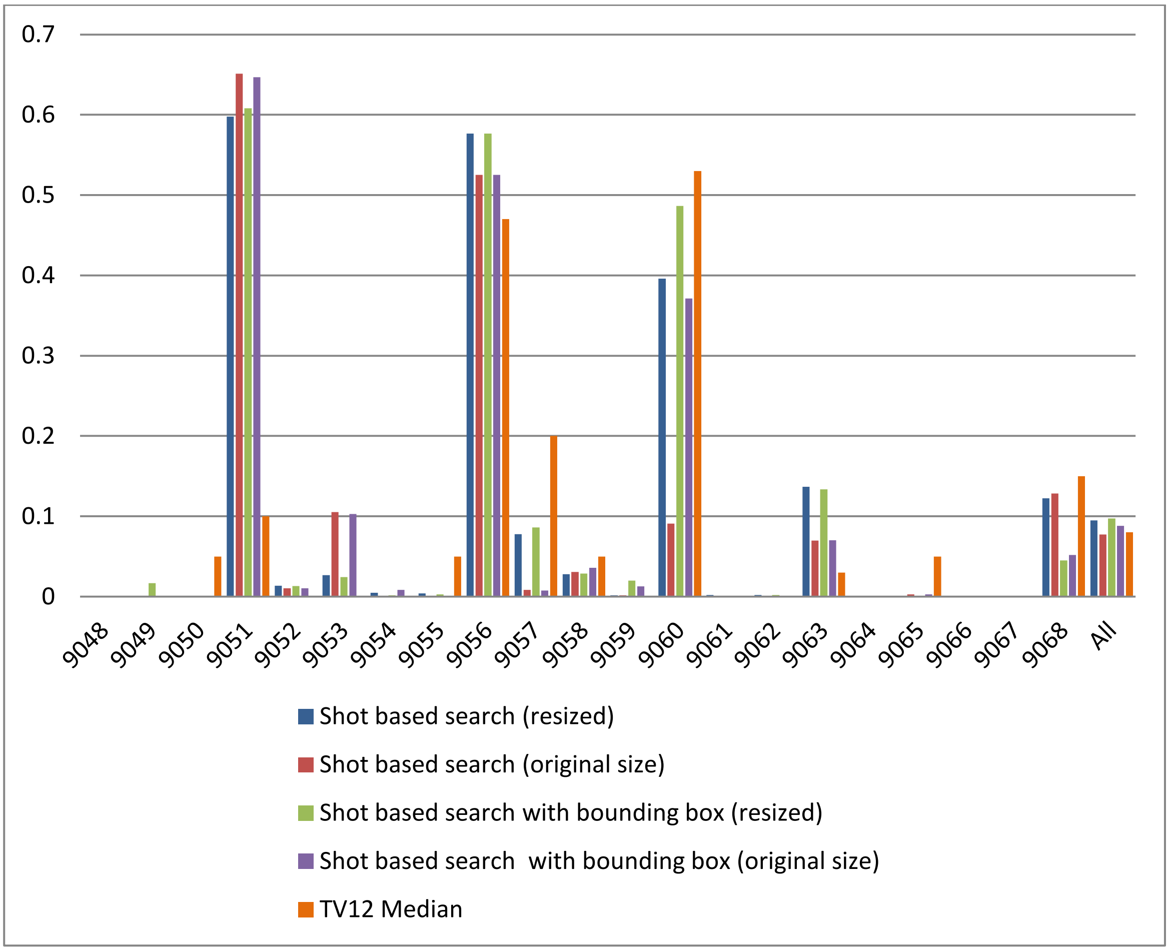


Figure 6.19. Overview of results on Flickr dataset using original and resized videos.

We can notice that the results for both methods are comparable, with a slight higher performance for the resized videos (approximately 0.01 in MAP). In addition the context information brought by the bounding box increases the performances of both approaches with 0.005-0.01 in MAP. The MAP for all runs is illustrated in Table 6-14. The lower results for the original sized videos might be caused by the increased number of descriptors in each video clip. Thus many noisy descriptors from the BoW representation could deteriorate the score. In the following experiments we will employ the resized dataset which yielded higher performances.

We have also compared the performances of the shot based BoW with the frame based BoW on the Flickr dataset. In the case of the frame based BoW, we launch a query for each query example image and then aggregate the results into a single list by selecting the best score for each video among all queries. The MAP of the frame based run is 0.0352, which is significantly lower than the MAP of the shot based BoW, 0.0975. This drop in performance is due to the multiple aggregations required for compiling the results from multiple queries. Firstly from each video shot we select the frame yielding the best score for a query, then for each video we select again a best score among all runs. These steps can discard positive candidates that yield a weaker similarity score with one of the queries. In addition, since some topics display rather different instances of the same object, the top results and the similarity scores for the different queries might be different. A top score for one of the queries can be a weak score for another one, so this can cause results that are relevant for the other query to be discarded. Following we will be using the shot based BoW descriptors.

Table 6-14. MAP on Flickr dataset using original and resized videos.

Dataset type	**Run**	MAP
Original videos	**Object mask**	0.0774
	Object bounding box	0.0879
Resized videos	**Object mask**	0.0947
	Object bounding box	**0.0975**
TV12 Median		0.0795

Computational complexity

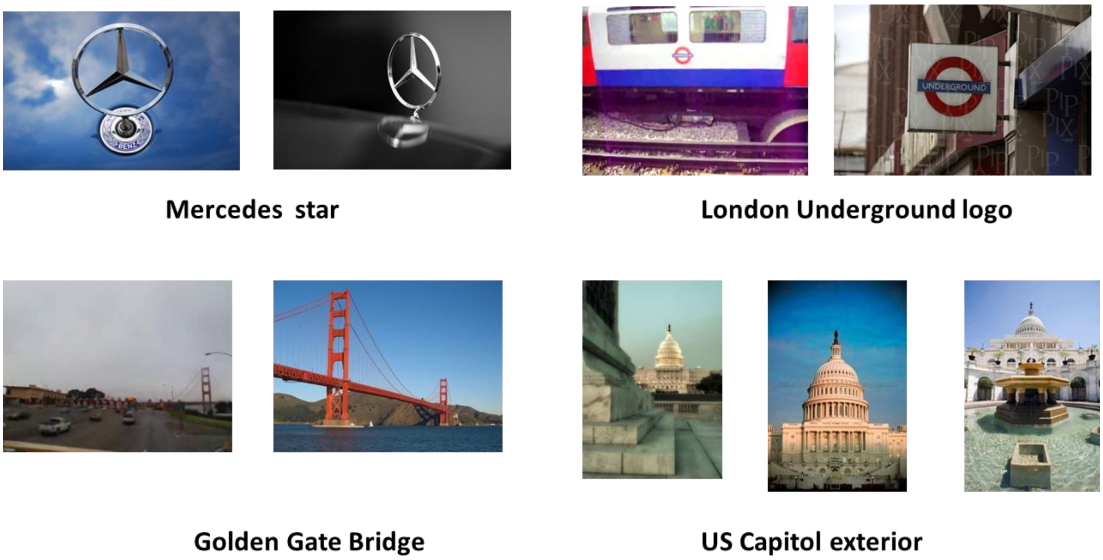
One of the main advantages of the BoW method concern it's speed during the live querying stage. In our experiments a typical query in the Flickr dataset performed among the BoW vectors (74958) took 3 minutes at most on a single machine on 8 paralellized threads. The time depends on the number of descriptors in the query and on their distinctiveness (*i.e.*, the number of videos where the descriptors have been retrieved through the inverted index) and it can drop to 20 second for a query with few interest points.

However during the off-line phase, the computational times are significantly higher. On the same machine, the detection of the Hessian affine regions and the extraction of their descriptors take up to a couple of days for 683,000 key-frames and the random sampling of 10% of the descriptors and their clustering takes up to one day and a half. The quantization of the video clips into BoW vectors can take up to 2 days.

Query expansion evaluation

We now proceed to the evaluation of the query expansion algorithm proposed in Section 4.6. We have downloaded from Flickr 3 sets of images of sizes $N_{download}$ = 25, 50, 75 using the textual information provided by the producers for each query. Let us note that since we have used the exact textual information provided (Table 6-1), some queries could not be retrieved in more than a couple of instances (*e.g.*, "Puma logo animal") making the topic not-searchable in the dataset as not enough visual information is available.

For each query, we determine a query graph and identify the representative images as connected components in this graph. Concerning the required minimum number of geometrically verified matched interest points (N_{min}) for two images to be connected with an edge in the query graph, we have experimented with values between 5 and 20 points. The performance was slightly better for N_{min} between 5 and 10. For the experiments presented below we considered N_{min} = 5. Some examples of representative images determined are illustrated in Figure 6.20.



Mercedes star

London Underground logo

Golden Gate Bridge

US Capitol exterior

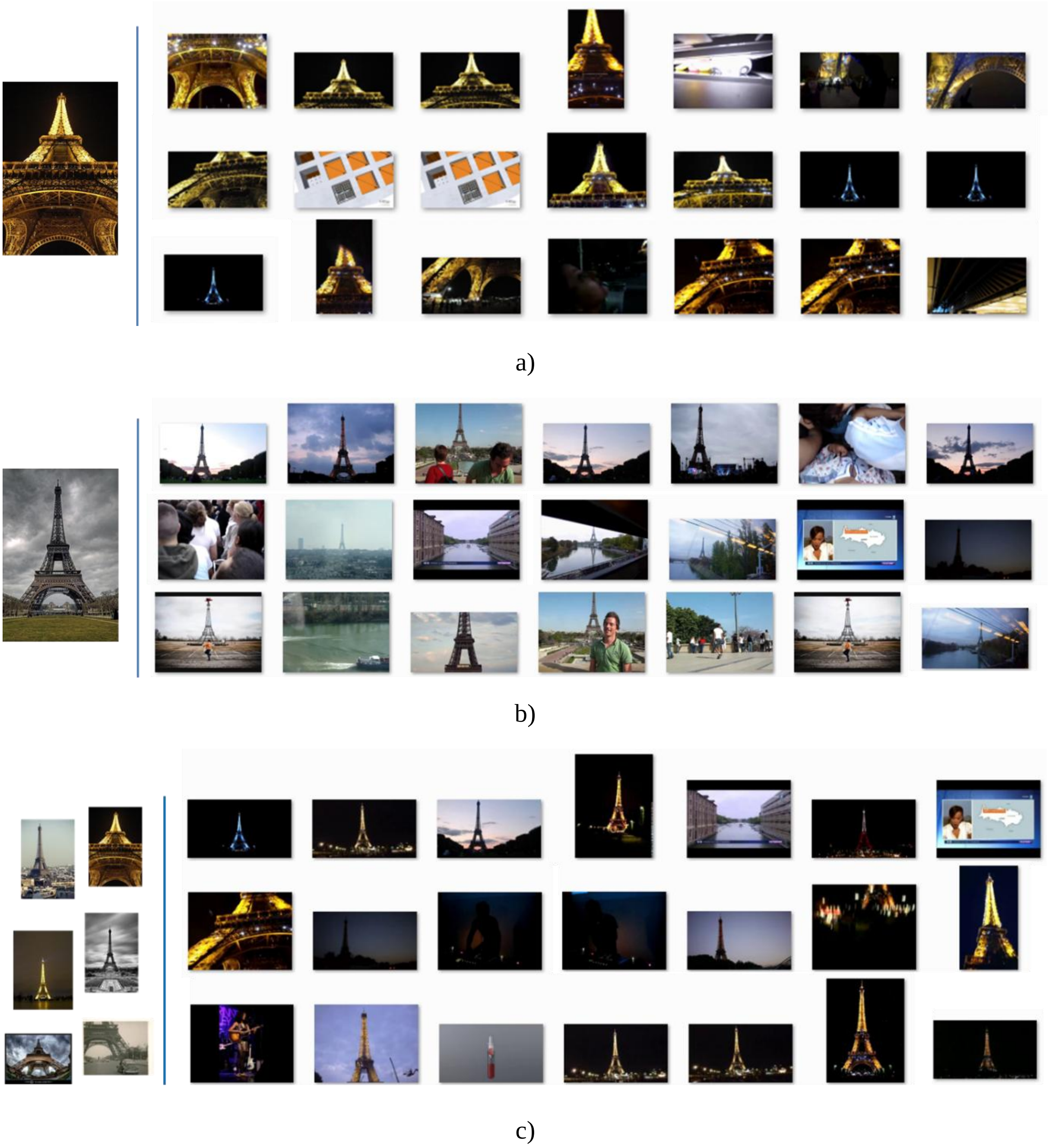
Figure 6.20. Representative images identified for different queries.

In order to describe the representative images and their matches, we test two complementary methods: centered and distributed representative queries. For the former we have considered solely the descriptors from the representative image that have been matched. In this way, points that have been matched multiple times will have their descriptors added for quantization multiple times. For the distributed representation, we consider the descriptors from the images that have been matched with the iconic image. In this manner more diverse descriptors will be considered in the query.

The images that we have downloaded from Flickr are visually different from the video frames. In order to see if there is any improvement brought by the TRECVID query images, we introduce them in the query graph generation.

The results of our experiments are summarized in Table 6-15. For each representative image, we issue a separate query and then the results for representative images associated to the same topic are aggregated in a single ranked list, by selecting the best score for each video among all runs. In Figure 6.21, we illustrate some results for different iconic queries belonging to the same topic (*i.e.*, “Eiffel tower” - 9050). We can notice that object instances similar with the one in the representative image are retrieved. These results are then aggregated in a single list for the topic.

We can notice that the best results are obtained for the collections of 50 images. For smaller collections; the number of identified collections is too low. In such cases, if no representative image has been identified, we choose the pair of images with the highest number of matches and construct an iconic descriptor from the matched descriptors from both images. The performance is still below the collections of size $N_{download}$ = 50. For large collections, the increased number of representative images and descriptors become noisy and degrade the performance.



a)

b)

c)

Figure 6.21. Results for different iconic queries (left column) belonging to the "Eiffel tower" topic – 9050. Each image corresponds to the first frame extracted from the retrieved shot. The lower section illustrates the aggregated list from a set of 6 queries.

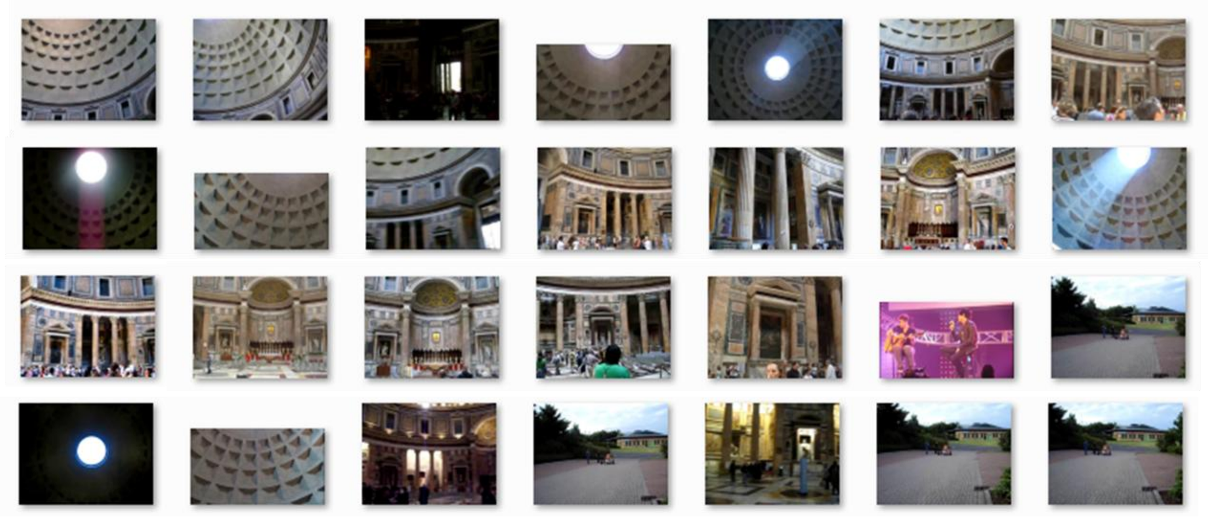
The influence of introducing the TRECVID 2012 queries within the query collection is positive, the retrieval performance in terms of MAP score being very slightly improved (with up to 0.01). Let us note that we have here used the query images only for building the matching graph and not for issuing queries in the dataset.

Table 6-15. Experimental results on query expansion.

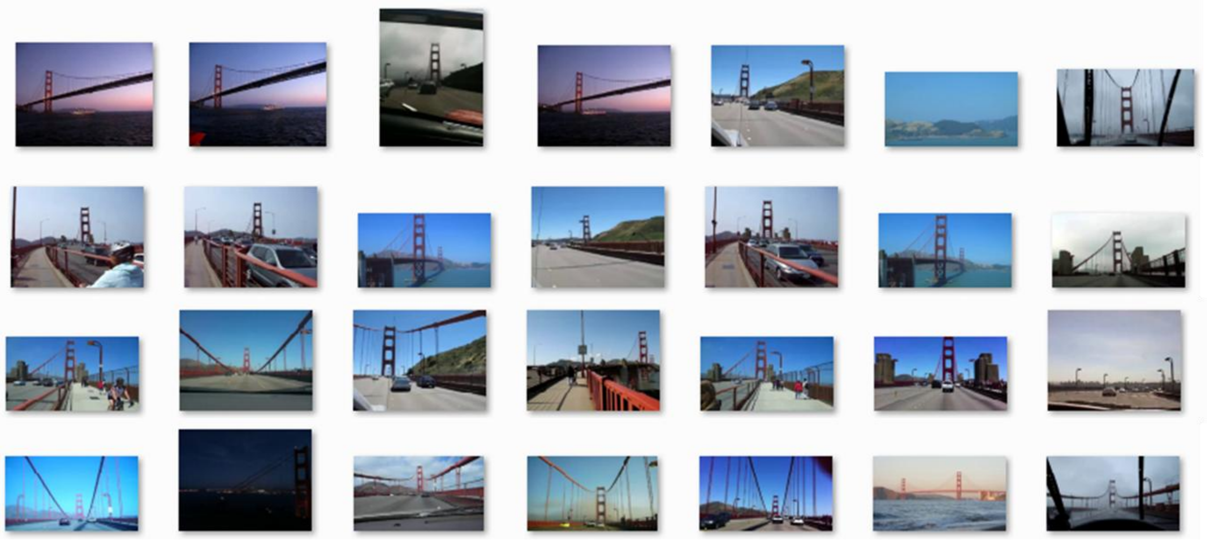
Query expansion method	Number of Flickr images	Using Trecvid2012 queries	MAP
Centered representative query	25	No	0.0430
		Yes	0.0510
	50	No	0.0684
		Yes	**0.0768**
	100	No	0.0479
		Yes	0.0522
Distributed representative query	25	No	0.0449
		Yes	0.0660
	50	No	0.0712
		Yes	0.0756
	100	No	0.0552
		Yes	0.0601

We have seen in Section 4.5 that grouping the frames of the same video shot provides a significant computational boost by reducing the number of BoW descriptors. In a similar manner, we have grouped the representative images in a single BoW representation. The advantage is that in this case we issue solely a single query. The results obtained are presented in Table 6-16. We can notice an improvement in the retrieval MAP scores for all the considered runs. Furthermore, the positive influence of the original queries is increased in this case. In Figure 6.22, we illustrate some visual examples of the results obtained for two distributed iconic query topics.

We compare the query expansion runs with a classical BoW run, relying on the original images and their masks. Using the regions under the masks we construct a topic BoW descriptor for multiple queries and use it for search in the database. The results are presented in Figure 6.23.



a)



b)

Figure 6.22. Results for different iconic query topics: a) “Pantheon interior” – 9056; b) “Golden Gate Bridge” - 9050. Each image corresponds to the first frame extracted from the retrieved shot.

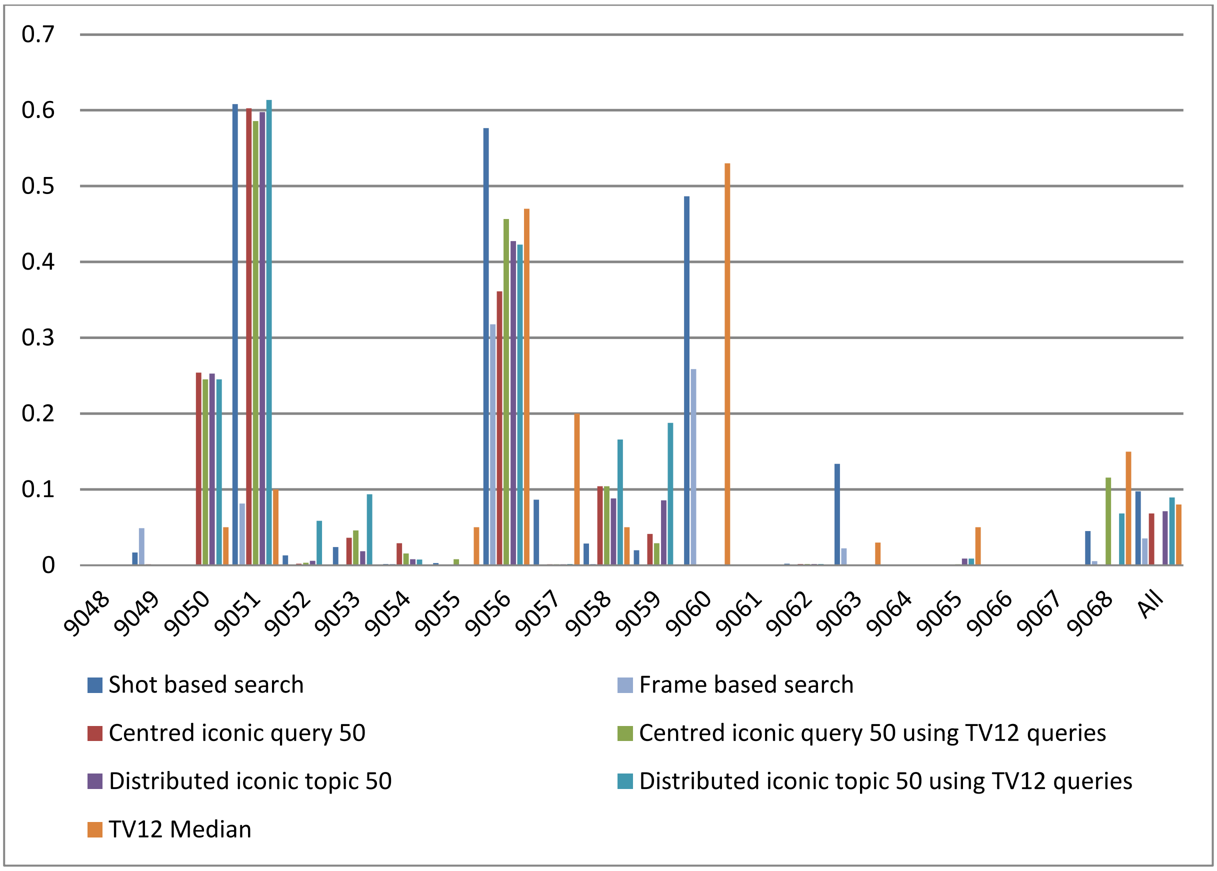


Figure 6.23. Overview of query expansion results on Flickr dataset.

We notice that the query expansion results are similar with the ones leveraging on the original queries and their binary masks. In many cases, even the expanded queries that do not use the original queries provide very good results, outperforming the median MAP value for the TRECVID2012 runs (which is of 0.08). The results are encouraging as they show that simple textual queries make it possible to determine relevant images that can serve then to retrieve relevant video content.

Table 6-16. Experimental results on query expansion with queries grouped in topics.

Query expansion method	Number of Flickr images	Using Trecvid2012 queries	MAP
Centered representative topic	50	No	0.0702
		Yes	0.0819
Distributed representative topic	50	No	0.0712
		Yes	**0.0893**

Finally, we test our method in a more standard query expansion manner [Chum 2007] and use the representative image descriptors to enrich the original query descriptors. The new queries and their corresponding descriptors are then searched within the dataset. Similarly with our previous experiments, we test the influence of adding the query images in the construction of the query graph. The results are summarized in Figure 6.24. We notice that for all runs the improvement of the results with up to 0.03 in MAP. The distributed iconic topic with the original query images in the query graph featured the best performance of 0.1295 MAP. Let us note that overall the MAP scores are similar (less than 0.01 difference) among all considered runs.

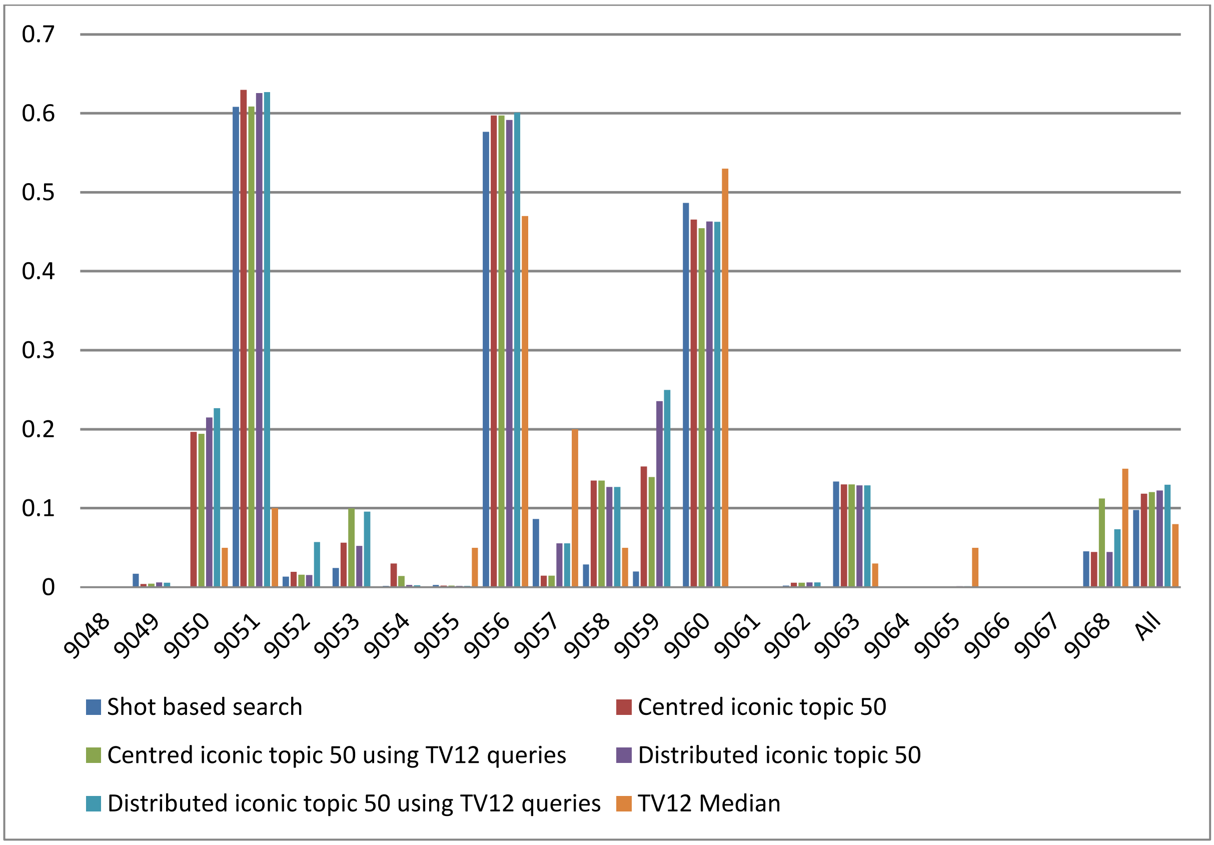


Figure 6.24. Overview of standard query expansion results on Flickr dataset.

Soft assignment influence

In the following; we have tested the influence of the soft assignment of the query descriptors on the retrieval results. We have followed the procedure proposed in Section 4.4.2 and considered for quantization the first five visual words correspondences for each query point descriptor. The results are illustrated in Figure 6.25. Overall the MAP is lower for the original queries (- 0.015) and better for the query expansion runs (+ 0.007 for the centered queries and unchanged for the distributed queries). The centered queries benefit from the soft assignments since they have interest points from only one image and they are less diverse. The mapping to multiple visual words can be regarded in this case as an additional query expansion with multiple visual words assigned to the query image. In addition, since the query expansion images are obtained from a different dataset than the one used for the construction of the vocabulary, descriptors can be miss-quantized often and in such cases the soft-assignments become handy.

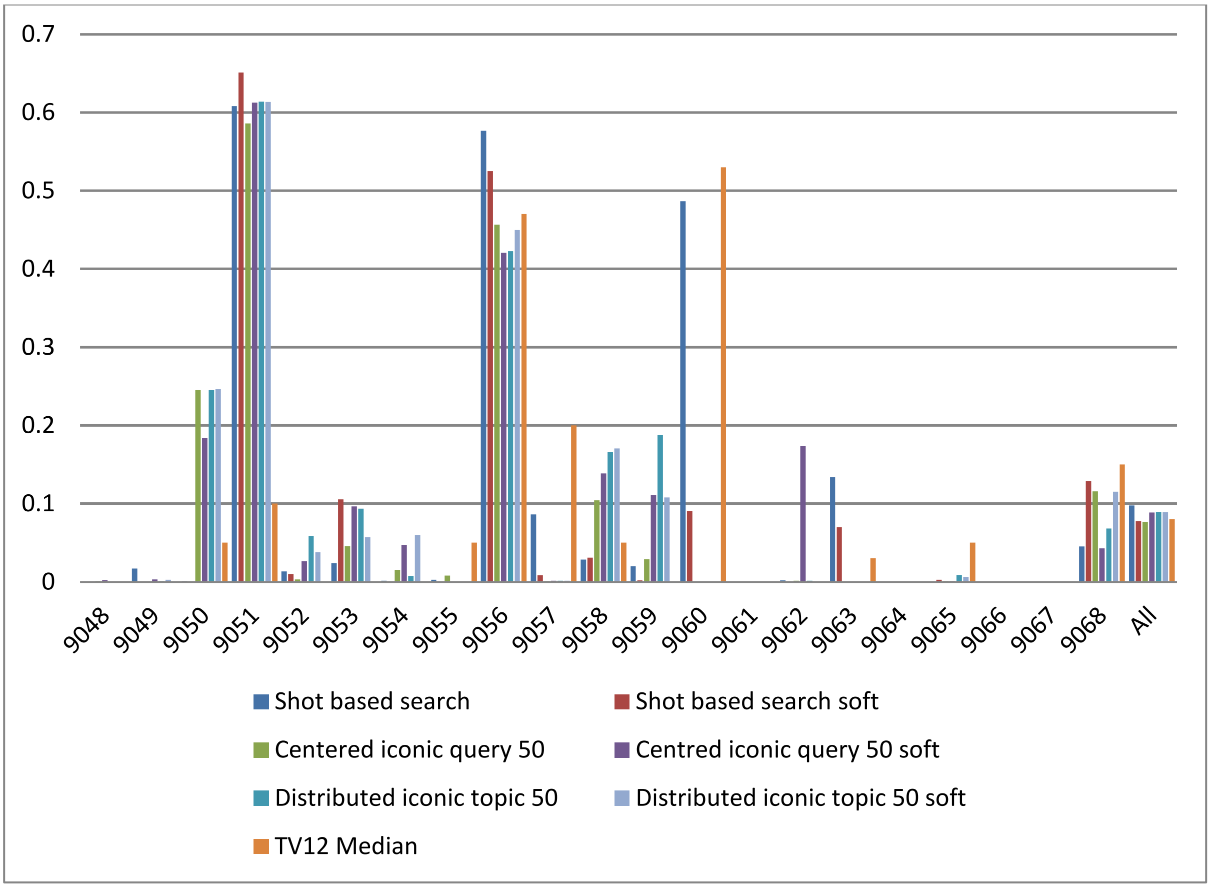


Figure 6.25. Results for hard and soft assignment based runs on Flickr dataset.

Query expansion generalization

Finally we have tested the generalization of our query expansion method to other popular image search engines on the web, such as Google Images [Google 2012c] and Bing Images [Bing 2012]. Such search engines are very popular for image content search retrieve and leverage on powerful image retrieval algorithms, user annotations and relevance feedback. The first results are in most of the cases relevant and usually contain diverse content (*e.g.*, different poses of the object of interest, different styles – photo, painting, line drawing …). In the same time this can be a drawback as representative instances of an object are more difficult to identify. In order to overcome this drawback and to collect a more relevant initial set of image to build the query graph, we have selected the top 10, 15 and 20 results from the same textual query on Bing, Flickr and Google, leading to sets of 30, 45 and 60 images. In this manner, the most representative instances of the object of interest can be identified among the results from the different search engines and aggregated in a mixed expanded query descriptor. Let us note, that some images retrieved by Bing or Google have expired links or cannot be accessed for download. We discard such images from our set and select the next images available for download from the list of results.

In this set of experiments, for each topic we have computed a BoW descriptor over all the respresentative images of the topic and its corresponding descriptors. Let us note that original query images have not been included in the construction of the query graph. The results for the runs using 45 images are summarized in Figure 6.26.

We can notice that the results are slightly lower than the ones of the query expansion generated only from the Flickr images (approx. 0.07 MAP for our expansion and 0.10 MAP for the standard expansion). This lack of improvement in the results is due to the increased variety of images retrieved by the web search engines. This reduces the number of representative images and matched points leading to a weaker expanded query. The advantage of this approach is the increased speed, as the download of the images is parallelized on the three engines and the number of images to download from each engine is reduced (*i.e.*, 15).

6.3.3.3 Discussion

In this section we have evaluated the performances of our BoW setting on the Flickr dataset. We have seen that videos of relatively reduced sized allow the retrieval of different object instances even in large scale datasets. In addition the effectiveness of a shot based descriptor for video search coupled with a query descriptor obtained from multiple query descriptors; has been proven. This setting has provided results superior to the classical frame-based BoW video object retrieval.

Concerning our query expansion algorithm, we have seen that good performances can be obtained even by using only textual queries to generate a visual query to search. The main drawback relates to the formulation of the textual query in order to retrieve plausible images containing the object of interest. In our examples we have considered the textual descriptions provided in the evaluation campaign and this affected a certain queries. For example, the Pantheon topic 9056 had the textual descriptor “Pantheon interior”. The Flickr search returned images of the Pantheon from Rome, but also many images of the Pantheon from Paris; affecting negatively the results. A more specific textual query such as “Pantheon Rome interior” would have provided a strong base for a visual query about the Pantheon.

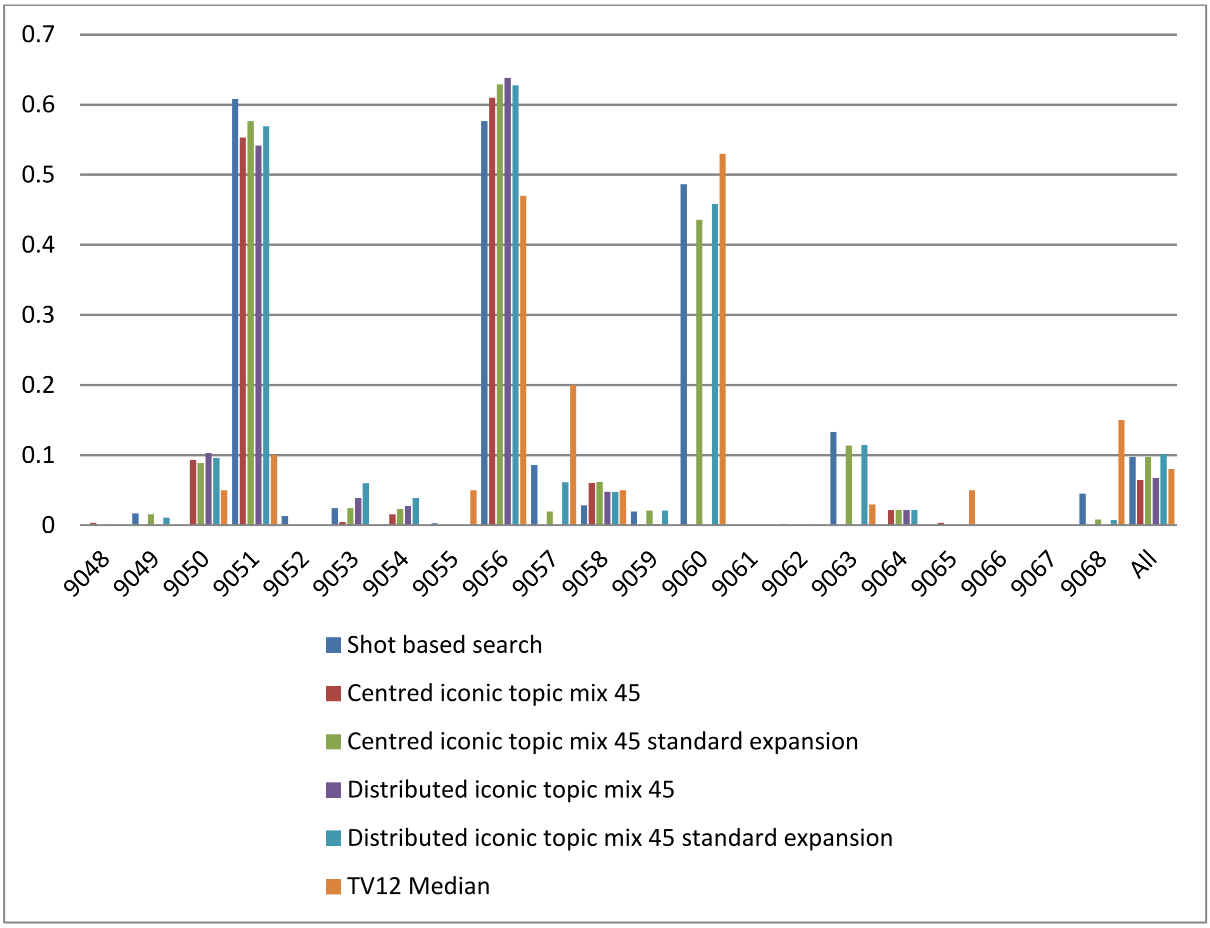


Figure 6.26. Results for query expansion using images from multiple search engines.

The perspectives of improvement for the query expansion algorithm concern the use of fewer images for building the query descriptors. For 50 Flickr images, the download takes 60 -120 seconds and the generation of the query graph takes approximately 120 seconds. A possible extension could to consider fewer images (*e.g.*, 15-20) and to identify multiple pairs of matched images and to feed them to an ad-hoc SVM classifier [Boser 1992, Cortes 1995], similarly with [Arandjelovic 2012], which would then rank the frames of the dataset according to the learned weights.

Another issue to address is the pertinence of the images retrieved by the search engines. Since most of them use textual annotations given by users, many images could be mislabeled and thus not relevant for our query. A possible solution would be to launch the textual query over multiple image search engines (*e.g.*, Flickr, Google Images, Bing Images) and to retrieve only the top results which are usually positive. However, these engines usually retrieve very diverse content in the top results, making it difficult to retrieve representative images. In such cases, the condition of representative image (*i.e.*, at least two matched images) can be relaxed and consider pairs of matched images. Yet in this manner we would rely very much on the pertninence of ther results retrieved by the search engines. Other mechanisms for validating pairs of images and points need to be addressed in future work.

PART II. OVIDIUS: A Web-Based Video Indexing and Retrieval Platform

7. State of the art on video browsing and retrieval platforms

Abstract: *This chapter proposes a comprehensive review of the state of the art in video browsing and retrieval systems and platforms. A multitude of video systems for both personal and professional use have been elaborated recently. Various parameters and factors need to be considered when designing a video browsing and retrieval system, including the richness of the video data; the devices that use the system, the design and ergonomics of the user interface, the navigation and search functionalities. We study these parameters across multiple systems and platforms that have been proposed in the last years and summarize their main functionalities. Both desktop-dedicated systems and mobile platforms are here described. Four families of approaches are identified, according to targeted functionality: Known-Item Search systems, video discovery and consumption systems, video archive and content organization systems and research prototypes for video navigation and discovery. Finally, a set of criteria that need to be taken into account when elaborating video browsing and retrieval platforms is proposed.*

Keywords: *video databases, multimedia indexing platforms, web systems, video navigation and browsing, search and retrieval.*

Résumé: *Ce chapitre propose une revue complète de l'état de l'art dans les systèmes et plates-formes de navigation et recherche de contenu vidéo. Une multitude de systèmes vidéo pour utilisation personnelle et professionnelle ont été élaboré récemment. Des différents paramètres et facteurs doivent être considérés lors de la conception d'un système de navigation et recherche de vidéo, notamment la richesse des données vidéo, les dispositifs qui sont utilisés pour accéder au système, le design et l'ergonomie de l'interface utilisateur, les fonctionnalités de navigation et de recherche. Nous étudions ces paramètres sur plusieurs systèmes et plates-formes qui ont été proposées dans les dernières années et nous résumons leurs principales fonctionnalités. Systèmes dédiés aux ordinateurs de bureau et plates-formes mobiles sont décrites ici. Quatre familles d'approches sont identifiées, selon la fonctionnalité ciblée: des systèmes de recherche d'éléments connus, des systèmes pour la découverte et la consommation des vidéos, des systèmes pour des archives vidéo et pour l'organisation du contenu et de prototypes de recherche pour navigation et découverte de vidéo. Un ensemble de critères qui doivent être pris en compte lors de l'élaboration des plates-formes pour la navigation et la recherche des vidéos est proposé.*

Mots clés: *bases de donnnés vidéo, plates-formes d'indexation multimédia, sysèmes web, navigation de vidéos, recherche et récupération.*

Multimedia search engines, and notably image and video search engines, represent the fruit of joint efforts and advancements in many different research areas such as temporal decomposition and chaptering, audio-visual feature extraction and description, machine learning, as well as visualization, navigation, interaction and user interface ergonomics and design. Existing popular video search engines (*e.g.*, YouTube [YouTube 2012], VideoSurf [VideoSurf 2011], Blinkx [Blinkx 2012], Vimeo [Vimeo 2012], DailyMotion [DailyMotion 2012]) are based on lexicons of semantic concepts and perform keyword-based queries. Such systems usually involve simple web interfaces that present the results of a query as a ranked list of preview icons. Additionally, such systems offer little interaction and flexibility, as they do not allow the users to perform complex queries based on the visual content and as search options are limited to the tags and keywords assigned by the users.

Several surveys in the field have been recently conducted in order to gather a set of necessary user requirements that a video search engine should fulfill [VidiVideo 2012, IM3I 2012]. Such surveys collected opinions and feedback from both scientists and video archive professionals. The analysis first showed that there is a strong need of a web-based indexing system, remotely accessible from both fixed and mobile devices over Internet. This would be necessary for increasing the accessibility and interoperability of the archives in both intra and inter-organization scenarios. A second strong requirement revealed by the user requirement analysis concerns the possibility to formulate complex, composite queries and to expand them with the help of ontology reasoning mechanisms [Wang 2004].

With the explosion of available multimedia content and content-producing devices, consumers gather impressive amounts of images and videos in their personal collections. Functionalities that were until now demanded mostly by professional producers, are becoming equally necessary for consumers, as there are very few available tools for organizing their archives in a structured manner. As a consequence, the multimedia search engines should be adapted to the needs of the general public user, offer ergonomic and user-friendly interaction mechanisms.

In this chapter, we propose a critical review of the existing technologies and approaches.

Our work draws upon research in several areas concerning multimedia content management: content-based image and video retrieval, multimedia content management, user feedback management and distributed content sharing. With the ever increasing amount of available multimedia content, researchers have been undergoing tremendous work on developing navigation and retrieval systems and identifying different use cases to leverage on this data.

Systems and interfaces suitable for video content navigation have been studied since the early years of video retrieval [Manske 1998]. The last years have seen significant advancements of the web technologies, correlated with the emergence of the Web 2.0, which pointed a milestone in the field multimedia content distribution and consumption. Since 2006, we have evolved from pioneer web image search systems such as PARAgrab [Joshi 2006]; exploiting visual features and textual meta-data for search, to general public applications for content-based multimedia retrieval such as Google Goggles [Google 2012c].

The various systems proposed in the literature are dedicated to either PC/desktop systems or to mobile terminals, since the associated capabilities in terms of storage/bandwidth and memory capacity represent strong constraints that have to be appropriately taken into account when elaborating a video search engine. Within this framework, let us first analyze the desktop-dedicated systems.

7.1 Desktop-dedicated systems

In this case, we identify three main families of existing navigation and retrieval systems that are categorized according to their main functionality:

1. "Known-Item-Search" systems,
2. Video discovery and consumption,
3. Video archive organization and annotation,
4. Experimental systems for new ways of navigation and discovery of the video content.

Each family of approaches, together with representative methods and systems, is described in detail in the following sections.

7.1.1 Known-Item Search systems

A first category of systems is represented by the so-called "Known-Item Search" (KIS) paradigm. The KIS task models the situation in which a user have knowledge of a video already seen, believes it is contained in a collection, but does not know where to look. To begin the search process, the user formulates a text-only description, expressed with the help of a few related keywords. Alternatively, whenever the user retrieves an element similar with the content of the target video, the user can perform a query-by-example in order to retrieve other instances of the content/objects from the current shot. Such systems usually target professional users offering a variety of tools for quick navigation of a database and search. Competitions and benchmarking campaigns such as TRECVID [Smeaton 2006], VideOlympics [Snoek 2008] or Video Browser Showdown [Schoeffmann 2012b] offer the occasion of testing such prototype platforms in a competition-like environment, under strong time constraints. Although such platforms currently address highly advanced use-cases, they provide an insight to the future visual search and navigation technologies.

First, let us mention the hierarchical video retrieval system developed in the Fudan University [Sun 2008], which uses an adaptive multi-modal fusion method and enables the user to browse the results hierarchically across different levels of temporal granularity. The main browsing space is mapped onto a sphere and offers 3 degrees of freedom: the X axis, representing the temporal degree of freedom, the Y axis, where results are ranked by the degrees of confidence; and the Z axis which provides zooming in/out mechanisms for browsing across different levels of search granularity. The interface and its navigation features have proven to be more effective in an interactive TV implementation using a remote control.

In [Vrochidis 2008], the MKLab interactive retrieval system brings back in the spotlight the traditional MPEG-7 descriptors. The system takes advantage of the MPEG-7 visual descriptors capturing different aspects of human perception such as colour and texture in order to provide a content-based similarity search. This search system is built in a web environment (php, JavaScript and a mySQL database) providing a graphical user interface for performing retrieval tasks over the internet. Retrieval results are presented ordered by rank in descending order with links to the temporally neighbouring shots of each one. The interface mimics the functionality of the shopping cart encountered in electronic commerce sites.

The LIGVID System for Video Retrieval [Ayache 2010] features two types of interfaces with different degrees of complexities, corresponding to expert and novice users. This web-based system displays the video shots on a classical 2D grid view, with an adjustable number of lines and columns. The search mechanism considers a temporal similarity, which introduces in the result list positive shots in the neighbourhood of the positive samples. The content retrieval process is enhanced by considering a large collection of concept names to use for queries.

A different approach of video search engine with an emphasis on the interactivity aspects of the application is proposed in [Zhe 2009]. Here, the VisionGo platform brings a very fast navigation system along with an increased interactivity, while taking full advantage of the users' *a priori* knowledge. A user can simply draw a mouse stroke across the object to obtain its bounding box and the subsequent visual search process is performed based on this object region.

Other approaches put the stress on the video browsing aspects and design the user interface accordingly, aiming to provide to the user a quick overview of a video or of a collection of videos. The VideoMap application [Cao 2009] displays the similarity relationships among the whole video collection using a map-based interface.

The AAU video browser [Fabro 2012] features a parallel and a tree-like browsing interface for navigating in a hierarchical and non-sequential manner through the content of single videos or within small video collection. No video structuring or segmentation is here required, since authors adopt a simple strategy: each video is uniformly divided into *n* segments of equal length. The user can navigate hierarchically in the video, each time dividing the current video segment in other *n* subparts.

The MediaMill group brings a series of innovative contributions to the field. In [Rooij 2007] the concept of "thread" for navigation through various video shots is introduced. A thread is defined as a linked sequence of camera shots represented as iconic images from various videos in some specified order. Threads are the basis for navigation in this system. They have been given different utilities, according to the type of navigation that they are offering: query result thread, visual thread (based on visual similarity), semantic thread (based on semantic similarity), time thread (based on the time-line), textual thread (based on similarity of textual annotations). Several thread-based browsers have been proposed, each of them aiming to cover different use-cases: the RotorBrowser, the CrossBrowser [Snoek 2007] and the combination of the previous two, so-called the ForkBrowser [Rooij 2009].

The RotorBrowser starts from an initial query result and the user can select a so-called focal shot *S* that is displayed in the center of the screen. The RotorBrowser provides then several navigation paths according to that focal shot *S* by displaying all the video threads that contain *S* in a star formation (Figure 7.1). The CrossBrowser is a lightweight version of the RotorBrowser suitable for non-professional users. The simplified CrossBrowser interface solely provides horizontal and vertical navigation. The time thread is visualized in the horizontal direction while the visually similar shots of *S* are displayed in the vertical one (Figure 7.1).

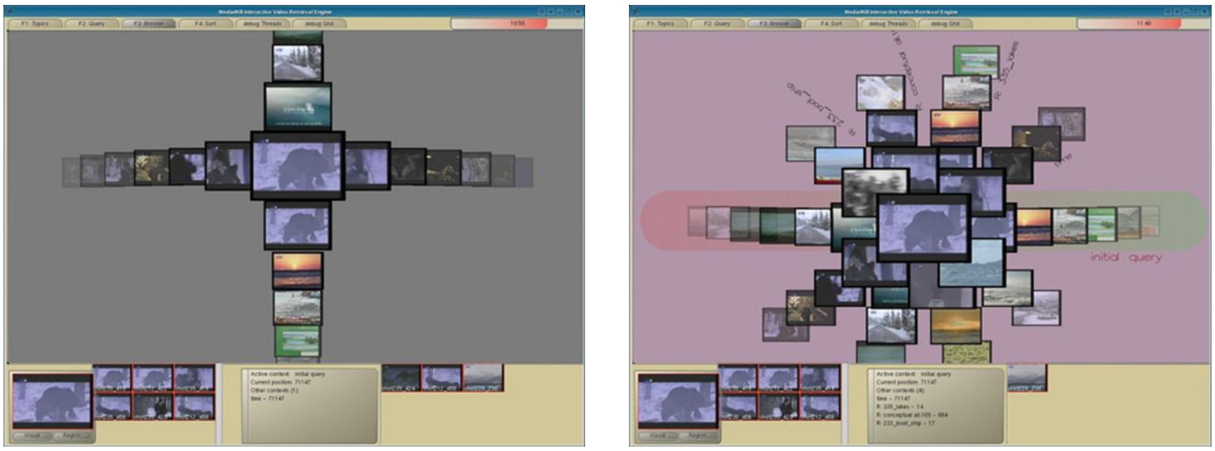
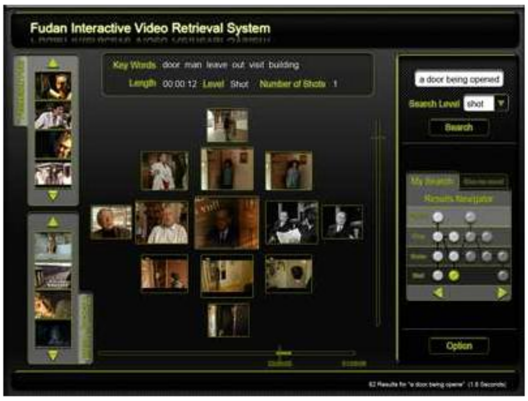
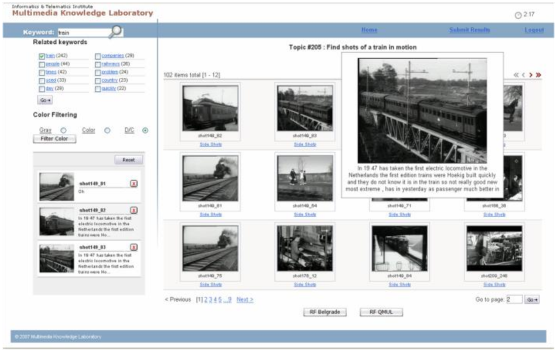
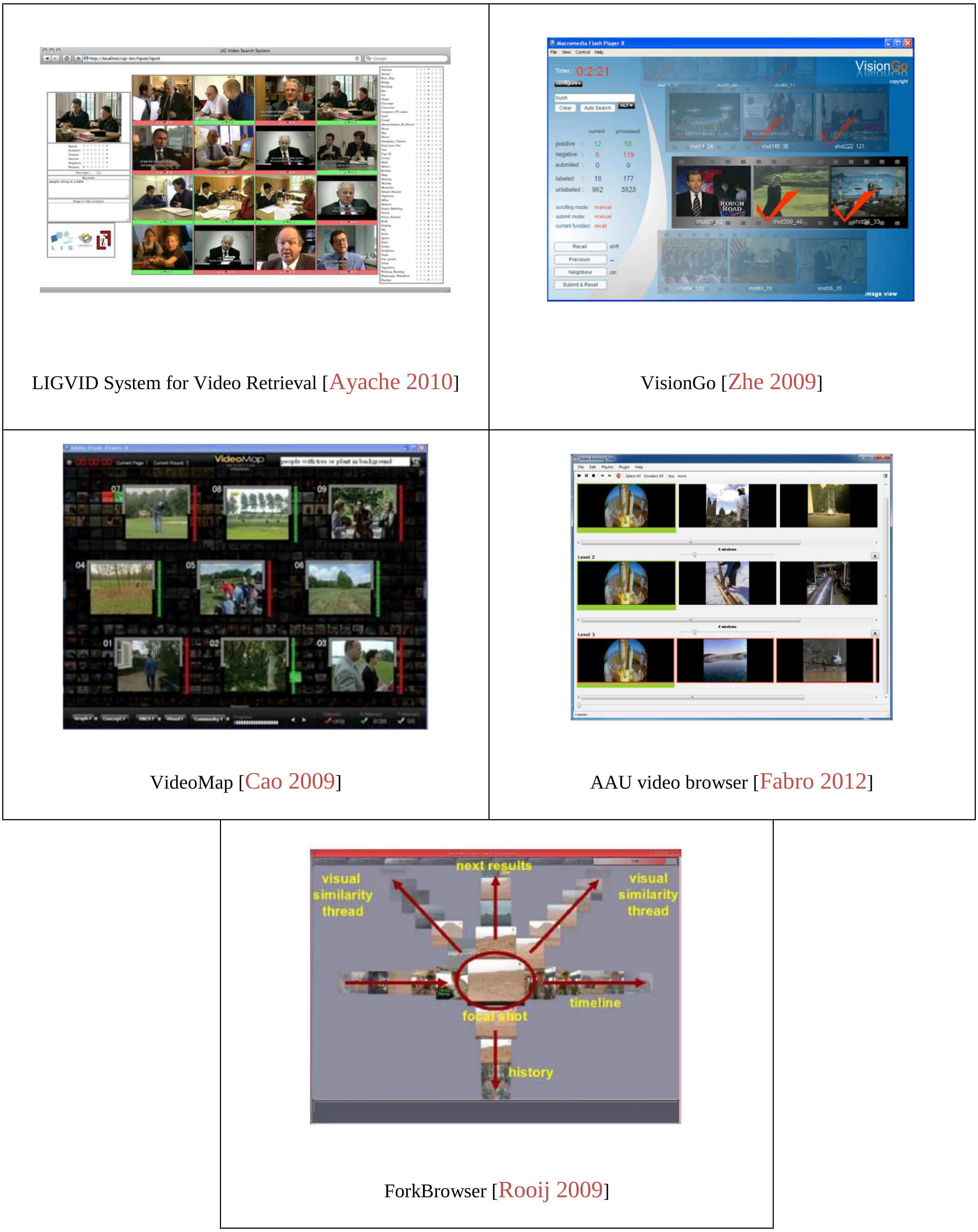


Figure 7.1. Screen captures of the CrossBrowser (left) and RotorBrowser (right) interfaces for multi thread visualization and navigation [Snoek 2007]

Such tools have been further extended in the ForkBrowser system, similar with the RotorBrowser, but displaying fewer threads related the timeline, visual similarity and navigation history. All systems have achieved top results during the TRECVID 2006-2008 evaluation campaigns [Smeaton 2006].

Figure 7.2 illustrates the user interfaces of the mentioned *Known Item Search* video navigation and retrieval systems and Table 7-1 presents a synthetic overview on their functionalities.

 Fudan University system [Sun 2008],	 MKLab system [Vrochidis 2008]



LIGVID System for Video Retrieval [Ayache 2010]

VisionGo [Zhe 2009]

VideoMap [Cao 2009]

AAU video browser [Fabro 2012]

ForkBrowser [Rooij 2009]

Figure 7.2. Overview of the interfaces of Known Item Search systems.

Table 7-1. Overview of the features of Known Item Search systems.

System	Web access	Navigation	Querying	Content description	Main features
Fudan University - Hierarchical Video Retrieval [Sun 2008]	No	Yes	Yes	- video decomposed into shots - Gabor, Gray Level Co-occurrence Matrix, MPEG-7 descriptors (Edge Histogram, Color Layout, Scalable Color) trained using SVM - textual description from audio transcript	- hierarchical video browsing scheme - visual similarity search, transcription based search
MKLab system [Vrochidis 2008]	Yes	Yes	Yes	- videos decomposed into shots - MPEG-7 visual descriptors (color, texture)	- interactive video retrieval using MPEG-7 descriptors - 2D grid web interface mimics functionality of shopping cart encountered in e-commerce sites
LIG Multi-Criteria System [Ayache 2010]	Yes	Yes	Yes	- videos decomposed into shots - textual description from audio transcript, phonetic string, similarity to example images, semantic categories, and relevance feedback strategies	- video retrieval system embedding multiple descriptors and search methods - two types of interfaces with different degrees of complexities - 2D grid view of video shots
VisionGo [Zhe 2009]	No	Yes	Yes	- videos decomposed into shots - supervised learning of concepts using BoW and SVM - learning from users' relevance feedback	- fast navigation system leveraging user feedback for search -keystroke based navigation -visualization of groups of 3 shots
Fork Browser [Rooij 2009]	No	Yes	Yes	- videos decomposed into shots - supervised learning of concepts using BoW and SVM - visual similarity using BoW	- multiple thread-based video navigation - number of threads adjustable to the type of user -fast keystroke based navigation
The AAU video browser [Fabro 2012]	No	Yes	No	- videos decomposed in slices of equal lengths	- visualization of multiple keyframes and hierarchical navigation among equal length video slices

7.1.2 Video discovery and consumption systems

This family of approaches concerns video platforms dedicated to the general public. The ergonomics of the associated user interfaces is usually subject of thorough studies of user needs, behavior and feedback. This typically results in user-friendly interfaces and interaction mechanisms. The queries are most often performed with the help of textual tags. In addition, the associated search engine is able to integrate different features such as textual query, popularity, novelty of the content, and user viewing history, in order to rapidly provide relevant content.

In what concerns the video visualization and representation, the documents are usually represented by a thumbnail extracted randomly from the video along with the title given by the author. Eventually, existing tags, such as those manually inserted by the user can also be displayed. Such platforms accurately reflect the current needs of the users, their knowledge and preferences in terms of navigation, search and video content.

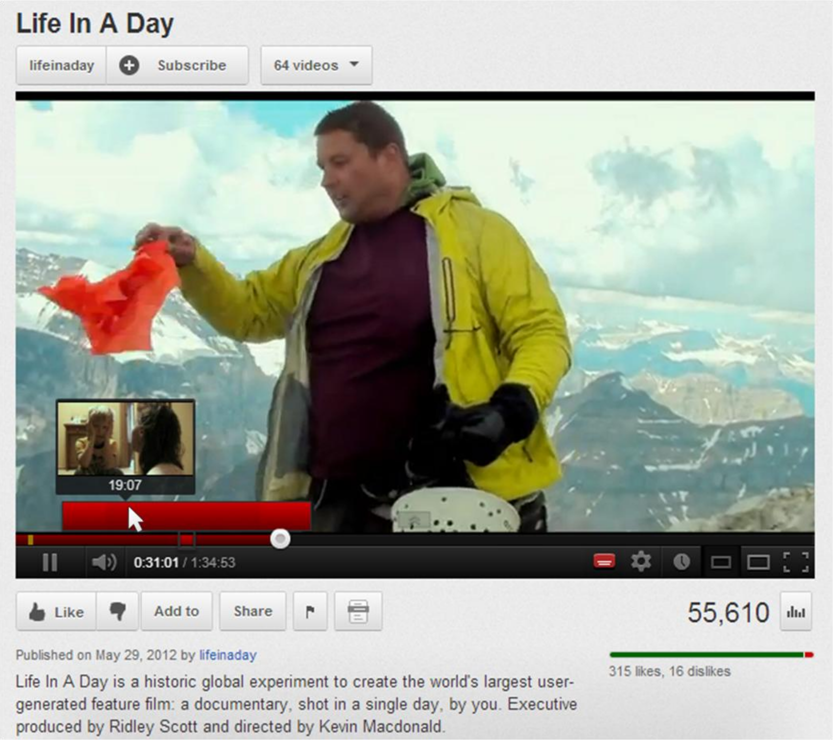
Among existing video consumption platforms, let us first mention the famous YouTube platform [YouTube 2012], which became the second most popular search engine on the web (after Google). YouTube features today the largest repository of videos on the web and the highest growth rate with 72 hours of video uploaded every minute [YouTube 2012c]. Youtube's search functionalities are mostly based on textual tagging combined with complex user behavior and feedback analysis.

In its first versions, the videos are treated as global, monolithic documents and their corresponding spatial-temporal structure is not all taken into account. However, very recently (in 2012), the platform integrated an innovative intra-video navigation functionality which has been proposed to the general public [YouTube 2012b]. In this case, a simple mouse roll-over on the video progress bar instantly displays the thumbnail of the key-frame corresponding to the selected timestamp. Moreover, a secondary progress bar makes it possible to zoom-in on the temporal neighborhood of the selected time stamp. (Figure 7.3). This shows the interest of getting access, within a given video, to finer elements of information corresponding to specific actions and events. However, for the time being, the YouTube solution only allows user-guided, manual browsing and does not provide any tool for intra-video search and retrieval.

Similarly to YouTube, VideoSurf [VideoSurf 2011], another main stream video platform, allows users to navigate through a video using a thread of uniformly sampled frames from the video. The same display style is used for the search function giving the possibility to visually navigate through the search results in order to find specific scenes, people or events and decide whether the videos contain shots of interest or not.

Let us mention that there are numerous other video search engines such as DailyMotion [DailyMotion 2012], Truveo [Truveo 2012], Blinkx [Blinkx 2012], Trooker [Trooker 2012] that offer similar services.

Finally, let us mention the Voxalead News [Law-To 2009] platform. Here, the user can search for news videos by specifying free text keywords or by selecting from a list of pre-defined keywords, corresponding to a basic ontology (including elements such as location, name of person, institution…). The application then displays a set of preview thumbnails for the news video, their date, a short description and their source. The user has also the possibility to navigate inside the video at the precise moment where a certain keyword is mentioned. This feature is achieved with the help of a s*peech-to-text*, technology which interprets the audio track in order to provide rich textual annotations that are synchronized with the video content.



Life In A Day

Published on May 29, 2012 by lifeinaday

Life In A Day is a historic global experiment to create the world's largest user-generated feature film: a documentary, shot in a single day, by you. Executive produced by Ridley Scott and directed by Kevin Macdonald.

Figure 7.3. Advanced fine video preview and navigation using the video player time bar on YouTube [YouTube 2012]. A sliding window on the time bar grants finer access to the video segment underneath it. The second time bar is displayed above the main bar and mouse-over moves display key-frames from selected video time stamps.

7.1.3 Video archive organization and annotation

Recent years have been marked by massive campaigns of digitization of video and audio archives around the world. Some of the national archives with the richest multimedia archives in Europe, such as BBC (*British Broadcast Corporation*) [BBC 2012], INA (*Institut National d l'Audiovisuel*) [INA 2012] or Beeld en Geluid (*Netherlands Institute for Sound and Vision*) [SoundVision 2012], which are disposing of enormous amounts of non-digitized multimedia content, have been undergoing campaigns for digitization and re-usage of their video archives. The digitization process includes a transcription of the audio channel, providing a good basis for content retrieval by keywords. Yet, retrieval of video content using solely textual queries in such vast collections of content can get very difficult due to textual ambiguities, repetitive words or audio transcription errors. Dedicated tools for visualization, annotation and search of such archives have been developed in order to complement the audio transcriptions.

Within this context, let us first mention the approach introduced in Bailer *et al.* [Bailer 2012]. Authors propose a web-based video browsing tool displaying video key-frames on a light table and illustrating clusters of similar key-frames with the help colored highlighters. Several features are considered for measuring the similarity between key-frames, including camera motion parameters, visual activity descriptors, global color descriptions and object trajectories. They are represented with the help of the MPEG-7 standard [MPEG 2002, MPEG 2003] and stored in a SQLite database [SQLite 2012].

Other post-production platforms such as *Lignes de temps* [IRI 2009] address the annotation aspects in more detail, introducing tools for manual annotation of particular video segments. The user has the possibility to browse the videos, select/display video segments of interest, and to perform collaborative annotation.

MediaTable [Rooij 2010] is a complex system designed for large multimedia collections and featuring three main functionalities. Notably, MediaTable makes it possible to provide an insight into a multimedia collection, to categorize the collection into several clusters, and to perform efficient video searches. Users can easily explore a multimedia collection through a table-based interface (Figure 7.4). In addition, a grid and a point cloud viewer can be employed for the exploration of the archive. Popular semantic concept detection algorithms are here employed for video search purposes. Content of interest can be stored temporarily in so-called “bucket-lists” that allow the user to view, tag, and mark video items as *relevant/non-relevant*.

Bertini *et al.* [Bertini 2011] introduced MediaPick, a web-based system featuring an interface adapted for large surface multi-touch devices. Users have the possibility to explore ontology graphs and perform queries-by-concept in a large video repository starting from elements of these graphs. In addition, they can visualize, organize and annotate the videos in a personalized manner. Let us also note that the authors propose a PC counterpart, more suitable for multi-user usage. The search engine exploits a bag-of-words implementation based on interest point descriptors for automatic semantic annotation, and MPEG-7 visual and audio descriptors for the queries-by-example.

Recently, TexMix [Bréhinier 2012] has tackled the issue of search and retrieval of large collection of broadcast news videos. They exploit speech transcription, natural language processing and image retrieval techniques. The HMTL5 implementation increases the interactivity of the interface allowing the user to navigate through keywords search or through time maps. Several hyperlinks are automatically created either between various reports related to a given subject or to the Web, in order to find out additional information about a given story, person or fact. Two TexMix features are of particular interest. The first one concerns the generation of *a dynamic summary*, based on keyword extraction, which gives the possibility to catch up news quickly on a specific topic. The second one is the geo-tagging capability, which makes it possible to look up for news related to specific locations.

A synthesis of the above-cited video archiving approaches is presented in Figure 7.5 and in Table 7-2.

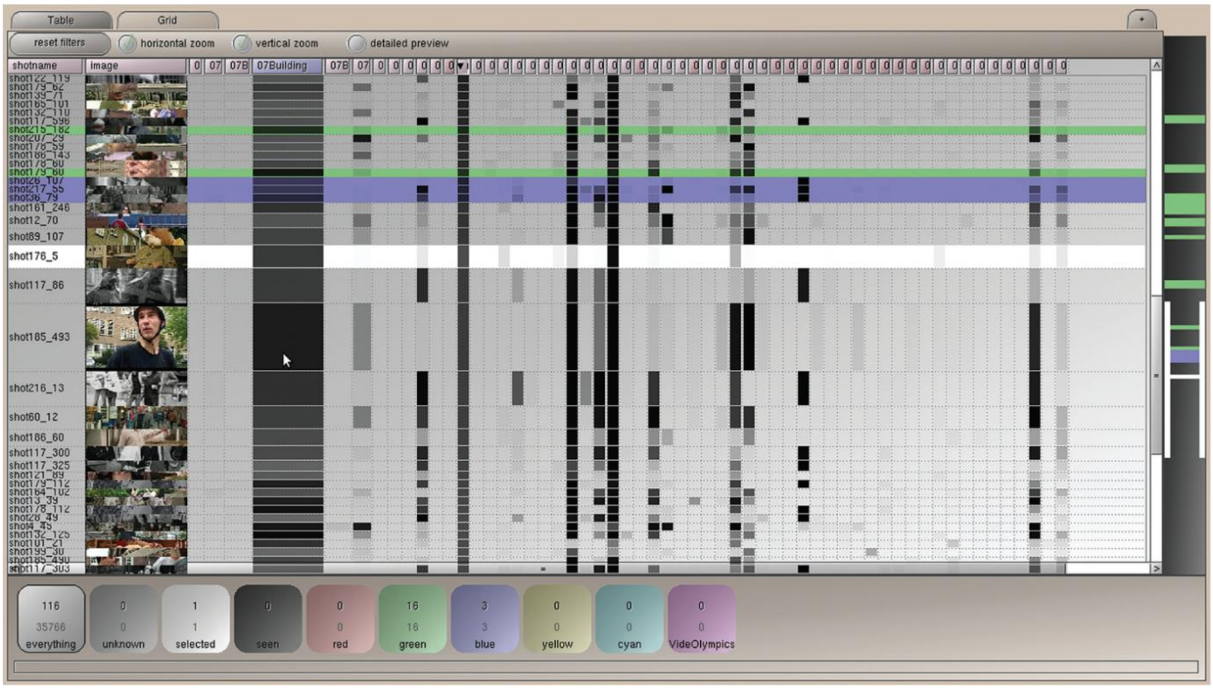
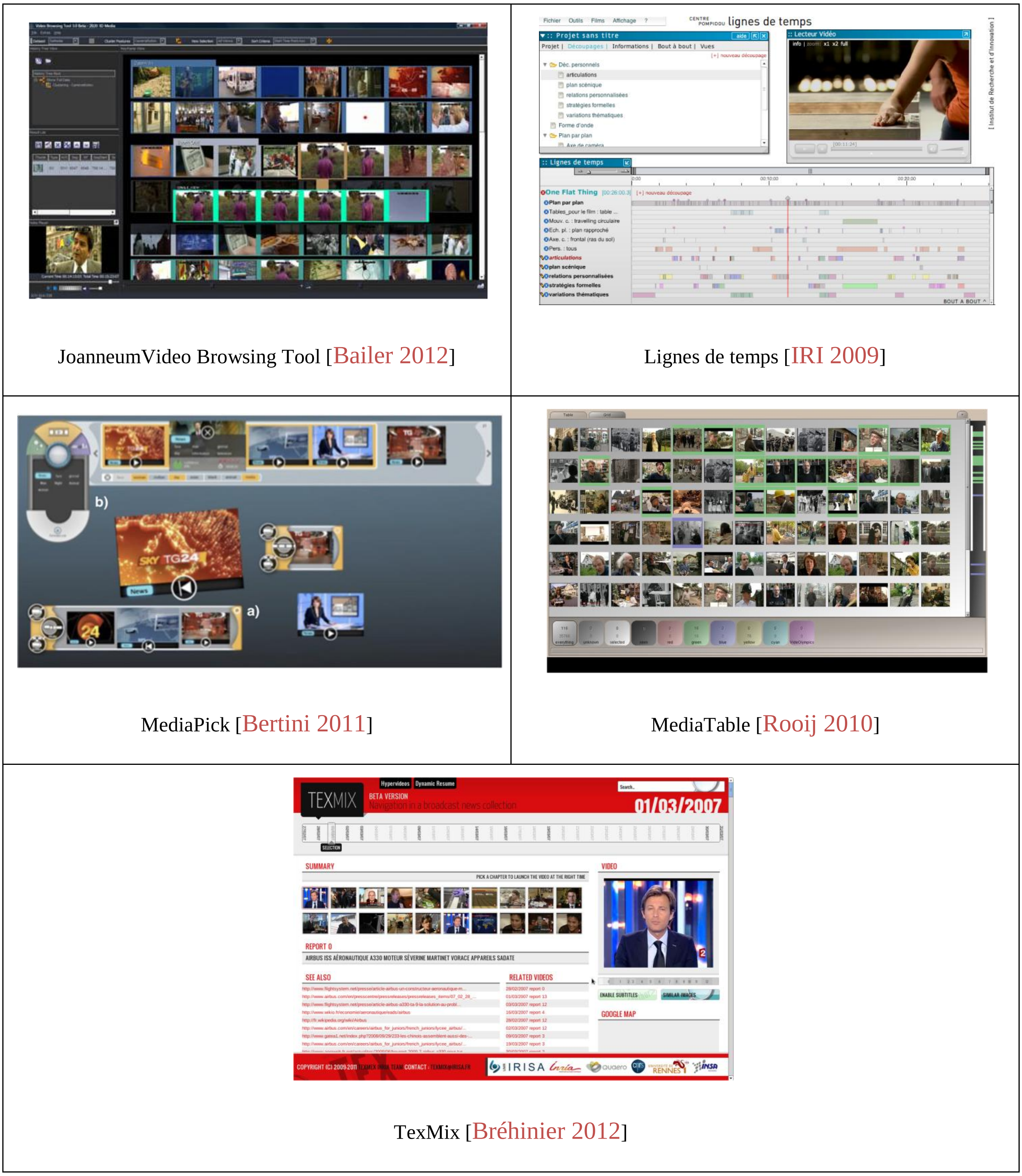


Figure 7.4. The MediaTable interface [Rooij 2010]. The table interface takes up most of the figure, with the overview pane to the far right and the bucket list on the bottom. The series of filled cells indicate each detected concept's presence or absence in each video fragment. The colors in the table and in the overview indicate in which buckets each fragment is currently categorized.

Let us note that, so far, most of the existing systems have been focusing on professional, expert users, with a specific goal of organization and reuse of video content. Yet, as the multimedia archives are gradually becoming available to the general public, as in the case of the vast European television heritage [EUscreen 2012], the next years will require significant advancements in the field. Thus, advanced tools for navigation and retrieval of such multi-lingual content should become affordable to the general public.

Identifying the important socio-economic challenges associated with such evolutions, several research prototypes have been recently proposed in the literature. They are described in the following section.



JoanneumVideo Browsing Tool [Bailer 2012]

Lignes de temps [IRI 2009]

MediaPick [Bertini 2011]

MediaTable [Rooij 2010]

TexMix [Bréhinier 2012]

Figure 7.5. Overview of the interfaces of systems for video archive organization and annotation.

Table 7-2. Overview of systems for video archive organization and annotation.

System	Web access	Navigation	Querying	Content description	Main features
JoanneumVideo Browsing Tool [Bailer 2012]	Yes	Yes	Yes	- MPEG-7 descriptors: camera motion estimation, visual activity estimation, global color features, object trajectories	- video browsing on a light table - clusters of similar keyframes illustrated with colored highlighters
Lignes de temps [IRI 2009]	No	Yes	No	- manual annotations and slicing of the video in chunks	- collaborative annotation - parallel visualization of multiple threads of annotations
MediaTable [Rooij 2010]	No	Yes	Yes	- Popular semantic concept detection algorithms (Bag of Words + Support Vector Machine classification)	- 3 types of interface(table, grid, point cloud) - automatic organization of videos on rows, grids, clouds according to queries - "bucket-list" for storing videos temporarily for further annotation, relevance evaluation, viewing
MediaPick [Bertini 2011]	Yes	Yes	Yes	- Bag of Words for automatic semantic annotation - MPEG-7 descriptors and audio descriptors for queries-by-example	- web based implementation for large surface multi-touch devices and desktops - ontology-graphs for concept visualization and querying - video organization and annotation can be performed by the users
TexMix [Bréhinier 2012]	Yes	Yes	Yes	- keywords and concepts extracted from the audio transcript and EPG - Bag of Words from interest points descriptors - geographical information for each video	- interactive user interface using HTML5 specification - navigation through keywords, maps and links between videos - dynamic summary generated from keywords - geo-tagging used for finding news related to a location

7.1.4 Research prototypes for video navigation and discovery

Many researchers have experimented with more innovative designs of the video search systems and of the associated user interfaces, envisioning new ways of discovery and exploration of video content.

In [Schoeffmann 2012a], thumbnail images representing video shots are projected onto a 3D ring for the purpose of enhancing the browsing in a large dataset. The images are organized in a ring structure based on their dominant colors, represented in the HSV color space. Groups of similar images, with respect to their color representation, are then presented in the same color-sequence.

In [Schoeffmann 2011], multiple 3D projection rings/carousels are used for hierarchical browsing of video content. Key-frames are linearly sampled from videos and displayed on the carousel. A click on a given key-frame generates another carousel displaying the first hierarchy level. Further levels of detail can be accessed similarly until frame level granularity is reached (Figure 7.6).

Halvorsen *et al.* [Halvorsen 2010] propose a web-based video search system, with video streaming delivery, for searching videos included in PowerPoint presentations, using the associated metadata. However, the specificity of the considered application limits its applicability to a large video indexing framework.

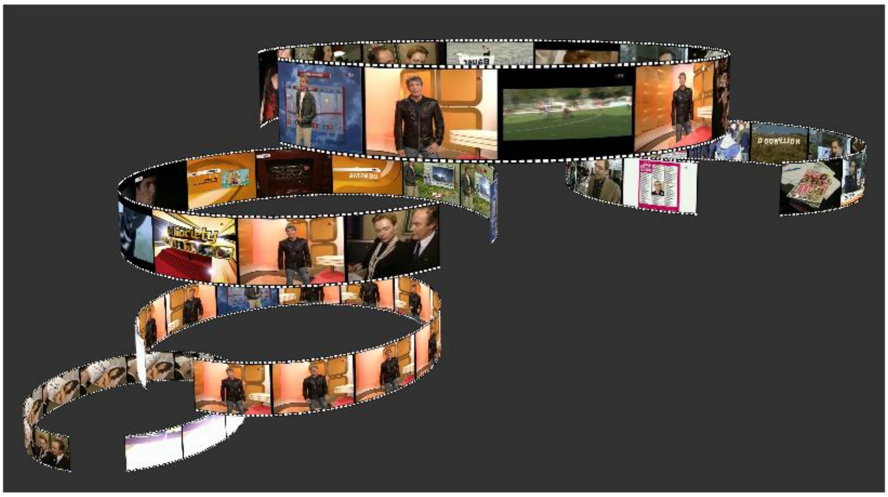


Figure 7.6. Hierarchical video browsing using 3D projection carousels [Schoeffmann 2011].

Jansen *et al.* [Jansen 2008] introduce the concept of VideoTrees, which offer a hierarchical tree-like temporal presentation of a video through associated key-frames. The key-frames are placed adjacently to their parents and siblings such that no edge lines are required to show the affiliation of a node. With each depth level, the level of detail increases as well. A user may navigate from a semantic root segment to one of the subjacent scenes, then to one of the subjacent shot groups, and finally to one of the subjacent shots. The current selected node in the tree is always centered, showing the context (*i.e.*, a few adjacent nodes) in the surrounding area.

A dynamic hierarchical browsing tool for single camera videos is also proposed in [Girgensohn 2011]. The hierarchical structure is here obtained with the help of key-frame clustering methods, based on both temporal closeness and visual similarities, in order to enforce a well-balanced tree structure. The balanced structure of the tree makes it possible to reach any key-frame of the video after equal number of steps so-called *short-paths*. The short-paths allow the users to efficiently reach the most promising clusters and to visualize their representative key-frames. The interface facilitates the quick browsing of the structure of the video for retrieving images or short clips of interest.

The above-mentioned approaches with associated features are summarized in Table 7-3.

Table 7-3. Overview on prototypes for video navigation and discovery.

System	Navigation entity	Main Features
3D thumbnail ring [Schoeffmann 2012a]	- 3-dimensional ring of keyframes clustered by colors	-frames are organized by HSV dominant color similarity
Hierarchical video browsing [Schoeffmann 2011]	- multiple 3-dimensional carousels of keyframes	- hierarchical video decomposition in slices of equal lengths until frame granularity is reached - a new carousel is used for every level
VideoTrees [Jansen 2008]	- hierarchical tree-like temporal presentation of a video through keyframes	- keyframes are placed adjacently to their parents and siblings and no edge lines are required to show the affiliation of a node - the current node in the tree is always centered and shows just his adjacent nodes
Adaptive clustering and visualization [Girgensohn 2011]	- balanced tree structure of videos for quick navigation	- balanced tree structure decomposition using temporal and visual proximity - tree structure adapted to task (image search or video shot search)
vESP[Halvorsen 2010]	- Web based grid interface	- Search engine for videos obtained from PowerPoint presentations

Let us now present the video browsing and search engines dedicated to mobile usages.

7.2 Mobile platforms

Currently we are witnessing a proliferation of powerful mobile phones, so-called “smartphones”, capable of fast connections and high-fidelity multimedia rendering. According to the consultancy company Nielsen, the amount of smartphone subscribers in the United States has augmented from 14% at the end of 2008 to 29% at the beginning of 2010. The same study predicts that by the end of 2011 in the U.S. there will be more smartphones than feature phones [Nielsen 2011].

Meanwhile, average mobile users seem to take more advantage of the new features provided by the phone, such as the faster internet connection. Thus, the use of Wi-Fi increases 10 times from 5% for feature phone owners to 50% for smartphone users [Nielsen 2011]. A recent study made by

Nielsen illustrates that in the U.S. active mobile video users grew by 57% from the fourth quarter of 2008 to the fourth quarter of 2009, from 11.2 million to 17.6 million [Nielsen 2010]. This means that mobile subscribers are also developing a growing appetite for online video content.

When considering video-related applications, the functionalities of such devices are however constrained by their size and by the computational/memory capacities. Existing studies have put an emphasis on the consumer's behavior, aiming to build user-driven interfaces and applications [Wang 2007]. Within this context, an important technological challenge concerns the issue of accessing/retrieving video content from mobile devices. The critical point relates to the high complexity of video content, in terms of amount of heterogeneous information included. In order to tackle the issue of complexity, appropriated presentation and search engines, as well as novel interaction modalities need to be developed. In addition, in a mobile framework, it is of outmost interest to ensure a personalized access to *segments* of interest, defined as parts of an audio-visual document. User should have the possibility of rapidly browsing the content and identify/access solely the segments of interest in order to limit the bandwidth and storage capacity required.

Due to the high level of complexity of designing a video navigation and retrieval system on mobile terminals, most of the early developments have been focusing rather on image-based approaches. Within this context, let us first mention the Multimodal Automatic Mobile Indexing (MAMI) system [Anguera 2008], which allows users to annotate and search for digital photos via speech input combined with time, date and location information.

Jesus *et al*. use geographical queries to retrieve personal pictures from various locations of interest [Jesus 2007]. Zhu *et al*. integrate the above-mentioned features in their user-centric system, called iScope [Zhu 2009]. The iScope system uses multi-modality clustering of both content and context information for efficient image management and search. In addition, online learning techniques are exploited for predicting images of interest, while supporting distributed content-based search among networked devices.

Kim *et al*. use CBIR techniques for visual-content recommendation [Kim 2004] while Yeh *et al*. exploit mobile images for content-based object search [Yeh 2005]. The CLOVER system [Kim 2005] searches for sketches or photos of leaf images on a server starting from a mobile phone. Photo-to-Search performs queries directly on the web using images acquired with a mobile device [Jia 2006].

Recently, a handful of mobile image recognition systems have conquered the smartphone world as applications for instant object recognition from pictures taken by the user and uploaded to a dedicated web server. The server processes the image descriptors and launches a query on a search server and the results are then send back to user. Let us mention the pioneer application PlinkArt [PlinkArt 2010] that is able to recognize works of art from museums starting from pictures acquired with a mobile phone . Recognition of film posters, product catalogs, printed ads, billboards or product packaging can be performed with the help of systems such as such as pixlinQ [PixlinQ 2012], Kooaba [Kooaba 2012], Moodstocks [Moodstocks 2012] or Google Goggles [Google 2012c]. Google Goggles can additionally recognize logos and landmarks by making use of the GPS functionalities of the phones. Such systems provide API's for advertisers, publishers and consumers for indexing their own content and make it retrievable by visual queries through different mobile applications.

Let us now present and discuss the mobile visual content retrieval systems that specifically address the issue of video content navigation and retrieval on handheld devices.

The issue of incompatibility between video resolution and certain mobile phones is addressed by the researchers from Zhejiang University [Liu 2009]. The user feedback is here used in order to update the metadata for video clips, including the acceptable resolution. In addition, redundant versions with lower resolution are generated for video clips by estimating their popularities.

The Multimedia Content Creation Platform (MMCP) [Jarvinen 2009] takes full advantage of the context information provided by the mobile device each time a new picture or video is created and added immediately as a metadata (date, time, location, etc.). The platform can be used both for generating new contact and retrieve content as well.

The system proposed in [Chen 2008] allows users to retrieve video content starting from a mobile photo uploaded by the user. The search engine in the background returns videos containing key frames that are similar with the uploaded picture. An automatic key frame extractor and the Contrast Context Histogram (CCH) are here used as descriptors.

Let us also mention the innovative approach proposed by Miller *et al.* Their mobile media content browser, so-called MiniDiver, has been developed and installed on an iPhone device. MiniDiver [Miller 2009] is based on four user interfaces serving mobile context sensitive video. After selecting the desired video or program via a simple interface, the user can visualize the content from one or two camera perspectives simultaneously. The MiniDiving user history is saved and can also be retrieved. Some semantic search mechanisms are provided for improving the navigation experience. In the case of live broadcasts, users can get highlighted a certain number of moving objects (*e.g.*, hockey players). They can select the favorite camera angles, in a multi-camera setting.

The emergence of tablet PCs, featuring powerful hardware configuration and larger screens than typical smartphones, has opened the path for a new generation of mobile video navigation and retrieval systems. Let us mention the recent advancements of Scott *et al*. [Scott 2012] who introduced the Clipboard system for visual search and browsing on both tablet and PC. The cross-platform deployment is ensured by the HTML5 implementation of the platform ensuring access from a large variety of handheld devices, while the search process is performed by selecting concepts from a set pre-defined dataset. Queries-by-example can be also performed by simply selecting a key-frame. The videos are decomposed into scenes and shots. MPEG-7 visual and audio descriptors are used for representing content. The search engine relies on several approaches, including a Bag-Of-Words model, a SVM framework for concept-based search, the fusion of MPEG-7 descriptors and Automatic Speech Recognition (ASR) transcripts of video-shots.

The various systems presented in this section, with associated features and functionalities are summarized in Table 7-4.

Table 7-4. Overview of mobile multimedia navigation and retrieval systems.

System	Image/Video	Device	Web access	Navigation	Querying	Main Features
MAMI [Anguera 2008].	image	mobile phone	no	yes	yes	- annotation and search for photos on the phone - audio tags, MPEG7 descriptors combined with time, date and location information are fusioned for search
Geographic Image Retrieval [Jesus 2007].	image	PDA	yes	yes	yes	- image retrieval using geographic queries - geographical coordinates, maps and images can be used as queries to retrieve images related to a touristic landmark
iScope [Zhu 2009].	image	smartphone	yes	yes	yes	- visual descriptors (color histogram, wavelets, region colors, SIFT) are fusioned with GPS and time information for image clustering and retrieval
pixlinQ [PixlinQ 2012], kooaba [Kooaba 2012], Moodstocks [Moodstocks 2012], Google Goggles [Google 2012c]	image/video	smartphone/tablet	yes	no	yes	- fast recognition of works of art, film posters, commercial products, ads, book covers, logos, landmarks - GPS information of the device is considered in the search - typically function as APIs for various commercial applications
Zhejiang University system [Liu 2009]	video	mobile phone	yes	yes	yes	- system capable to recommend to users videos adapted to their resolutions and bandwidth
Multimedia Content Creation Platform [Jarvinen 2009]	image/video	mobile phone	yes	yes	yes	- web-based platform for managing user-generated content - content can be retrieved from context information
Mobile Device Video Retrieval [Chen 2008]	video	mobile phone	yes	no	yes	- images taken by the user are uploaded and queried on a video database using the Contrast Context Histogram -the query image is matched with video keyframes
MiniDiver [Miller 2009]	video	smartphone	yes	yes	no	- context sensitive video browser environment - provides mechanisms to select multiple camera views, navigate hyperlinked video, save and retrieve navigation history
Clipboard [Scott 2012]	video	tablet	yes	yes	yes	- HTML5 implementation available to multiple handheld devices - BoW and SVM are used for concept-based search - MPEG-7 descriptors and AST transcripts are used for visual similarity search

7.3 Discussion

The analysis of the state of the art shows that multiple bottlenecks need to be addressed and solved in order to come out with a video indexing platform able to jointly meet different user requirements, different levels of expertise of the users and various use cases, while fully exploiting the intrinsic spatial-temporal structure of a video document. Typical user requirements that need to be addressed are the fast navigation/browsing of the video content, intuitive and efficient querying techniques, fast retrieval of relevant content displayed in a format allowing quick identification of the targeted content, ease of access from various types of terminals and operating systems (OSs). Complex and professional services additionally require elaborated and multi-modal video descriptions, combining multiple types of descriptors (audio, visual, textual, semantic).

Concerning the mobile systems, the majority of the current approaches address the issue of image retrieval, as the deployment of a video browsing and retrieval platform on an enormous variety of terminals is still a highly challenging issue. Until recently, most of the mobile systems were implemented as native applications, with different specifications for each smartphone, OS and hardware configurations. The recent advancements brought by the HTML5 standard have opened new paths for developing such systems as web applications, taking advantage of the new features provided by the mobile web browsers. So far, the existing pioneer approaches offer interesting preliminary solutions to the issue of mobile video access. However, most of them are basically mobile versions of their desktop-based counterpart and hence, do not take into account the specificity of mobile devices, environments and usages/services. In addition, they massively focus on textual queries, while an efficient search process should consider rich and multimodal queries, combining text, image, audio and video features.

Another drawback of existing systems comes from the fact that videos are often considered in a monolithic manner, without taking into account the intrinsic spatial-temporal structure of the video content. However a common video document may include a huge amount of heterogeneous information that needs to be identified, described and accessed independently.

Thus, elaborating dedicated tools for query formulation, metadata driven visualization/navigation and ergonomic user interaction in both fixed and mobile environments is still a challenge that needs to be addressed and solved.

The above-mentioned problems are addressed by the proposed OVIDIUS system. We begin by defining the required design criteria on which we will built upon. Rooij *et al.* [Rooij 2008] identified six criteria for designing video search engine interfaces:

1. The interface should always indicate to the user where he/she can go;
2. The interface should not overwhelm the user;
3. The interface should avoid switching between different interfaces. Also, the faster the user is able to select;
4. The interface must use a clear mapping between navigation and visualization;
5. The interface must be able to browse within a shot;
6. The interface must support queries demanding explicit motion.

Out of these criteria we retain and adapt three of them (*i.e.* 1, 2 and 4) as being highly important for a video search engine interface. In addition, we propose four other criteria that we consider essential for a video browsing and retrieval system. The retained design criteria are explained here below.

Criterion 1: The interface should always indicate the possible directions for browsing videos/databases.

When using an intra-video navigation system users can easily get lost inside the video. In such cases, for each new operation the user have to restart their navigation or search. In order to avoid this, the interface should provide information about the current position of the user inside the video or the search results and it should also display directions for further navigation or for returning at an earlier point of choice.

Criterion 2: The interface should not overwhelm the user.

Contrary to the first criterion, an interface could display extra directions and information overloading the displayed space and overwhelming the user's attention span. If an interface is too rich in information, the user requires more time to adapt to it, to use it and to process the available visual information before a decision is possible.

Criterion 3: The system should provide a balance between the visualization, navigation and search features

When searching for a certain piece of content, users can skim through results and typically take a closer look at some of the results in order to check whether the respective result is the targeted one or not. The interface should allow the visualization of the results and the possibility to return to the navigation of the results. In addition, the search should not occlude the navigation and visualization functionalities. Search could be done in parallel while the user can still navigate in video portions of interest.

Criterion 4: The user should be able to browse within the video hierarchically.

Videos are rich containers of information and a global description of its content is difficult to obtain. For an easier understanding and faster navigation of the video, a hierarchical structure of the video needs to be elaborated. A three-level tree decomposition of the video in scenes, shots and key-frames can be suitable for different use cases. The grouping of the shots in meaningful scenes is also an issue to be addressed for a better navigation and search experience.

Criterion 5: The query formulation should be performed in a user-friendly and intuitive manner.

Most users are accustomed to performing textual queries whenever they are looking for information (*e.g.* documents, images, videos ...). Such a search mechanism has to be integrated in the system in an easy to spot place. A list of choices in terms of keywords to be used for search can be provided in order to guide the user in his search. For video search engines, visual queries can prove more effective than its textual counterparts. Yet, in the majority of cases this is a new concept for the user. Therefore, a visual search mechanism should both serve as a search tool and as a tool for training the user in using visual queries. It should be seamlessly integrated in the platform and easy to use.

Criterion 6: The system should be remotely accessible.

Both personal and industrial video archives can expand quickly and require massive amounts of storage space. In such cases, users typically store their content on a personal server or on cloud-based service allowing simultaneous multiple user access. A video navigation and search system should be then remotely accessible granting access to online video repositories to any Internet connected device.

Criterion 7: The interface should be plugin-free and available to multiple terminals and operating systems

Currently we are witnessing the existence of a myriad of computers, devices, operating systems with different screen sizes, functionalities, web browsers and operating systems versions. Developing applications adapted to the specificity of each device requires the elaboration of different versions, each one dedicated to a certain device. An obvious alternative is to develop a web based system that can be accessed from a multitude of devices. Yet, many popular web technologies (*e.g.* Flash) require the installation of a web browser plugin for content rendering. Such plugins are becoming unavailable to many portable devices. In order to ensure the availability to a higher number of terminals, the interface should be implemented with standard open web technologies (*e.g*., HTML5, JavaScript) compatible with most of the Internet connected devices.

Such criteria have been taken into account in the design of the OVIDIUS platform, described in the following chapter.

8. The OVIDIUS platform

Abstract: *In this chapter we introduce a novel on-line video browsing and retrieval platform, so-called OVIDIUS (On-line VIDeo Indexing Universal System). In contrast with traditional and commercial video retrieval platforms, where video content is treated in a more or less monolithic manner (i.e., with global descriptions associated with the whole document), the proposed approach makes it possible to browse and access video content in a finer, per-segment basis. The hierarchical metadata structure exploits the MPEG-7 approach for structural description of video content. The MPEG-7 description schemes have been here enriched with both textual/semantic and content-based metadata. The platform is accessible on the web from various, both desktop and hand-held devices. Moreover, the system exploits a modular and distributed architecture that makes it possible to distribute the workload on various servers, with heterogeneous operating systems. The various communication protocols adopted are also presented in this chapter. Finally, OVIDIUS integrates the DOOR method and allows fast video object retrieval.*

Keywords: *multimedia platforms, browsing and retrieval, video, web services, MPEG-7, HTML5, multi-terminal access.*

Résumé: *Dans ce chapitre, nous introduisons une nouvelle plate-forme en ligne pour la navigation et la recherche de vidéo, dits OVIDIUS (On-line VIDeo Indexing Universal System). Contrairement aux plates-formes traditionnelles et commerciales de recherche de vidéo, où le contenu vidéo est traité d'une manière plus ou moins monolithique (c'est à dire, avec des descriptions globales associées à l'ensemble du document), l'approche proposée permet de naviguer et d'accéder au contenu vidéo dans une manière plus fin, par-segment. La structure hiérarchique des métadonnées exploite l'approche de la norme MPEG-7 pour la description structurelle du contenu vidéo. Les schémas de description MPEG-7 ont été ici enrichie de métadonnées textuelle/sémantique et basées sur le contenu. La plate-forme est accessible sur le web depuis différents appareils, fixes et portables. En outre, le système exploite une architecture modulaire et distribuée qui permet de répartir la charge de travail sur des serveurs différents, avec des systèmes d'exploitation hétérogènes. Les protocoles différents de communications adoptées sont également présentés dans ce chapitre. Enfin, OVIDIUS intègre la méthode DOOR et permet la recherche rapide des objets vidéo.*

Mots clés: *plates-formes multimédias, navigation et recherche, vidéo, services Web, MPEG-7, HTML5, accès multi-terminal.*

The proposed OVIDIUS (*On-line VIDeo Indexing Universal System*) platform aims at solving the limitations of existing systems, by ensuring all the interaction and navigation capabilities needed to access video content at a fine level of granularity, from both fixed and mobile devices.

The main contributions proposed and integrated in the OVIDIUS system are the following:

- Modular and distributed architecture, achieved with the help of web services, which makes it possible to easily upgrade the system in order to keep pace with inherent future technological advances,
- Fine granularity access to video content, based on the MPEG-7 structural approach for video content description [MPEG 2003]
- Core interoperability achieved with open MPEG-7 standard technologies,
- Enrichment of MPEG-7 structural description schemes with semantic and content-based descriptors,
- Advanced interaction functionalities integrating browsing, search, hierarchical navigation and visualization capabilities on an HTML5 plug-in free interface,
- 3 different types interfaces for different types of use (lightweight, professional, mobile)
- Support of both textual, content-based and hybrid queries,
- Compatibility with a vast variety of terminals and operating systems.

8.1 OVIDIUS platform: system overview

Figure 8.1 illustrates the distributed architecture adopted in the OVIDIUS platform. It includes a content management module (*i.e.*, storage and editing of content and metadata), a metadata extraction engine, a MPEG-7 search engine and a web interface which can be remotely accessed from mobile and fixed environments.

All the components of the system can be distributed on various servers. We consider an Apache web server for the frontend [Apache 2012a], while for the background processing we employ Apache Tomcat servers [Apache 2012b]. The communication between servers and modules is achieved through HTTP requests and Java Servlets, while the data is transferred as XML files. Such a modular approach facilitates the future extensions of the platform as well as the replacement of individual components (*e.g.*, video players, search engine, type of content descriptors, etc.).

An in-depth diagram of the employed technologies together with the associated communication protocols between the various modules of the system is presented in Figure 8.2.

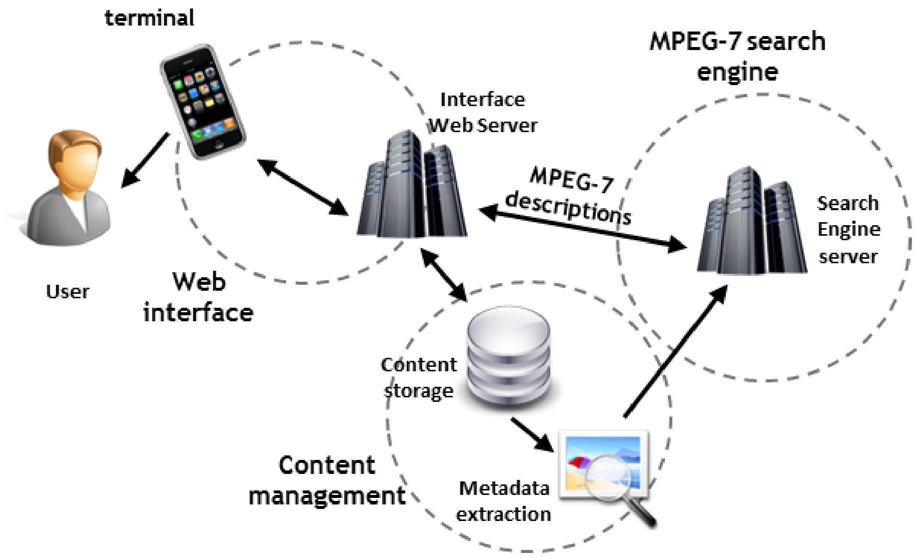


Figure 8.1. Overview of the architecture of the OVIDIUS platform.

For the client-side, our implementation relies on the HTML5 specification [HTML5 2012a]. The visual effects and the interactivity functionalities are enhanced with JavaScript functions. In addition we employ that jQuery JavaScript library [jQuery 2012] for the management of page events and AJAX (*Asynchronous JavaScript and XML*) methods [AJAX 2012] enabling the exchange of XML files between the client and the server and quick updates of sections of the web page. This makes it possible to quickly update individual sections of the web page, according to the user input. The communication with the web server is performed through HTTP requests and responses and with AJAX based HTTP POST commands.

For the server-side, we have considered an Apache web server [Apache 2012a], which handles all the communication with the client. The Apache server relies on php functions. In addition, we insert some supplementary servers in the architecture of the system with specific roles. The workload is thus distributed on several machines and the architecture can be further extended with other servers with new functionalities. This frontend Apache server is connected to a Java Apache Tomcat server [Apache 2012b] (*MPEG-7 search engine server* from Figure 8.2) implementing Java Servlets. The MPEG-7 server manages the video metadata (video IDs, EPG information; video structure, time stamps, keywords, visual descriptors) in XML format. As it stores the video metadata, the MPEG-7 server is also employed as a search engine. The communication between the frontend server and the MPEG-7 server enables the quick navigation in the video and the update of the displayed video content and information according to the user's input and decisions. Once the user performs an action requiring the display of additional content, the Apache servers sends to the MPEG-7 server a HTTP request containing the parameters given by the user's action. The MPEG-7 server processes this request and returns the result in XML format. The Apache server parses the XML in php and displays the required content on the web page. Note that the AJAX method from the web interface can also trigger requests and communications between the Apache and the MPEG-7 server. This is typically a HTTP POST request and the result is displayed by updating a specified section of the web page, without reloading or manually refreshing the whole page.

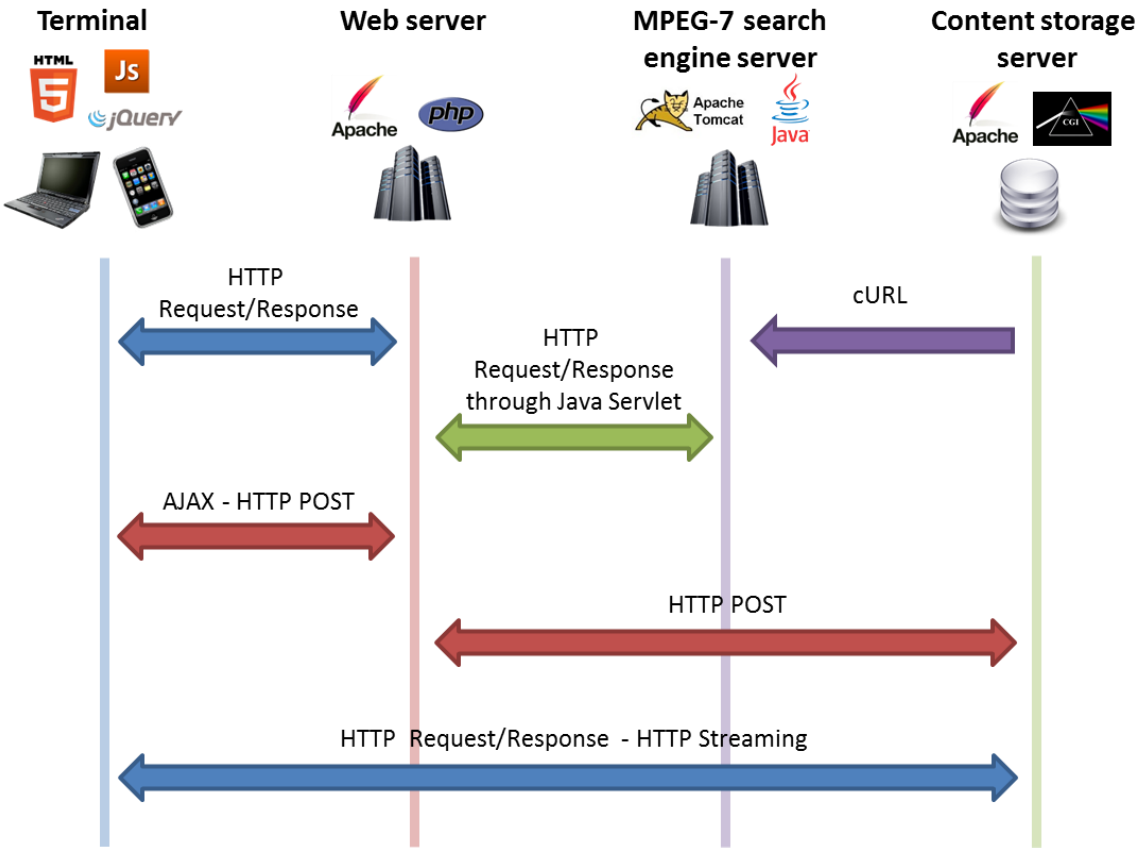


Figure 8.2. Technologies and communication protocols employed for the modules and the communication between them.

The Apache frontend server communicates with a Content storage server (which is also an Apache web server). The main purpose of this server is to store the videos from the dataset and their associated keyframes. In addition, the content storage server supports the metadata extraction function. Whenever a new piece of content is added to the dataset, the processing of the content and the description extraction are performed on this server with C/C++ programs. Once extracted, the metadata is then uploaded to the MPEG-7 server through cURL commands [CURL 2012]. The content storage server also handles the processing of the object retrieval methods presented in Section 3. The search application is a C++ program and it's launched with an HTTP request sent to its CGI interface [CGI 2012]. This request is performed by the Apache server and the results are returned in XML format and displayed to the user. The content storage server communicates with the terminal as well, sending key-frames and streaming the video content to the client.

The following sections describe the functionalities of each of the components considered.

8.1.1 MPEG-7 structural video content description

The ISO/WG11/MPEG-7 standard [MPEG 2001], is a multimedia content description standard, emerged in the early 2000 years. MPEG-7 is formally called *Multimedia Content Description Interface*. Unlike the previous MPEG standards (MPEG-1/2/4), which deal with the issue of multimedia//video content encoding, MPEG-7 provides a complementary functionality, related to image/audio/video metadata representation. Thus, the scope of MPEG-7 concerns the standardization of multimedia content descriptions.

MPEG-7 can be used independently of the other MPEG standards and is also independent of the representation used to store the content. In a general manner, the objectives of the MPEG-7 standard are the following:

- Provide fast and efficient searching, filtering and content identification technologies;
- Describe main issues about the content (low-level audio and visual characteristics, structure, semantics, models, collections….);
- Support a wide range of applications,
- Describe the relationships between objects included in a scene,
- Ensure independence between the description and the information itself.

In order to achieve such objectives, a number of different tools have been developed. They include a set of descriptors (the elements), description schemes (the structures), a Description Definition Language (DDL) (for extending the predefined set of tools) and a number of Systems tools [Pereira 2002].

Such tools are recalled here-below:

- **Descriptors (D):**
    - A descriptor is a representation of a feature, where a feature is a distinctive audio or visual characteristic of the data (*e.g.*, color, texture, motion, 2D/3D shape…)
    - A descriptor allows an evaluation of the corresponding feature via the descriptor value. It is possible to have several descriptors representing a single feature, in order to address different relevant requirements/functionalities.
- .**Description Schemes (DS):**
    - A description scheme specifies the structure and semantics of the relationships between its components, which may be both descriptors (D) and description schemes (DS).
    - A description scheme provides a solution to model and describe multimedia content in terms of structure and semantics.
- **Description Definition Language (DDL):**
    - The DDL is a language that allows the creation of new description schemes (DS) and, possibly, descriptors (D).In the case of MPEG-7, an XML Schema approach has been adopted
- **System tools:**
    - Tools related to the binarization, synchronization, transport and storage of descriptions, as well as to the management and protection of intellectual property.

MPEG-7 fulfills a key function in the forthcoming evolutionary steps of multimedia. As much as MPEG-1, MPEG2 and MPEG-4 provided the tools through which the current abundance of audiovisual content could happen, MPEG-7 provides the means to navigate through this wealth of content. The strong point of MPEG-7 is its generality. Today we are facing the age in which content companies are by no means constrained by their own traditional delivery mechanisms, while content consumers are no longer tied to a single source of content. This generality of MPEG-7 is a value that is in line with past MPEG standards: the objective is to provide a generic solution that is technically agnostic of the environment [MPEG 2001].

Even without MPEG-7, there are many ways to describe multimedia content in use today in various digital asset management systems. Such systems, however, generally suffer from several interoperability issues and creating/adopting a standard is the only way to address them. The main results of the MPEG-7 standard are the increased interoperability, the prospect to offer lower cost products through the creation of a sizable market with new, standard-based services and a rapidly growing user base.

In our work, we have considered the MPEG-7 structural approach for video description [MPEG 2003], which is based on an abstract class of *Segment*. An MPEG-7 Segment represents an arbitrary part of a video and includes generic descriptors (*e.g.*, textual annotations, keywords, temporal localization elements for specifying the starting and the ending time stamps of a segment, spatial-temporal locators for specifying arbitrary-shaped objects of interest….).

Starting from this abstract structure, which cannot be directly instantiated, a set of media-specific segments is derived by applying an inheritance mechanism. In our developments, we have considered the following MPEG-7 segments: AudioVisualSegment DS, AudioSegment DS, Video DS, StillRegionDS, and MovingRegionDS.

Each segment can include dedicated, both textual and content-based descriptors, adapted to each segment type. Examples are color features for still region/video segment, audio features for audio segments, motion parameters for moving regions…. Let us underline that the MPEG-7 visual descriptors [MPEG 2002] can play here an important role for query by example purposes.

A second MPEG-7 mechanism exploited is the Segment Decompostion DS, which allows the partition of a segment into sub-segments. In combination with a recursive decomposition mechanism (which may be temporal, spatial-temporal or spatial), such an approach makes it possible to create a hierarchical and multi-granular video content description, represented as a tree of segments and adapted to both navigation/browsing and search functionalities. Although arbitrary numbers of levels are supported, the most widely used video structure utilizes a three-level hierarchy made of scenes, shots, and key-frames/objects of interest (Figure 8.3).

The adopted MPEG-7 language for specifying all these descriptors and descriptions schemes is XML Schema. This choice facilitates the parsing and interpretation of the descriptions, since various XML utilities are available and can be directly used (*e.g.*, Xerces parser). The XML files are notably well-suited as an exchange data format between the different modules of our system through HTTP requests. XML files can be easily formatted to the MPEG-7 specification, making it possible to communicate with other systems and modules on the web, exchanging metadata, descriptors, queries and results.

Let us note that, recently, several lightweight data-interchange formats have emerged on the web as alternatives to XML, which has been criticized for being large and less suitable for quick transfer of small files. Among the most popular such formats is JSON [JSON 2012], which is often used for serializing and transmitting structured data over a network connection. Native JSON support is included in most of the general public browsers. JSON is designed for human-readable data interchange and discards the closing tags typically included in XML files. When data is encoded in XML, the result is typically larger than an equivalent encoding in JSON. Yet, if the XML data is compressed using a standard algorithm such as *gzip* [GZIP 2012] the gains obtained with the JSON approach become negligible. Moreover, the adoption of the JSON format would require a conversion module to the MPEG-7 format on each server. For these reasons we have adopted the classical MPEG-7 XML format for data exchange between the various OVIDIUS modules.

OVIDIUS can exploit arbitrary extraction methods and utilities, provided that the representation of the description is compliant with the MPEG-7 specification. In our work, we have adopted the video temporal and spatial-temporal segmentation/structuring framework recently proposed in [Tapu 2011], which makes it possible to automatically extract scenes, shots, key-frames and objects of interest (Figure 8.3).

Concerning the visual descriptors, we have adopted an updated version of the MPEG-7 reference software, exploited in [Zaharia 2010]. In this way, the totality of MPEG-7 visual descriptors is supported by the system. In addition, we have also considered other descriptors, outside MPEG-7 and corresponding to various video analyzers that can be integrated (*e.g.*, the TextureLEP descriptor [Cheng 2003]). The only constraint in this case is to specify for each descriptor a XML-based representation.

Disposing of metadata is a first and essential step in the indexing process. However, appropriate video visualization and interaction capabilities need to be elaborated in order to efficiently exploit such a description. The OVIDIUS user interface integrates all the necessary interaction, navigation and search capabilities, as described in the following sections.

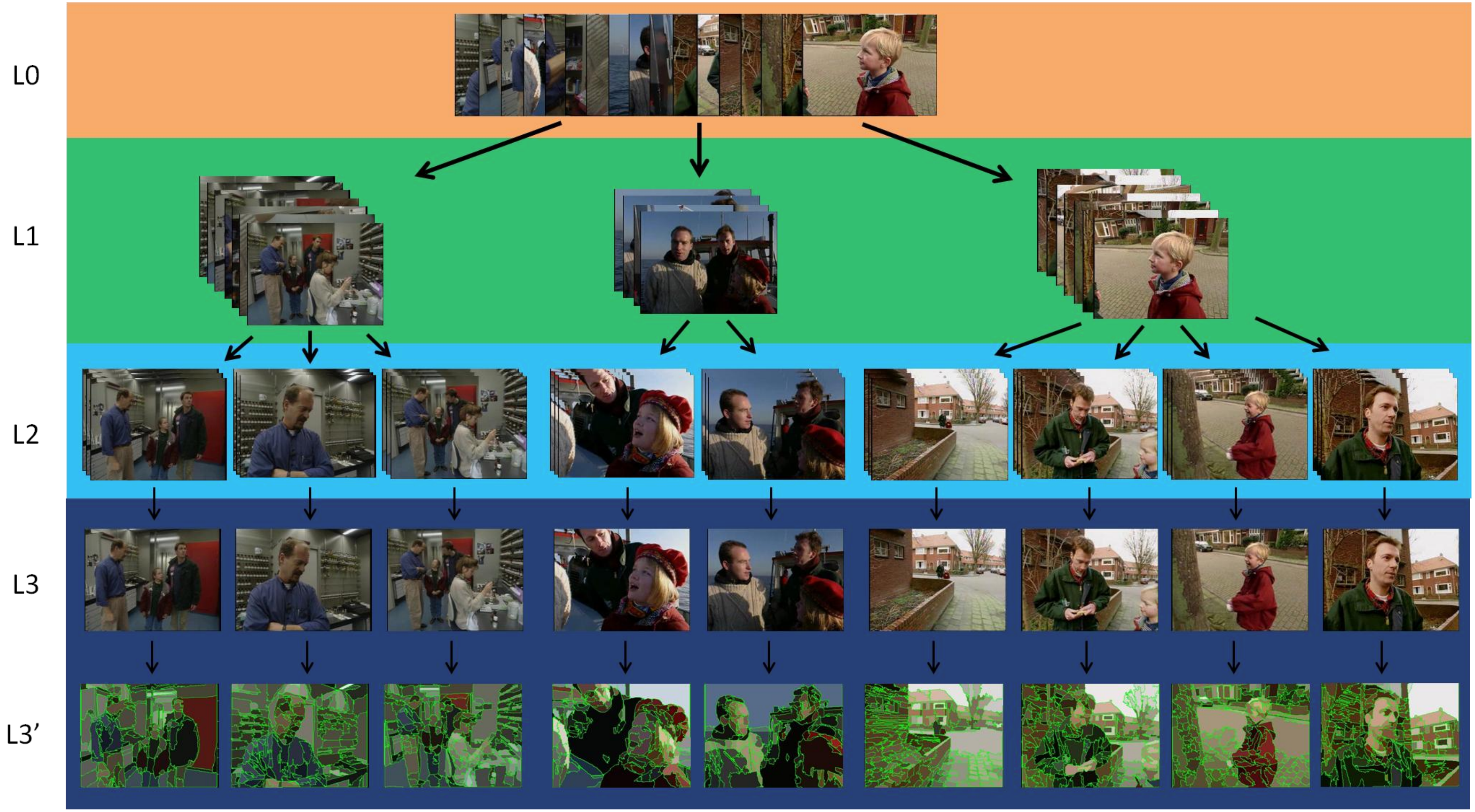


Figure 8.3. Hierarchical decomposition of a video (L0) in scenes (L1), shots (L2) and keyframes(L3). Level L3' concerns the regions from the keyframes, which are the entities used for video similarity in this case.

8.1.2 Video players

With the exponential increase of multimedia content available today on the Internet, a large variety of video coding formats have emerged and are today extensively used by various multimedia applications.

The explosion of such formats makes it possible to visualize a large variety of content from fixed or mobile terminals. However, the richness of the video formats available today is in the same time a drawback from the point of view of the application developer. Obviously, it is non-realistic to implement specific encoders for all these formats which, in addition, are constantly evolving. Which are the “good” formats to choose and implement? How to ensure the maintenance of applications by taking into account the rapid evolutions of such formats? How to ensure interoperability with other formats? These are fundamental questions that have to be considered when designing a multimedia system.

From the renderer’s perspective, it is almost impossible to find nowadays a “one-fits-all” solution. There is large spectrum of reliable players available, but it is quite difficult to find a unique one able to perfectly fit any media format around.

This is why many stakeholder portals content providers (*e.g.*, YouTube [YouTube 2012], Facebook [Facebook 2012], DailyMotion [DailyMotion 2012] …), allow users to post digital content and to transcode it all into a single format (H.264), in order to guaranty the correct rendering of all the media clips proposed by their services. In most of the cases the video player of choice is the Adobe FlashPlayer [Adobe 2012], which is largely the most popular video player on the web. However, the Adobe FlashPlayer requires the users to download a browser plug-in in order to visualize the video content.

Recently, with the considerable advancements of HTML5 format [HTML5 2012a], the introduction of the so-called *video element* [HTML5 2012b] in the HTML5 syntax, along with its improved rendering and interactivity features, Internet browsers can now render videos natively without the necessity of downloading or updating a video player plugin.

Yet, despite the significant progress of the HTML5 technologies and the perspectives that it unfolds, currently there is no codec supported by all the major web browser developers (Table 8-1). The two most popular formats on the web are the H.264, supported by Apple and Microsoft, and WebM [WebM 2012] using the VP8 codec owned by Google.

Most of the general public video platforms (*e.g.*, YouTube [YouTube 2012], DailyMotion [DailyMotion 2012], Vimeo [Vimeo 2012]) have started to transcode their videos in such formats making a part of their content available for the HTML5 native player. The incompatibilities between the browsers and the video formats are solved in the HTML code which automatically verifies these compatibilities and provides the right content with the help of a *fallback* function (Figure 8.4).

Table 8-1. Supported video formats by popular web browsers and mobile operating systems

Web Browser	Chrome	Firefox	Internet Explorer	Safari	Opera	Android	iOS
							iOS
Version	21	15	9	5	12	4	5
H.264	Yes	No	Yes	Yes	No	Yes	Yes
WebM	Yes	Yes	No	No	Yes	Yes	No
Ogg/Theora	Yes	Yes	No	No	Yes	Yes	No

```
<video width="640" height="480">
      <!-- if Chrome/Safari -->
      <source src="video.mp4" type="video/mp4">
      <!-- if Firefox/Opera -->
      <source src="video.webm" type="video/webm">
      <!-- if the browser doesn’t understand the <video> element, then reference
a Flash file and open the Flash player -->
      <embed  src="video.flv"  type="application/x-shockwave-flash"  width="640"
height="480" allowfullscreen="true" allowscriptaccesss="always"></embed>
</video>
```

Figure 8.4. HTML.5 syntax for fallback to different video formats

The OVIDIUS platform features a scene/shot/keyframe-based navigation in videos in order to provide a fine granularity access to specific items of interest. Therefore, the considered players should offer an enhanced configurability and notably advanced features for random access to video segments: a video should be accessed and visualized in a precise and reliable manner at given time instants/intervals specified by the description. However, such a random access to arbitrary time instants and intervals is not yet fully supported by existing video players. Our experiments demonstrated that the Flash Player and the HTML5 video player ensure a most reliable random access. The VLC player could allow it only for a limited category of video formats, namely MPEG-1 and MPEG-2, but fails for other formats.

A highly interesting feature of the HTML5 player is that it allows fast seeking to not-yet downloaded parts of the video, within a streaming scenario. This can be achieved with HTTP 1.1 Byte Range requests [HTTP 2012], supported by most common web browsers. This is the standard way in which web browsers receive HTML5 media resources from the web servers. In this case, the browser first retrieves the initial byte ranges of a media resource, which specify the audio and video decoding pipelines that enable to play the content. The audio and video data in a media resource are then provided in a multiplexed stream. If the browser asks for a byte range on the resource, it will retrieve both, the audio and video data that belongs to the same time range together. It can thus progressively download byte ranges and at the same time decode and start playing back the audio-visual content already received. If the network is fast enough to feed the decoding pipeline quicker than the decoder and the graphics engine can play back the video in real time, this will give a smooth playback impression to users. In this manner, the video is rendered as it is downloaded and there is no need to wait for the whole video to be completely downloaded before playing it. When the user seeks to a certain time stamp and the seek time is not yet buffered, the browser will stop the download and request a byte range that starts at the user-specified time stamp. This feature becomes very useful in the context of mobile devices, where the available bandwidth is a typical issue to be dealt with. In this case, the amount of video data to be streamed is significantly reduced and the quick access to video segments of interest is ensured. The seeking operation is done automatically in javascript according to the selections made by the user.

The OVIDIUS platform should allow an intuitive and interactive querying mechanism. In this respect, the HTML5 video player can offer advanced interactivity features. Notably, the user can specify/draw selections of arbitrary shapes on the video content [Pfeiffer 2010]. This can be achieved by projecting each video frame on the canvas element which can then support several transformations (*e.g.*, color changes, division of content on a grid with each cell representing another frame…) and editing tools (*e.g.*, drawing shapes). We leverage on these functionalities and provide an interactive querying technique which gives to the user the possibility of selecting a region of interest directly on the video (Figure 8.17).

An analysis of the functionalities of the existing video players is presented in Table 8-2.

The criteria that we have mentioned in Section 7.3 are satisfied in the highest measure by the HMTL5 video player that offers an open, plug-in free, quick and user-friendly solution for the OVIDIUS requirements.

As a drawback, let us mention the non-uniformity of the supported video formats, with respect to the various browsers considered. In order to achieve a stable solution, adapted for a large majority of browsers, we have considered the H.264 and WebM video formats.

In addition, let us note that recent advancements in the field make it possible to connect the HTML5 video player with other portals. Thus, it becomes possible to share the specific shots from the video on the web thanks to the W3C Media Fragments framework [W3C 2012]. The *Popcorn.js* HTML5 media framework [PopcornJS 2012] is a JavaScript library dedicated to HTML5 media which allows synchronization of the video other types of content (*e.g.*, text, hyperlinks, Google Maps places, SlideShare slides, Twitter queries, Wikipedia content) displayed on the same web pages at specified video moments. Keywords or clickable subtitles can be displayed while watching a video, while known objects can be indicated on the video during its rendering.

Table 8-2. Analysis of functionalities of web browser video players

Video Player	**Adobe Flash** [JWPlayer 2012]	**VideoLan** [VLC 2012a]	**HTML5** [HTML5 2012b]
Supported video formats	H.264, Sorenson H.263, On2 VP6	Most of the video formats	H.264, WebM, Ogg/Theora
Seeking	Yes	Partially	Yes
Interactivity features	Yes	No	Yes
Proprietary	Yes	No	No
Browser support	Most of the browsers	Most of the browsers	Most of the browsers
Plugin Required	Yes	Yes	No

More recently, the Dynamic Adaptive Streaming over HTTP (DASH) standard from MPEG has been integrated into the web using the HTML5 video element as a java-script and WebM-based library for Google Chrome [Rainer 2012]. The video data is the most bandwidth resource consumer of a video platform. Such an extension would increase the accessibility capabilities of the OVIDIUS platform, by adapting it to the available bandwidth fluctuations of the user's connection.

Such perspectives show that HTML5 offers an interesting solution, with respect to the future extensions in terms of functionalities that can be considered. For all these reasons, we have privileged the HTML5 approach in our developments.

A video search engine requires appropriate user-friendly interfaces, able to specify queries and to present the search results to the user in a pertinent manner. Such aspects are discussed in the following section.

8.1.3 OVIDIUS user interface

The OVIDIUS platform interface attempts to offer a good balance between navigation, visualization and search features in an accessible web-based environment as mentioned in Section 7.3. In order to ensure a large interoperability, the interface has been developed by using HTML5, JavaScript and php technologies. While several animations and transitions effects have been used to ensure a smooth user experience, no plugins are required and the platform can be accessed from most of the popular web browsers. Due to its construction, OVIDIUS can be accessed from multiple types of terminals (desktop, mobile). The single requirement is a JavaScript-compliant Internet browser.

Let us note that the interface server (Figure 8.1) automatically detects the operating system of the user terminal and adapts accordingly (*i.e.*, if the user terminal is a mobile phone, the user is directed to a small screen adapted interface).

Depending on the usage of the OVIDIUS platform and the type of terminal from which it is accessed we propose three different designs of interfaces:

- **OVIDIUS *explorer*:** recommended for quick visualization of the content of a video and including navigation and querying functionalities,
- **OVIDIUS *finder*:** recommended for Known Item Search cases, for quick retrieval of specific objects/segments of interest inside a video,
- **OVIDIUS *mobiler*:** recommended for handheld devices for navigation and querying of a video adjusted for small screens.

The various interfaces proposed are detailed in the following sections.

8.1.3.1 OVIDIUS explorer

The *OVIDIUS explorer* is recommended for the use-case of average users looking for a video to watch. The interface proposed various tools for navigation and quick visualization/browsing of the video that help the user in getting quickly a general idea of the content of a video.

The interface features two menus:

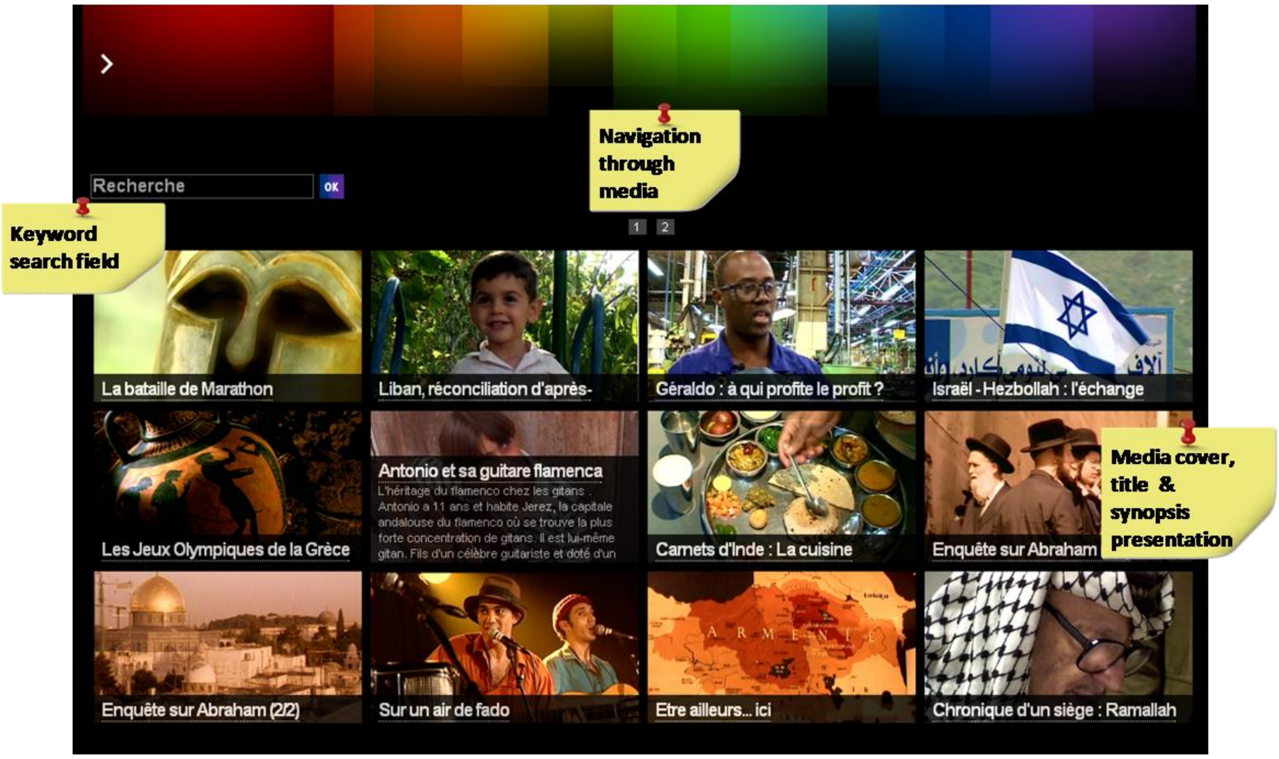
- a home portal for exploration of all the media existing in the database (Figure 8.5), and
- a navigation menu, which makes it possible to explore and visualize a given video selected by the user (Figure 8.7).

In this case, users can easily visualize iconic representations of the available media along with their title, summary and associated keywords. A simple query by keywords is proposed in order to find the most relevant content. For a better understanding of the functioning of the OVIDIUS platform on different servers, we illustrate the timeline of events performed, whenever a user accesses the home screen of the platform (Figure 8.6).

Let us further detail the navigation menu proposed. The OVIDIUS GUI makes it possible to obtain a visual and interactive representation of the MPEG-7 descriptive structure and integrates the following components:

- video player,
- selector of the segment type (VideoSegment, AudioSegment, AudioVisualSegment, StillRegionDS – *cf.* Section 8.1.1),
- selector of the hierarchical level of each segment,
- threads of iconic representations of segments, which are dynamically derived from the media without any content duplication,
- navigation/browsing buttons,
- summary and keyword visualization elements,
- selection of a timeline scrollbar for instant access to scenes from different parts of the video.

This menu aims at offering a complete user experience when searching and accessing media. While watching the video, the user can simultaneously view a summary of the content along with the corresponding keywords describing the video content.


Recherche
OK
1
2
La bataille de Marathon
Liban, réconciliation d'après-
Géraldo : à qui profite le profit ?
Israël - Hezbollah : l'échange
Les Jeux Olympiques de la Grèce
Antonio et sa guitare flamenca
Carnets d'Inde : La cuisine
Enquête sur Abraham
Enquête sur Abraham (2/2)
Sur un air de fado
Etre ailleurs... ici
Chronique d'un siège : Ramallah

L'héritage du flamenco chez les gitans . Antonio a 11 ans et habite Jerez, la capitale andalouse du flamenco où se trouve la plus forte concentration de gitans. Il est lui-même gitan. Fils d'un célèbre guitariste et doté d'un

Navigation through media

Keyword search field

Media cover, title & synopsis presentation

Figure 8.5. Exploring a video database with *OVIDIUS explorer*

The lower part of the interface is dedicated to the structured visualization of each video document. In particular, this zone is a visual illustration of the structural MPEG-7 description associated with a video document as a thread of iconic representations of the segments. Scenes, shots, and still regions can be here browsed and accessed in a hierarchical manner, as MPEG-7 segments. As illustrated in Figure 8.8, this thread of iconic images provides information about the visual content of each segment (preview image), as well as information regarding the classification and hierarchy (identifier, time stamp, order, type). With a simple click, users can access the specified segment and visualize it along with the keywords that describe it. A color code is used in order to inform the user about the current hierarchical level of access. Video scenes are represented on a thread with a green background, shots with a light blue background and key-frames with a dark blue background.

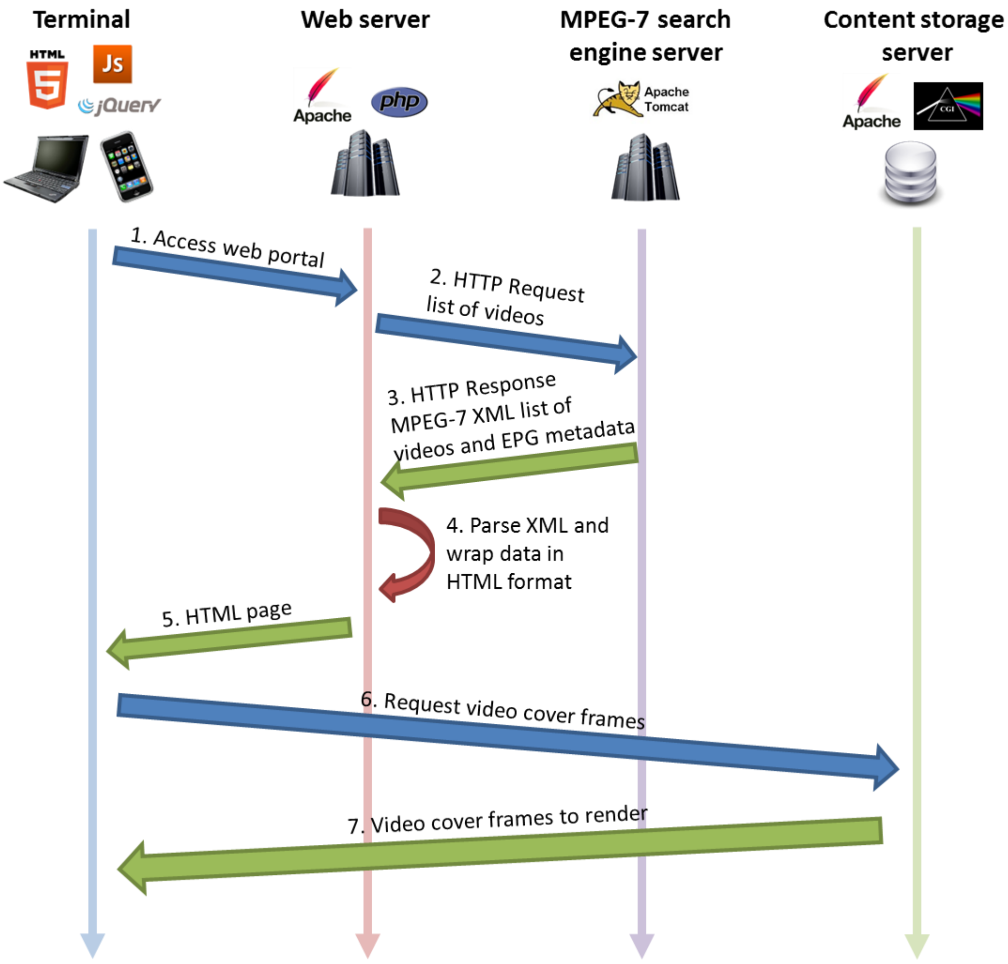
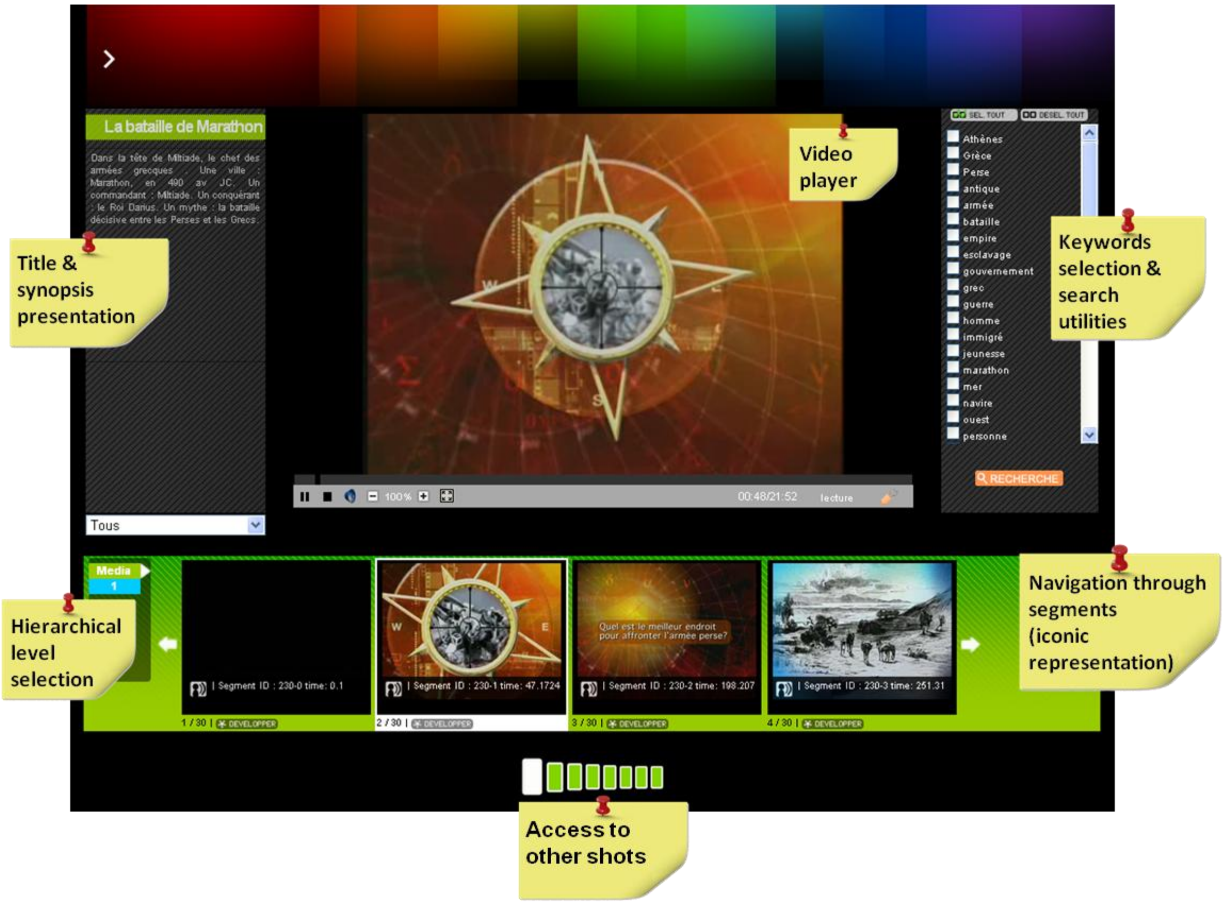

Terminal
HTML
JS
jQuery
Web server
Apache
php
MPEG-7 search engine server
Apache Tomcat
Content storage server
Apache
CGI
1. Access web portal
2. HTTP Request list of videos
3. HTTP Response MPEG-7 XML list of videos and EPG metadata
4. Parse XML and wrap data in HTML format
5. HTML page
6. Request video cover frames
7. Video cover frames to render

Figure 8.6. Timeline of events triggered when displaying OVIDIUS home screen.

The displayed keywords describe the content of the video segments depending on current level of segmentation. For the scenes, the keywords capture information related to the events occurring in that segment obtained from the soundtrack of the video, while for shots and single frames these keywords describe the physical information extracted from the content (*e.g.*, indoor, outdoor, person, vegetation …).

Users can navigate “horizontally” on the timeline of the video with a random access to any section of the video by a single click. If, instead, the user would like to go deeper in the hierarchical tree of the media, he can navigate through the “children” segments of a selected segment, with the possibility of reaching the finest level of granularity, which in our case is represented by a StillRegion (single frame).

The navigation experience through the hierarchical structure of the video segments is improved with the usage of a hierarchy panel displaying the path of segment identifiers followed by the user, with the implementation of the background color code for each level of segmentation and with a customized timeline scrollbar for instant access to scenes from different parts of the video.



La bataille de Marathon

Dans la tête de Miltiade, le chef des armées grecques . Une ville : Marathon, en 490 av JC. Un commandant : Miltiade. Un conquérant : le Roi Darius. Un mythe : la bataille décisive entre les Perses et les Grecs.

Title & synopsis presentation
Video player
Keywords selection & search utilities
Hierarchical level selection
Navigation through segments (iconic representation)
Access to other shots
SEL. TOUT
DESEL. TOUT
Athènes
Grèce
Perse
antique
armée
bataille
empire
esclavage
gouvernement
grec
guerre
homme
immigré
jeunesse
marathon
mer
navire
ouest
personne
RECHERCHE
Tous
100%
00:48/21:52
lecture
Media
1
Quel est le meilleur endroit pour affronter l'armée perse?
| Segment ID : 230-0 time: 0.1
| Segment ID : 230-1 time: 47.1724
| Segment ID : 230-2 time: 198.207
| Segment ID : 230-3 time: 251.31
1 / 30 | DEVELOPPER
2 / 30 | DEVELOPPER
3 / 30 | DEVELOPPER
4 / 30 | DEVELOPPER

Figure 8.7. *OVIDIUS explorer* media navigation and retrieval interface

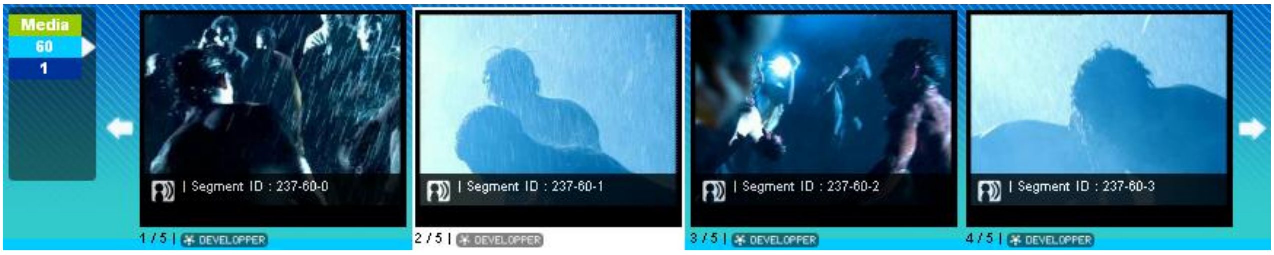


Figure 8.8. Iconic navigation panel for video shots

The timeline of the events running in the background during the video navigation is illustrated in Figure 8.9.

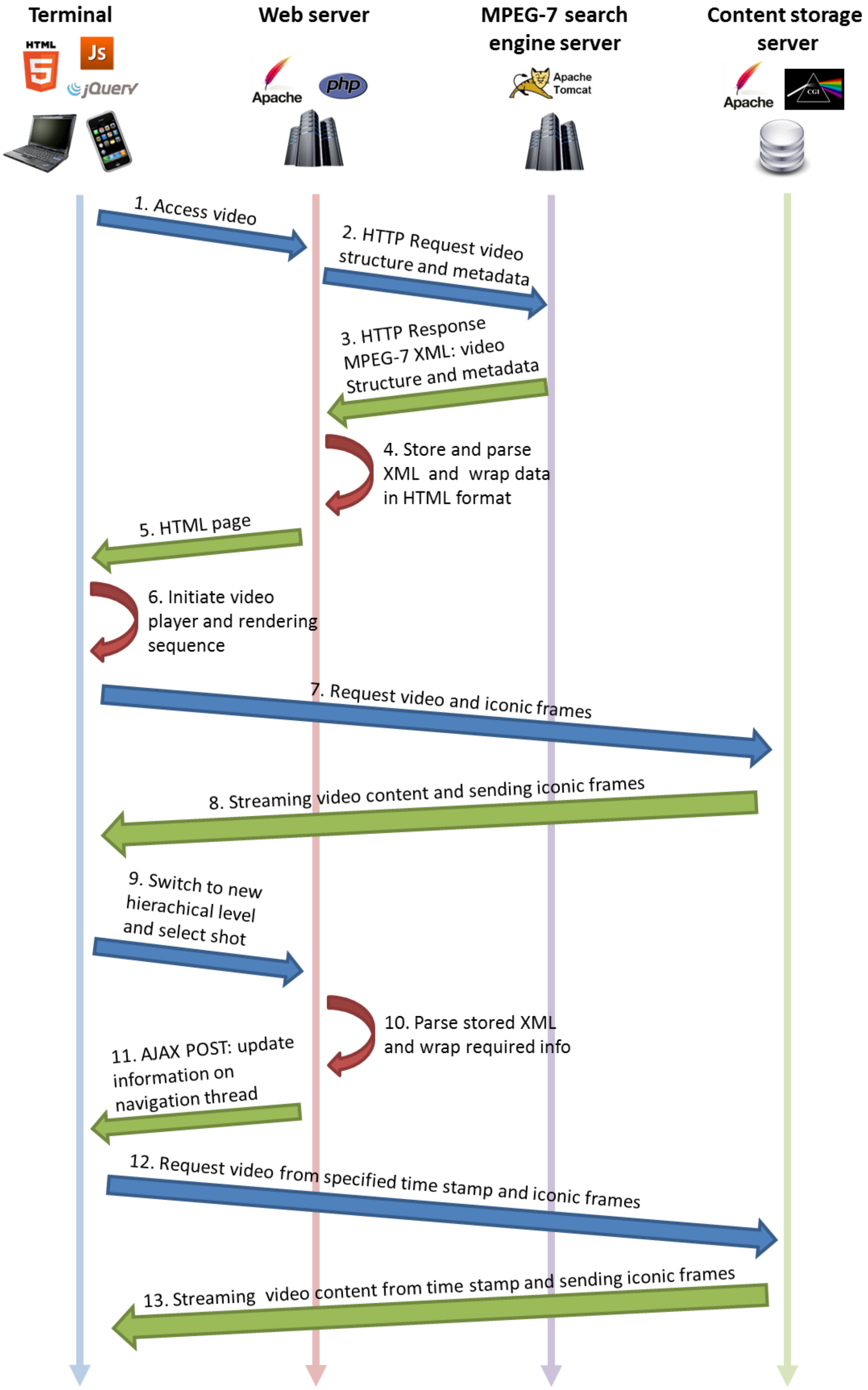

Terminal
HTML
JS
jQuery
Web server
Apache
php
MPEG-7 search engine server
Apache Tomcat
Content storage server
Apache
CGI
1. Access video
2. HTTP Request video structure and metadata
3. HTTP Response MPEG-7 XML: video Structure and metadata
4. Store and parse XML and wrap data in HTML format
5. HTML page
6. Initiate video player and rendering sequence
7. Request video and iconic frames
8. Streaming video content and sending iconic frames
9. Switch to new hierachical level and select shot
10. Parse stored XML and wrap required info
11. AJAX POST: update information on navigation thread
12. Request video from specified time stamp and iconic frames
13. Streaming video content from time stamp and sending iconic frames

Figure 8.9. Timeline of events triggered when navigating in a video on the OVIDIUS platform.

8.1.3.2 OVIDIUS finder

The *OVIDIUS finder* (Figure 8.10) is adapted for the Known Item Search task, where the main objective is to retrieve one or more specific shots/elements of interest inside a video. In this respect, OVIDIUS *finder* emphasizes and enhances the navigation functionalities.

OVIDIUS *finder* is derived from the OVIDIUS *explorer* and features the same home screen. In addition, it allows the parallel visualization of the threads of iconic representations corresponding to the different hierarchical levels of the structure of the video. While visualizing the video, the user can simultaneously skim through neighboring scenes and shots. The user can also add other navigation rows/threads of iconic representations from a different hierarchical level of the considered segmentation. One thread contains the scene previews, while the other one displays the previews of the shots corresponding to the current scene. Users can switch between different segments horizontally (on the same hierarchical level) and vertically (between different hierarchical levels). Whenever a new scene is selected, the shots thread is updated instantly with the shot previews corresponding to the selected scene. Whenever a user launches an object-based query, the search is performed in the background on the server and the user can continue his navigation while the query is processed. The query results are then displayed on a new (third) thread in decreasing order of similarity. When the user has finished skimming through the search results he can either launch a new query from the video or from one of the results or continue the navigation on the other threads. The search thread can be closed if the user does not employ it anymore.

These functionalities are illustrated in Figure 8.10. Notice the three horizontal threads and their dedicated color codes. They enable the user to get a quick overview on the video and its content, without overwhelming the user and overcharging the interface. The simultaneous visualization of different hierarchical levels allows to user to get an increased amount of information and to take quick decisions in for further navigation and browsing.

In order to speed-up the navigation in the video, the navigation of the threads can be done by using keystrokes. Skimming through neighboring shots or scenes can be performed with simple keystrokes, each thread having its own dedicated keys for the various actions considered. The functionality is illustrated in Figure 8.11. In this case the keyboard based navigation is significantly faster than the mouse based navigation, as each thread has its dedicated keys and their navigation can be made in parallel. The navigation involving only the mouse would have required the user to press the navigation arrows from the interface for each thread separately.

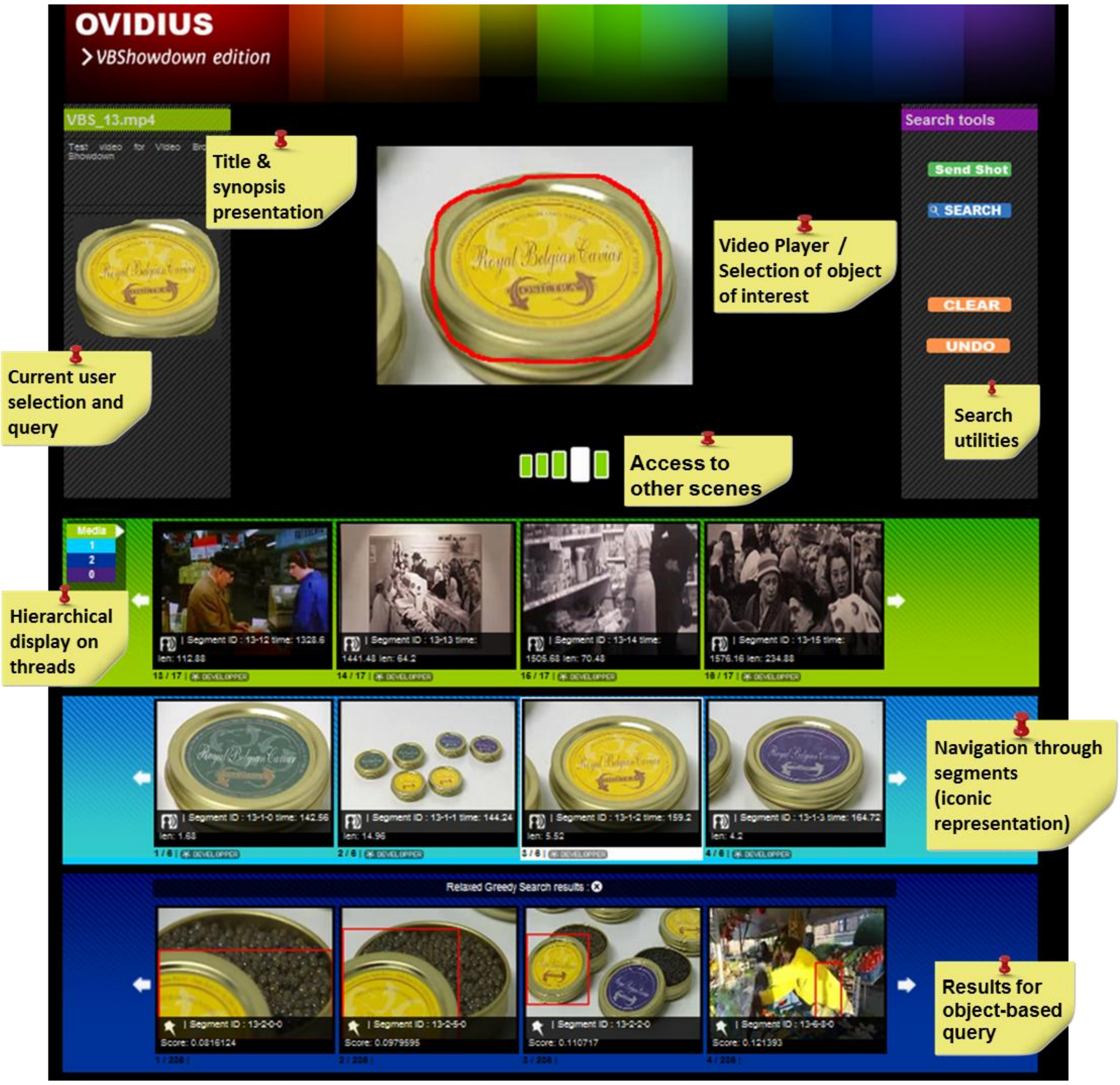

OVIDIUS
VBShowdown edition
VBS_13.mp4
Search tools
Send Shot
SEARCH
CLEAR
UNDO
Title & synopsis presentation
Video Player / Selection of object of interest
Current user selection and query
Search utilities
Access to other scenes
Hierarchical display on threads
Navigation through segments (iconic representation)
Relaxed Greedy Search results
Results for object-based query

Figure 8.10. OVIDIUS *finder* interface.

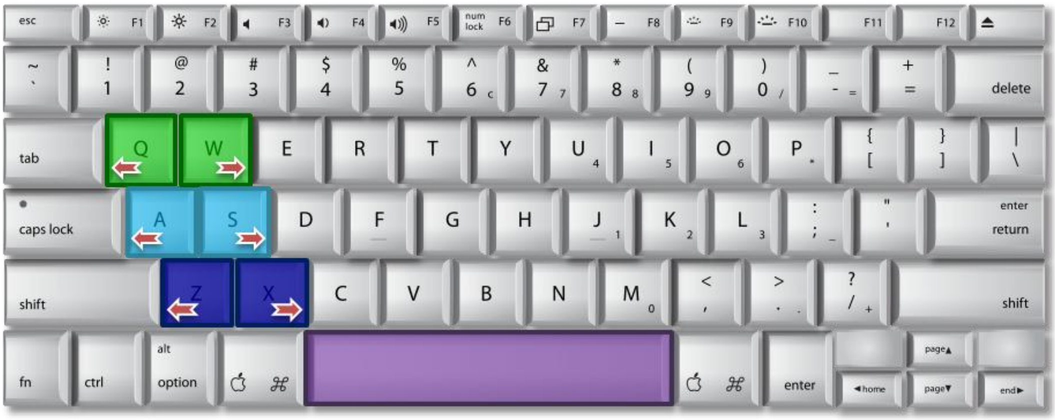


Figure 8.11. Operating the navigation on OVIDIUS *finder* GUI with the keyboard. Left-Right horizontal navigation can be performed on each thread. Each thread has its dedicated keys marked with the corresponding color code. A key press displays on the thread the next or the previous sequence of scenes/shots/keyframes. The spacebar is employed for launching a query by example search, once an object of interest has been specified.

8.1.3.3 OVIDIUS mobiler

The OVIDIUS *mobiler* (Figure 8.12) interface is adjusted for video navigation and querying from handheld devices (*e.g.*, smartphones) which usually have small screens and where the interaction is mainly performed by taping. As mentioned in Section 7.2, the development of a mobile video navigation and retrieval interface able to preserve and visualize the fine structure and information of the video, is still an open issue of research.

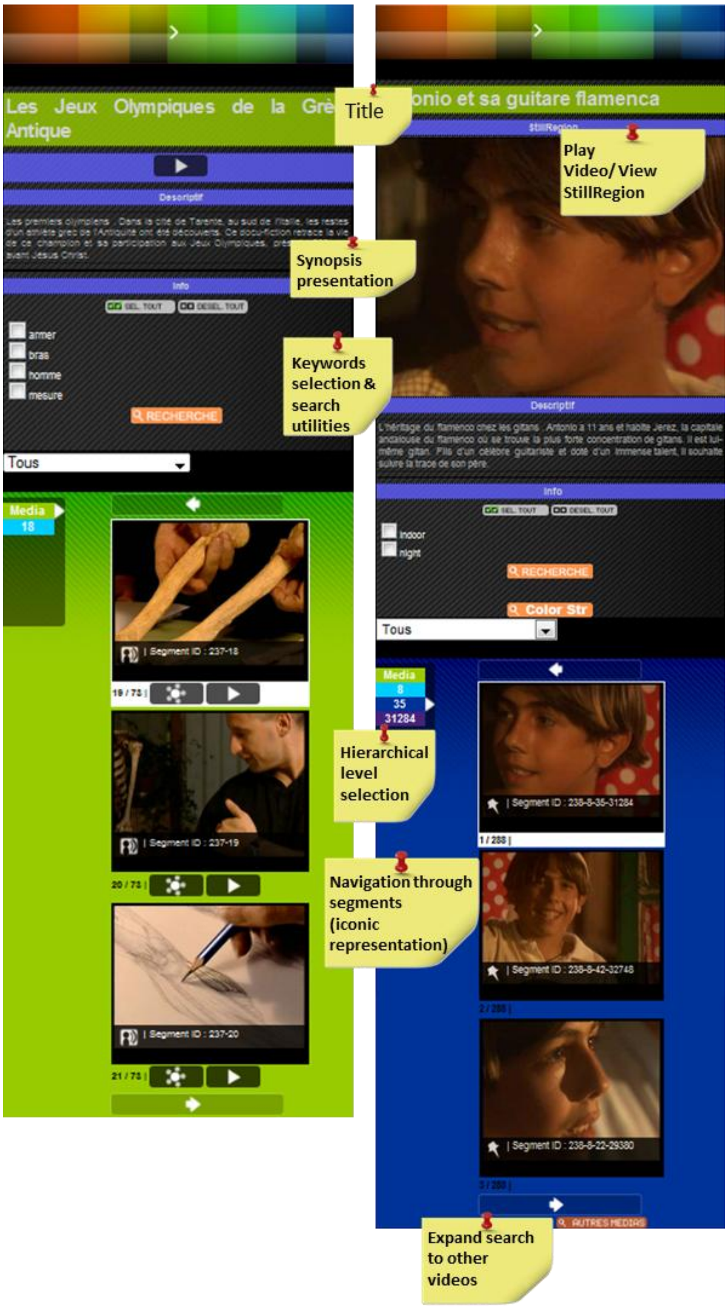
The OVIDIUS *mobiler* is derived from the OVIDIUS *explorer*. It preserves the main functionalities of its desktop counterpart for navigation (hierarchical navigation and color codes for hierarchical levels, threads of iconic images displaying the preview image, video summary, identifier, time stamps, order, type…) and querying. The various sections of the interface are here re-arranged in order to fit in the constraints imposed by the small touch screen of the handheld devices and their corresponding interaction tools (*e.g.*, taping, touch scrolling…).

The interface layout is set to a portrait mode, allowing navigation by vertical scrolling. Different levels of visualization can be achieved by zooming in and out, which is typical for most of the currently existing mobile interfaces.

Some parts of the sections (summary, keywords, video player, StillRegion) can be minimized, by taping on their header, in order to leave more space for the other sections of the interface (Figure 8.12). These sections can be restored at any moment by taping again on their header. Different instances and functionalities of OVIDIUS *mobiler* deployed on an iPhone device are illustrated in Figure 8.12

A specific feature has been added, envisioning time and bandwidth efficiency and compatibility issues. The HTML5 video player is still a rather recent technology and its video player it is still not as popular as it is on the desktops, although latest versions of both Android and IOS endorse it. For devices not yet supporting the HTML5 video element, we employ the FFMPEG library [FFMPEG 2012] that makes it possible to automatically cut a segment from the video at the user's request. Whenever a user wants to access a certain video segment, this video segment will be automatically cut from the media and sent to user. In this way, the user will receive the exact requested video segment and optimize the required bandwidth.

Most of the main video systems and portals (*e.g.*, YouTube, Vimeo, DailyMotion...) make their content available to the smartphones through applications (so-called *apps*) that need to be downloaded and installed on the smartphone. The access to the servers and the exchange of data is performed by the app. The apps fully benefit from the hardware performances of the devices, but different versions need to be developed for each type of mobile operating systems and, in some cases, even for each type of device. In contrast with them, the OVIDIUS system is available directly from the browser, whatever the OS or the type of device. Let us also mention that the OVIDIUS *mobiler* is based on a pure HTML5 and lightweight JavaScript implementation and does not use complex operations or animations that are characteristic to native app. Thus, the user experience is not affected by its web-based implementation and the platform is guaranteed to be accessible from a multitude of devices.


Les Jeux Olympiques de la Grè
Antique
Title
onio et sa guitare flamenca
Play Video/ View StillRegion
Synopsis presentation
Keywords selection & search utilities
armer
bras
homme
mesure
RECHERCHE
Tous
Media
18
Segment ID : 237-18
19 / 73
Segment ID : 237-19
20 / 73
Segment ID : 237-20
21 / 73
Descriptif
Info
indoor
night
Color Str
Media
8
35
31284
Hierarchical level selection
Navigation through segments (iconic representation)
Segment ID : 238-8-35-31284
1 / 288
Segment ID : 238-8-42-32748
Segment ID : 238-8-22-29380
AUTRES MEDIAS
Expand search to other videos

a) b)

Figure 8.12. Components and different instances of the OVIDIUS mobiler interface: (a) browsing through scenes of a video; (b) browsing through results of a query by ColorStructure descriptor at the still regions level.

8.1.3.4 *Search functionalities*

The search engine is a core element of any video navigation and retrieval platform. In the field of visual search, the queries can be formulated in various manners, including both textual and content-based (*i.e.*, queries by example). In addition, there is no generic search method that can fulfill all user requirements. Often, the techniques and descriptors need to be adapted to the query. OVIDIUS integrates multiple search functionalities. In the same time, we have paid a specific attention to not overwhelming the user in terms of required interaction and parameters to be set.

Three different types of querying functionalities with corresponding search strategies are proposed:

- Query-by-keywords,
- Query-by-image,
- Query-by-region.

The user has the possibility of launching a query either in the whole video dataset or just in the current video, depending on his use case and intentions. Let us also note that for all the descriptors considered, dedicated XML schema representations have been elaborated and used to enrich the MPEG-7 schema definition.

The proposed querying functionalities are detailed in the next sections.

8.1.3.5 Query-by-keywords

Ever since web search engines such as Google [Google 2012a] emerged to the general public usage, textual queries have become the mean of choice for retrieving any type of content (documents, images, videos) on the web, in document archives or computers. Users are today accustomed to find and use such functionalities on any system. The OVIDIUS platform features text based search functionalities adapted for each hierarchical level of the video tree (*cf.* Section 8.1.1).

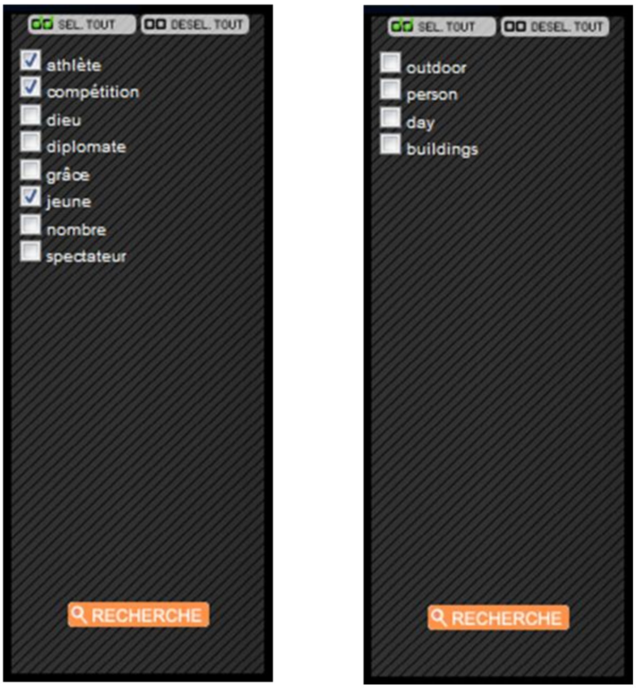
In order to illustrate the textual functionalities supported, we have considered corpus from the Médi@TIC project [Mediatic 2010], consisting of 16 television documentaries and 45 radio shows. The EPG information (title, synopsis) and the keywords globally describing the content have been provided by the authors. Other textual metadata have been derived from the text analysis of the transcriptions segments, represented as keywords and associated with their corresponding MPEG-7 segments. The speech-to-text tools and the textual analysis have been provided within the framework of the Médi@TIC project by CEA-LIST. Details about the corresponding technologies can be found in [Delezoide 2008]. Our work consisted in integrating the textual metadata within the corresponding MPEG-7 segments. A semantic description of the visual content of the considered keyframes was also provided. Here, for each concept considered, (*e.g.*, indoor, outdoor, person, day, night, vegetation …), a score representing its probability of occurrence has been indicated. The set of concepts retained is the one used for the ImageClef CVDT evaluation [IClef 2008]. The probability of occurrence has been determined with the help of Fast Shared Boosting [Le Borgne 2010] in a first stage, followed by joint classification Bayesian model [Delezoide 2008].

A *bag of words* approach, combined with a *tf-idf* weighting mechanism [Salton 1988, Sparck 2000] (*cf.* Section 4.4.1) have been employed for performing queries. The cosine correlation measure has been adopted as similarity measure. The *tf-idf* scheme offers a simple yet efficient algorithm for matching textual queries with concepts included in the description.

The queries can be performed at any hierarchical level within a given video. The search process can be applied either globally, at the level of the entire video database (inter-video search), or locally, within the current video (intra-video search), depending on the user's purposes and of the considered application.

For each video segment and at each hierarchical level, OVIDIUS displays the list of available keywords. The user can select a sub-set of them through checkboxes (Figure 8.14) and launch a query.

Let us also note that a free-text keyword specification is also supported, the user having the possibility to introduce a set of personalized tags. In all cases, the search is performed based on the existing keywords of each MPEG-7 segment. Such keywords are naturally included in the description at the segment level. In the case of global queries, performed at the level of the root video segment, the keywords considered are the words included in the title and synopsis.



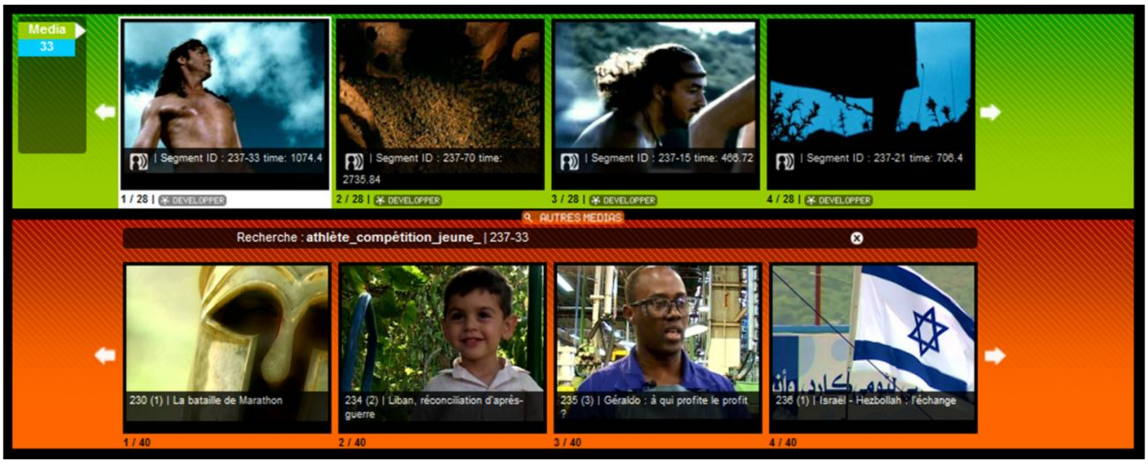
☑ athlète
☑ compétition
□ dieu
□ diplomate
□ grâce
☑ jeune
□ nombre
□ spectateur

□ outdoor
□ person
□ day
□ buildings

Figure 8.13. Keywords panel. Left: keywords extracted from the audio transcript of a video scene are illustrated. User can select with a checkbox the words he wants to use for querying. Right: Keywords describing concepts detected in a key-frame. Concepts can be selected and used for composing a new query.

Figure 8.14 illustrates the results of a textual query. The search is performed at the level of granularity of the query. For example, a query launched at the scene level will retrieve video scenes, while a query at the shot level will retrieve shots. For better structuring and visualization of results we employ the same color code. In addition, we represent results from other videos on a dedicated thread with orange background. The results from other videos are grouped per video and only a representative frame of the video is illustrated. The retrieved video segments can be then accessed by clicking on this representative frame.

The limitations of keyword-based search are well-known: availability of reliable annotations, linguistic barriers, subjectivity, ambiguity…The second type of search paradigm considered and integrated within OVIDIUS concerns content-based representations and query-by-example approaches.



Recherche : athlète_compétition_jeune_ | 237-33

Media
33
| Segment ID : 237-33 time: 1074.4
1 / 28 | DEVELOPPER
| Segment ID : 237-70 time: 2735.84
2 / 28 | DEVELOPPER
| Segment ID : 237-15 time: 466.72
3 / 28 | DEVELOPPER
| Segment ID : 237-21 time: 706.4
4 / 28 | DEVELOPPER
AUTRES MEDIAS
230 (1) | La bataille de Marathon
1 / 40
234 (2) | Liban, réconciliation d'après-guerre
2 / 40
235 (3) | Géraldo : à qui profite le profit ?
3 / 40
236 (1) | Israël - Hezbollah : l'échange
4 / 40

Figure 8.14. Results displayed after a query by keywords selected in Figure 8.13 (*athlète*, *compétition*, *jeune*) within the scenes of the current video content (upper row) and within all the other contents available in the data base (lower row)

8.1.3.6 Query-by-example

Query-by-example is increasing in popularity among users of various visual search systems (*e.g.*, Google Images [Google 2012c], TinEye [TinEye 2012], kooaba [Kooaba 2012]). In this case, the principle consists of considering exclusively the low-level visual descriptors associated with the content ant to exploit their corresponding similarity measures. Let us underline that such an approach is completely agnostic of the semantics of the content and exploits solely the perceptual characteristics.

Concerning the visual descriptors considered, OVIDIUS can exploit arbitrary MPEG-7 visual descriptors since one of the extraction engine adopted is based on the MPEG-7 reference software. In the current query-by-image context, we have considered solely the color and texture descriptors that can be associated with a global still image.

Figure 8.15 presents an example of a query performed with the help of the MPEG-7 ColorStructure descriptor. As it can be noted, the Color Structure descriptor is highly effective in retrieving instances of the same person and of the same environment.

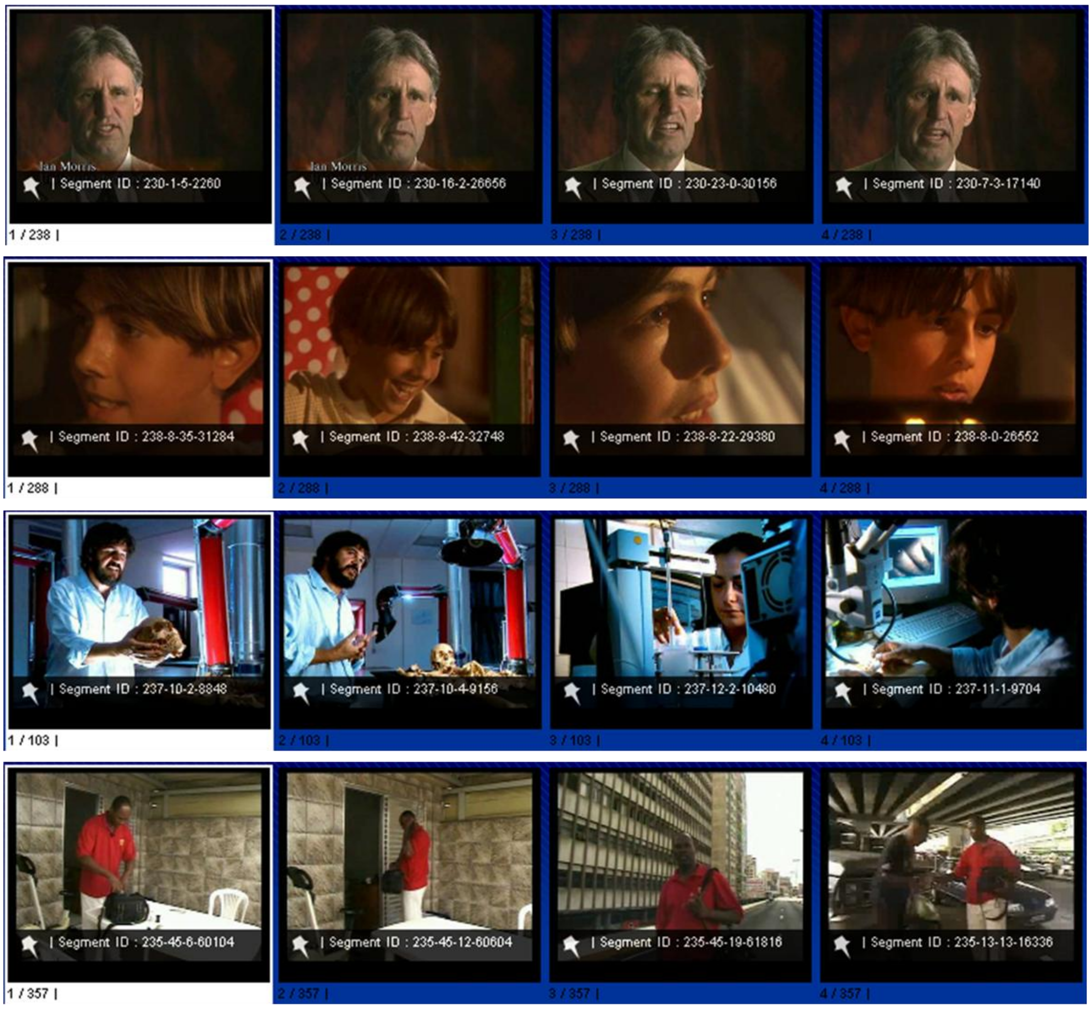

Jan Morris
| Segment ID : 230-1-5-2260
1 / 238 |
Jan Morris
| Segment ID : 230-16-2-26656
2 / 238 |
| Segment ID : 230-23-0-30156
3 / 238 |
| Segment ID : 230-7-3-17140
4 / 238 |
| Segment ID : 238-8-35-31284
1 / 288 |
| Segment ID : 238-8-42-32748
2 / 288 |
| Segment ID : 238-8-22-29380
3 / 288 |
| Segment ID : 238-8-0-26552
4 / 288 |
| Segment ID : 237-10-2-8848
1 / 103 |
| Segment ID : 237-10-4-9156
2 / 103 |
| Segment ID : 237-12-2-10480
3 / 103 |
| Segment ID : 237-11-1-9704
4 / 103 |
| Segment ID : 235-45-6-60104
1 / 357 |
| Segment ID : 235-45-12-60604
2 / 357 |
| Segment ID : 235-45-19-61816
3 / 357 |
| Segment ID : 235-13-13-16336
4 / 357 |

Figure 8.15. Search results based on MPEG-7 ColorStructure descriptor queries by example (first image of each row is the query image).

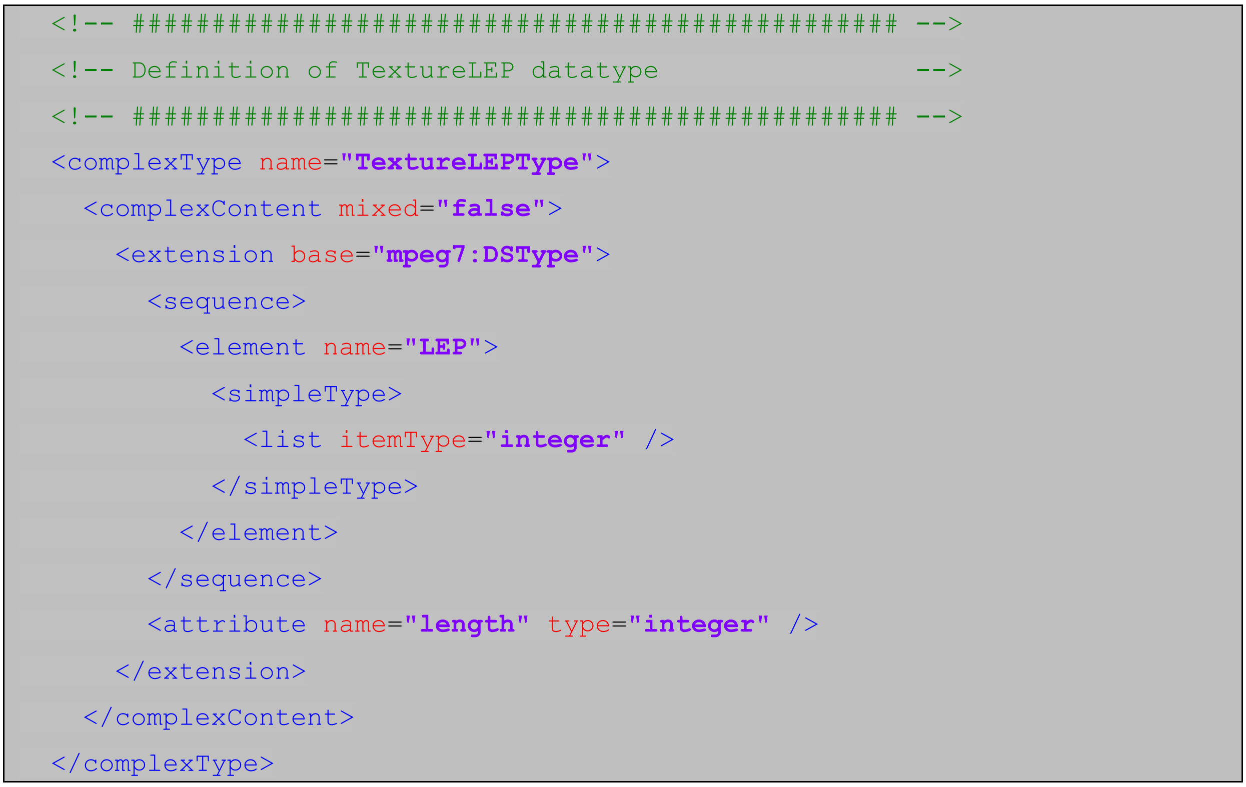
Let us also note that other descriptors, outside MPEG-7, can be easily integrated within OVIDIUS. In order to achieve this goal, the following steps need to be accomplished:

- Update of the MPEG-7 Schema with the syntax and semantics of the new descriptor to be integrated,
- Activation of the communication protocol between and server engine with respect to the new descriptor,
- Update of the existing MPEG-7 description associated with the considered videos in the database,
- Integration of the corresponding similarity measure at the level of the search engine server.

In order to validate such an approach, we have considered the TextureLEP descriptor [Cheng 2003], exploited within the framework of the Médi@TIC project and developed by partners from CEA-LIST.

The adopted XML specification of the TextureLEP descriptor is presented in Figure 8.16:

A third query paradigm concerns the object-based retrieval, presented in the following section.



```
<!-- ################################################ -->
<!-- Definition of TextureLEP datatype                -->
<!-- ################################################ -->
<complexType name="TextureLEPType">
  <complexContent mixed="false">
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="LEP">
          <simpleType>
            <list itemType="integer" />
          </simpleType>
        </element>
      </sequence>
      <attribute name="length" type="integer" />
    </extension>
  </complexContent>
</complexType>
```

Figure 8.16. XML specification of the TextureLEP descriptor.

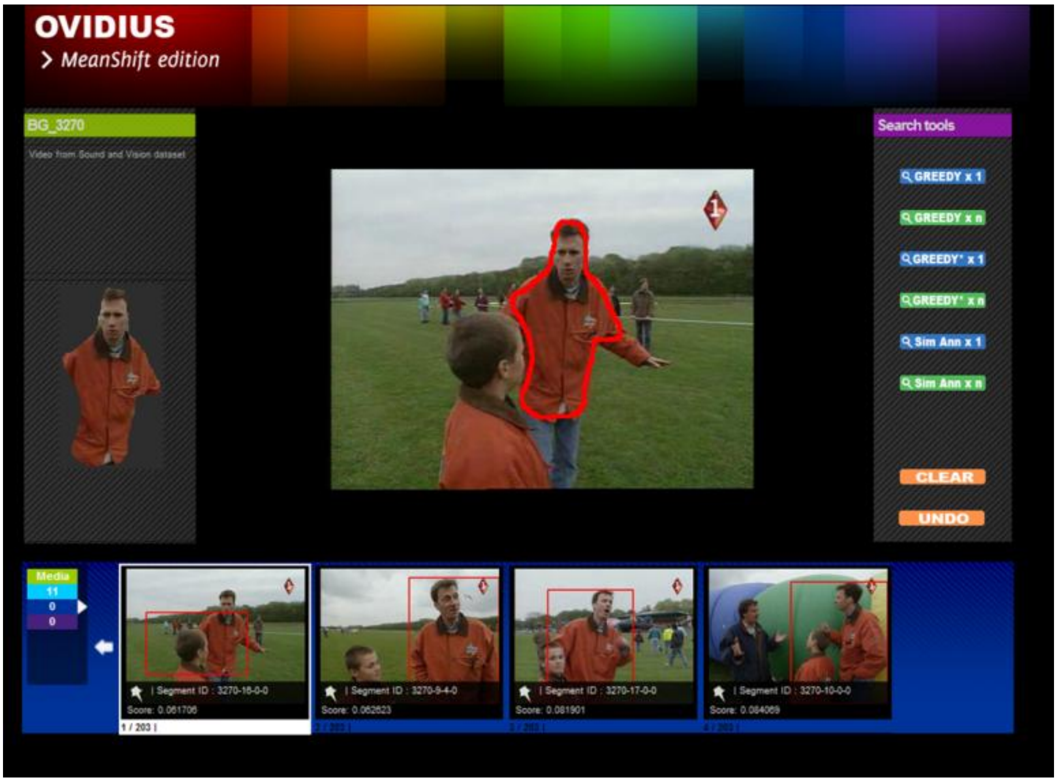
8.1.3.7 Query-by-region

In this case, only a part of an image representing a pattern, an object or a group of objects is considered as a query for retrieving other videos containing different instances of the entity of interest. Here, OVIDIUS integrates the DOOR framework introduced in part 1 of this manuscript in order to provide the video object instance retrieval functionality in an on-line environment.

The main advantage of the object-based approach is that the user can directly access specific shots of the video containing the object of interest. Usually, a text based query on a general public video platform (*e.g.*, YouTube, DailyMotion, Vimeo) retrieves a collection of videos according to the user specification. Yet, no information about the localization of the object in the video is provided and users have to skim through the video in order to retrieve the item or object of interest. In our approach, this step is discarded as the results of a query-by-region search lead to the exact shot/moment in the video where the item of interest has been encountered.

In order to support the object-based retrieval functionality, the only functional constraint that needs to be satisfied concerns the interactivity issues. Thus, the user should be allowed to interactively specify a region of interest, in order to formulate a query of arbitrary shape. The adopted solution is based on a HTML5 drawing/selection tool, which gives to the user the possibility to select and define interactively a region or object of interest in the video. The user selection is then used as query and represents the input data of the DOOR method.

Figure 8.17 illustrates a typical query-by-region case. The user selects with a red marker the object of interest (*i.e.*, man with orange jacket) and launches a query in the video. The user selection is then displayed on the right. The representative key-frames of the relevant video shots determined are displayed in decreasing order of similarity to the query on a dedicated thread. The object candidates retrieved are represented with red bounding boxes on the preview frames along with their similarity score.



| Segment ID : 3270-18-0-0
Score: 0.061706
1 / 203 |

| Segment ID : 3270-9-4-0
Score: 0.062623
2 / 203 |

| Segment ID : 3270-17-0-0
Score: 0.081901
3 / 203 |

| Segment ID : 3270-10-0-0
Score: 0.084069
4 / 203 |

Figure 8.17. The DOOR method used for an object-based query using with the OVIDIUS platform.

This completes the description of the three different search functionalities supported by OVIDIUS. In order to provide some hints on how the OVIDIUS platform can be used in practice, we have developed several use-cases, presented in the following section.

8.2 OVIDIUS use-cases

In order to better emphasize the possible applications of the OVIDIUS platform let us outline three use case scenarios. The first use-case concerns the mobility framework and is addressed with the help of OVIDIUS *mobiler* (*cf.* Section 8.1.3.3). This use-case has been presented in [Bursuc 2010].

8.2.1 Mobile video retrieval and consumption

A person using a smartphone connected to a 3G/4G or wireless network is on his way home with a train. He wants to watch a small documentary or take advantage of the time spent on travelling in order to find a movie to watch when getting home. The user can start his search by typing some keywords describing the desired content. OVIDIUS *mobiler* (Figure 8.12; Figure 8.18) matches then these keywords with the ones from the EPG of the videos in the database and returns a collection of related videos.

At this stage, the user can visualize for each video a representative image, the title and a short summary added by the creators of the content. After selecting one of these videos, OVIDIUS displays the intra- video navigation interface. The user can now navigate through the video horizontally and view the scenes along with a representative iconic image and descriptive keywords. Also a vertical in-depth navigation is possible, by accessing the hierarchical structure of the scenes. Shots can be visualized in a similar manner. With a few clicks the user has general idea over the considered video content.

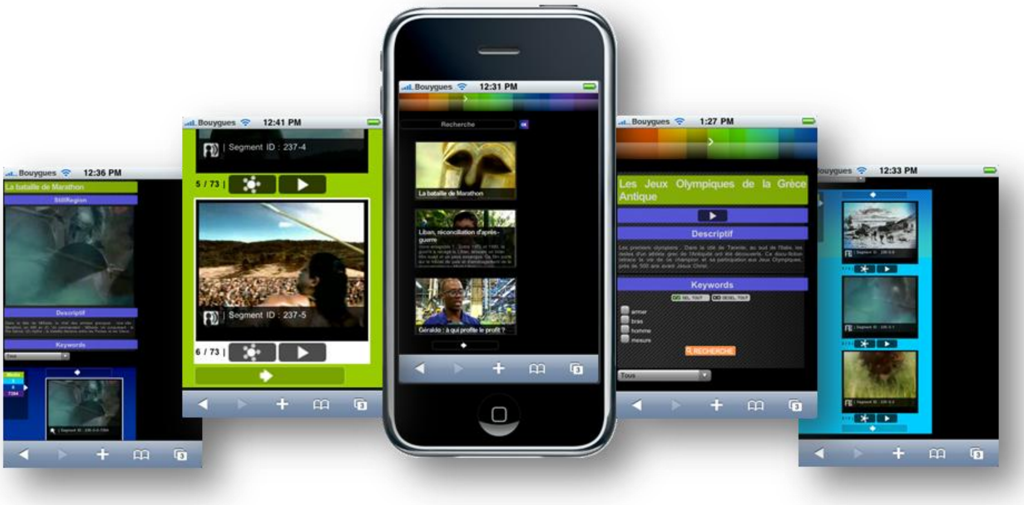


Figure 8.18. Screen captures of the OVIDIUS *mobiler* on an iPhone smartphone

Moreover, if he would like to group similar scenes and shots or to find similar content within the other videos in the database, he can choose a sample scene and request a search by a set of keywords of its choice. The user can then access the results (*i.e.*, scenes/shots retrieved and presented in decreasing order of similarity with the query) in order to verify whether they are pertinent or not.

The advantage of the OVIDIUS approach is that solely short scenes/shots of interest are returned and the user can rapidly check their content. Time and bandwidth are saved in this manner.

Nevertheless, if the user prefers some specific shots in the video (*e.g.*, war scenes or a certain anchor person in a news bulletin) he can use the visual search. Thanks to the MPEG-7 ColorStructure descriptor similar shots are made available to the user after a query by example. In this case, supplementary navigation inside the video is no longer necessary, the system being able to directly locate the desired elements of interest. This proves to be very effective in a mobile context when different hardware and connection constraints make complex operations were difficult.

A second use-case scenario concerns the know item search paradigm.

8.2.2 Known Item Search in videos

When a user needs to find and access a specific video segment in a video, having just a general idea of the content of the segment, he can employ OVIDIUS *finder*. This is a standard scenario for Known Item Search tasks and competitions [Schoeffmann 2012b].

Given a short video sequence to find in a current video, the user can start browsing the scenes and shots looking for the query video or for objects or textures similar with the ones in the query (*e.g.*, the T-shirts of the children in the example presented in Figure 8.19). Let us note that the navigation by keystrokes significantly speeds up the visualization of the video.

Once such a frame is found, the user can select the area of interest and ask the system to retrieve it. Alternatively, when the user wants to use all the information in the current frame for querying, he can perform a query-by-image using the MPEG-7 ColorStructure descriptor (*cf.* Section 8.1.3.4). When the query processing ends, key-frames from different moments of the video, containing similar patterns or objects can be accessed instantly or can be used as additional queries to narrow the area of search.

This functionality has been validated in the Video Browser Showdown competition [Schoeffmann 2012b, Bursuc 2012] with good results (4th position out of 12 participants). The scenario for the Video Browser Showdown was in the KIS task spirit. Given a short video clip presented on a shared screen that is visible to all participants, after the video clip has stopped playing the participants have used their own equipment to perform an interactive search in the specified video file to retrieve the sequence of video shots that have been just displayed. The performance of the participating systems has been evaluated in terms of successful answers and average search time. The participants were allowed to perform any content analysis that supported interactive browsing in the video (e.g., through novel content visualization, content clustering, or advanced seeker-bars etc.). The search process was supposed to be interactive without any text queries allowed. The systems have been evaluated in an *expert run*, where the developers of the systems acted as searchers, and in *novice run*, where the searchers were volunteers from the audience.

This scenario mimicked a real-life use case when users disposing of a collection of non-indexed video content, wish to retrieve a specific video sequence quickly, without the need to watch all the videos.

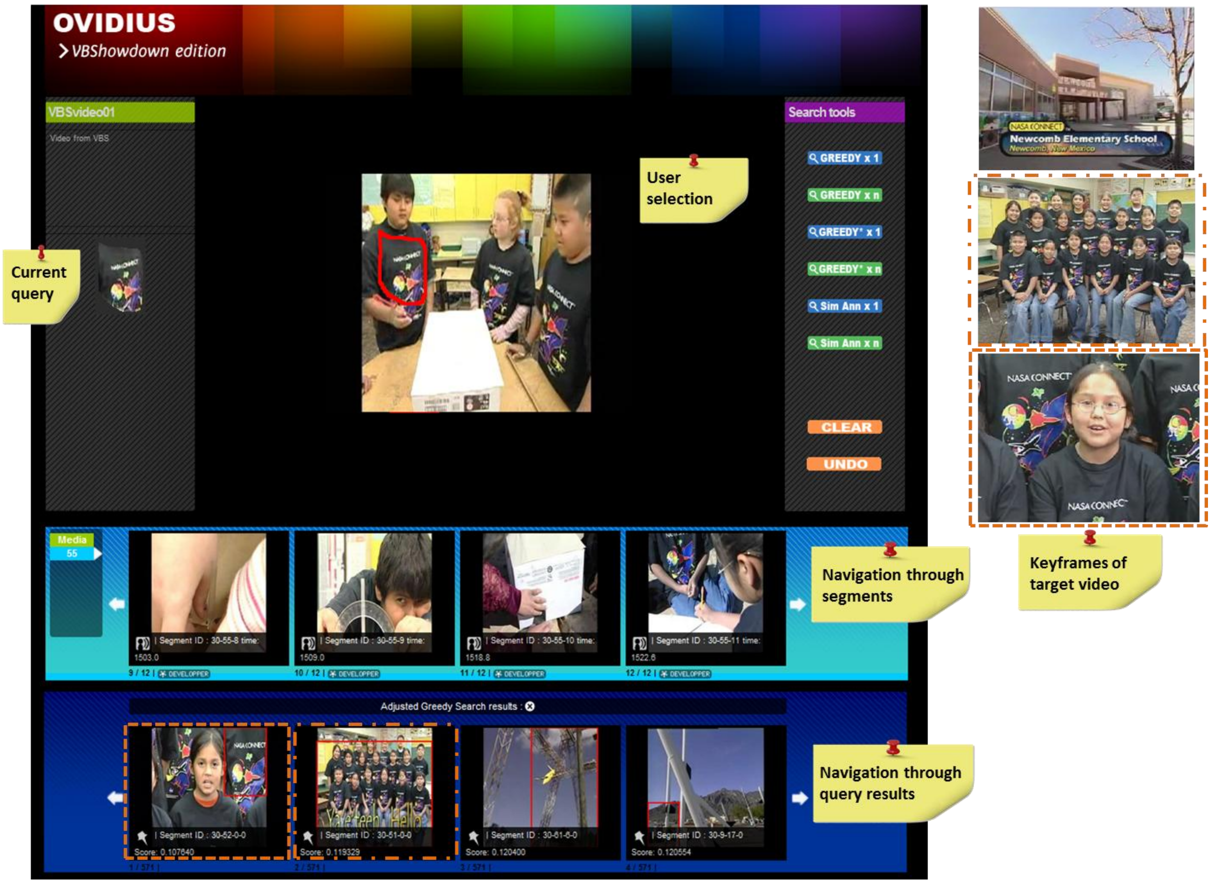


Figure 8.19. OVIDIUS finder use for the Known Item Search task

The final use-case considered concerns rather professional environments and is about semi-automatic video annotation.

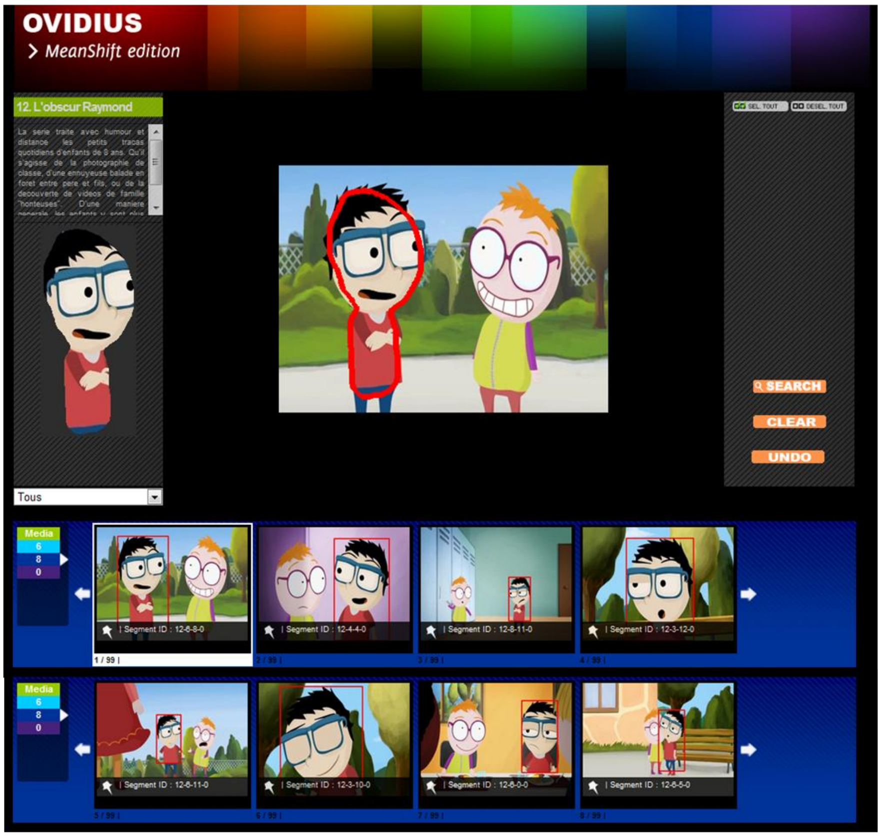
8.2.3 Semi-automatic video annotation

Professional content editors, producers and documentalists dispose of large repositories of content that need to be indexed and organized. In most of the cases, this process is done manually by the editors themselves. Elaborating tools able to at least partially automate this process would be of outmost interest

However, because of the fine description that is usually required (*e.g.* character name, description, context…), a complete automatic annotation of the content is a highly difficult challenge. Existing methods involve a learning mechanism and the development of a ground truth dataset every time a new element of content is created and added to the repository.

An alternative solution would be to leverage on both the experience of professional and the effectiveness of an object-based retrieval approach such as the one proposed by the DOOR framework. Editors can perform visual queries in the dataset for retrieving different instances of user-specified objects. As most of the first results are positive (*cf.* Section 6.3), the editor can apply the same tags to all of the first results, thus propagating the annotations to the corresponding video shots. The tags are written directly to the MPEG-7 XML Schema, weighted with the *tf-idf* method and become available for other users accessing the server. Textual queries can be then performed based on these tags for retrieving the same video shots for further re-use.

This functionality is illustrated in Figure 8.20. The OVIDIUS *explorer* can be updated to display an additional thread of results for a better visualization of the retrieved video shots. We can observe that the character selected by the user has been retrieved in various instances in all the first 8 results. The user can now associate the same tags (*e.g.*, boy, character's name, dark hair, blue glasses, red shirt) to all of these shots. From now on, such shots can be retrieved by using one or more of the associated tags.



12. L'obscur Raymond

La serie traite avec humour et distance les petits tracas quotidiens d'enfants de 8 ans. Qu'il s'agisse de la photographie de classe, d'une ennuyeuse balade en foret entre pere et fils, ou de la decouverte de videos de famille "honteuses". D'une maniere generale, les enfants y sont plus

Figure 8.20. The results of a query on a cartoon dataset using the DOOR method on the OVIDIUS platform. The retrieved results are displayed on 2 threads and ordered in decreasing similarity with the query.

Figure 8.21 presents some examples of different queries and retrieval results, in the case of a cartoon dataset, specified within the framework of the HD3D-IIO project [HD3D 2011].

In the same spirit of automatic content annotation, we propose an additional functionality of the OVIDIUS platform, which concerns the automatic hierarchical decomposition of the videos in scenes, shots and key-frames. In this respect, we employ the video temporal and spatial-temporal segmentation/structuring framework proposed in [Tapu 2011]. We allow the users to upload their content and have it decomposed spatial-temporally for enhanced navigation. Figure 8.22 illustrates the video uploading functionality. The user's video collection is shown on the home screen in the upper part, while the lower part contains a section for videos to be uploaded. The user can simply drag and drop a video from his PC in the dedicated page section and have his content uploaded to the Content storage server, transcoded and further processed.

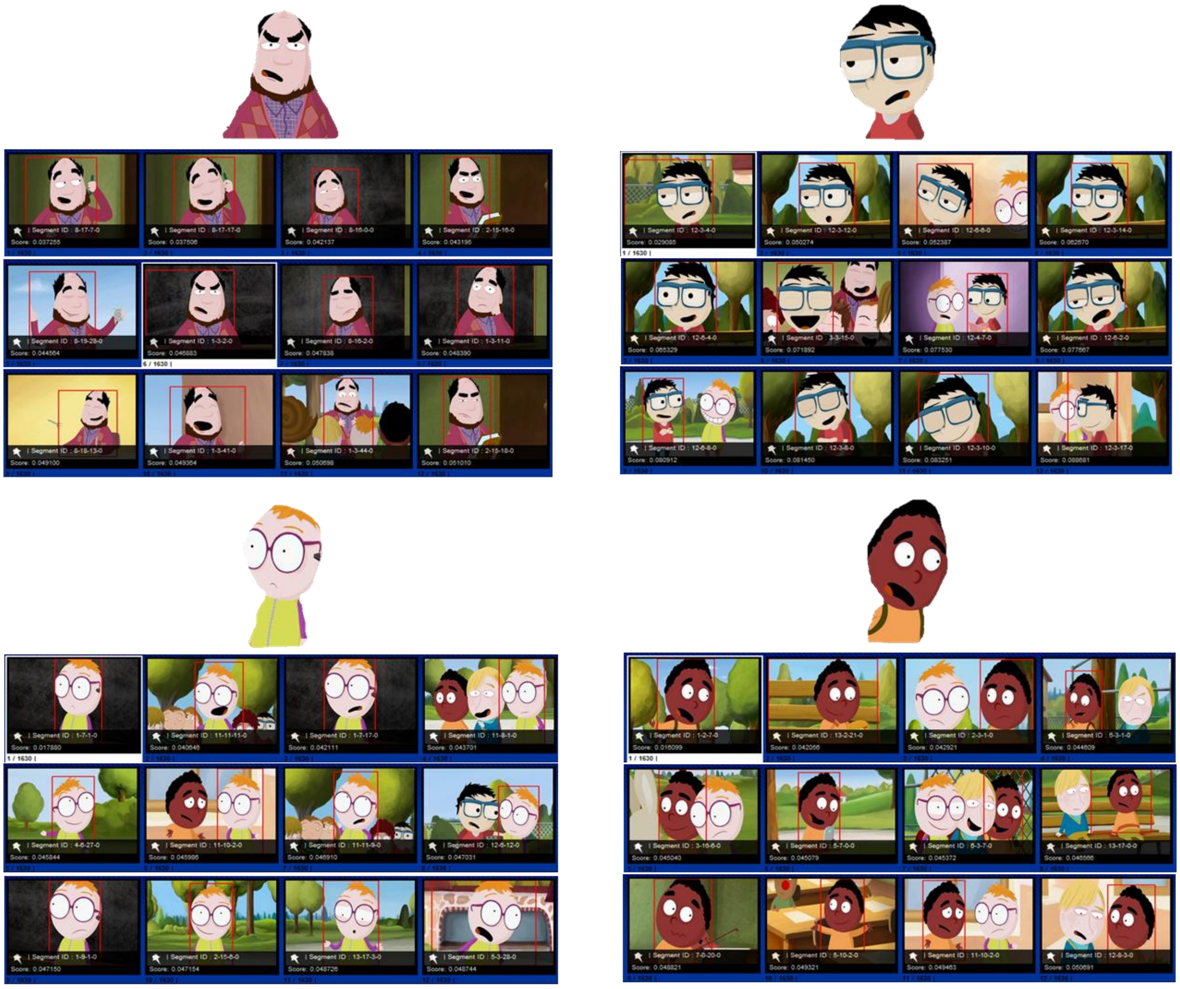


Figure 8.21. Results for different cartoon character queries.

Once the *upload* effected, the video is transcoded and the user is redirected to the OVIDIUS *explorer* navigation screen (Figure 8.23). Here the user can view his video online, while it is being processed. The evolution of the video decomposition sequence is illustrated in orange thread from Figure 8.23, where frames from the shots that have been just detected are displayed. The decomposition method takes less than the duration of the video (less than half of the duration of the video) and the displayed frames are ahead of the video player. After the processing has been concluded, the user can navigate in his video using the freshly computed structure and annotate it with tags as described above.

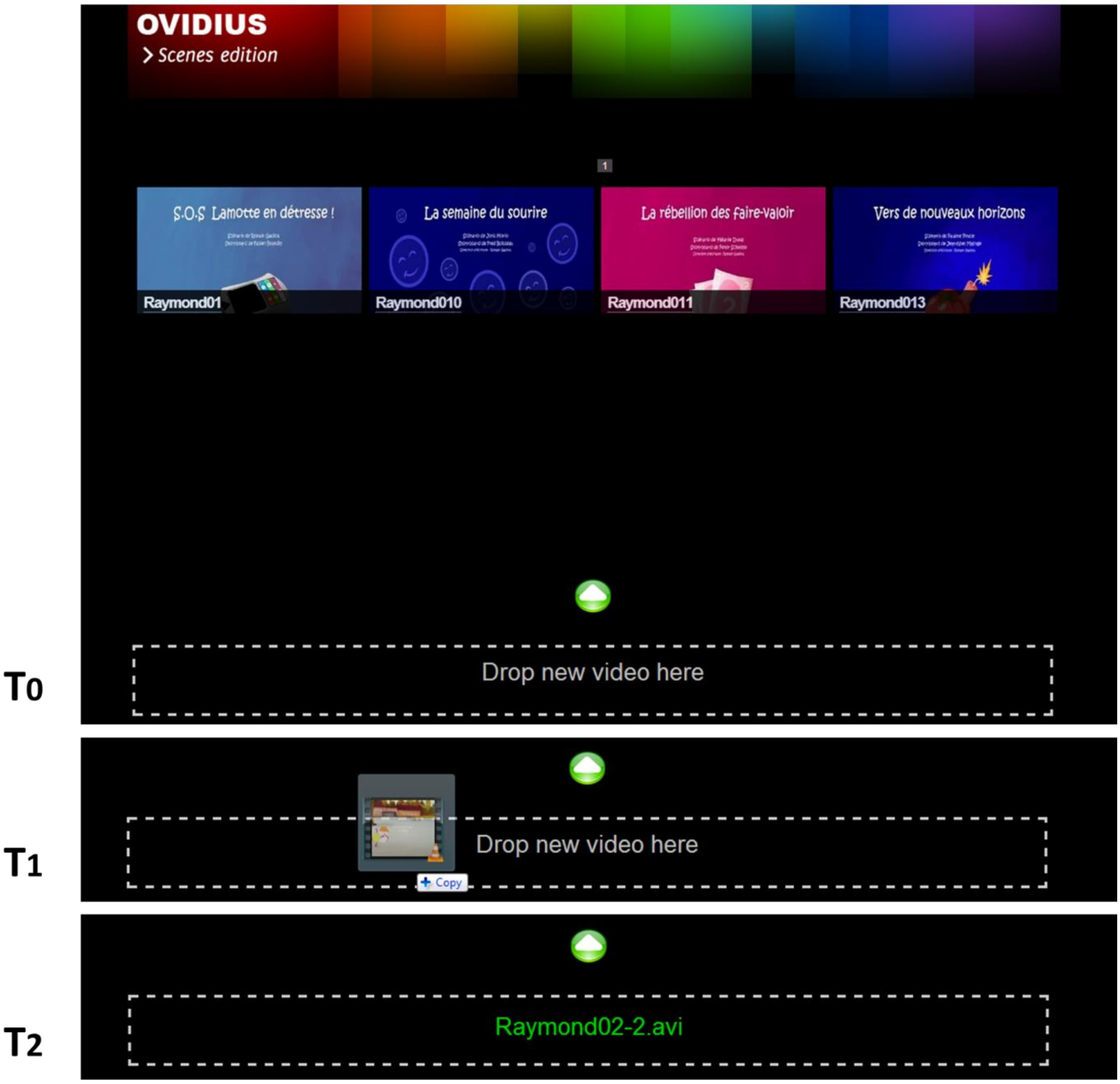


Figure 8.22. Video uploading on OVIDIUS. Users can simply drag and drop their content in a dedicate section on the web page. Once the video is dropped the user can press the upload green button and have their content decomposed in a hierarchical structure.

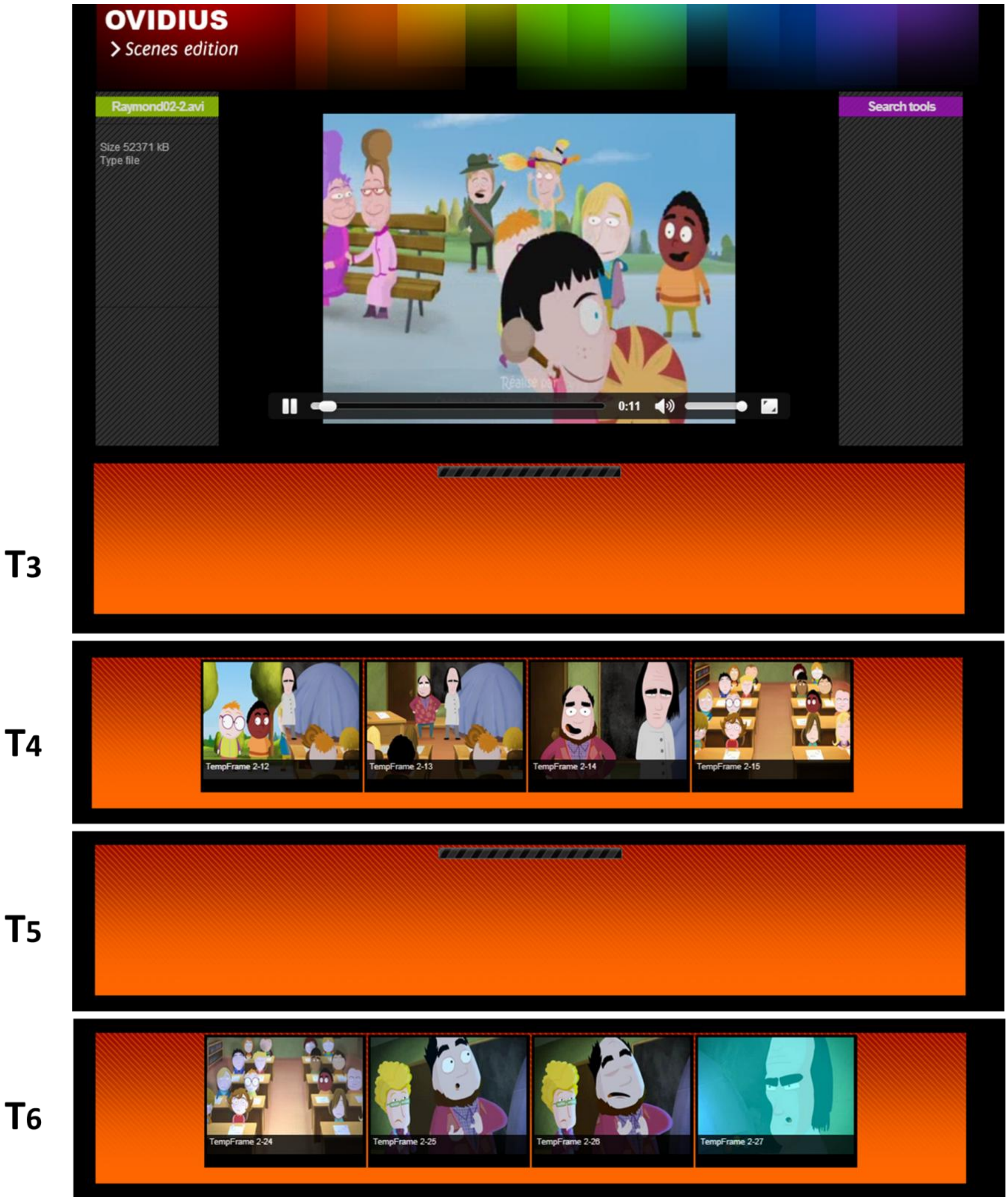


Figure 8.23. Online video hierarchical the composition. While the uploaded can be watched on the platform it is structured in scenes, shots and key-frames. The shots which have been just computed are illustrated in the orange thread. The video structuring technique is faster than the video rendering and the content is processed in less than half of its duration.

8.3 Conclusion and perspectives

This chapter introduced the OVIDIUS (*On-Line VIDeo Indexing Universal System*) platform for video content navigation and retrieval from both desktop and mobile in a web environment. We showed the advantages of a web system with a modular architecture relative to scalability and maintenance aspects.

The feasibility and the benefits of the HTML5 implementation considered have been also discussed. The MPEG-7 description schemes have been here successfully employed. They ensure the compatibility the different modules considered in the system. In addition, they provide a basis for facilitating the upgrades of the system.

The DOOR framework described in Part I of this manuscript has been seamlessly integrated in the OVIDIUS platform ensuring the selection and retrieval of user-defined video objects in an on-line database.

Three different user interfaces (OVIDIUS *explorer*, *finder* and *mobiler*), in order to cover different use cases of interest. A set of use case scenarios has also been presented.

The perspectives concern the development of a tablet adapted interface which would allow touch and drag interactions and display additional content on the screen. Additional functionalities can be integrated in the video player in order to display on the video bounding boxes of recognized object entities, as the video plays. Moreover, such functionality will then enhance the interactivity of the video and users could simply click on object to search it or to have displayed additional information on it. In a setting similar with the one of the PopcornJS [PopcornJS 2012] framework, information specific to each sequence can be displayed in dedicated sections of the interface and updated as the video plays. Such information could concern the concept keywords, the audio transcript, the objects and characters in the video sequence.

Concerning the annotation functionalities, different drawing tools can be integrated in order to allow the user to accurately annotate regions of the frames, similarly with LabelMe [Russell 200] and LabelMe video [Yuen 2009] framework, yet in an open HTML5 implementation. In addition, a relevance feedback tool can be integrated in order to let the user update or modify the automatic labelling of the video.

9. Conclusion and perspectives

9.1 Conclusion

In this thesis we have tackled the issue of retrieving different instances of an object of interest within a video collection. We have approached the issue from two different perspectives. The first one concerned the representation of an object and the search mechanism that could exploit these representations in order to retrieve relevant object instances in video content. The second one concerned the support offered for video retrieval, namely the video navigation and retrieval interface and its underlying architecture.

In the first part of our thesis we approach the issue of visual object search and retrieval. After an analysis of the state of the art we identify the existing challenges in object instances retrieval and propose a methodological framework so-called DOOR (*Dynamic Object Oriented Retrieval*) exploiting a semi-global image representation obtained with the help of an over segmentation of the frames. The advantage of region-based approaches comes from the possibility of directly exploiting the connectivity information (*i.e.*, adjacency between regions), which can be highly useful in the matching stage. In order to overcome the challenges related with the identification of sub-graphs as object entities, we proposed four optimization strategies: Greedy, Relaxed Greedy, Simulated Annealing and GraphCut.

In the following, we have considered the interest point representations and proposed a comprehensive study and analysis of existing methods, detailing the existing interest point detectors, interest point descriptors, and related clustering approaches. Based on this analysis, we have retained a reference interest point technique that has been used as baseline in our experiments for benchmarking purposes. Moreover, we have enriched the Bag of Words representation with a query definition and expansion mechanism based on a multi-modal, text-image-video principle.

Finally, we have introduced a novel hybrid representation which integrates both a region based approach and a set of interest points. The similarity between two images is achieved with the help of a spectral matching technique integrating both colorimetric and interest points descriptors.

The proposed methods have been evaluated and compared experimentally on four different datasets under different experimental settings with various color spaces and segmentation methods.

In the second part of the thesis we have introduced an online video browsing and retrieval platform (OVIDIUS – *On-Line VIDeo Indexing Universal System*).

First we have reviewed the numerous video navigation and retrieval platforms developed in the last years and elaborated a set of criteria for designing an effective system. Then, we have introduced the proposed OVIDIUS platform. The major advantage of the proposed system concerns its modular architecture which makes it possible to deploy the system on various terminals (both fixed and mobile), independently of the exploitation systems involved. The choice of the technologies employed for each composing module of the platform has been argued in comparison with other technological options. Finally different scenarios and use cases for the OVIDIUS platform have been presented.

9.2 Perspectives

The perspectives of future work are numerous.

9.2.1 Region-based representation

Future work for the region based representation concerns the enrichment of the segment descriptors. The dominant color representation is sensible to multiple intensity variations that typically occur during videos and other local descriptors can be considered to overcome such drawbacks. A straightforward solution would be to extract one SIFT-like descriptor from each segmented region. In this case the Superpixels segmentation would be more suitable due to their similarly sized regions, whose size will can define the scale of the extracted feature. Alternatively more complex mid-level descriptors can be computed over the superpixels segments. For example the features proposed by Boureau *et al*. [Boureau 2010] and based on sparse coding and max pooling can be employed. In this case, the image is divided into overlapping regions of 32 × 32 pixels and four 128-dimensional SIFT descriptors are extracted and concatenated into a 512-dimensional vector. The vector is then decomposed as a sparse linear combination of atoms of a learned dictionary. The vectors of the coefficients of this sparse decomposition are used as local sparse features, which are then summarized over larger image regions by taking, for each dimension of the vector of coefficients, the maximum value over the region. Such descriptors can be assigned to segmented regions. Alternatively these descriptors can be extracted from regular Superpixels instead of 32x32 pixels regions. The advantage of such a representation is that descriptors can be computed over user defined regular image divisions or grids.

For segments with variable sizes (*e.g.*, MeanShif, EGBS) a descriptor similarly with the one proposed in [Kim 2011] can be employed. First a gPb edge map is computed over the image [Gu 2009] and a PHOG descriptor [Bosch 2007] can be then computed over the area defined by the boundaries of each segmented region. This descriptor can represent the outline of the shape and partially its inner texture. Alternatively, HOG descriptors [Dalal 2005] can be computed from the image surface defined by the segments, leading to HOG descriptors of similar sizes.

Adding new region descriptors in our object representation would enforce the replacement of the global energy function. The identification of a correspondence measure between two sub-graphs is a challenging problem. Duchenne *et al.* [Duchenne 2011] have shown that the energy obtained from their complex GraphCut optimization can be used to define a kernel describing the similarity between the two images and then further used to train a classifier from all image matchings. While outside of the scope of this work, a similar kernel could be computed from an optimization on our region based representation to be further used to train classifiers.

In order to measure the deformations and displacements of corresponding pairs of regions from the two images, a regular grid can be defined over the superpixels representations. Since the superpixels have similar sizes, they can be fitted in a cell covering the majority of its surface. The displacements of the regions can be then quantified by analyzing the changes in the position on this grid. In [Duchenne 2011] such displacements have been integrated in the energy function by employing a variant of a multi-label MRF optimization technique [Ishikawa 2003] adapted to two-dimensional labels.

Concerning the perspectives for an extension to large scale search for our region based representation we foresee the development of a Bag-Of-Regions framework. Segmented regions can be described by one of the descriptors mentioned above and quantized in a visual vocabulary. Recent advances in clustering techniques for large datasets [Shindler 2011] make it possible to generate a vocabulary for descriptors with an increased number of dimensions (above 128). Let us note that a Bag-of-Regions can be computed also by considering only the dominant color for each region and quantize regions according to the color distances. Recently, Wengert *et al.* [Wengert 2011] have shown that a Bag-of-Colors obtained from the mean colors of the cells of rectangular grid image division can provide competitive performances with interest point based BoW. The advantage of our proposed Bag-of-Regions would be that the color information is extracted from meaningful image entities and the region adjacency can be exploited in the re-ranking stage.

9.2.2 BoW representation

The short term perspectives for the query expansion algorithm using the BoW representation concern the integration of images from multiple search engines (*e.g.*, Google Images, Bing) in a more complex manner for identifying examples of instances of the object of interest to be used for generating a visual query. In order to ensure that there are enough inliers in the expanded query, the condition for representative images to have at least two matched images can be relaxed. However, other mechanisms for validating the query descriptors need to be further developed. Another extension of this method would enrich the multimodal aspect by adding an audio block in the context of mobile terminals. Most of the major smartphone operating systems [Android 2012, IOS 2012] feature advanced speech to text technologies. They can be employed by the users to launch queries by voice, which are then transcripted to text, which is used to launch queries over search engines and fetch images, which are then used to generate a visual query and then the relevant videos are retrieved.

Another aspect to study is the use of an ad-hoc classifier trained on the images and descriptors identified in the query graph. In this manner a linear SVM classifier could be computed from the images of the query graph used as positive examples and some random images used as negative examples for each query. The advantages of such a classifier would be that more geometrical constraints can be introduced in the learning stage. The search can be then performed similarly with the simple BoW search by employing an inverted index and leaving the classifier to decide the relevancy of each of the considered images.

An important aspect to be further studied for the BoW representation is the size of the vocabulary and hence, the size of the BoW vectors. The largest vocabulary size that it is usually considered for large scale object retrieval is around 1M [Nister 2006, Philbin 2007, Zhu 2012], while the majority of the approaches settle to 200k sized datasets [Jégou 2010]. Recently Jégou *et al.* [Jégou 2012] have proposed a different representation of the images using the VLAD (Vector of Locally Aggregated Descriptors) descriptors based on Fisher kernels [Jaakola 1998] with a significantly lower number of dimensions for the image representation vectors (down to 128 dimensions). The Fisher kernels can transform an incoming variable-size set of independent samples into a fixed-size vector representation. This makes it possible to search within large scale datasets of 100 million images in less than one second. This solution offers interesting perspectives for more compact frame representations. However different aspects specific to object retrieval such as geometric consistency check or quantification of groups of interest points (*i.e.*, visual phrases [Sadeghi 2011]) need to be considered.

There are multiple perspectives for the adaptation of the BoW framework for video clip representation. As we have seen in Section 6.3.3 the shot based BoW representation reduced significantly the dimensionality of the dataset and improved the performances for multiple queries video search. However, many of the key-frames that we have considered were blurry or had different artifacts due to the fixed sampling. The perspectives concern the identification of the noisy frames and the extraction of clear ones. In the case of the H.264 [H264 2012] compression such frames could be identified as the *I* frames which usually show an increased stability. Another possibility would be to analyze the evolution of the affine covariant regions [Sivic 2003] or to perform the geometric verification of the regions (*cf.* Section 4.4.4) over consecutive images and keep only the most stable ones. The maximum number of key-frames a shot based representation needs also to be studied.

Another aspect that hasn’t been solved yet is the generalization of the visual vocabularies. In most of the cases, vocabularies obtained from a dataset do not generalize well for other datasets. An alternative would be to optimize the matching algorithms and nearest neighbor metrics with machine learning approaches and then perform the search in the descriptor space without any clustering. Such aspects and questions need to be further studied for improving the performances of object instance retrieval methods.

9.2.3 Hybrid representation

The hybrid region representation needs to be further studied on different datasets with more various content and objects to be retrieved.

Concerning the matching strategies for the hybrid graphs the Spectral Graph Matching can be improved by performing the matching in an iterative manner, by using for example the Integer Quadratic Programming method in the discrete domain [Leordeanu 2009b]. Different affine constraints can be further introduced into the spectral decomposition with respect to the unary matching constraints [Cour 2006]. Moreover, the identification of the most promising pairs of points can be also performed by employing probabilistic voting techniques over multiple iterations, in order to test different sub-graph configurations [Cho 2012].

Other developments concern the extension of the graph matching problem to higher order, hyper-graph matching [Duchenne 2009, Lee 2011], which makes it possible to integrate constraints relative to tuples of matching points in the matching score.

Let us also note that the computational cost of this method can be reduced significantly by pre-computing the affinity matrices offline. Currently; this stage is the most expensive part of our algorithm and can be easily transferred to the offline stage. At query time, only the affinity matrix data concerning the interest points involved (*i.e.*, the points under the query mask and all the points from the test image) can be read from the memory, leaving only the principal vector to be computed at query time.

The adjacency of the interest points can be employed for identifying important regions of the video frames corresponding to objects or object parts. In this respect, the connectivity information of the graph structure can be employed to identify the most influential nodes on this graph (*i.e.*, the nodes corresponding to the most inter-connected sub-graphs) similarly with the Google PageRank algorithm [Page 1999] which identifies the most relevant documents from a collection of inter-connected web pages. The identified influential nodes and their adjacent nodes can be then considered separately as object entities. Such object entities can be further used to define possible object configurations, locations and sizes [Sande 2011] or can be described by a dedicated BoW vector inside the image. Images can be then represented by multiple BoW vectors, each describing a meaningful region configuration. This makes it possible to employ the BoW representation at the object level and to match solely relevant object entities in order to obtain more accurate retrieval results.

9.2.4 OVIDIUS

On the short term the perspectives for the OVIDIUS platform concern the development of a tablet adapted interface which would allow touch and drag interactions and display additional content on the screen.

Concerning the navigation functionalities, a mechanism that would allow the user to define his own threads according to his criteria of choice (navigation history, color similarity, concept similarity) can be developed. The number of such threads can be chosen by the user. Depending on the size of the screen and the device of the user and its orientation, the orientation of the threads can be selected (*e.g.*, vertical threads for tablets in portrait mode and small screens, horizontal for wide screens and tablets in landscape mode).

Additional functionalities enhanced by the advancements of the HTML5 technical specifications can be integrated in the video player in order to display on the video bounding boxes of recognized object entities, as the video plays. Moreover, such functionality will then enhance the interactivity of the video and users could simply click on the object in order to launch a query or to display additional information about it. Let us note that hyperlinks leading to other videos or content on the web can be also integrated in the video content and associated to different objects in the video. In a setting similar with the one of the PopcornJS [PopcornJS 2012] framework, information specific to each sequence can be displayed in dedicated sections of the interface and updated as the video plays. Such information could concern the concept keywords, the audio transcript, the objects and characters in the video sequence.

Another perspective concerns the enhanced access of the OVIDIUS platform. The functionalities that we have presented are based on HTTP requests and responses, so they can be compiled and used in an OVIDIUS API. Different interfaces and systems can be further proposed and developed with the navigation and search information provided by the OVIDIUS API.

10. Dissemination

10.1 Publications

- A. Bursuc, T. Zaharia, and F. Prêteux, “Dynamic detection of visual entities,” *Proc. 20th European Conference on Signal Processing Conference*, pp. 2392-2396, IEEE Conference Publications, Bucharest, Romania, 2012.
- A. Bursuc, T. Zaharia, and F. Prêteux, “Retrieval of Multiple Instances of Object in Videos,” *Proc. International Conf. on Advances in Multimedia Modelling*, pp. 358-369, Springer LNCS, Klagenfurt, Austria, 2012.
- A. Bursuc, T. Zaharia, and F. Prêteux, “OVIDIUS: a web platform for video browsing and search,” *Proc. International Conf. on Advances in Multimedia Modelling*, pp. 649-651, Springer LNCS, Klagenfurt, Austria, 2012.
- A. Bursuc, T. Zaharia, and F. Prêteux, “Detection of Multiple Instances of Video Objects,” *Proc. IEEE/ACM .International Conference on Signal Image Technology and Internet-based systems*, pp. 446-453, IEEE Conference Publications, Dijon, France, 2011.
- A. Bursuc, T. Zaharia, and F. Prêteux, “Interactive region-based retrieval;” *Proc. SPIE 8136*, 81360E, SPIE Digital Library, San Diego, USA, 2011.
- A. Bursuc, T. Zaharia, and F. Prêteux, “Online interactive video content retrieval,” *Proc. IEEE International Conference on Consumer Electronics*, pp.215-216, IEEE Conference Publications, Las Vegas, USA, 2011.
- A. Bursuc, T. Zaharia, and F. Prêteux, “Mobile Video Browsing and Retrieval with the OVIDIUS Platform”, *Proc. ACM International Conference on Multimedia*, pp. 1659-1662, Florence, Italy, 2010.
- A. Bursuc, T. Zaharia, and F. Prêteux, ”Mobile video navigation and retrieval services with the OVIDIUS platform”, *Proc. NEM Summit 2010 - Towards the future Media Internet*, Barcelona, Spain, 2010.
- A. Bursuc, T. Zaharia, and F. Prêteux, “OVIDIUS: an on-line video indexing universal system,” *Proc. SPIE 7799*, 77990C, SPIE Digital Library, San Diego, USA, 2010.
- A. Bursuc, T. Zaharia, and F. Prêteux, “OVIDIUS: An on-line video retrieval platform for multi-terminal access,” *Proc. International Workshop on Content-Based Multimedia Indexing*, pp. 1-6, IEEE Conference Publications, Grenoble, France, 2010.

10.2 Competitions

- Participation in the TRECVID 2012 Campaign in the Instance Search Task.
- Video Browser Showdown, Klagenfurt, Austria, January 2012
- Participation in the TRECVID 2011 Campaign in the Instance Search Task.

11. References

[2minutes 2012] www.2minutes.fr, last accessed in September 2012.

[Adobe 2012] http://www.adobe.com/products/flashplayer.html, last accessed in September 2012.

[Adobe 2012b] http://www.adobe.com/products/aftereffects.html, last accessed in September 2012.

[Agarwal 2002] S. Agarwal and D. Roth, "Learning a sparse representation for object detection," *Proc. European Conference on Computer Vision*, pp. 113–130, 2002.

[Aherne 1998] F. Aherne, N. Thacker, and P. Rockett, "The Bhattacharyya metric as an absolute similarity measure for frequency coded data," *Kybernetika*, vol. 34(4), pp. 363–368, 1998.

[AJAX 2012] http://api.jquery.com/category/ajax/, last accessed in September 2012.

[Alahi 2012] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast Retina Keypoint," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2012.

[Android 2012] http://www.android.com/, last accessed in September 2012.

[Anguera 2008] X. Anguera, J. Xu, and N. Oliver, "Multimodal photo annotation and retrieval on a mobile phone," *Proc. ACM 1st International Conference on Multimedia Information Retrieval*, pp. 188-194, 2008.

[Apache 2012a] http://httpd.apache.org/, last accessed in September 2012.

[Apache 2012b] http://tomcat.apache.org/, last accessed in September 2012.

[Arandjelovic 2012] R. Arandjelovic, A. Zisserman, "Three things everyone should know to improve object retrieval," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp.2911-2918, 2012.

[Arbelaez 2009] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "From contours to regions: An empirical evaluation," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[Ayache 2010] S. Ayache, G. Quénot, and A. Tseng, "The LIGVID system for video retrieval and concept annotation," *Proc. International Conference on Multimedia Information Retrieval*, pp. 385-388, 2010.

[Baeza 1999] R. Baeza-Yates and B. Ribeiro-Neto, "Modern Information Retrieval," *ACM Press*, ISBN: 020139829, 1999.

[Bai 2007] X. Bai, G. Sapiro, "A geodesic framework for fast interactive image and video segmentation and matting," *Proc. IEEE International Conference on Computer Vision*, 2007.

[Bai 2009] X. Bai, J. Wang, D. Simons, and G. Sapiro, "Video SnapCut: robust video object cutout using localized classifiers," Proc. *ACM SIGGRAPH Conference*, 2009.

[Bailer 2012] W. Bailer, W. Weiss, C. Schober, and G. Thallinger, "A Video Browsing Tool for Content Management in Media Post-Production," *Proc. International Conf. on Advances in Multimedia Modelling*, pp. 658-659, 2012.

[Bay 2006] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Proc. European Conference on Computer Vision*, pp 404-417, 2006.

[Bay 2008] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346-359, Jun. 2008.

[BBC 2012] http://www.bbc.co.uk/, last accessed in September 2012.

[Berg 2005] A. C. Berg, T. L. Berg, and J. Malik, “Shape matching and object recognition using low distortion correspondences,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 26–33, 2005.

[Bertini 2011] M. Bertini, A. Del Bimbo, A. Ferracani, L. Landucci, D. Pezzatini, “Interactive multi-user video retrieval systems,” *Multimedia Tools and Applications*, pp. 1-27, October 2011.

[Besag 1986] J. Besag, “On the Statistical Analysis of Dirty Pictures,” J*ournal of the Royal Statistical Society*, Series B, vol. 48, no. 3, pp. 259-302, 1986.

[Bing 2012] http://images.bing.com/, last accessed in September 2012.

[Blei 2002] D. Blei, A. Ng, A. and M. Jordan, “Latent Dirichlet allocation,” *Proc. Neural Information Processing Systems Conf.*, 2002.

[Blinkx 2012] http://www.blinkx.com/, last accessed in September 2012.

[Bosch 2007] A. Bosch, A. Zisserman, and X.Munoz, “Representing Shape with a Spatial Pyramid Kernel,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[Bosch 2008] A. Bosch, A. Zisserman, and X. Munoz, “Scene Classification Using a Hybrid Generative/Discriminative Approach,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 4, pp. 712-727, Apr. 2008.

[Boser 1992] B.E. Boser, I.M. Guyon, and V.N. Vapnik, “A training algorithm for optimal margin classifiers”. *Proc. 5th Annual ACM Workshop on COLT*, pp. 144–152, 1992.

[Boureau 2010] Y. Boureau, F. Bach, Y. LeCun, and J. Ponce, “Learning mid-level features for recognition,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2010.

[Boykov 2001] Y. Boykov, O. Veksler, R. Zabih, “Fast Approximate Energy Minimization via Graph Cuts,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222-1239, Nov. 2001.

[Boykov 2001b] Y. Boykov and M.P. Jolly, “Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images,” *Proc. IEEE International Conference on Computer Vision*, 2001.

[Boykov 2004] Y. Boykov and V. Kolmogorov, “An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124-1137, Sep. 2004.

[Boykov 2006] Y. Boykov and G. Funka-Lea, “Graph cuts and efficient N-D image segmentation,” *International Journal of Computer Vision*, vol. 70, no. 2, pp.109-131, Nov. 2006.

[Brainard 2003] D.H. Brainard, ”Color Appearance and Color Difference Specification,” *The Science of Color, 2nd edition*, S. K. Shevell (ed.), Optical Society of America, Washington D.C., pp. 191-216, 2003.

[Bréhinier 2012] M. Bréhinier, S. Campion, and G. Gravier. “Texmix: an automatically generated news navigation portal,” *Proc. 2nd ACM International Conference on Multimedia Retrieval*, 2012.

[Brown 2003] M. Brown and D. Lowe. “Recognizing panoramas”. *Proc. IEEE International Conference on Computer Vision*, pp. 1218–1225, 2003.

[Burghouts 2009] G.J. Burghouts and J.M. Geusebroek, “Performance Evaluation of Local Color Invariants,” *Computer Vision and Image Understanding*, vol. 113, no. 1, pp. 48-62, Jan. 2009.

[Bursuc 2010] A. Bursuc, T. Zaharia, F. Prêteux. “Mobile Video Browsing and Retrieval with the OVIDIUS Platform”, *Proc. ACM International Conference on Multimedia,* 2010.

[Bursuc 2011] A. Bursuc, T. Zaharia, and O. Martinot, “ARTEMIS-UBIMEDIA at TRECVid 2011: Instance Search,” *Proc. TRECVid 2011 - Text REtrieval Conference TRECVid Workshop*, 2011.

[Bursuc 2012] A. Bursuc, T. Zaharia, and F. Prêteux, “OVIDIUS: a web platform for video browsing and search,” *Proc. International Conf. on Advances in Multimedia Modelling*, pp. 649-651, 2012.

[Caetano 2009] T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola, “Learning graph matching,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 6, pp. 1048–1058, Jun. 2009.

[Calonder 2010] M. Calonder, V. Lepetit, C. Strecha, P. Fua, ”BRIEF: Binary Robust Independent Elementary Features,” *Proc. 11th European Conference on Computer Vision*, 2010.

[Campbell 2006] N. Campbell, B. Williamson; R. J. Heyden, “Biology: Exploring Life,” *Boston, Massachusetts: Pearson Prentice Hall*, ISBN 0-13-250882-6, 2006.

[Cao 2009] J. Cao, Y. Zhang, J. Guo, L. Bao, and J. Li, “VideoMap: an interactive video retrieval system of MCG-ICT-CAS,” *Proc. ACM International Conference on Image and Video Retrieval*, 2009.

[CGI 2012] http://www.w3.org/CGI/, last accessed in September 2012.

[Chen 2008] C. Chen, Y. Wang, H. Wang, and C. Chiu, “Digital Video Retrieval via Mobile Devices,” *Proc. 4th IEEE International Conference on eScience*, pp. 376-377, 2008.

[Cheng 2003] Y.-C. Cheng and S.-Y. Chen, “Image classification using color, texture and regions,” I*mage and Vision Computing*, vol. 21, no. 9, pp.759–776, Sep. 2003.

[Chevalier 2007] F. Chevalier, J.P. Domenger, J. Benois-Pineau, M. Delest, “Retrieval of objects in video by similarity based on graph matching,” *Pattern Recognition Letters*, vol. 28, no. 8; pp. 939-949, Jun. 2007.

[Cho 2009] M. Cho, J. Lee, Kyoung, M. Lee, “Feature Correspondence & Deformable Object Matching via Agglomerative Correspondence Clustering,” *Proc. 12th IEEE International Conference on Computer Vision*, 2009.

[Chiariglione 2002] L. Chiariglione, “Chapter 1. Introduction to MPEG-7: Multimedia Content Description Interface,” *Introduction to MPEG-7 Multimedia Content Description Interface*, p.3-6, 2002.

[Cho 2009b] M. Cho, K.M. Lee, “Bilateral Symmetry Detection and Segmentation via Symmetry-Growing,” *Proc. 20th British Machine Vision Conference*, 2009.

[Cho 2012] M. Cho and K. M. Lee, “Progressive Graph Matching: Making a Move of Graphs via Probabilistic Voting,“ *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2012.

[Christoudias 2002] C. M. Christoudias, B. Georgescu, and P. Meer, “Synergism in low level vision,” *Proc. International Conference on Pattern Recognition*, 2002.

[Chum 2003] O. Chum, J. Matas, and J. Kittler. “Locally optimized RANSAC,” *Proc. DAGM Symposium*, pp. 236-243, 2003.

[Chum 2005] O. Chum and J. Matas, “Matching with PROSAC - progressive sampling consensus,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[Chum 2007] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, “Total recall: Automatic query expansion with a generative feature model for object retrieval," *Proc. IEEE 11th International Conference on Computer Vision*, pp. 1-8, 2007.

[Chum 2008] O. Chum and J. Matas, "Optimal randomized ransac," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 8, pp. 1472–1482, 2008.

[Chum 2008b] O. Chum, J. Philbin, and A. Zisserman, "Near duplicate image detection: min-hash and tf-idf weighting," *Proc. British Machine Vision Conference*, 2008.

[Chum 2011] O. Chum, A. Mikulik, M. Perdoch, and J. Matas, "Total recall II: Query expansion revisited," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 889-896, 2011.

[Comaniciu 1999] D. Comaniciu and P. Meer, "Mean shift analysis and applications," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1197-1203, 1999.

[Comaniciu 2002] D. Comaniciu, P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603-619, May, 2002.

[Cortes 1995] C. Cortes, and V.N. Vapnik, "Support-Vector Networks", *Machine Learning*, 20, 1995.

[Cour 2006] T. Cour, P. Srinivasan, and J. Shi, "Balanced graph matching," *Proc. Neural Information Processing Systems Conf.*, pp. 313–320, 2006.

[Cook 1998] W. Cook, W. Cunningham, W. Pulleyblank, and A. Schrijver, "*Combinatorial Optimization*," *John Wiley & Sons*, 1998.

[CURL 2012] http://curl.haxx.se/, last accessed in September 2012.

[DailyMotion 2012] http://www.dailymotion.com/, last accessed in September 2012.

[Dalal 2005] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[Delezoide 2008] B. Delezoide, G. Pitel, H. Le Borgne, G. Greffenstette, P.A. Moellic, C. Millet, "Object/Background Scene Joint Classification in Photographs Using Linguistic Statistics from the Web," *Proc. 2nd International Language Resources for Content-Based Image Retrieval Workshop*, 2008.

[Delong 2011] A. Delong, "Advances in Graph-Cut Optimization: Multi-Surface Models, Label Costs, and Hierarchical Costs." *PhD Thesis. University of Western Ontario*, 2011.

[Duchenne 2009] O. Duchenne, F. Bach, I. Kweon, and J. Ponce, "A tensor based algorithm for high-order graph matching," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[Duchenne 2011] O. Duchenne, A. Joulin, and J. Ponce, "A Graph-Matching Kernel for Object Categorization," *Proc. IEEE International Conference on Computer Vision*, 2011.

[EUscreen 2012] http://euscreen.eu/, last accessed in September 2012.

[Everingham 2010] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303-338, Jun. 2010.

[Fabro 2012] M.D. Fabro and L. Böszörmenyi, "AAU video browser: non-sequential hierarchical video browsing without content analysis," *Proc. International Conf. on Advances in Multimedia Modelling*, 2012.

[Facebook 2012] https://www.facebook.com/, last accessed in September 2012.

[Fei-Fei 2005] L. Fei-Fei, and P. Perona, “A bayesian hierarchical model for learning natural scene categories”, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

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

[Felzenszwalb 2007] http://www.cs.brown.edu/~pff/segment/, last accessed in September 2012.

[Felzenszwalb 2008] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object Detection with Discriminatively Trained Part Based Models,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

[Ferrari 2006] V. Ferrari, T. Tuytelaars, and L. Gool, “Simultaneous object recognition and segmentation from single or multiple model views,” *International Journal of Computer Vision*, vol. 67, no. 2, pp. 159-188, Apr. 2006.

[Fergus 2003] R. Fergus, P. Perona, and A. Zisserman, “Object class recognition by unsupervised scale-invariant learning,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 264–271, 2003.

[Fergus 2007] R. Fergus, P. Perona, and A. Zisserman, “Weakly supervised scale-invariant learning of models for visual recognition”, *International Journal of Computer Vision*, vol. 71, no. 3, pp. 273–303, Mar. 2007.

[FFMPEG 2012] http://ffmpeg.org/, last accessed in September 2012.

[Fischler 1981] M.A. Fischler and R.C. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”, *Comm. of the ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[Flickr 2012] http://www.flickr.com/, last accessed in September 2012

[Flickr 2012b] http://www.flickr.com/cameras/, last accessed in March 2012

[Foley 2010] C. Foley, *et. al*, “TRECVID 2010 Experiments at Dublin City University,” *TRECVid 2010 - Text REtrieval Conference TRECVid Workshop*, 2010.

[Ford 1962] L. Ford and D. Fulkerson, “Flows in Networks,” *Princeton University Press*, 1962.

[Freedman 2005] D. Freedman and P. Drineas, “Energy minimization via graph cuts: Settling what is possible,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 939-946, 2005.

[Friedman 1977] J.H. Friedman, J.L. Bentley, and R.A. Finkel, ”An algorithm for finding best matches in logarithmic expected time,” *ACM Transactions on Mathematical Software*, vol. 3, no. 3, pp. 209-226, 1977.

[Fulkerson 2009] B. Fulkerson, A. Vedaldi, S. Soatto. “Class segmentation and objectlocalization with superpixel neighborhoods,” *Proc. IEEE International Conference on Computer Vision*, 2009.

[Gabriel 2005] P. Gabriel, J-B. Hayet, J. Piater, and J. Verly, “Object tracking using color interest points,” *Proc. IEEE Conf. on Advanced Video and Signal Based Surveillance*, 2005.

[Garding 1996] J. Garding and T. Lindeberg, “Direct computation of shape cues using scale-adapted spatial derivative operators,” *International Journal of Computer Vision*, vol. 17, no. 2, pp. 163–191, Feb. 1996.

[Garey 1979] M. R. Garey and D. S. Johnson, “Computers and Intractability: A Guide to the Theory of NP-Completeness,” *W. H. Freeman*, New-York, 1979.

[Geman 1984] S. Geman and D. Geman, “Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721-741, Jun. 1984.

[Gemert 2010] J.C. van Gemert, C.J. Veenman, A.W.M. Smeulders, and J.-M. Geusebroek, “Visual Word Ambiguity,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1271-1283, Jul. 2010.

[Girgensohn 2011] A. Girgensohn, F. Shipman, and L. Wilcox, “Adaptive clustering and interactive visualizations to support the selection of video clips,” *Proc. 1st ACM International Conference on Multimedia Retrieval*, 2011.

[Goldberg 1988] A. Goldberg and R. Tarjan, “A new approach to the maximum-flow problem,” *Journal of the Association for Computing Machinery*, vol. 35, no. 4, pp.921-940, Oct. 1988.

[Goldberg 2011] A.V. Goldberg, S. Hed, H. Kaplan, R.E. Tarjan, and R.F. Werneck, “Maximum Flows by Incremental Breadth-First Search.” *Algorithms ESA*, 2011.

[Golub 1996] G. H. Golub and C. F. Van Loan, “Matrix computations (3rd ed.),” *Johns Hopkins University Press*, 1996.

[Google 2012a] http://www.google.com/, last accessed in September 2012.

[Google 2012b] http://www.google.com/mobile/goggles, last accessed in September 2012.

[Google 2012c] http://images.google.com/, last accessed in September 2012.

[Gorisse 2010] D. Gorisse, *et al.*, “IRIM at TRECVID 2010: Semantic Indexing and Instance Search,” *TRECVid 2010 - Text REtrieval Conference TRECVid Workshop*, 2010.

[Grauman 2005] K. Grauman and T. Darrell, “The pyramid match kernel: discriminative classification with sets of image features,” *Proc. IEEE International Conference on Computer Vision*, 2005.

[Greig 1989] D. Greig, B. Porteous, and A. Seheult, “Exact Maximum A Posteriori Estimation for Binary Images,” *Journal of Royal Statistical Society*, Series B, vol. 51, no. 2, pp. 271-279, 1989.

[Gu 2009] C. Gu, J. Lim, P. Arbelaez, and J.Malik, “Recognition using regions,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[GZIP 2012] http://www.gzip.org/, last accessed in September 2012.

[H264 2012] http://www.itu.int/rec/T-REC-H.264

[Hafner 1995] J. Hafner, H.S. Sawhney, W. Equitz, M. Flickner, W. Niblack, “Efficient color histogram indexing for quadratic form distance functions,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no.7, pp. 729–736, Jul. 1995.

[Halvorsen 2010] P. Halvorsen, D. Johansen, B. Olstad, T. Kupka, S. Tennøe, “vesp: enriching enterprise document search results with aligned video summarization,” *Proc. ACM International Conference on Multimedia*, pp 1603–1604, 2010.

[Harris 1988] C. Harris and M. Stephens, “A combined corner and edge detector,” *Proc. 4th Alvey Vision Conference*, pp. 147–151, 1988.

[Hauptmann 2007] A. Hauptmann, R. Yan, W.-H. Lin, M. Christel, and H. Wactlar, “Can high-level concepts fill the semantic gap in video retrieval? A case study with broadcast news,” *IEEE Trans. Multimedia*, vol. 9, no. 5, pp. 958-966, May. 2007.

[HD3D 2011] http://www.hd3d.fr/, last access September 2011.

[HDF5 2012] http://www.hdfgroup.org/HDF5/whatishdf5.html, last accessed in September 2012.

[HTML5 2012a] http://www.w3.org/TR/html5/, last accessed in September 2012.

[HTML5 2012b] http://www.w3.org/wiki/HTML/Elements/video, last accessed in September 2012.

[HTML5 2012c] http://www.w3.org/wiki/HTML/Elements/canvas, last accessed in September 2012.

[HTTP 2012] http://www.w3.org/Protocols/rfc2616/rfc2616.html, last accessed in September 2012.

[Hummel 1983] R.A. Hummel, S.W. Zucker, "On the foundations of relaxation labeling processes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 5, no. 3, pp. 267–287, May 1983.

[IM3I 2012] http://www.im3i.eu, last accessed in September 2012.

[IClef 2008] http://www.imageclef.org/2008/vcdt, last accessed in September 2012.

[IClef 2011] http://imageclef.org/2011/photo, last accessed in September 2012.

[IClef 2012] http://imageclef.org/2012/photo-flickr, last accessed in September 2012.

[INA 2012] http://www.ina.fr/, last accessed in September 2012.

[IOS 2012] http://www.apple.com/iphone/ios/, last accessed in September 2012.

[IRI 2009] http://www.iri.centrepompidou.fr/outils/lignes-de-temps/, last accessed in September 2012.

[Ishikawa 1998] H. Ishikawa and D. Geiger, "Occlusions, Discontinuities, and Epipolar Lines in Stereo," *Proc. European Conference on Computer Vision*, pp. 232-248, 1998.

[Ishikawa 2003] H. Ishikawa, "Exact Optimization for Markov Random Fields with Convex Priors," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1333-1336, Oct. 2003.

[Jaakola 1998] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," *Proc. Neural Information Processing Systems Conf.*, 1998.

[Jansen 2008] M. Jansen, W. Heeren, B. van Dijk, "Videotrees: Improving video surrogate presentation using hierarchy," *Proc. International Workshop Content-Based Multimedia Indexing*, 2008.

[Jarvinen 2009] S. Järvinen, J. Peltola, J. Lahti, and A. Sachinopoulou, "Multimedia service creation platform for mobile experience sharing," *Proc. 8th ACM International Conference on Mobile and Ubiquitous Multimedia*, 2009.

[Jégou 2007] H. Jégou, H. Harzallah, and C. Schmid, "A contextual dissimilarity measure for accurate and efficient image search," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[Jégou 2008] H. Jégou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," *Proc. European Conference on Computer Vision*, pp. 304-317, 2008.

[Jégou 2010] H. Jégou, M. Douze, and C. Schmid, "Improving bag-of-features for large scale image search," *International Journal of Computer Vision*, vol. 87, no. 3, pp. 316-336, May 2010.

[Jégou 2012] H. Jégou, F. Perronnin, M. Douze, J. Sanchez, P. Pérez and C. Schmid, "Aggregating local images descriptors into compact codes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1704-1716, Sep. 2012.

[Jesus 2007] R. Jesus, R. Dias, R. Frias, and N. Correia, "Geographic image retrieval in mobile guides," *Proc. 4th ACM Workshop on Geographical Information Retrieval*, 2007.

[Jia 2006] M. Jia, *et al.*, "Photo-to-Search: Using camera phones to inquire of the surrounding world," *Proc. 7th International Conference on Mobile Data Management*, 2006.

[Jiang 2007] H. Jiang, M.S. Drew, Z. Li, "Matching by linear programming and successive convexification," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 959–975, Jun. 2007.

[Joshi 2006] D. Joshi, R. Datta, Z. Zhuang, W.P. Weiss, M. Friedenberg, J. Li, J.Z. Wang, "Paragrab: a comprehensive architecture for web image management and multimodal querying," *Proc. International Conference on Very Large Data Bases*, pp 1163–1166, 2006.

[jQuery 2012] http://jquery.com/, last accessed in September 2012.

[JSON 2012] http://www.json.org/, last accessed in September 2012.

[Julesz 1981] B. Julesz, "Textons, the elements of texture perception, and their interactions," *Nature*, vol. 290, no. 5802, pp.91–97, 1981.

[Jurie 2005] F. Jurie, B. Triggs, "Creating efficient codebooks for visual recognition," *Proc. IEEE International Conference on Computer Vision*, 2005.

[JWPlayer 2012] http://www.longtailvideo.com/players/jw-flv-player/, last accessed in September 2012.

[Kalogerakis 2009] E. Kalogerakis, O. Vesselova, J. Hays, A. Efros, A. Hertzmann, "Image Sequence Geolocation with Human Travel Priors," *Proc. IEEE International Conference on Computer Vision*, 2009.

[Kang 2011] H. Kang; M. Hebert; T. Kanade, "Discovering object instances from scenes of Daily Living," *Proc. IEEE International Conference on Computer Vision*, pp.762-769, 2011.

[Karp 1972] R.M. Karp , R.E. Miller and J.W. Thatcher, "Reducibility among combinatorial problems", *Complexity of Computer Computations*, Plenum Press, NY, pp. 85-103,1972.

[Ke 2004] Y. Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.

[Kim 2004] C. Y. Kim, *et al.*, "VISCORS: A visual-content recommender for the mobile web," *IEEE Intelligent Systems*, vol. 19, no. 6, pp. 32-39, 2004.

[Kim 2005] S. Kim, Y. Tak, Y. Nam, and E. Hwang, "mCLOVER: mobile content-based leaf image retrieval system," *Proc. 13th ACM International Conference on Multimedia*, pp. 215-216, 2005.

[Kim 2011] K. Kim, K. Grauman, "Boundary Preserving Dense Local Regions," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2011.

[Kirkpatrick 1983] S. Kirkpatrick, C.D. Gelatt, M.P. Vechi, "Optimization by simulated annealing," *Science*, vol. 220, 1983.

[Kohli 2007] P. Kohli and P.H. S. Torr, "Dynamic Graph Cuts for Efficient Inference in Markov Random Fields," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.29, no. 12, pp. 2079–2088, Dec. 2007.

[Kooaba 2012] http://www.kooaba.com/, last accessed in September 2012.

[Kolmogorov V. Kolmogorov, R.Zabih, "What Energy Functions can be Minimized via Graph Cuts?"

2004] *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp.147-159, Feb 2004.

[Lampert 2008] C. Lampert, M.B. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1-8. IEEE, 2008.

[Law-To 2009] J. Law-To, G. Grefenstette, and J.L. Gauvain, "VoxaleadNews: robust automatic segmentation of video into browsable content," *Proc. 17th ACM International Conference on Multimedia*, pp. 1119-1120, 2009.

[Lazebnik 2003] S. Lazebnik, C. Schmid, and J. Ponce, "Affine-invariant local descriptors and neighborhood statistics for texture recognition," *Proc. IEEE International Conference on Computer Vision*, pp. 649–655, 2003.

[Lazebnik 2006] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.

[Le Borgne 2010] H. Le Borgne, N. Honnorat, "Fast shared boosting: application to large-scale visual concept detection," *Proc. International Workshop on Content-Based Multimedia Indexing*, 2010.

[Lee 2011] J. Lee, M. Cho, and K.M. Lee, "Hyper-graph matching via reweighted random walks," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2011, pp. 1633-1640. IEEE, 2011.

[Leibe 2008] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *International Journal of Computer Vision*, vol. 77, no.1-3, pp. 259–289, May 2008.

[Levinshtein 2009] A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, K. Siddiqi. "TurboPixels: Fast superpixels using geometric flows," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no.12, pp. 2290-2297, Dec. 2009.

[Leordeanu 2005] M. Leordeanu, M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," *Proc. IEEE International Conference on Computer Vision*, 2005

[Leordeanu 2009] M. Leordeanu and M. Hebert, "Unsupervised learning for graph matching," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[Leordeanu 2009b] M. Leordeanu, M. Herbert, "An integer projected fixed point method for graph matching and map inference," *Proc. Neural Information Processing Systems Conf.*, 2009.

[Lepetit 2005] V. Lepetit, P. Lagger, and P. Fua, "Randomized trees for real-time keypoint recognition," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[Leung 2001] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International Journal of Computer Vision*, vol. 43, no. 1, pp. 29–44, Jun. 2001.

[Leutenegger 2011] S. Leutenegger, M. Chli, and R. Siegwart, "Brisk: Binary robust invariant scalable keypoints," *Proc. IEEE International Conference on Computer Vision*, 2011.

[Li 2004] Y. Li, J. Sun, C. Tang and H. Shum, "Lazy Snapping," *Proc. ACM SIGGRAPH Conference*, pp. 303-308, 2004.

[Li 2010] H. Li, E. Kim, X. Huang, L. He, "Object matching with a locally affine-invariant constraint," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp.1641-1648, 2010.

[Li 2011] H. Li; J. Huang; S. Zhang; X. Huang, “Optimal object matching via convexification and composition,” *Proc. IEEE International Conference on Computer Vision*, pp.33-40, 2011.

[Lindeberg 1998] T. Lindeberg, “Feature detection with automatic scale selection,” *International Journal of Computer Vision*, vol. 30, no. 2, pp. 77–116, Nov. 1998.

[Liu 2009] Y. Liu, Z. Yang, X. Deng, J. Bu, and C. Chen, “Media Browsing for Mobile Devices Based on Resolution Adaptive Recommendation,” *Proc. WRI International Conference on Communications and Mobile Computing*, pp.285-290, 2009.

[Lowe 1985] D. Lowe, “Perceptual Organization and Visual Recognition*,” Kluwer Academic Publishers*, Boston, 1985.

[Lowe 1999] D. Lowe, “Object recognition from local scale-invariant features,” *Proc. IEEE 7th International Conference on Computer Vision*, pp. 1150–1157, 1999.

[Lowe 2004] D. Lowe, “Distinctive image features from scale-invariant keypoints”, *International Journal of Computer Vision*, vol. 2, no. 60, pp. 91–110, Nov. 2004.

[LSCOM 2006] “LSCOM Lexicon Definitions and Annotations Version 1.0”, *DTO Challenge Workshop on Large Scale Concept Ontology for Multimedia*, Columbia University ADVENT Technical Report #217-2006-3, March 2006.

[Lundy 1986] M. Lundy, A. Mees, “Convergence of an annealing algorithm,” *Mathematical Programming*, vol. 34, pp. 111–124, 1986.

[Mahamud 2003] S. Mahamud and M. Hebert, “The optimal distance measure for object detection,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[Maire 2008] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik, “Using Contours to Detect and Localize Junctions in Natural Images,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

[Malisiewicz 2007] T. Malisiewicz, A. Efros, “Improving spatial support for objects via multiple segmentations,” *Proc. British Machine Vision Conference*, 2007.

[Manske 1998] K. Manske, “Video browsing using 3d video content trees”. *Proc. ACM Workshop on New Paradigms in Information Visualization and Manipulation*, 1998.

[Matas 2002] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide baseline stereo from maximally stable extremal regions”, *Proc. British Machine Vision Conference*, pp. 384–393, 2002.

[Mediatic 2010] http://www.media-tic.org , last accessed March 2010.

[Metropolis 1953] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, ”Equations of state calculation by fast computing machines,” *Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[Mikolajczyk 2001] K. Mikolajczyk and C. Schmid, “Indexing based on scale invariant interest points,” *Proc. IEEE 8th International Conference on Computer Vision*, pp. 525–531, 2001.

[Mikolajczyk 2002] K. Mikolajczyk, and C. Schmid, “An affine invariant interest point detector,” *Proc. European Conference on Computer Vision*, 2002.

[Mikolajczyk K. Mikolajczyk and C. Schmid, “Scale and affine invariant interest point detectors,” 2004] *International Journal of Computer Vision*, 60:63–86, 2004.

[Mikolajczyk 2005] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir and L. Van Gool, “A comparison of affine region detectors,” *International Journal of Computer Vision*, vol. 65, no. 1-2, pp. 43–72, Nov. 2005.

[Miller 2009] G. Miller, S. Fels, M. Finke, W. Motz, W. Eagleston, and C. Eagleston, “MiniDiver: A Novel Mobile Media Playback Interface for Rich Video Content on an iPhoneTM,” *Proc. 8th International Conference on Entertainment Computing*, pp. 98-109, 2009.

[MIR 2012] http://press.liacs.nl/mirflickr/, last accessed in September 2012.

[Mishra 2009] A. Mishra and Y. Aloimonos, “Active segmentation with fixation,” *Proc. IEEE International Conference on Computer Vision*, 2009.

[Moodstocks 2012] http://www.moodstocks.com/, last accessed in September 2012.

[Moravec 1977] H. Moravec, “Towards automatic visual obstacle avoidance,” *Proc. International Joint Conference on Artificial Intelligence*, 1977.

[MPEG 2001] MPEG-7 Requirements Goup, *“MPEG-7 Requirements”*, Doc. ISO/MPEG N4320, Sydney MPEG Meeting, 2001.

[MPEG 2002] International standard ISO/IEC 15938-3:2002, Information technology - Multimedia Content Description. Interface - Part 3: Visual. 2002.

[MPEG 2003] International standard ISO/IEC 15938-5:2003, Information technology - MultimediaContent Description. Interface-Part 5: Multimedia Description Schemes. 2003.

[MPEG 2009] MPEG Home Page, http://www.chiariglione.org/mpeg/ , last accessed August 2009.

[MPEG 2012] MPEG Home Page, http://www.chiariglione.org/mpeg/ , last accessed September 2012.

[MPI 2012] http://www.open-mpi.org/, last accessed in September 2012.

[Muja 2009] M. Muja and D. Lowe, “Fast approximate nearest neighbors with automatic algorithm configuration,” *Proc. International Conference on Computer Vision Theory and Applications*, 2009.

[Nevatia 1977] R. Nevatia and T.O. Binford, “Description and recognition of complex curved objects,” *Artificial Intelligence Journal*, vol. 8, pp. 77–98, 1977.

[Ni 2009] K. Ni, H. Jin, and F. Dellaert, “GroupSAC: Efficient Consensus in the Presence of Groupings,” *Proc. IEEE International Conference on Computer Vision*, 2009.

[Niblack 1993] W. Niblack, R. Barber, W. Equitz, M. Fickner, E. Glasman, D. Petkovic, and P. Yanker, “The QBIC project: Querying images by content using color, texture and shape,” *Proc. SPIE Conference on Geometric Methods in Computer Vision II*, 1993.

[Nielsen 2010] http://blog.nielsen.com/nielsenwire/online_mobile/three-screen-report-q409/, last accessed April 2010.

[Nielsen 2011] http://blog.nielsen.com/nielsenwire/consumer/smartphones-to-overtake-feature-phones-in-u-s-by-2011/, last accessed April 2010.

[Nister 2006] D. Nister and H. Stewenius, “Scalable recognition with a vocabulary tree,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.

[Ojala 2002] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classiffication with local binary patterns,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971-987, Jul. 2002.

[Oliva 2001] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *International Journal of Computer Vision*, vol. 42, no. 3, pp.145–175, May 2001.

[Page 1999] L. Page, S. Brin, M. Rajeev and T. Winograd, “The PageRank citation ranking: bringing order to the web,” *Technical Report, Stanford InfoLab*, 1999.

[Pantofaru 2006] C. Pantofaru, G. Dorko, C. Scmid and M. Hebert, ”Combining Regions and Patches for Object Class Localization,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.

[Pantofaru 2008] C. Pantofaru, C. Schmid, and M. Hebert, “Object Recognition by Integrating Multiple Image Segmentations,” *Proc. European Conference on Computer Vision*, 2008.

[Pascal 2006] The pascal object recognition database collection. Website, 2006. http://www.pascalnetwork.org/challenges/VOC/.

[Perdoch 2009] M. Perdoch, O. Chum, and J. Matas, “Efficient Representation of Local Geometry for Large Scale Object Retrieval,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[Pereira 2002] F. Pereira and B. Koenen, “Chapter 2. Context, Goals and Procedures,” *Introduction to MPEG-7 Multimedia Content Description Interface*, p.7-30, 2002.

[Pfeiffer 2010] S.Pfeiffer, “The Definitive Guide to HTML5 Video,” *Apress*, ISBN: 978-1-4302-3090-8, 2010.

[Philbin 2007] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, ”Object retrieval with large vocabularies and fast spatial matching,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[Philbin 2008] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, ”Lost in quantization: Improving particular object retrieval in large scale image databases,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

[Philbin 2010] J. Philbin, “Scalable Object Retrieval in Very Large Image Collections,” *PhD thesis*, University of Oxford, 2010.

[Philbin 2011] J. Philbin, J. Sivic, A. Zisserman, “Geometric Latent Dirichlet Allocation on a Matching Graph for Large-scale Image Datasets,” *International Journal of Computer Vision*, vol. 95, no. 2, pp. 138-153, Nov. 2011.

[Philbin 2012] J. Philbin. “FASTCLUSTER: A library for fast, distributed clustering,” http://github.com/philbinj/fastcluster, last accessed in September 2012.

[Pingdom 2012] http://royal.pingdom.com/2012/01/17/internet-2011-in-numbers/, last accessed in September 2012.

[PixlinQ 2012] http://www.pixlinq.com/home, last accessed in September 2012.

[PlinkArt 2010] http://www.smashapp.com/plinkart, last accessed in September 2010.

[PopcornJS 2012] http://popcornjs.org/, last accessed in September 2012.

[Potts 1952] R. Potts, “Some Generalized Order-Disorder Transformation,” *Proc. Cambridge Philosophical Society*, vol. 48, pp. 106-109, 1952.

[Raguram 2008] R. Raguram, J.M. Frahm, and M. Pollefeys, "A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus," *Proc. European Conference on Computer Vision*, pp. 500–513, 2008.

[Rainer 2012] B. Rainer, S. Lederer, C. Müller and C. Timmerer, "A Seamless Web Integration of Adaptive HTTP Streaming," *Proc. 20th European Signal Processing Conference*, 2012.

[Ren 2003] X. Ren, J. Malik, "Learning a classication model for segmentation," *Proc. IEEE International Conference on Computer Vision*, 2003.

[Robertson 1977] S. Robertson, "The probability ranking principle in information retrieval," *Journal of documentation*, vol. 33, pp. 294 – 304, 1977.

[Rooij 2007] O.D. Rooij, C.G.M. Snoek, and M. Worring, "Query on demand video browsing", *Proc. the ACM International Conference on Multimedia*, pp. 811-814, 2007.

[Rooij 2008] O.D. Rooij, C. G. M. Snoek, and M. Worring, "Balancing thread based navigation for targeted video Search," *Proc. ACM International Conference on Image and Video Retrieval*, pp. 485 – 494, 2008.

[Rooij 2009] O.D. Rooij, C.G.M. Snoek, and M. Worring, "MediaMill: guiding the user to results using the ForkBrowser," *Proc. of the ACM International Conference on Image and Video Retrieval*, 2009.

[Rooij 2010] O.D. Rooij and M. Worring, "MediaTable: a tool for categorizing multimedia collections," *Proc. ACM International Conference on Multimedia*, pp. 1633-1636, 2010.

[Rosten 2006] E. Rosten and T. Drummond, "Machine learning for highspeed corner detection," *Proc. European Conference on Comuter Vision*, 2006.

[Roy 1998] S. Roy and I. Cox, "A Maximum-Flow Formulation of the n-Camera Stereo Correspondence Problem," *Proc. IEEE International Conference on Computer Vision*, 1998.

[Rublee 2011] E. Rublee; V. Rabaud; K. Konolige, G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *Proc. IEEE International Conference on Computer Vision*, pp. 2564-2571, 2011.

[Russell 2008] B. Russell, A. Torralba, K. Murphy, W. T. Freeman, "LabelMe: a database and web-based tool for image annotation," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 157-173, May 2008.

[Sadeghi 2011] M.A. Sadeghi, and A. Farhadi, "Recognition using visual phrases," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1745-1752, 2011.

[Salembier 2000] P. Salembier and O. Avaro, "*MPEG-7: Multimedia Content Description interface*," 2000, http://gps-tsc.upc.es/imatge/_Philippe/demo/MPEG21_MPEG7.pdf

[Salton 1988] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Proceeding and Management*, p.513-523, 1988.

[Sande 2010] K.E.A. van de Sande, T. Gevers, C.G.M. Snoek, "Evaluating Color Descriptors for Object and Scene Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp.1582-1596, Sep. 2010.

[Sande 2011] K.E.A. van de Sande, J.R.R. Uijling and A.W.M. Smeulders, "Segmentation As Selective Search for Object Recognition," *Proc. IEEE International Conference on Computer Vision*; 2011

[Schaffalitzky 2002] F. Schaffalitzky and A. Zisserman, "Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?" *Proc. 7th European Conference on Computer Vision*, pp. 414–431, 2002.

[Schiele 2000] B. Schiele and J. Crowley, "Recognition without correspondence using multidimensional receptive field histograms," *International Journal of Computer Vision*, vol. 36, no. 1, pp. 31–50, Jan. 2000.

[Schneiderman 2004] H. Schneiderman and T. Kanade, "Object detection using the statistics of parts," *InternationalJournal of Computer Vision*, vol. 56, no. 3, pp.151–177, Feb. 2004.

[Schoeffmann 2011] K. Schoeffmann and M.D. Fabro, "Hierarchical video browsing with a 3D carousel," *Proc. 19th ACM International Conference on Multimedia*, pp. 827-828, 2011.

[Schoeffmann 2012a] K. Schoeffmann, D. Ahlström, and L. Böszörmenyi, "Video browsing with a 3d thumbnail ring arranged by color similarity," *Proc. International Conf. on Advances in Multimedia Modelling*, pp. 646-648, 2012.

[Schoeffmann 2012b] K. Schoeffmann and W. Bailer, "Video browser showdown," *SIGMultimedia Records*, vol.4, no. 2, pp. 1-2, Jul. 2012.

[Scott 2012] D. Scott, J. Guo, H. Wang, Y. Yang, F. Hopfgartner, and C. Gurrin, "Clipboard: A Visual Search and Browsing Engine for Tablet and PC," *Proc. International Conf. on Advances in Multimedia Modelling*, pp. 646-648, 2012.

[Shi 2000] J. Shi, J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, Aug. 2000.

[Shindler 2011] M. Shindler, A. Meyerson, and A. Wong, "Fast and accurate k-means for large datasets," *Advances in Neural Information Processing Systems* vol. 24, pp. 2375-2383, 2011.

[Sivic 2003] J. Sivic, and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos", *Proc. IEEE International Conference on Computer Vision*, 2003.

[Sivic 2006] J. Sivic, F. Schaffalitzky, A. Zisserman, "Object Level Grouping for Video Shots," *International Journal of Computer Vision*, vol. 67, no. 2, pp. 189-210, Apr. 2006.

[Sivic 2009] J. Sivic, and A. Zisserman, "Efficient visual search of videos cast as text retrieval," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 4; pp. 591-606, Apr. 2009.

[Smeaton 2006] A.F. Smeaton, P. Over, and W. Kraaij, "2006. Evaluation campaigns and TRECVid"; *Proc. 8th ACM International Workshop on Multimedia Information Retrieval*, pp. 321-330, 2006.

[Smith 1996] J.R. Smith, and S.F. Chang, "VisualSEEk: a fully automated content-based image query system," *Proc. ACM International Conference on Multimedia*, 1996.

[Snoek 2007] C.G.M. Snoek, M. Worring, D.C Koelma, and A.W.M Smeulders, "A learned lexicon-driven paradigm for interactive video retrieval," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 280-292, Feb. 2007.

[Snoek 2008] C. G. M. Snoek, M. Worring O.D. Rooij, K.E.A. van de Sande K., R. Yan, and A.G. Hauptmann, "VideOlympics: Real-Time Evaluation of Multimedia Retrieval Systems," *IEEE Trans. Multimedia*, vol. 15, no. 1, pp. 86-91, Jan. 2008.

[Snoek 2008b] C.G.M. Snoek, M. Worring, "Concept-Based Video Retrieval," F*oundation and Trend in Information Retrieval*, vol.2, no.4, pp. 215-322, 2008.

[SoundVision 2012] http://www.beeldengeluid.nl/en, last accessed in September 2012.

[Sparck 2000] K. Spärck Jones, S. Walker, and S. E. Robertson, "A Probabilistic Model of Information Retrieval: Development and Comparative Experiments," *Information Processing and Management*, vol. 36, no. 6, pp. 779-840, 2000.

[SQLite 2012] http://www.sqlite.org/, last accessed in September 2012.

[Sun 2008] Z. Sun, Y. Song, Y. Zheng, H. Yu, C. Jin, H. Lu, and X. Xue, “Fudan University: hierarchical video retrieval with adaptive multi-modal fusion,” *Proc. ACM International Conference on Content-Based Image and Video Retrieval*, pp. 549-550, 2008.

[Tapu 2011] R, Tapu, T. Zaharia, “A complete framework for temporal video segmentation,” *Proc. IEEE International Conference on Consumer Electronics*, 2011.

[TinEye 2012] http://www.tineye.com/, last accessed in September 2012.

[Tirilly 2010] P. Tirilly, V. Claveau, and P. Gros, “Distances and weighting schemes for bag of visual words image retrieval,” *Proc. ACM International Conference on Multimedia Information Retrieval*, pp. 323-332, 2010.

[Tola 2008] E. Tola, V. Lepetit, and P. Fua, “A fast local descriptor for dense matching,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

[Torr 2000] P. Torr, and A. Zisserman, “Feature based methods for structure and motion estimation,” *Vision Algorithms: Theory and Practice*, pp. 278-294, 2000.

[Tuytelaars 2000] T. Tuytelaars and L. Van Gool, “Wide baseline stereo matching based on local, affinely invariant regions,” *Proc. 11th British Machine Vision Conference*, pp. 412–425, 2000.

[Tuytelaars 2007] T. Tuytelaars, and C. Schmid, “Vector quantizing feature space with a regular lattice,” *Proc. IEEE International Conference on Computer Vision*, 2007.

[Tuytelaars 2008] T. Tuytelaars and K. Mikolajczyk, “Local Invariant Feature Detectors: A Survey,” *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177-280, 2008.

[Tuytelaars 2010] T. Tuytelaars, “Dense Interest Points,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* , pp. 2281-2288, 2010.

[Trooker 2012] http://www.trooker.com/, last accessed in September 2012.

[Truveo 2012] http://www.truveo.com/, last accessed in September 2012.

[Ullman 2001] S. Ullman, E. Sali, and M. Vidal-Naquet, “A fragment-based approach to object representation and classification,” *Proc. International Workshop on Visual Form*, pp. 85–100, 2001.

[Varma 2004] M. Varma and A. Zisserman, “Unifying statistical texture classification frameworks,” *Image and Vision Computing*, vol. 22, no. 14, pp.1175–1183, Dec. 2005.

[Vedaldi 2009] A. Vedaldi, V. Gulshan, M. Varma and A. Zisserman, “Multiple Kernels for Object Detection,” *Proc. IEEE International Conference on Computer Vision*, 2009.

[Veksler 2010] O. Veksler, Y. Boykov, P. Mehrani, “Superpixels and Supervoxels in an Energy Optimization Framework,” *Proc. European Conference on Computer Vision*, 2010.

[VideoSurf 2011] http://www.videosurf.com/, last accessed in August 2010.

[VidiVideo 2012] http://www.vidivideo.info, last accessed in September 2012.

[VLC 2012a] http://www.videolan.org/, last accessed in September 2012.

[VLC 2012b] http://wiki.videolan.org/Documentation:WebPlugin#Javascript_API_description, last accessed in September 2012.

[Vieux 2012] R.Vieux, J. Benois-Pineau, and J.P. Domenger, “Content based image retrieval using bag-of-regions,” *Proc. International Conference on Advances in Multimedia Modeling*, pp. 507-

517, 2012.

[Vijayanarasimhan 2011] S. Vijayanarasimhan, K. Grauman, "Efficient region search for object detection," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp.1401-1408, 2011.

[Vimeo 2012] http://vimeo.com/, last accessed in September 2012.

[Vodeo 2012] http://vodeo.tv/, last accessed in September 2012.

[Vrochidis 2008] S. Vrochidis, P. King, L. Makris, A. Moumtzidou, V. Mezaris, and I. Kompatsiaris, "MKLab interactive video retrieval system," *Proc. ACM International Conference on Content-Based Image and Video Retrieval*, pp. 563-564, 2008.

[W3C 2012] http://www.w3.org/TR/media-frags/, last accessed in September 2012.

[Wang 2004] X.H. Wang, D.Q. Zhang, T. Gu, and H.K. Pung, "Ontology based context modeling and reasoning using OWL," Proc. *2nd IEEE Conference on Pervasive Computing and Communications Workshops*, pp. 18-22, 2004.

[Wang 2007] L. Wang, D. Tjondrongoro, and Y. Liu, "Clustering and visualizing audiovisual dataset on mobile devices in a topicoriented manner," *Proc. 9th International Conference on Advances in Visual Information Systems*, pp. 310-321, 2007.

[WebM 2012] http://www.webmproject.org/, last accessed in September 2012.

[Weijer 2006] J. van de Weijer, T. Gevers, and A. Bagdanov, "Boosting Color Saliency in Image Feature Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 150-156, Jan. 2006.

[Wengert 2011] C. Wengert, M. Douze, and H. Jégou "Bag-of-colors for improved image search," *Proc. 19th ACM International Conference on Multimedia*, pp. 1437-1440, 2011.

[Wills 2006] J. Wills, S. Agarwal, and S. Belongie, "A feature-based approach for dense segmentation and estimation of large disparity motion," *InternationalJournal of Computer Vision*, vol. 68, no. 2, pp. 125-143, Jun. 2006.

[Wired 2010] http://www.wired.com/magazine/2010/08/ff_webrip/all/1, last accessed in September 2012

[Wu 2009] Z. Wu, Q. Ke; M. Isard, J. Sun, "Bundling features for large scale partial-duplicate web image search," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp.25-32, 2009

[Yang 2007] L. Yang, P. Meer, and D. Foran, "Multiple Class Segmentation Using A Unified Framework over Mean-Shift Patches," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[Yang 2007b] J. Yang, J. Yu-Gang, A.G. Hauptmann, and C.W. Ngo, "Evaluating bag-of-visual-words representations in scene classification," *Proc.ACM International Workshop on Multimedia Information Retrieval*, pp. 197-206, 2007.

[Yang 2008] N.C. Yang, W.H. Chang, C.M. Kuo, T.H. Li, "A fast MPEG-7 dominant color extraction with new similarity measure for image retrieval," *Journal of Visual Communication and Image Representation*, vol. 19, no. 2, pp. 92-105, Feb. 2008.

[Yeh 2005] T. Yeh, K. Grauman, K. Tollmar, and T. Darrell, "A picture is worth a thousand keywords: image-based object search on a mobile platform," *Proc. ACM Computer Human Interaction Conference*, pp. 2025-2028, 2005.

[YouTube 2012a] http://www.youtube.com/, last accessed in September 2012.

[YouTube 2012b] http://youtube-global.blogspot.fr/2012/03/looking-ahead-in-youtube-player.html, last accessed in September 2012.

[YouTube 2012c] http://www.youtube.com/t/press_statistics, last accessed in September 2012.

[Yuen 2009] J. Yuen, B. Russell, C. Liu, and A. Torralba, "Labelme video: Building a video database with human annotations," *Proc. IEEE 12th International Conference on Computer Vision*, pp. 1451-1458, 2009.

[Zaharia 2010] T. Zaharia, A. Vaucelle, T. Laquet, F. Preteux, "INVENIO: An MPEG-7 image indexing platform for content re-use within audio-visual production chains," *Proc. International Workshop on Content-Based Multimedia Indexing*, 2010.

[Zhe 2009] Y. Zheng, S. Neo, X. Chen, and T. Chua, "VisionGo: towards true interactivity," *Proc. ACM International Conference on Image and Video Retrieval*, 2009.

[Zhu 2009] C. Zhu, K. Li, Q. Lv, L. Shang, and R.P. Dick, "iScope: personalized multi-modality image search for mobile devices," *Proc. 7th International Conference on Mobile Systems, Applications and Services*, pp. 277-290, 2009.

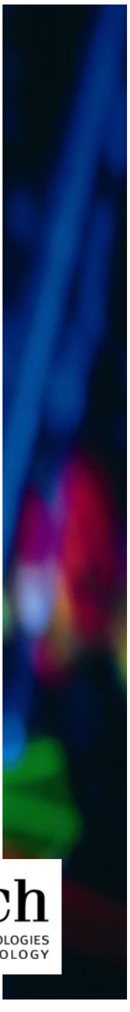
[Zhu 2012] C.Z. Zhu and S. Satoh, "Large vocabulary quantization for searching instances from videos," *Proc. 2nd ACM International Conference on Multimedia Retrieval, 2012.*

[Zin 2009] T.T. Zin, P. Tin, T. Toriu, H. Hama, "Dominant Color Embedded Markov Chain Model for Object Image Retrieval," *Proc. International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2009.

Object based visual content indexing and retrieval

Abstract: The issue of content-based video objects retrieval is growing in difficulty and becomes a mandatory feature for video search engines.The present thesis advances a user defined video object retrieval framework and brings two major contributions. The first contribution is a methodological framework for user selected video object instances retrieval, entitled **DOOR** (*Dynamic Object Oriented Retrieval*), while the second one concerns the support offered for video retrieval, namely the video navigation and retrieval system and interface and its underlying architecture. Under the DOOR framework, the objects comport a hybrid representation obtained by over-segmenting the frames, constructing region adjacency graphs and aggregating interest points. The identification of object instances across multiple videos is formulated as an energy optimization problem approximating an NP-hard problem. Object candidates are sub-graphs that yield an optimum energy towards the user defined query. Four optimization strategies are proposed: Greedy, Relaxed Greedy, Simulated Annealing and GraphCut. This object representation is further improved by the aggregation of interest points into a hybrid representation, where the similarity metric relies on a spectral matching technique integrating multiple types of descriptors. The DOOR framework is extended to large scale video archives through the use of Bag-of-Words representation enriched with a query definition and expansion mechanism based on a multi-modal, text-image-video principle. The proposed techniques are evaluated on multiple TRECVID video datasets prooving their effectiveness.The second contribution is related to the user support for video retrieval - video navigation, video retrieval, graphical interface - and consists in the **OVIDIUS** (*On-line VIDeo Indexing Universal System*) on-line video browsing and retrieval platform, integrating the DOOR framework. The OVIDIUS platform features hierarchical video navigation functionalities that exploit the MPEG-7 approach for structural description of video content. The major advantage of the proposed system concerns its modular architecture which makes it possible to deploy the system on various terminals (both fixed and mobile), independently of the exploitation systems involved. The choice of the technologies employed for each composing module of the platform is argued in comparison with other technological options.

Keywords: multimedia indexing, object retrieval, object representation, energy minimization, Greedy, Simulated Annealing, GraphCut, MPEG-7, Bag-of-Words, query expansion, graph matching, multi-modal search, TRECVID, multimedia indexing platform, video browsing, search and retrieval, HTML5, multi-terminal access.



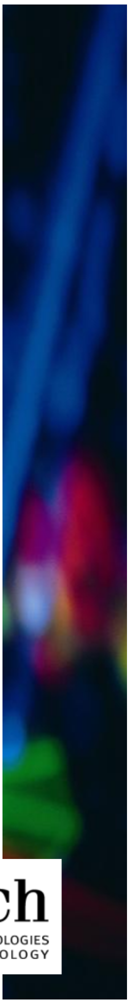
MINES ParisTech

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

Indexation et recherche de contenus par objet visuel

Résumé: La question de recherche des objets vidéo basés sur le contenu lui-même, est de plus en plus difficile et devient un élément obligatoire pour les moteurs de recherche vidéo. Cette thèse présente un cadre pour la recherche des objets vidéo définis par l'utilisateur et apporte deux grandes contributions. La première contribution, intitulée **DOOR** (*Dynamic Object Oriented Retrieval*), est un cadre méthodologique pour la recherche et récupération des instances d'objets vidéo sélectionnés par un utilisateur, tandis que la seconde contribution concerne le support offert pour la recherche des vidéos, à savoir la navigation dans les vidéo, le système de récupération de vidéos et l'interface avec son architecture sous-jacente.Dans le cadre DOOR, l'objet comporte une représentation hybride obtenues par une sur-segmentation des images, consolidé avec la construction des graphs d'adjacence et avec l'agrégation des points d'intérêt. L'identification des instances d'objets à travers plusieurs vidéos est formulée comme un problème d'optimisation de l'énergie qui peut approximer un tache NP-difficile. Les objets candidats sont des sous-graphes qui rendent une énergie optimale vers la requête définie par l'utilisateur. Quatre stratégies d'optimisation sont proposées: Greedy, Greedy relâché, recuit simulé et GraphCut. La représentation de l'objet est encore améliorée par l'agrégation des points d'intérêt dans la représentation hybride, où la mesure de similarité repose sur une technique spectrale intégrant plusieurs types des descripteurs. Le cadre DOOR est capable de s'adapter à des archives vidéo a grande échelle grâce à l'utilisation de représentation sac-de-mots, enrichi avec un algorithme de définition et d'expansion de la requête basée sur une approche multimodale, texte, image et vidéo. Les techniques proposées sont évaluées sur plusieurs corpora de test TRECVID et qui prouvent leur efficacité.La deuxième contribution, **OVIDIUS** (*On-line VIDeo Indexing Universal System*) est une plate-forme en ligne pour la navigation et récupération des vidéos, intégrant le cadre DOOR. Les contributions de cette plat-forme portent sur le support assuré aux utilisateurs pour la recherche vidéo - navigation et récupération des vidéos, interface graphique. La plate-forme OVIDIUS dispose des fonctionnalités de navigation hiérarchique qui exploite la norme MPEG-7 pour la description structurelle du contenu vidéo. L'avantage majeur de l'architecture propose c'est sa structure modulaire qui permet de déployer le système sur terminaux différents (fixes et mobiles), indépendamment des systèmes d'exploitation impliqués. Le choix des technologies employées pour chacun des modules composant de la plate-forme est argumentée par rapport aux d'autres options technologiques.

Mots clés: indexation multimédia, recuperation d'objets, représentation d'objet, minimization d'énergie, Greedy, recuit simulé, GraphCut, MPEG-7, Sac-de-Mots, extension de requête, appariement de graphes, recherche multi-modale, TRECVID, plates-formes d'indexation multimédia, navigation de vidéos, recherche et récupération, HTML5, accès multi-terminal.



MINES ParisTech

ParisTech INSTITUT DES SCIENCES ET TECHNOLOGIES PARIS INSTITUTE OF TECHNOLOGY