N° d'ordre : 4244

THÈSE

PRÈSENTÉE

L'UNIVERSITÉ BORDEAUX I

ÉCOLE DOCTORALE DE MATHEMATIQUES ET D'INFORMATIQUE

Par **Rémi VIEUX**

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : INFORMATIQUE

Extraction de Descripteurs Pertinents et Classification pour le Problème de Recherche des Images par le Contenu

Soutenue le : 30 Mars 2011

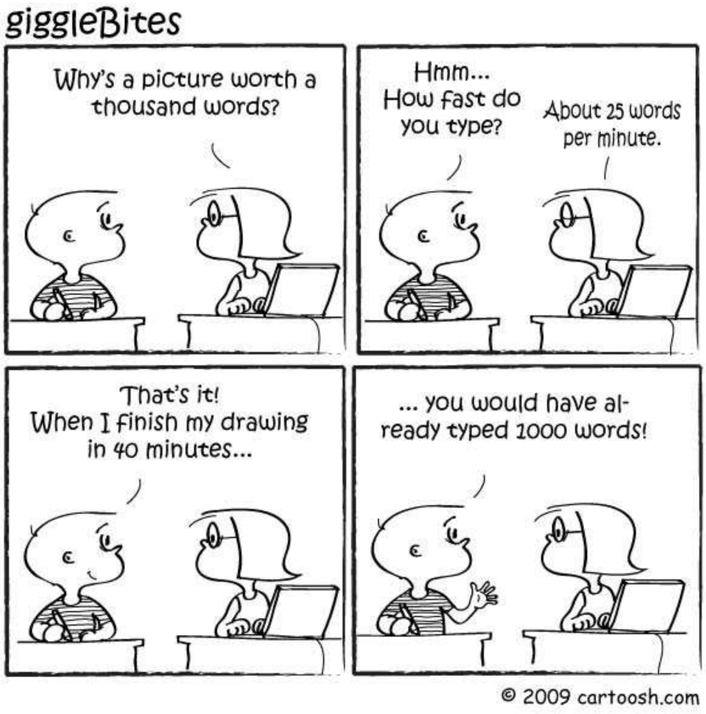
Après avis des rapporteurs :

M. Bernard MERIALDO	Professeur, Institut EURECOM	
M. Yannis AVRITHIS	Senior Researcher, Nat. Tech. Univ. of Athens	

Devant la commission d'examen composée de :

M. Achille BRAQUELAIRE	Professeur, Université Bordeaux 1	Directeur de thèse
Mme Jenny BENOIS-PINEAU	Professeur, Université Bordeaux 1	Co-Directrice de thèse
M. Jean-Philippe DOMENGER	Professeur, Université Bordeaux 1	Examinateur
M. Georges QUÉNOT	Chercheur CNRS, LIG	Président

– 2011 –


giggleBites
Why's a picture worth a thousand words?
Hmm... How fast do you type?
About 25 words per minute.
That's it! When I finish my drawing in 40 minutes...
... you would have already typed 1000 words!
© 2009 cartoosh.com

A Gaelle et Elisa

Remerciements

Je tiens bien sûr a remercier mes directeurs de thèse, Mme Jenny Benois-Pineau et M. Achille Braquelaire, pour m'avoir accueilli dans leur équipe, et m'avoir fait confiance dans le cadre des missions qui m'ont été données lors du projet **X-MEDIA**.

Merci à M. Jean Philippe Domenger, qui fut mon plus fidèle partenaire lors des meetings dans les coins les plus reculés d'Europe, et avec qui j'ai encore la chance de collaborer sur de nouveaux projets toujours plus passionnants.

Je remercie M. Bernard Merialdo et M. Yiannis Avrithis, pour avoir accepté d'être les rapporteurs de ces travaux de thèse, et le temps qu'ils y ont consacré.

Je n'oublie pas l'ensemble des partenaires du projet **X-MEDIA**, pour l'expérience que j'ai acquise, les nombreux voyages que j'ai pu effectuer et la bonne ambiance dans laquelle ces années se sont déroulées.

Enfin, je remercie très chaleureusement Claire, Nico(Maïkeul), Hugo, Ronan, Svebor, Boris, Eliana, Chris, Petra et Daniel (1 et 2) pour avoir partagé bureau, repas, soirée, cafés, idées, rires, grippes, . . . Merci à ceux que j'ai oublié.

Seeking for Relevant Descriptors and Classification for Content Based Image Retrieval

Abstract: The explosive development of affordable, high quality image acquisition devices has made available a tremendous amount of digital content. Large industrial companies are in need of efficient methods to exploit this content and transform it into valuable knowledge. This PhD has been accomplished in the context of the **X-MEDIA** project, a large European project with two major industrial partners, FIAT for the automotive industry and Rolls-Royce plc. for the aircraft industry. The project has been the trigger for research linked with strong industrial requirements. Although those user requirements can be very specific, they covered more generic research topics. Hence, we bring several contributions in the general context of Content-Based Image Retrieval (CBIR), Indexing and Classification.

In the first part of the manuscript we propose contributions based on the extraction of global image descriptors. We rely on well known descriptors from the literature to propose models for the indexing of image databases, and the approximation of a user defined categorisation. Additionally, we propose a new descriptor for a CBIR system which has to process a very specific image modality, for which traditional descriptors are irrelevant. In the second part of the manuscript, we focus on the task of image classification. Industrial requirements on this topic go beyond the task of global image classification. We developed two methods to localize and classify the local content of images, *i.e.* image regions, using supervised machine learning algorithms (Support Vector Machines). In the last part of the manuscript, we propose a model for Content-Based Image Retrieval based on the construction of a visual dictionary of image regions. We extensively experiment the model in order to identify the most influential parameters in the retrieval efficiency.

Keywords: **X-MEDIA**, CBIR, Indexing, Image Classification, SVM.

Discipline: Computer Science

LaBRI (UMR CNRS 5800)
Université Bordeaux 1
351, cours de la Libération
33405 Talence Cedex (FRANCE)

Contents

List of Main Notations

Colour Spaces

HMMD Hue Max-Min-Difference color space. , page 13

HSV Hue, Saturation, Value color space. , page 12

Luv Color space which attempt to perceptual uniformity. , page 13

RGB Red, Green, Blue color space. , page 12

YUV Color space defined in terms of Luminance (Y) and Chrominance (u,v). , page 13

Visual Descriptors

ART MPEG-7 Angular Radial Transform Descriptor, see equation (2.11), page 21

CLD MPEG-7 Color Layout Descriptor , page 15

CSD MPEG-7 Color Structure Descriptor, see equation (2.3), page 15

CSS MPEG-7 Curvature Scale Space descriptor), see equation (2.10), page 21

DCD MPEG-7 Dominant Color Histogram, see equation (2.2), page 14

EHD MPEG-7 Edge Histogram Descriptor, see equation (2.5), page 18

H Histogram, see equation (2.1), page 14

HTD MPEG-7 Homogeneous Texture Descriptor, see equation (2.5), page 17

LBP Local Binary Pattern, a texture descriptor by Ojala *et al.* [127], see equation (2.7), page 20

SCD MPEG-7 Scalable Color Descriptor , page 15

TBD MPEG-7 Texture Browsing Descriptor, see equation (2.4), page 16

SIFT Scale Invariant Feature Transform , page 22

SURF Speeded Up Robust Features, see equation (2.20), page 28

Industrial Partners - Use cases

CRF	Centro di Richercha FIAT, page 2
RR	Rolls Royce plc., page 2
CSF	Competitor Scenario Forecast, **X-Media** use case from FIAT , page 2
RREV	Rolls Royces Experimental Vibration use case, page 5
RRIR	Rolls Royce Issue Resolution use case, page 4

Performance Metrics

AP	Average Precision, see equation (2.50), page 39
F_1	F1-Score, see equation (2.48), page 39
F_β	F-Score, see equation (2.48), page 39
JC	Jaccard coefficient, see equation (2.44), page 38
MAP	Mean Average Precision, see equation (2.51), page 40
N	Set of Negative examples , page 38
$\mathrm{P}_{(\mathrm{n})}$	Precision at n, see equation (2.49), page 39
P	Set of positive examples (classification), page 38
P	Precision, see equation (2.47), page 38
R	Recall, see equation (2.46), page 38
FN	False Negative, page 38
FP	False Positive, page 38
TN	True Negative, page 38
TP	True Positive, page 38
YC	Yule coefficient, see equation (2.45), page 38

Distance Metrics

$\mathbf{\Sigma}^{-1}$	Inverse Covariance Matrix, see equation (2.29), page 32
L_∞	Chebyshev distance, see equation (2.28), page 32
χ^2	χ^2 statistics, see equation (2.32), page 33
L_2	Euclidean distance, see equation (2.25), page 32
L_1	Manhattan distance, see equation (2.26), page 32
L_p	Minkowski distance, see equation (2.27), page 32
EMD	Earth Movers Distance, see equation (2.36), page 33

Chapter 1

Introduction

Figure 1.1: A horse from Lascaux caves. Replica in the Brno museum Anthropos. Extracted from Wikimedia Commons.

> « *One picture is worth more than ten thousand words.* »

It is not a surprise that this quote is the very first sentence in the book by Gonzales and

Woods, *Digital Image Processing* [66]. Images, pictures and drawings have been used by individuals to express ideas, feelings or to transmit heritage even before formal written language (figure 1.1). The democratization of affordable, high quality acquisition devices (cameras, phones, scanners) and the drastic reduction of memory and storage cost have changed our traditional connection with multimedia information. In companies, the amount of data produced has dramatically increased. These data need to be interpreted and retrieved. While in the past, medium size, mainly textual, centralized archives used to be the only sources for knowledge management, nowadays large companies handle large quantities of multimedia information in distributed archives. Intranets in large enterprises have reached the dimension of mini Webs. The increased use of the World Wide Web as a source of information narrows down the boundary between *intra* and *inter*-net. This dramatically increases the dimension of the information space. Nowadays, the information is distributed across different media and sources of visual information such as images available in digital form are unavoidable: they allow fast visual apprehension of phenomena, which are described, or not, in accompanying texts and annotations.

1.1 The context of the work

1.1.1 X-Media project

This PhD has been accomplished in the context of the **X-Media**[1] project. **X-Media** addresses the issue of knowledge management in complex distributed environments. Its goal is to study, develop and implement large scale methodologies and techniques for knowledge management able to support sharing and reuse of knowledge that is distributed in different media (images, documents and data) and repositories (data bases, knowledge bases, document repositories, etc.).

During the time period of the **X-Media** project, we have experienced close collaboration with industrial partners coming from different backgrounds and having different requirements. The research problems that have been tackled in this PhD are initially driven by the industrial requirements defined during the project. The industrial partners were Rolls Royce plc. (**RR**) for the aeronautic industry, and Centro di Richercha **FIAT** (**CRF**) for the car industry. In such large organizations, several departments are potentially interested in the results of a research project such as **X-Media**. The requirements from the industrial point of view have been gathered into four use cases (two for **RR** and two for **CRF**). Three of these use cases contained image related material. We report these use cases in the following sections.

1.1.1.1 Competitor Scenaria Forecast - (CRF use case)

Scenario description Competitor analysis represents a key success factor for every kind of organization. This particular activity should provide a constant monitoring of the existent competitors' products, particularly analyzing technological innovations and solutions. Considering car manufacturing environment, it is vital to understand market trends and

[1] `http://www.x-media-project.org/`

fashions, in order to follow the flow or, even better, trying to anticipate customer needs providing smart and usable new solutions.

Competitor analysis has the goal to monitor the market and its voices or rumors, to be daily informed of its evolution, specifically concerning:

- Trends, *e.g.* evolution of shapes of products or parts
- New products: concepts, prototypes, new productions
- New components
- New materials that become available or are adopted for specific products

This daily activity is done keeping in mind the final goals:

- Cost reduction
- Weights reduction
- Performances
- Comfort (usability, driver, acoustic, vibration)
- Low fuel consumption
- Reliability
- Innovation (style,. . .)
- Safety
- Mobility enhancement

In **FIAT** Auto, there are many different kinds of activities connected with Competitors Analysis. **FIAT** has a Competitor Analysis (CA) department that leads two main data acquisition processes:

1. In-house testing and or dismantling of competitors' cars (synchronous process): this process consists in purchasing, renting or exchanging with competitors' cars in order to carry out performance testings, components measurements, photo shooting and car dismantling (if the car is purchased).

2. Collecting any kind of world-wide information related to competitors (asynchronous process): this includes information extraction from competitors' press release, web monitoring and screening from a predefined list of reliable websites, and information extraction from specialized magazines.

Concerning the most time and cost consuming activity (car dismantling), CA department processes on average 5 competitors' cars per year. The dismantling process consists of a really articulated activity that reduces the car into all its components, until the individual component (*e.g.* screw). Moreover 15 to 20 competitors' cars are rented and tested every year. Considering all the aforementioned activities, CA department creates and makes available a considerable amount of data. Nowadays they have to manage a growing rate of 15 GB/per year. All these data and the outputs produced analyzing them (daily reports) are stored in a unique central database where each information is classified with a unique key, car manufacturer, car model and car version. Other information is organized in folders, mainly stored in the company Document Management Systems and classified only by car name. Other data may reside in the local repository managed by each department collecting and producing it. More than 200 users from different **FIAT** departments take advantage of the data collected by the competitor analysis department, many of them also contribute in collecting and producing data.

Industrial Requirements The synchronous and asynchronous processes produces a large amount of image material, coming from the Internet (news, articles, reports) or from photos taken directly during the dismantling process. Currently, there exists no system to properly store and retrieve images in a meaningful way. In the best case, the images collected from automotive websites are saved in a shared folder named after the car model name. Following the fact that even such a simplistic approach of image storing is time consuming, it has been identified as a potential breakthrough to be able to recognize and automatically classify images according to some meta-data, such as the car view (interior/exterior), car components (*e.g.* air ducts, gear lever, steering wheel, . . .), brand, model.

1.1.1.2 Issue Resolution - (RR use case)

Scenario description During the product development phase of an engine program it is essential that every opportunity is taken to identify potential shortfalls in the products capability and ensure effective resolution prior to entry into service. The issues identified on one product are also a valuable source of lessons for the design phase of subsequent products and the operational phase of other existing products. The sources of knowledge information on the issue, the root cause understanding and the problem solution may all be represented in a variety of media (photographic, video, text, graphical). In many instances the process of investigation and solution will require the involvement of multiple organizations. Following entry into service, additional issues (and investigations) will continue to occur. These too require effective problem management and may identify weaknesses in the original development process (where issues were not identified prior to service operation). In order to progress from an issue to resolution, it is necessary for the team working on the problem to work in data, information and knowledge levels. As they progress the problem, there comes a need to communicate progress out to the other stakeholders in the issue. These may include the engine project, customer interfaces and the customers themselves. These communications generally need to be at an informational level (where are we, when will it be closed, what do I need to do) and in some cases knowledge (what are the implications of this issue, what could it lead to, where does it sit in priority terms compared to other issues). In parallel with the progression of the basic issue and after its completion, the problem, its understanding, its root cause and validation strategy provide a valuable knowledge resource for other existing engine types and future product design. The root cause can cover both the problem root cause and, if applicable, the root cause for it failing to have been identified and resolved earlier in the design process.

Industrial Requirements Finding the root cause of an engine failure is a long, time consuming process that requires the conjoint work of several people or groups of people. During this process, several sources of information are investigated and evidence is collected and shared among the people in the organisation across different media: emails with attachments, multimedia documents (presentations), photos, technical reports or engine test reports (raw data). The **X-MEDIA** system has been foreseen as a potential breakthrough within this process, by providing a uniform and coherent way to create, store, retrieve and share knowledge across users, time and space. As far as the requirements for image analysis are concerned, a desirable feature from the industrial point of view would be the ability to recognize potential defects or defects causes from the engine pictures.

1.1.1.3 Experimental Vibration - (RR use case)

Scenario description Every component or structure has a series of natural frequencies and associated mode-shapes. If the structure is excited at the same frequency as the natural frequency, then the structure will resonate with a response many times greater than if excited at a non-natural frequency. Gas turbines like many other lightweight high performance structures have to be carefully designed to ensure the resonance of the structure does not cause loss of structural integrity. This failure mechanism is known as high cycle fatigue. To complete this engineering task, it is important to understand the mode-shape of the component (image), the type of test that the component has been subject to (test schedule and test log text documents) and the vibration characteristics seen in the measured data (data and image). We must also understand the maximum response seen in the engine and the maximum safe running amplitude to ensure we do not see failures in service. A theoretical finite element model is often required to get all this information together in a coherent manner. There are many different components within a gas turbine and many different instrumentation techniques that can be used to measure the vibration characteristics of a component. The Experimental Vibration (EV) department forms part of the Operations, Engineering and Technology business. The department is primarily a service function to both military and civil aerospace, although does complete work for marine and other parts of the Rolls-Royce business. The department has approximately 40 staff. They can be broken into Project Vibration Engineers and laboratory Technologists. The Project Vibration Engineers are located on Bristol and Derby. They routinely define and interpret engine and laboratory experiments to understand the vibration characteristics of key components. The data is used to make safety decisions and directly used in the certification of an engine type. This work is mainly in support of new products, although some work is to understand change in manufacturing process or product improvement. The department has its own laboratory (located in Derby) that routinely tests engine components, often to destruction. The department also has very close links with the engine test organisation that is responsible for completing the whole engine experiments. Rolls-Royce is a global organisation and these engine tests can be performed in many different parts of the world, ranging from Europe (*e.g.* England, Spain), to the United States of America (*e.g.* Florida, Indiana), across to the Far East (*e.g.* Japan).

Industrial Requirements Rolls-Royce engineers have considerable knowledge and expertise in the field of vibration understanding. This is based on the $50+$ years of testing gas turbines. However this knowledge takes considerable time to accumulate for a new Vibration Engineer. It is intended that the **X-Media** Experimental Vibration Use Case will present historical knowledge to *novice* graduate engineers, enabling them to act with the authority of engineers who have in excess of 15 years experience, *i.e.* those with experience of 3 or 4 engine design cycles. Such a system would rely extensively on image modality: images produced by the new engine tests are confronted to historical data, *i.e.* images from previous tests. The expert could browse historical knowledge using images, and infer from previous cases the reliability of the newly tested engine components.

1.1.2 Research context

The formal introduction of image processing and analysis algorithms in industrial environment was in the development of very specific, control oriented or quality oriented applications such as the automated visual inspection of manufactured goods [65]. The emergence of new imaging facilities, easiness of creating image content and the access to external sources have triggered new challenges for the management of image data in industrial environment. The industrial requirements defined in the previous section can be expressed in terms of well know topics in the research community, namely **Content Based Image Retrieval** (CBIR) and **Image Classification**.

CBIR has been an extensively studied topic in the last 20 years [153]. Yet, the research in this field is still growing with the emergence of new needs due to the democratization of digital content creation. Nowadays, CBIR systems are growing into much broader systems that trigger the association of researchers from weakly related fields. The paper from Datta *et al.* [40] presents a survey of almost 300 key theoretical and empirical contributions in the current decade related to image retrieval and automatic image annotation The goal of CBIR is to perform efficient retrieval on — possibly large scale —, image data-sets based on the image content. In the case of industrial domains, images can come from highly heterogeneous environments. In this context, the adequate descriptor of image content is of primary importance. This is a key-point in general CBIR research, as illustrated by Datta *et al.* in figure 1.2.

One of the difficulties of the formulation of an appropriate image signature is defined as the *semantic gap*:

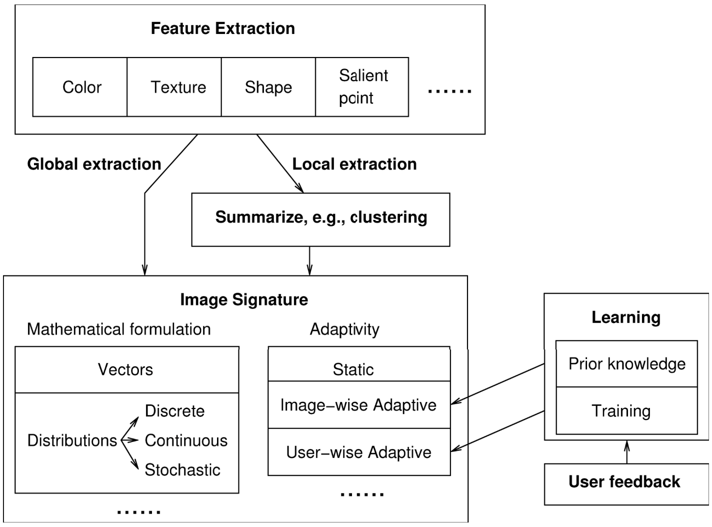


Figure 1.2: Overview of image signature formulation. Image from [40], ©ACM

> « *The semantic gap is the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation [153].* »

The goal of image classification is to train a *classifier* to be able to recognize a particular object or object class. The topic of image classification is linked with both machine learning and CBIR. Image classification relies intensively on machine learning algorithms as classifiers. Like CBIR, image classification needs an adequate description of the image content, *i.e.* an adequate signature, which is fed to the classifiers. Intuitively, image classification and CBIR are linked by the concept of *similarity* in the description space. Moreover, CBIR systems can make use of image classification to provide keyword or category based entry to image retrieval.

The relationships between the industrial requirements gathered in the **X-MEDIA** use cases and the broader research topics of CBIR and image classification are the following:

- **Competitor Scenario Forecast Use Case (CSF):** the task of recognizing car views or car components is a typical image classification task. One of the challenging aspects of this task to handle a multi-class detection and localisation of components, as several car components can be present in an image.

- **Issue Resolution Use Case (IR):** the task of recognizing defects from image is also an image classification task. However, after intensive discussion with the use case owners, it is clear that the recognition of defects not only relies on purely visual inspection of the components, but on the background expertise of skilled engineers. Methods employing evidences gathered across the different media, with a strong emphasize on text should be employed and are beyond the scope of this work. Nevertheless, CBIR can be useful in this scenario for the defects that can be identified by a specific visual

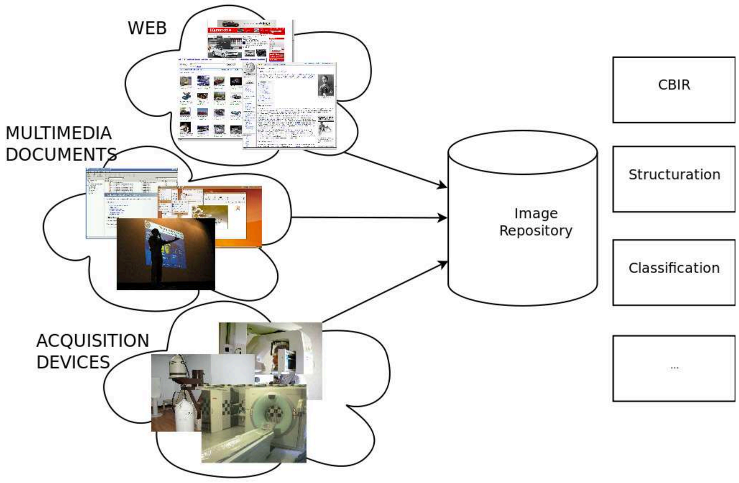


Figure 1.3: Challenges for image management in large industrial environments

appearances (*e.g.* burns, leaks, ...). The system could retrieve *similar* pictures, which are linked with the history of investigation of the engine failure cause. Ideally, such a system should allow to perform local queries, *i.e.* queries using just a part of the image where the defect is identified by an expert.

- **Experimental Vibration Use Case (EV):** The image requirements for this use case are a crucial step toward narrowing the gap between novice and experienced engineers. It is expected that the images collected during the vibration testing can be used to access historical data, *i.e.* the history of tests of all the components that have been previously processed, in order to quickly identify the reliability of the newly tested components. This is a typical CBIR task, where the **X-MEDIA** system keep the links between the image data and the textual and data reports.

An additional implicit requirement from the industrial partners is the need for structuring the visual data. At the moment, their best practice in terms of image management consists in a manual categorization of images into folders. Automatic *structuring* of image databases based on visual features, *e.g.* for image browsing, is a desirable feature. The challenges for image management in large industrial companies are illustrated in figure 1.3. Companies *gather* image content from very heterogeneous sources such as the web, internal documents or their own acquisition devices. All these sources fill a huge image repository, on top of which research applications can be built. This includes, but does not limit to, CBIR, classification and structuring of the image database.

1.2 Structure of the manuscript

The **X-MEDIA** project has been a powerful opportunity to investigate the benefits of image management process within the context of large industrial environments. From the image analysis point of view, the requirements from the industrial partners are challenging and diverse. From the industrial point of view, discussing the requirements was also a good opportunity to see the benefits of more powerful image management systems. Hence, we identified that an efficient management of visual data in large organization should rely on a collection of tools for the organization, classification, indexing and retrieval schemes for image databases. All such aspects have been covered in this work.

The chapter 2 is an introduction to several concepts that have been used throughout this work. The core of general research in CBIR and image classification is the adequate description of the image content. We start chapter 2 by reviewing the state-of-the-art in low level feature extraction and matching. We continue this chapter by introducing the concept of **data clustering** a key notion referred throughout this manuscript. Finally, we present the metrics for the evaluation of the performances of CBIR and classification systems.

The organisation of the next chapters follows a progression in our work concerning the level of granularity we have been working on at the image level. In chapter 3, we focus on systems which operates at the full image level, *i.e.* using global image descriptors. The chapter is divided in three sections. In section 3.1, we propose a system for the indexing of images based on multiple clustering. Rather that responding to a specific **X-MEDIA** use case requirement, this system is a generic solution for a flexible indexing mechanism. In section 3.3, we present a CBIR system that was designed having in mind the specific requirements of the **RR** Experimental Vibration use case. In this use case, the nature of the images processed makes CBIR systems relying on traditional low level features useless. We propose an alternative feature representation based on the Radon Transform and an appropriate similarity measure in this feature space. In section 3.2, we propose an algorithm that finds the best approximation of a user-defined categorization, based on low level descriptors. The proposed approach is based on solving the well know assignment problem.

Chapter 4 narrows down the analysis of the image content to local content. We adopted two approaches for this task. In section 4.3, we propose a two-stage object detection and localisation schema. The localization of objects is first achieved using an algorithm for Regions Of Interest (regions-of-interest) detection. Then candidate regions-of-interest are submitted to a classifier build to recognize particular object classes. In section 4.4, a different approach is adopted. In this section, we propose a multi-class object detection and localization algorithm which operates on segmented regions of the image. Using image segmentation as a pre-processing step, the whole image is processed but each region is classified independently which enables the localisation of different objects, or different instances of the same object class.

In chapter 5, we propose a CBIR system relying on an hybrid local and global characterization of the image content. In this work, we are among the rare representatives of the research community using region-based features in the now popular Bag-Of-Words (BoW) approach. We present the Bag-of-Regions(BoR) framework and demonstrate its efficiency for both global and localized CBIR.

Finally, we conclude this manuscript in chapter 6, by drawing the strength and weaknesses of our approaches and proposing openings for future works.

Chapter 2

State of the art in image content description

2.1 Introduction

In this chapter, we introduce the basic elements that we use throughout this manuscript. As a fundamental step, the extraction of features is the process of computing, from the image, a descriptor which is representative of the image content. In section 2.2, we review the state of the art of visual features in terms of color, texture, shape and the most recent keypoint-based features.

Once the image content is expressed in terms of visual features, the matching between the features is usually achieved using some standard similarity or distance measure. They are introduced in section 2.5.

In section 2.4 we introduce the notion of **data clustering** and present some algorithms that are used throughout this manuscript.

Finally, we describe in section 2.5 the standard methodology and metrics for the evaluation of classification and retrieval systems.

2.2 Visual Features

Since the emergence of computer imaging, the researchers have always been interested in finding an accurate description of the content of images. However, as pointed out by Smeulders *et al.* [153], one of the main difficulties of such a task relies on the *semantic gap*: Visual features are useful for Content-Based Image Retrieval (CBIR) applications, image classification but also for extracting the objects that compose an image. The large range of applications of visual features combined with the problems of semantic gap have led the way to the development of a lot of different visual features. We can consider four different types of visual features:

1. Color features
2. Texture features
3. Shape features
4. Local (or keypoint-based) features

In this section, we review some of the literature about visual feature extraction, with a special focus on the MPEG-7 standard. The MPEG-7 standard defines a set of nine visual descriptors for 2D image representation. Although MPEG-7 has never become a wide-spread standard for content description, the results of the fundamental work for content descriptors has become widely used for content based indexing and retrieval. Most of our approaches have been based on the descriptors from the MPEG-7 standard.

2.2.1 Colour features

2.2.1.1 Preliminaries: Colour spaces

Before going further in the description of colour features it is necessary to remind the notion of colour spaces. A colour space is a mathematical model that enables the representation of colors, usually as a tuple of color *components*. There exists several such models, some motivated by the application background, some by the perceptual background of the human vision system.

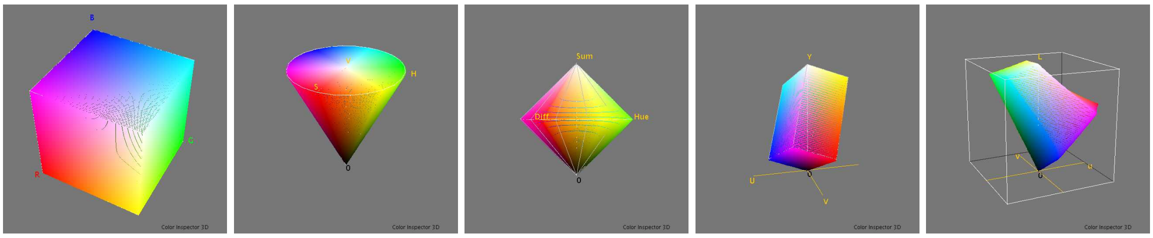
The most commonly used color space is the **RGB** space, where a color is defined by the additive amount of the primary colors **R**ed, **G**reen and **B**lue. The design of this color space is closely related to the way the colors are reproduced on hardware devices such as computer screens, television, *etc*. A classic representation of the **RGB** color space is the 3-dimensional cube, where each axis correspond to the amount or red, green and blue components (figure 2.1(a)).

The **HSV** (for **H**ue, **S**aturation, **V**alue) color space was designed in an attempt to describe the perceptual color relationships more accurately than **RGB**, while remaining simple. It is defined by a unique, non-linear mapping from the **RGB** space. The colors in **HSV** are traditionally represented in a 3D-cone (figure 2.1(b)). The hue takes values from 0 to 360 representing the color wheel. The saturation represented by the distance from the center of a circular cross-section of the cone, corresponds to the purity of the color (pure red, green, yellow, …). The value component corresponds to the brightness/darkness of the color. It is located on the color cone as the distance from the pointed end of the cone. Saturation and Value usually take values in the interval $[0, 1]$.

The **HMMD** (**H**ue-**M**ax-**M**in-**D**ifference) Color space is yet another color space obtained from a non-liner transformation of the RGB color space. It is defined by the MPEG-7 standard as beeing closer to a perceptually uniform color space. Hue as the same semantics as in the **HSV** color space. Max, which is the max over the **RGB** values, specifies how much black color is present. Min (min of **RGB** component) specifies how much white color is present. Diff (Max - Min) specifies how much a color is close to pure colors. Sum ((Max + Min)/2) specifies the brightness of the color. Although 5 components can be extracted, {Hue, Max, Min} or {Hue, Diff, Sum} are sufficient to specify any color point in this color space.

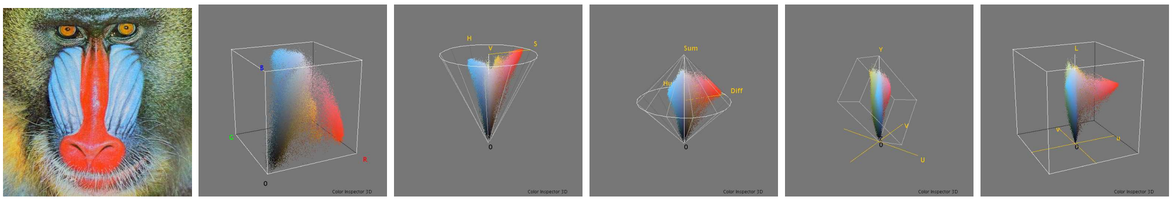
The emergence of the color in television has motivated the usage of color spaces that separate the pixel luminance (brightness) and chrominance (color) values, such as **YUV** (figure 2.1(d)). Such a definition of the color enabled the cohabitation of black and white and color for analog television. **YUV** is also the standard in video encoding, since the chrominance component can be encoded using a reduced bandwidth without loss of perceptual quality.

Finally, some efforts have been made in order to build color spaces that attempt perceptual uniformity. One such color space is **Luv** (figure 2.1(e)). **Luv** was designed so that the perceptual color difference can be computed in the **Luv** space using the euclidean distance.



(a) **RGB** (b) **HSV** (c) **HMMD** (d) **YUV** (e) **Luv**

Figure 2.1: Graphical representation of different color spaces. Figures created with Color Inspector 3D plugin for ImageJ.



(a) Original Image (b) **RGB** (c) **HSV** (d) **HMMD** (e) **YUV** (f) **YUV**

Figure 2.2: Color histogram of the baboon image in different color spaces. Figures created with Color Inspector 3D plugin for ImageJ.

2.2.1.2 Colour descriptors

Human visual perception mostly relies on colour information. Colour descriptors aim somehow at simulating this ability by describing and discriminating images through a set of accurate colorimetric properties. Such description tools are rather effective, easy to extract and meaningful. Most of them require the use of a particular colour space to be fully efficient.

Histogram The most common way to represent the distribution of colours (or gray level) in an image (or an image region) is the **histogram**. The histogram of an image is a discrete function (cf equation 2.1) that associates a color with its frequency in the image. Histograms are usually computed using a quantization of the color space: a histogram bin counts the number of pixels falling into its range of gray level or color values. The quantization can be linear (a bin represents a fixed number of values) or non-linear. The quantization steps can be different on each color component (for example, it is wise to use a bigger number of bins for the Luminance component of color spaces such as **YUV** and **Luv**). If we note N the total number of observations (*e.g.* the number of pixels in an image or a region), k the number the histogram bins, the histogram $\mathbf{H}$ meets the following condition:

$$\sum_{i=1}^{K} \mathbf{H}(i) = N \tag{2.1}$$

Histograms are invariant under geometrical transformations(*e.g.* translation, rotation, ...).

Moments Colour moments are another tool to characterize and compare colour distribution of images. The first order moment (mean), the second-order moment (variance) and the third order moment (skewness) have been proved to be both efficient and effective to represent the colour distribution of images [157]. The mean provides the average value of pixels of the distribution whereas variance consists in the square root of the variance of the distribution. Skewness yields an evaluation of the asymmetry degree of the distribution. The fourth order moment (kurtosis) is also used but merely in the context of texture analysis. The characterization of the colour distribution of an image through its first three moments is very compact as it contains exactly 9 values (3 moments computed for each component of the colour) [84, 102].

Dominant Color Descriptor The **Dominant Color Descriptor** (DCD) [43] is the first of a set a 8 descriptors described in the MPEG-7 standard [108] for still images. DCD provides a compact representation of salient colors in the image or an image region. Following the MPEG-7 notations, the DCD is defined as (equation 2.2):

$$\text{DCD} = \Big\{(\mathbf{c}_i, p_i, v_i), s\Big\}, \quad (i = 1, 2, \dots, N) \tag{2.2}$$

N is the number of dominant colors, $\mathbf{c}_i$ is a vector of the color components values in a particular color space, $p_i \in [0, \dots, 1], \sum_i p_i = 1$ is the percentage of pixels in the image corresponding to the color $\mathbf{c}_i$. v_i is an optional parameter describing the color variance of the color values of the pixels with respect to their representative color. Finally, s is a single number representing the spatial coherency of colors in the image. Up to 8 dominant colors can be computed for an image. MPEG-7 standard defines the semantics of the descriptors without bounding the implementation to a particular algorithm. For DCD, the reference implementation of MPEG-7 uses the Generalized Lloyd Algorithm [101] in the **Luv** color space to compute the dominant colors. Another implementation was proposed by Yang *et al.* [178]. DCD is mainly used to retrieve images with similar colours in large databases.

Scalable Color Descriptor The **Scalable Color Descriptor** (SCD) [14] is a 256-bin color histogram computed on the **HSV** color space with 16 bins for Hue, 4 bins Saturation and 4 bins for Value component. The standard imposes **HSV** color space and a 256-bin histogram to ensure interoperability. The focus for this descriptor was put in the scalability and storage efficiency, while remaining accurate as a descriptor for similarity search. Scalability is achieved by encoding the histogram using a Haar transform. The number of coefficients retained and the number of bits per coefficient are the two coefficients setting up the scalability of the descriptor. Summing adjacent bins in the histogram reduces the original histogram size by half. The sum and difference between adjacent histogram bins are respectively equivalent to a low and high pass filtering of the histogram. The matching between SCD can be computed using the L_1 distance metric (equation 2.26).

Color Structure Descriptor The **Color Structure Descriptor** (CSD) [112] is a modified version of a traditional histogram which captures both the distribution of colors in the image and the local spatial structure of the colors. CSD is defined as:

$$\mathrm{CSD} = \bar{\mathbf{H}}_s(m) \quad m \in \{1, \ldots, M\} \tag{2.3}$$

where $M \in 256, 128, 64, 32$ is the quantization step, $\bar{\mathbf{H}}_s(m)$ is the number of times a particular color is contained within a squared structuring element as the structuring element scans the image and s is the scale of the associated square structuring element. CSD aims at expressing the local structure of colours in the image through some histogram computation. CSD can be compared using L_1 metrics (equation 2.26).

Color Layout Descriptor The **Color Layout Descriptor** (CLD) [89] yields a compact representation of the spatial colour distribution, in **YCbCr** colour space. It consists in partitioning an image into 64 blocks (8 x 8). A representative colour is computed for each of these blocks (*e.g.* average colour or dominant colour). A Discrete Cosine Transform is then applied on this set of 64 values. A few low frequency coefficients are then selected and quantized through a zigzag-scanning pattern. The CLD is invariant to changes in resolution (scale) but not invariant with respect to rotation or translation.

2.2.2 Texture Descriptors

Image information does not only rely on its colour distribution but also on the textures it contains. Though no rigorous and formal definition of the term *texture* exists, everyone agrees that *texture* information brought by an image is of great importance for its understanding [35, 180]. One may say that a texture is generated by the quasi-periodic repetition of one

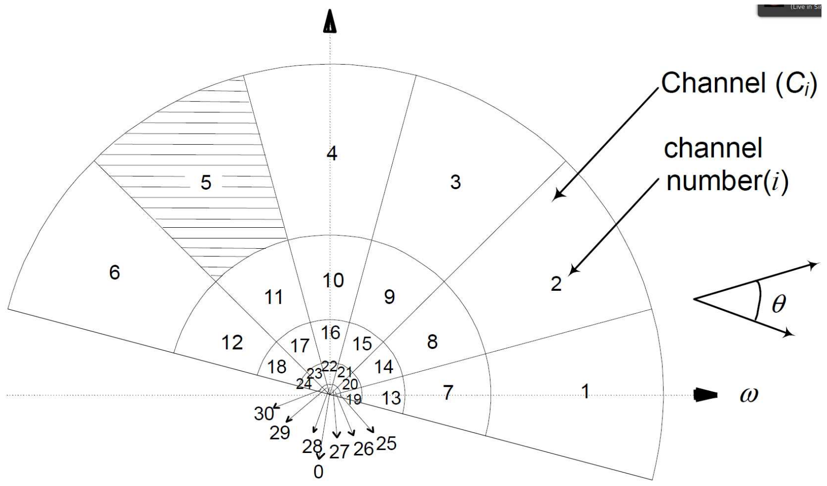


Figure 2.3: Frequency region division for HTD. Image from [141].

or more local patterns over some region or image that possess some stochastic structure. From a perceptual point of view, a texture may be defined by its *coarseness*, *repetitiveness*, *directionality* and *granularity*.

The MPEG-7 standard defines three texture descriptors for still images.

2.2.2.1 Texture Browsing Descriptor

The **Texture Browsing Descriptor** (TBD) [96] captures the perceptual characterization of a texture such as regularity, coarseness and directionality. The semantics of the descriptor is the following:

$$\text{TBD} = \{v_1, v_2, v_3, v_4, v_5\} \tag{2.4}$$

- $v_1 \in \{1, 2, 3, 4\}$ represents the regularity of the texture. A value of 4 means that the texture is highly regular.
- $v_2, v_3 \in \{1, \ldots, 6\}$ represents the directionality of the texture. Hence a texture may be associate with two dominant directions.
- $v_4, v_5 \in \{1, \ldots, 4\}$ represents the coarseness of the texture. 1 means a fine grain texture while 4 indicates coarse texture

The TBD is suitable for applications where a user can manually define the texture properties that a system must retrieve. However, it is neither invariant under rotation (directionality) nor under scaling (coarseness).

2.2.2.2 Homogeneous Texture Descriptors

The **Homogeneous Texture Descriptor** (HTD) [141] provides a quantitative representation of texture using local spatial frequencies statistics. The frequency plane is partitioned ac-

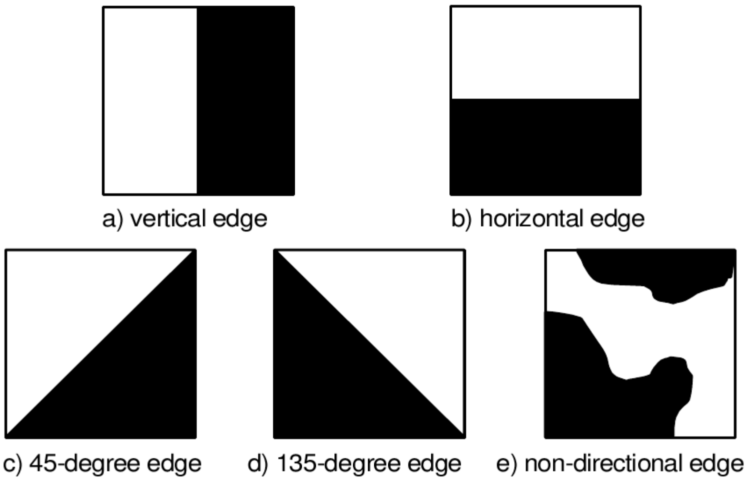


Figure 2.4: Edges orientations for MPEG-7 Edge Histogram Descriptor. Picture from [173].

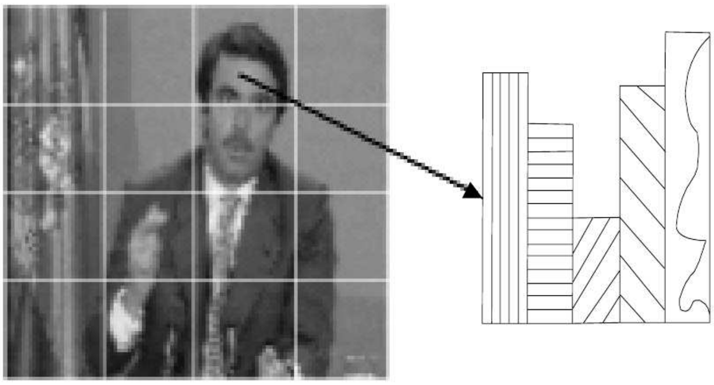


Figure 2.5: Five types of edge bins for each sub-image. Picture from [173].

cording to an angular and radial sampling (figure 2.3). The semantics of the HTD is the following:

$$\mathrm{HTD} = \{f_{DC}, f_{SD}, e_1, e_2, \ldots, e_{30}, d_1, d_2, \ldots, d_{30}\} \tag{2.5}$$

f_{DC} and f_{SD} are the first (mean) and second order (standard deviation) moments of the image and the e_i, d_i are, respectively, the mean and deviation of the energy in the frequency domain of the i^{th} channel in figure 2.3. HTD should be used in the context of similarity-based image-to-image matching because it yields a quantitative characterization of the texture.

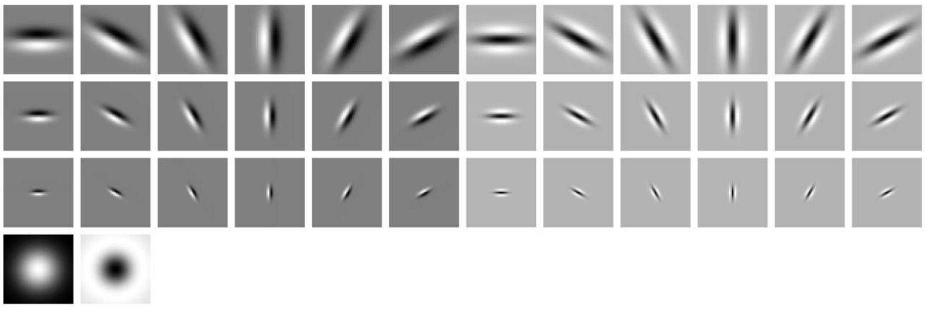


Figure 2.6: The MR8 filter bank consists of 2 anisotropic filters (an edge and a bar filter, at 6 orientations and 3 scales), and 2 rotationally symmetric ones (a Gaussian and a Laplacian of Gaussian). Picture from [167]

2.2.2.3 Edge Histogram Descriptor

The **Edge Histogram Descriptor** (EHD) [173] encodes the distribution of edges orientations in local areas of the image (sub-images). There are five types of edge orientations considered, as illustrated in figure 2.4. The image is divided into 4×4 equal-size non overlapping blocks which define the sub-images. Each sub-image is divided into a predetermined number of image-blocks, which are non-overlapping square blocks inside the sub image. The edge orientation is computed by applying oriented edge detectors using a 2×2 image-blocks neighborhood. The edge detector with the maximum strength is identified. If none of the edge detectors reaches a specified threshold, edge is considered as non-oriented. Hence, the EHD is a $4 \times 4 \times 5 = 80$-dimensional feature vector, where each dimension encodes the distribution of a specific edge orientation within a local area of the image (figure 2.5). EHD is particularly useful for matching regions with varying (*i.e.* non uniform) textures.

2.2.2.4 Bank of filters

Recent development in texture research has lead to the development of statistical approaches to texture characterization [97, 146] using a bank of filters. The image is convolved with a bank of filters of different scales and orientations (figure 2.6), and the texture is characterized by the filter response distribution. A dictionary of representative texture elements, called textons, is obtained by aggregating the filter bank responses via the K-Means clustering algorithm (which will be introduced in section 2.4 page 35). The texture is hence characterized by the filter response distribution with respect to the codebook. The distribution of texture elements can be seen as a histogram of texture elements where the quantification of the texture space into non overlapping bins is achieved using the clustering algorithm.

2.2.2.5 Image patches

Varma and Zisserman [167] formulated a very simple yet efficient approach to material classification using image patch exemplars. They demonstrated that materials can be classi-

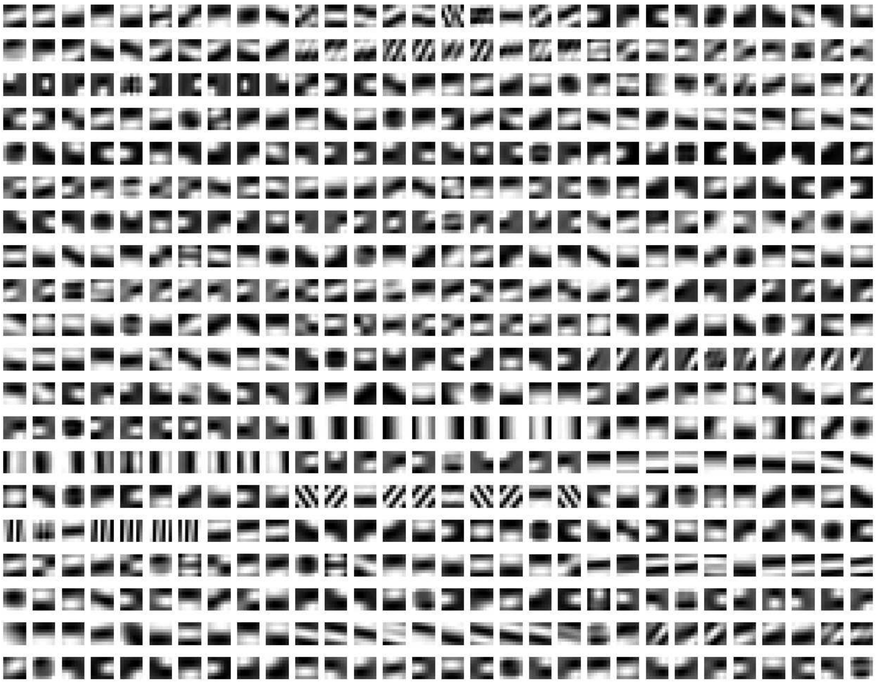


Figure 2.7: Example of an image patch textons dictionary over a 7×7 neighbourhood. Picture from [167].

fied using the joint distribution of intensity values over extremely compact neighbourhoods (starting from 3×3) with superior performances than the filter bank approaches. Concretely, instead of using the filter responses, one can directly use the pixels intensities of the source image over a small $N \times N$ neighbourhood. The raw pixel intensities are reordered to form a N^2-dimensional feature vector. Then the same procedure as with filter banks applies: a codebook of textons is obtained via clustering (figure 2.7) and the texture is characterized by its distribution with respect to the texton dictionary.

2.2.2.6 Local Binary Patterns

Local binary patterns (LBP) are a multi-resolution gray-scale and rotation invariant texture descriptors that was proposed by Ojala *et al.* for the task of texture classification [127]. A local binary pattern $LBP_{P,R}$ is extracted at a given position $c = x, y$ on a gray-scale image. Let us note the following P-dimensional binary vector:

$$\mathbf{t}_{P,R} = \{\text{sign}(I_0 - I_c), \text{sign}(I_1 - I_c), \ldots, \text{sign}(I_{P-1} - I_c)\} \tag{2.6}$$

$\text{sign}(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases}$, I_c is the gray level value of image I at position c, and the $I_0, I_1, \ldots, I_{P-1}$ are locations sampled on a circular symmetric neighbourhood of center c and radius R. This is illustrated by figure 2.8.

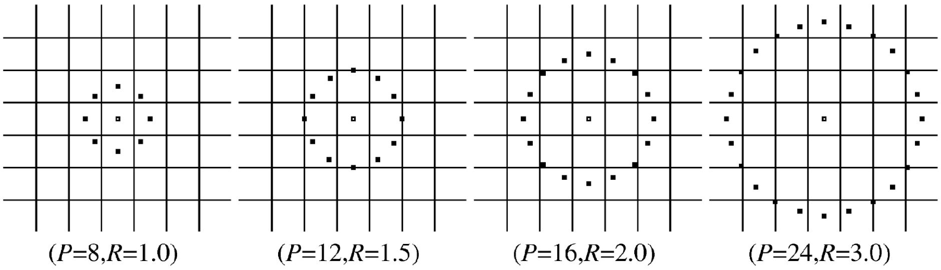


Figure 2.8: Examples of circular symmetric neighbourhood for different P,R. Image from [127].

$\mathbf{t}$ captures the texture information as the joint gray level distribution. $\mathbf{t}$ is invariant with uniform changes in the luminance as it considers the signed difference between the center and its neighbour. Discarding the amplitude of the differences and just keeping the sign of the differences makes $\mathbf{t}$ invariant to the scaling of the gray-scale values. The Local Binary Pattern is a number that characterizes the local spatial structure of the image texture defined as:

$$LBP_{P,R} = \sum_{p=0}^{P-1} (\mathbf{t}_p)2^p \tag{2.7}$$

The image texture is then characterized by the distribution of its LBP values. Ojala *et al.* have experimentally shown that the vast majority of LBP patterns are *uniform* patterns. The *uniformity* of a pattern is defined as the number of spatial transition in $\mathbf{t}$, that is the number of $0/1$ changes in the pattern. It is computed as:

$$U(LBP_{P,R}) = \sum_{p=0}^{P-2} |\mathbf{t}_p - \mathbf{t}_{p+1}| + |\mathbf{t}_{p-1} - \mathbf{t}_0| \tag{2.8}$$

Thus, they propose a rotation invariant binary pattern LBP^{ri} as:

$$LBP^{ri}_{P,R} = \begin{cases} \sum_{p=0} P - 1\mathbf{t}_p \text{ if } U(LBP_{P,R} \leq 2) \\ P + 1 \text{ otherwise} \end{cases} \tag{2.9}$$

$LBP^{ri}_{P,R}$ gives a finer quantization of the uniform patterns. Each uniform pattern is uniquely identified. There exists $P + 1$ such unique uniform patterns on a circular symmetric neighbourhood of P pixels. The non-uniform patterns are just put under a common label. In the rest of the manuscript, when we refer to the local binary patterns LBP, we refer to the definition of $LBP^{ri}_{P,R}$.

More texture descriptors are available in the literature, but their exhaustive overview would be long and out of the scope of this manuscript. We have focused our work in using the MPEG-7 descriptors which, as mentioned above, have been widely accepted by the CBIR community. To only cite one publication about the state-of-the-art, we point out the work of Tuceryan and Jain [162].

2.2.3 Shape Descriptors

Using shape descriptors implies being able to extract accurate shapes from an image. Shape descriptors may be based on their contour or on the region they contain. Shape description methods are particularly suitable for simple images, which contain one shape easily distinguishable from the background. Image segmentation is a usual pre-processing step before shape description extraction. MPEG-7 provides two shape descriptors which are invariant to scaling, rotation, and translation.

2.2.3.1 Curvature Scale Space

The **Curvature Scale Space** descriptor is the choice of the MPEG-7 expert group as a contour shape descriptor [117]. The idea of the curvature scale space representation is that a contour is well described by its inflection points. The CSS descriptor describes the evolution of the set of curvature zero-crossings of a contour when it is progressively smoothed until convexity. The semantics of the CSS descriptor is the following:

$$\mathrm{CSS} = \{n, c_1, e_1, c_2, e_2, h, \mathbf{x}, \mathbf{y}\} \tag{2.10}$$

n is the number of peaks in the CSS, c_1, c_2 are the circularity of, respectively, the original contour and the smoothed contour. e_1, e_2 are the eccentricity of the contour and the smoothed contour. h is the height of the highest peak, $\mathbf{x}$ and $\mathbf{y}$ are respectively the x position of a peak on the contour and the height of the peak. CSS is invariant to geometrical transformations of the object contour, and robust to noise.

2.2.3.2 Angular Radial Transform

The **Angular Radial Transform** (ART) is the Region-Based shape descriptor chosen by the MPEG-7 standard. ART belongs to the class of moment invariants methods[79, 139]. The descriptor uses a complex 2D Angular Radial Transform of the unit disc where the ART coefficients are given by:

$$\mathrm{ART}_{nm} = \int_0^{2\pi} \int_0^1 \mathrm{A}_m(\theta)\mathrm{R}_n(\rho) f(\rho,\theta)\rho \, d\rho \, d\theta \tag{2.11}$$

ART_{nm} are the ART coefficients of order n and m, $f(\rho,\theta)$ is an image in polar coordinates and A and R are the ART basis functions along the angular and radial directions:

$$\begin{cases} \mathrm{A}_m(\theta) = \frac{1}{2\pi} exp(jm\theta) \\ \mathrm{R}_n(\rho) = \begin{cases} 1 \text{ if } n = 0 \\ 2\cos(\pi n\rho) \text{ if } n \neq 0 \end{cases} \end{cases} \tag{2.12}$$

The ART descriptor is defined as a set of normalized magnitudes of the set of ART coefficients. Twelve angular and three radial functions are used, and the coefficients are divided by the magnitude of ART coefficient with $n = 0$ and $m = 0$. Hence the descriptor is a 35-dimensional vector ($12 \times 3 - 1$ because $n = 0, m = 0$ is constant after normalization). The descriptor is able to describe complex objects composed of several disconnected regions. Matching is performed with the L_1 metric.

2.2.3.3 Other shape descriptors

Simple geometrical attributes such as area (*i.e.* number of pixels in the region), perimeter (*i.e.* the length of the region boundary), compactness (*i.e.* $\frac{perimeter^2}{area}$) are often used to describe shapes [135]. Rather simple to compute, they can be gathered in attributes vector that may be compared through the use of some distance measures. However their characterization power is too weak to be used alone and they are often combined with more complex shape descriptors.

Again, a vast literature on shape descriptors exists [179, 57], but falls out of the scope of this manuscript.

2.2.4 Local descriptors

Another category of descriptors for image content are the local descriptors. Local descriptors compute local characteristics of the image from various locations. Computing such descriptors for all pixels in the image would create too much information. The local descriptors are sampled at specific locations resulting from an interest point detector. Interest points are local features at which the signal changes two-dimensionally. An interest point is good if it can be unambiguously located in different views of the scene. The use of interest points has advantages over features such as edges or regions including robustness to partial occlusion and high informational content [147]. Local interest points usage can be traced back to the work of Moravec [120] for stereo matching. Schmid and Mohr have been the first to propose a general image recognition scenario using local interest points [147]. The most widely known achievement using interest points has been published by Lowe [104, 103]. Lowe proposed a method for object recognition based on local scale invariant features. These scale invariant features are known as SIFT descriptors. As this work is the most popular of the interest point based descriptors, we will introduce it formally in this section. Similar works inspired by SIFT include GLOH (Gradient Location and Orientation Histogram) [116], PCA-SIFT [92] and SURF (Speeded Up Robust Features) [15]. The latter one we have mainly used throughout this PhD will also be introduced, highlighting the differences with the SIFT approach. A performance evaluation of local descriptors was carried by Mikolajczyk and Schmid [116]. The following sections on SIFT and SURF descriptors are mostly citations of the original papers [103, 15].

2.2.4.1 SIFT

SIFT features were proposed by Lowe in an attempt to provide features invariant to image scaling, rotation and partially invariant to changes in illumination and 3D camera viewpoint. Those features are well localized in both the spatial and frequency domain, reducing the probability of disruption by occlusion, clutter, or noise. In addition, the features are highly distinctive, which allows a single feature to be correctly matched with high probability against a large database of features, providing a basis for object and scene recognition. The computation of the set of SIFT image features follows four major steps:

1. **Scale-Space extrema detection:** the first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation.

2. **Keypoint localization:** at each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability.

3. **Orientation assignement:** one or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.

4. **Keypoint descriptor:** The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination.

Detection of Scale Space extrema The first stage of keypoint detection is to identify locations and scales that can be repeatably assigned under differing views of the same object. Detecting locations that are invariant to scale change can be accomplished by searching for stable features across all possible scales, using a continuous function of scale known as scale space. Lowe proposed to use the Gaussian function as the scale-space kernel. The scale-space of an image is defined as a function $L(x, y, \sigma)$ that is produced from the convolution of a variable-scale Gaussian $G(x, y, \sigma)$ with an input image $I(x, y)$:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{2.13}$$

where $*$ is the convolution operation in x, y and σ is the scale parameter:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp^{-(x^2+y^2)/2\sigma^2} \tag{2.14}$$

Stable keypoint locations in the scale space are extracted using scale-space extrema in the Difference-of-Gaussian function convolved with the image, $D(x, y, \sigma)$. It can be computed from the difference of two nearby scales separated by a constant multiplicative factor k:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \tag{2.15}$$

It is shown that the difference-of-Gaussian function provides a close approximation to the scale-normalized Laplacian of Gaussian, $\sigma^2\nabla^2 G$ [103]. Experiments showed that maxima and minima of $\sigma^2\nabla^2 G$ produce the most stable image features compared to a range of other possible image functions, such as the gradient, Hessian, or Harris corner function [115]. Moreover, as illustrated in figure 2.9, the difference of Gaussian function can be computed efficiently by simple image subtraction.

In order to detect the local maxima and minima of $D(x, y, \sigma)$, each sample point is compared to its eight neighbors in the current image and nine neighbors in the scale above and below (see figure 2.10). It is selected only if it is larger or smaller than all of these neighbors. The cost of this check is reasonably low due to the fact that most sample points will be eliminated following the first few checks.

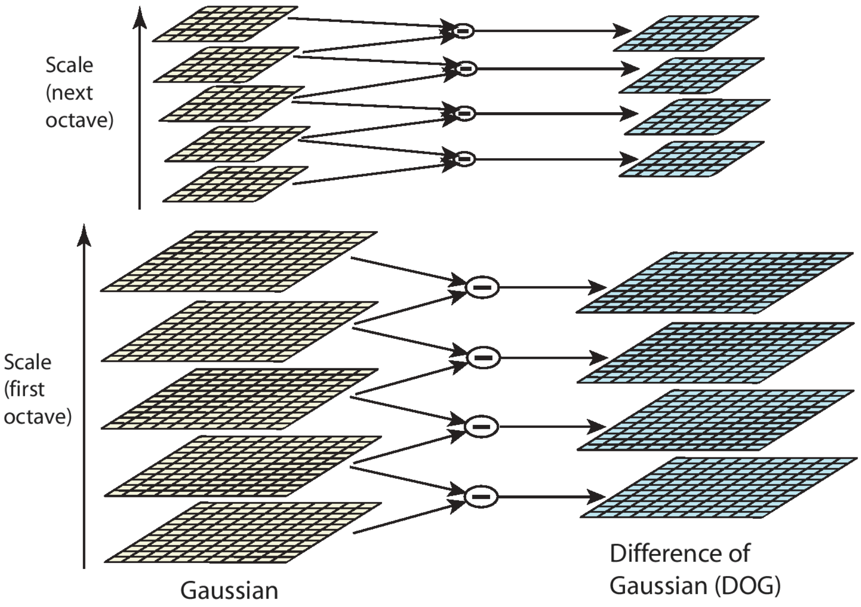


Figure 2.9: For each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images shown on the left. Adjacent Gaussian images are subtracted to produce the difference-of-Gaussian images on the right. After each octave, the Gaussian image is down-sampled by a factor of 2, and the process repeated. Image from [103]

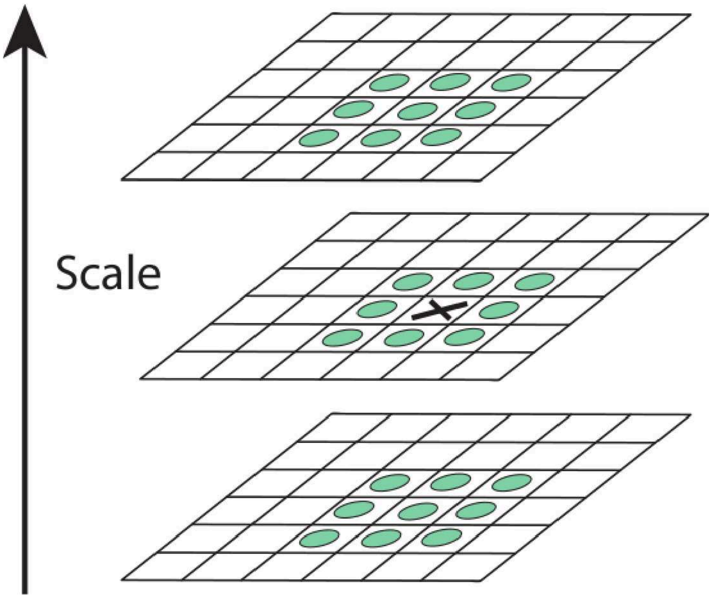


Figure 2.10: Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles).Image from [103]

Accurate keypoint localization Keypoints localized in the extrema of the scale space in the preceeding step are fit into a model for determining their location, scale and ratio of principal curvatures. This step carries a filtering out of keypoints that have low contrast (and are therefore sensitive to noise) or are poorly localized along an edge. The model for computing the interpolated location of the maximum has been proposed by Brown and Lowe [21]. His approach uses the Taylor expansion (up to the quadratic terms) of the scale-space function, $D(x, y, \sigma)$, shifted so that the origin is at the sample point:

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x} \tag{2.16}$$

where D and its derivatives are evaluated at the sample point and $\mathbf{x} = (x, y, \sigma)^T$ is the offset from this point. The location of the extremum x̂ is determined by taking the derivative of this function with respect to x and setting it to zero:

$$\hat{\mathbf{x}} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2}\frac{\partial D}{\partial \mathbf{x}} \tag{2.17}$$

The offset x̂ is added to the location its sample point to get the interpolated estimate for the location of the extremum. A filtering of the keypoints obtained after extrema localization is performed in order to keep the most meaningful keypoints. The function value at the extremum, $D(\hat{\mathbf{x}})$, is useful for rejecting unstable extrema with low contrast. Substituting equation 2.17 into equation 2.16, we have:

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2}\frac{\partial D^T}{\partial \mathbf{x}}\hat{\mathbf{x}} \tag{2.18}$$

Low contrast extrema, *i.e.* those with a value of $|D(\hat{\mathbf{x}})| < K$ with K a threshold value are filtered out. Finally, points located on the edges, which yields a strong response of the Difference-of-Gaussian function might be poorly located along the edges. A poorly defined peak in the difference-of-Gaussian function will have a large principal curvature across the edge but a small one in the perpendicular direction. Keypoints that have a ratio between their principal curvatures higher than a threshold are discarded. The different stages of keypoint selection are shown in figure 2.11.

Orientation assignment Invariance to image rotation is a desirable property for the keypoint descriptors. A consistent orientation is assigned to each keypoint based on local image properties. The method proposed by Lowe for orientation assignment is the following: the scale of the keypoint is used to select the Gaussian smoothed image L with the smallest scale so that all computations are performed in a scale invariant manner. For each image sample, $L(x, y)$ at this scale, the gradient magnitude (equation 2.19) and orientation (equation 2.20) is precomputed using pixel differences:

$$m(x, y) = \sqrt{(L(x-1, y) - L(x+1, y))^2 + (L(x, y-1) - L(x, y+1))^2} \tag{2.19}$$

$$\theta(x, y) = \tan^{-1}\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \tag{2.20}$$

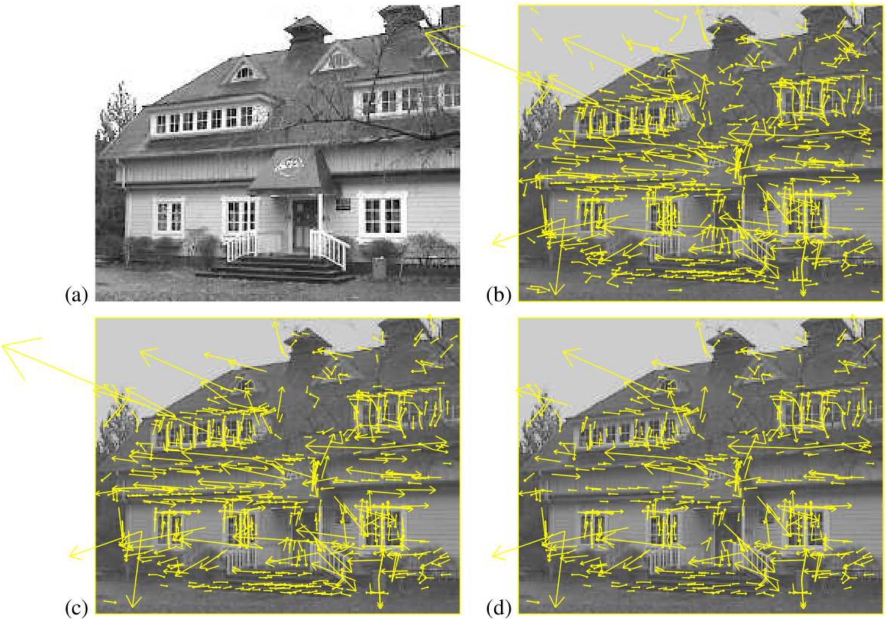


Figure 2.11: This figure shows the stages of keypoint selection. (a) The 233x189 pixel original image. (b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location. (c) After applying a threshold on minimum contrast, 729 keypoints remain. (d) The final 536 keypoints that remain following an additional threshold on ratio of principal curvatures. From [103]

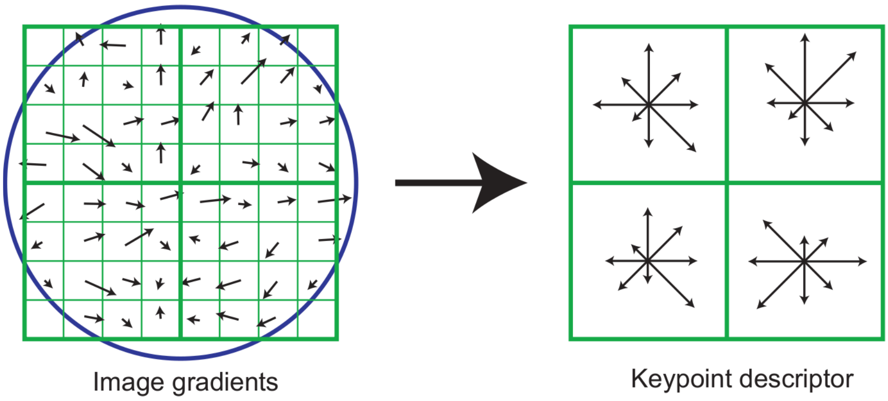


Figure 2.12: A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 sub-regions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2x2 descriptor array computed from an 8x8 set of samples, whereas the experiments in the original paper use 4x4 descriptors computed from a 16x16 sample array. From [103]

The gradient orientations of sample points within a region around the keypoint are collected into an histogram of 36 bins covering the full 360 degrees range of orientations. Each sample added to the histogram is weighted by its gradient magnitude and by a Gaussian-weighted circular window with a σ that is 1.5 times that of the scale of the keypoint. The orientation assigned to the keypoint corresponds to the orientation of the highest peak in the histogram. The orientation is accurately computed by fitting a parabola to the 3 bins surrounding the peak. If any other peak in the histogram is within 80 % of the highest peak, a new keypoint is created with the same location and this orientation. Lowe experimented that even if multiple orientation assignment is quite rare (about 15% of the keypoints) it contributes significantly to the stability of matching.

Keypoint description The previous steps have led to the computation of repeatable stable keypoints which are assigned a location, scale and orientation. The method of computation ensures robustness to affine transformations and noise in the image. The next step is to compute a descriptor for the local image region that is highly distinctive yet is as invariant as possible to remaining variations, such as change in illumination or 3D viewpoint. The descriptor computation was inspired by the work of Edelman, Intrator, and Poggio [50], which shown that in a model of biological vision, the perception of 3D objects is driven by the orientation and spatial frequency of gradients but the location of the gradient is allowed to be shifted. The computation of the keypoint descriptors is illustrated in figure 2.12.

First the image gradient magnitudes and orientations are sampled around the keypoint location, using the scale of the keypoint to select the level of Gaussian blur for the image. In order to achieve orientation invariance, the coordinates of the descriptor and the gradient orientations are rotated relative to the keypoint orientation. These are illustrated with small arrows at each sample location on the left side of figure 2.12. A Gaussian weighting function with σ equal to one half the width of the descriptor window is used to assign a weight to the magnitude of each sample point as illustrated by the circular window on the left of figure 2.12. The gaussian weighting avoid sudden changes in the descriptor with small changes of the position and decrease the influence of gradient samples that are far from the center of the descriptors. The keypoint descriptor is shown on the right side of 2.12. It allows for significant shift in gradient positions by creating orientation histograms over 4x4 sample regions. The figure shows eight directions for each orientation histogram, with the length of each arrow corresponding to the magnitude of that histogram entry. A gradient sample on the left can shift up to 4 sample positions while still contributing to the same histogram on the right, thereby achieving the objective of allowing for larger local positional shifts. The descriptor is formed from a vector containing the values of all the orientation histogram entries, corresponding to the lengths of the arrows on the right side of figure 2.12. The figure shows a 2x2 array of orientation histograms, whereas the original sampling for SIFT descriptors is achieved with a 4x4 array of histograms with 8 orientation bins in each. Therefore, the each keypoint is described using a 4x4x8 = 128 element feature vector.

2.2.4.2 Speeded Up Robust Features

Based on the success of the SIFT and SIFT related descriptors, Bay, Tuytelaars and Van Gool have recently proposed a novel scale and rotation invariant interest point detector and descriptor called SURF [15]. The main achievement of SURF descriptors compared to SIFT was to provide at least equal performances with respect to repeatability, distinctiveness and robustness, while being much faster to compute. Most of the computational efficiency of the SURF descriptors rely on the use of *integral images*. An integral image is a data structure for efficiently computing the sum of values in a rectangular subset of a grid. In the integral image, the value at any point $\hat{I}(x, y)$ is the sum of the pixel values that above and to the left of the position of the point:

$$\hat{I}(x,y) = \sum_{i \leq x, j \leq y} I(x,y) \tag{2.21}$$

The computation of the integral image is linear using an incremental algorithm:

$$\hat{I}(x,y) = I(x,y) + \hat{I}(x-1,y) + \hat{I}(x,y-1) - \hat{I}(x-1,y-1) \tag{2.22}$$

With such an integral image, computing the sum of the pixel values within a rectangular area is achieved in constant time. In example in figure 2.14, the sum of the pixel values within the rectangle delimited by the points A,B,C,D is obtained by:

$$\sum_{x_A < i \leq x_C, y_A < j \leq y_C} I(i,j) = \hat{I}(A) + \hat{I}(C) - \hat{I}(B) - \hat{I}(D) \tag{2.23}$$

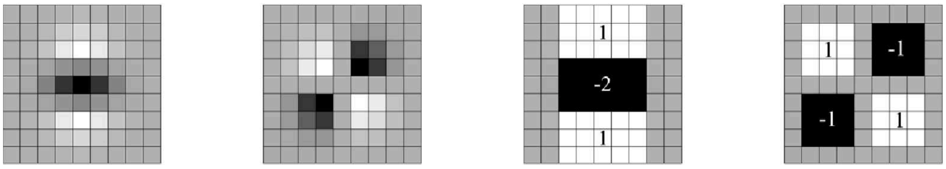


Figure 2.13: Left to right: the (discretized and cropped) Gaussian second order partial derivatives in y-direction and xy-direction and their approximations thereof using box filters. The grey regions are equal to zero. Figure from [15].

Detection of keypoint SIFT relies on the Difference-of-Gaussian (DoG) as an approximation of the Laplacian of Gaussian for scale space extrema detection. SURF detector relies on the determinant of the Hessian Matrix instead of the Laplacian-based detector:

$$H(\mathbf{x},\sigma) = \begin{bmatrix} L_{xx}(\mathbf{x},\sigma) & L_{xy}(\mathbf{x},\sigma) \\ L_{xy}(\mathbf{x},\sigma) & L_{yy}(\mathbf{x},\sigma) \end{bmatrix} \tag{2.24}$$

where $\mathbf{x} = (x, y)$ is a point in the image I, $H(\mathbf{x},\sigma)$ is the Hessian matrix in $\mathbf{x}$ at scale σ, $L_{xx}(\mathbf{x},\sigma)$ is the convolution of the Gaussian second order derivative $\frac{\partial^2}{\partial^2 x^2} g(\sigma)$ with the image I in point $\mathbf{x}$.

SURF relies on an approximation of the Hessian matrix called *fast Hessian* using box filters. This process is illustrated in figure 2.13, where the discretized Gaussian filters (on the left) are approximated using their box counterparts (on the right). The main advantage of this approximation is the gain in computational efficiency using the integral images rather than computing the discretized Gaussians. The authors shown that the experimental performances are comparable using the discretized Gaussians and their box filters approximation.

The scale space is analyzed by filtering the original image with increasing filter sizes, which save the time of convolution with a Gaussian kernel. A box filter of size 9×9 pixels approximates a Gaussian with $\sigma = 1.2$, which is the initial scale referred as $s = 1.2$. The next filter sizes are respectively $15 \times 15, 21 \times 21, 27 \times 27$. Accurate localization of maxima in the scale space is achieved using the same technique as with SIFT algorithm [21].

Orientation assignment SURF keypoint are assigned an orientation to ensure rotation invariance according to the following procedure: the Haar Wavelet response to x and y directions are computed in a circular windows of size $6s$ around the interest point, with s the scale of the keypoint. Using integral images, Haar Wavelet are computed efficiently. The wavelet responses are weighted with a Gaussian of $\sigma = 2.5s$ centered at the interest point. The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window covering an angle of $\pi/3$.

Keypoint description The extraction of the descriptor is done considering an oriented square window centered at the interest point of size $20s$ (figure 2.15. This region is split

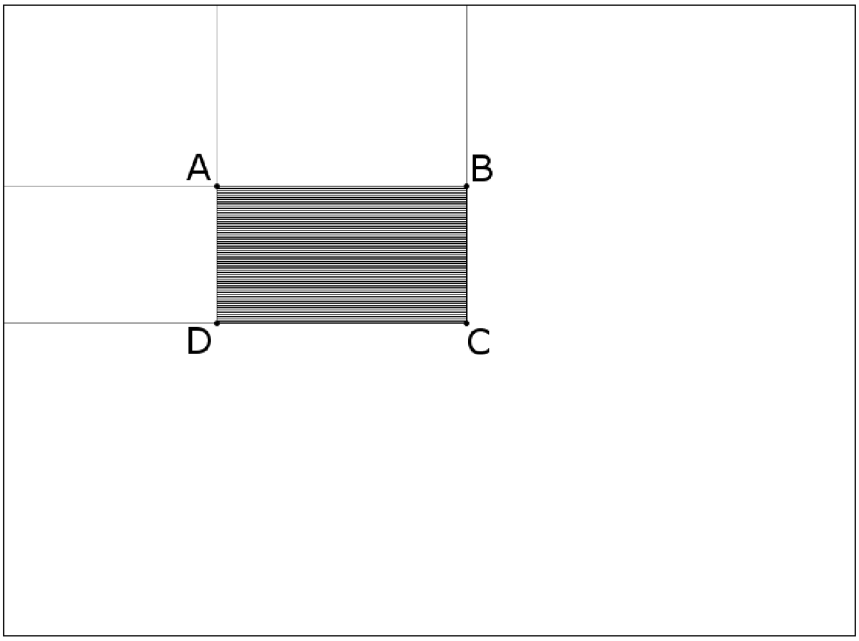


Figure 2.14: Finding the sum of a rectangular area using integral images

up into 4×4 sub-regions. The Harr wavelet response in the horizontal and vertical direction (with respect to the keypoint orientation) are computed within each sub-region. The responses are weighted with a Gaussian $\sigma = 3.3s$ centered at the interest point to increase robustness. Then, each sub-region yields a feature vector of size 4 consisting of the sum of the wavelet responses in the x and y directions, and the sum of the absolute value of the wavelet responses in the x and y directions. If we note d_x, d_y the wavelet responses, the feature vector for a sub-region is $\mathbf{x} = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$ (figure 2.16).

The total SURF descriptor hence is a $4 \times 4 \times 4 = 64$-dimensional feature vector. The descriptor is normalized to the unit vector to ensure invariance to contrast. An extended version of the SURF descriptor can be computed by summing the positive and negative wavelet responses separately. The extended feature vector is hence of size 128.

2.3 Similarity metrics

As described in the previous section, visual features can be expressed in a high dimensional vector space. Let us denote $\Omega = [0,1]^d$ the d-dimensional data space. In such a space, the distance between visual features can be computed with standard vector-space metrics. Let us consider two vectors $\mathbf{x}$ and $\mathbf{y} \in \Omega$.

The **Euclidean** distance, also known as L_2 distance is given in equation 2.25.

$$L_2(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{d} (\mathbf{x}_i - \mathbf{y}_i)^2} \tag{2.25}$$

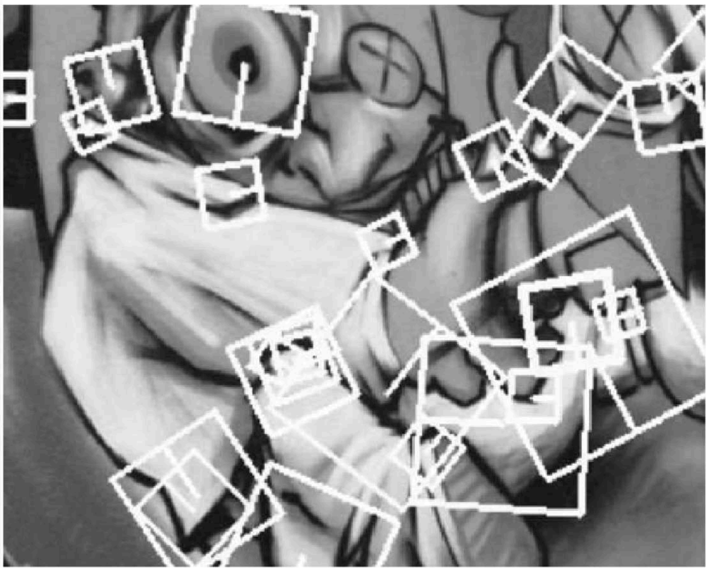


Figure 2.15: Example of the SURF descriptor window at different scales, within which the descriptor is computed. From [15]

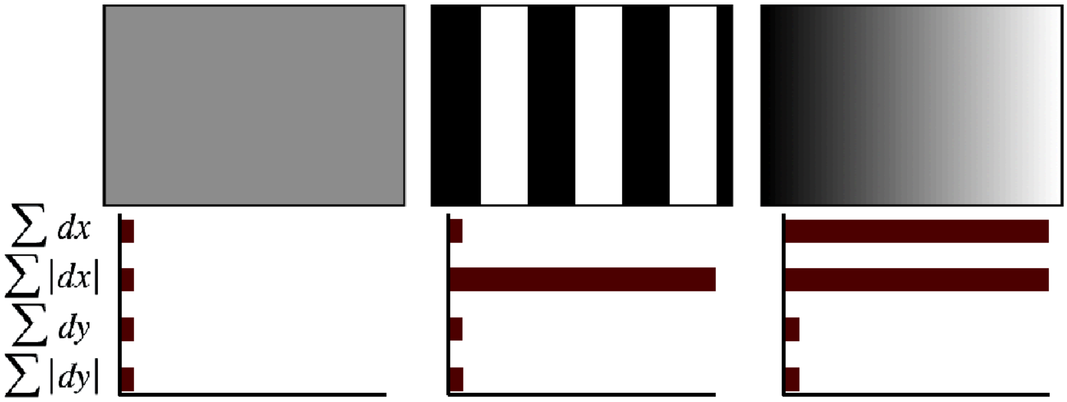


Figure 2.16: The descriptor entries of a sub-region represent the nature of the underlying intensity pattern. Left: In case of a homogeneous region, all values are relatively low. Middle: In presence of frequencies in x direction, the value of $\sum |d_x|$ is high, but all others remain low. If the intensity is gradually increasing in x direction, both values $\sum d_x$ and $\sum |d_x|$ are high (right). From [15].

The **Manhattan** distance (or L_1 distance) is given by equation 2.26.

$$L_1(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{d} |\mathbf{x}_i - \mathbf{y}_i| \tag{2.26}$$

Both distances are specific cases of the generalized **Minkowski** or L_p distance, which is given in equation 2.27.

$$L_p(\mathbf{x},\mathbf{y}) = \sqrt[p]{\sum_{i=1}^{d}(\mathbf{x}_i - \mathbf{y}_i)^p} \tag{2.27}$$

Another specific case of Minkowski distance is the **Chebyshev** distance, also known as Maximum or L_∞ distance (equation 2.28).

$$L_\infty(\mathbf{x},\mathbf{y}) = \lim_{p\to\infty} \sqrt[p]{\sum_{i=1}^{d}(\mathbf{x}_i - \mathbf{y}_i)^p} = \max_i \left\{|\mathbf{x}_i - \mathbf{y}_i|\right\} \tag{2.28}$$

The **Mahalanobis** distance is expressed in equation 2.29.

$$\textsc{Mahalanobis}(\mathbf{x},\mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\mathbf{y})} \tag{2.29}$$

$\boldsymbol{\Sigma}^{-1}$ is the inverse covariance matrix. Multiplying by the inverse covariance matrix as the effect of weighting each attribute by a coefficient inversely proportional to the noise on each component (assuming a Gaussian distribution of the component). Indeed, when one replaces $\boldsymbol{\Sigma}^{-1}$ with the identity matrix, the Mahalanobis distance is the same as the L_2 distance.

The *cosine* similarity (equation 2.30) gives the similarity between the vectors by computing the cosine of their angle.

$$\textsc{Cosine}(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}.\mathbf{y}}{||\mathbf{x}||\,||\mathbf{y}||} \tag{2.30}$$

. is the vector dot product and $||\mathbf{x}||$ the L_2 vector norm. The cosine similarity is a similarity value and not a distance as opposed to the previous measure. A cosine similarity of 1 indicates that the vectors point exactly in the same direction while a similarity of 0 indicates independence.

For special cases where $\mathbf{x},\mathbf{y}$ are strings, of equal lengths, the **Hamming** distance measures the minimum number of substitutions required to change one string into the other.

$$\textsc{Hamming}(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{d} \delta(x_i, y_i) \text{ with } \delta(x,y) = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{otherwise} \end{cases} \tag{2.31}$$

Varma [166] employ the χ^2 statistics to measure distances between the texton distributions with a nearest neighbour classifier:

$$\chi^2(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{d} \frac{(\mathbf{x}_i - \mathbf{y}_i)^2}{\mathbf{x}_i + \mathbf{y}_i} \tag{2.32}$$

Features such as normalised histograms provide the basis for an empirical estimate of the probability density function (pdf). Probabilistic approaches based on the measurement of the overlap between pdfs are equivalent to measuring a distance [30]. The Bhattacharyya distance (equation 2.33) measures the amount of overlap between two pdfs:

$$\textsc{Bhatta}(\mathbf{x},\mathbf{y}) = -\log \sum_{i=1}^{d} \sqrt{\mathbf{x}_i \times \mathbf{y}_i} \tag{2.33}$$

A closely related measure was given by Matusita [109] (equation 2.34)

$$\textsc{Matusita}(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{d} (\sqrt{\mathbf{x}_i} - \sqrt{\mathbf{y}_i})^2} \tag{2.34}$$

The Kullbach-Lieber divergence is a (non-symmetric) measure of the difference between pdfs [94].

$$\text{K-L}(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{d} \mathbf{y}_i \times \log(\frac{\mathbf{y}_i}{\mathbf{x}_i}) \tag{2.35}$$

The Earth Movers Distance (EMD) [145] is a measure of the distance between two distributions as the cost of transforming one distribution into the other one. It relies on the definition of a **ground truth distance**, which represents the cost of transforming a single feature into another one. We note $d(i,j)$ the cost of transforming feature i into feature j. Typically, for histograms, $d(i,i)$ will be null while $d(i,j)$ is not. Hence, the EMD is a measure which also takes into account inter-bin measures. The EMD is defined as equation 2.36:

$$\text{EMD}(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{d} (f_{i,j} d_{i,j}) \tag{2.36}$$

$f_{i,j}$ is the flow between x_i and y_j. The flow that minimises the overall cost is found by solving the transportation problem [87]. The EMD is a true metric if the ground truth distance is a metric and **x**, **y** are equal-weight distributions. Rubner *et al.* [145] have shown that the metric is perceptually sound for color and texture-based image retrieval. The main drawback is that solving the transportation problem is computationally expensive. Hence, the EMD is usually used for signatures, where the distribution is encoded using a few representative (possibly of different size) per image, and not suitable for histograms.

Most of the MPEG-7 features introduced in section 2.2 uses standard vector metrics (L_1 or L_2) for the computation of a distance between the feature vectors. However, the standard also propose some *ad-hoc* distance measures for some descriptors. Hence, given two DCD descriptors $\text{DCD}_1 = \Big\{(\mathbf{c}_{1i}, p_{1i}, v_{1i}), s_1\Big\}, i = \{1, 2, \ldots, N_1\}$ and $\text{DCD}_2 = \Big\{(\mathbf{c}_{2i}, p_{2i}, v_{2i}), s_2\Big\}, i = \{1, 2, \ldots, N_1\}$, three dissimilarity functions are defined depending on the parameters used for the computation: equation 2.37 gives the dissimilarity between the descriptors discarding the spatial coherence and variance parameters, equation 2.38 uses the spatial coherence field and equation 2.39 uses the variances.

$$D^2(\mathrm{DCD}_1, \mathrm{DCD}_2) = \textstyle\sum_{i=1}^{N_1} p_{1i}^2 + \sum_{j=1}^{N_2} p_{2j}^2 - \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} 2a_{1i,2j} p_{1i} p_{2j} \tag{2.37}$$

with $a_{k,l} = \begin{cases} 1 - d_{k,l}/d_{\max} & d_{k,l} \leq T_d \\ 0 & d_{k,l} > T_d \end{cases}$ the similarity coefficient between two colors $\mathbf{c}_k, \mathbf{c}_l$, $d_{k,l} = L_2(\mathbf{c}_k, \mathbf{c}_l)$ the Euclidean distance between the colors, T_d the threshold for two colors to be considered similar and $d_{\max} = \alpha T_d$. The standard recommend a value of T_d between 10 and 20 in the **Luv** color space and α from 1.0 to 1.5.

$$D_S(\mathrm{DCD}_1, \mathrm{DCD}_2) = w_1 \left| s_1 - s_2 \right| D + w_2 D \tag{2.38}$$

w_1, w_2 are fixed weights with a recommendation of 0.3 and 0.7 respectively by the standard.

The third distance function is based on a modeling of the colors with a mixture of Gaussian distributions, using the variance parameters. The squared difference between the distributions is given by:

$$D_v(\mathrm{DCD}_1, \mathrm{DCD}_2) = \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} p_{1i} p_{1j} f_{1i,1j} + \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} p_{2i} p_{2j} f_{2i,2j} - \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} 2p_{1i} p_{2j} f_{1i,2j} \tag{2.39}$$

where

$$f_{x_i,y_i} = \frac{1}{2\pi \sqrt{v_{x_i y_j}^{(l)} v_{x_i y_j}^{(u)} v_{x_i y_j}^{(v)}}} \times \exp -\frac{1}{2} \left(\frac{c_{x_i,y_j}^{(l)}}{v_{x_i,y_j}^{(l)}} + \frac{c_{x_i,y_j}^{(u)}}{v_{x_i,y_j}^{(u)}} + \frac{c_{x_i,y_j}^{(v)}}{v_{x_i,y_j}^{(v)}} \right)$$

and

$$c_{x_i,y_j}^{(l)} = (c_{x_i}^{(l)} - c_{y_j}^{(l)})^2, \quad v_{x_i,y_j}^{(l)} = (v_{x_i}^{(l)} + v_{y_j}^{(l)})$$

with $c_{x_i}^{(l)}$ and $v_{x_i}^{(l)}$ the dominant color values and color variances, x, y the query and target descriptors, i, j the descriptor components and l, u, v the components of the color space.

For CLD, the standard propose the following distance measure (equation 2.40):

$$\begin{aligned} D(\mathrm{CLD}_1, \mathrm{CLD}_2) = \quad & \sqrt{\textstyle\sum_i w_{yi} (Y_i^{(1)} - Y_i^{(2)})^2} \quad + \\ & \sqrt{\textstyle\sum_i w_{bi} (Cb_i^{(1)} - Cb_i^{(2)})^2} \quad + \\ & \sqrt{\textstyle\sum_i w_{ri} (Cr_i^{(1)} - Cr_i^{(2)})^2} \end{aligned} \tag{2.40}$$

The superscripts $(1), (2)$ identify the descriptors, the subscript i represents the scanning order of the coefficients of the DCT transform, Y, Cb, Cr represents DCT coefficients the luminance and chrominance and the w are weights to adjust the relative importance of each coefficient. A good weighting could assign more weight to the first coefficients of the DCT (low frequencies). No recommended values for the weights are given by the standard.

2.4 Clustering

Clustering is defined as the process of unsupervised grouping of data patterns into groups (clusters) [81]. The literature about data clustering is vast and some of its fundamental contributions include the Introduction to Cluster Analysis by Kaufman and Rousseeuw [91], the 1999 review of Jain *et al.* [81] and the more recent reviews of Xu and Wunsch [175] and Berkhin [19]. An exhaustive overview of clustering algorithms falls out of the scope of this manuscript. Nevertheless, some of the approaches that we have developed relied on clustering algorithms. The purpose of this section is to introduce them to the reader.

2.4.1 K-means clustering

Given the data set $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, the aim of k-means clustering is to partition the data into $k < N$ sets $\mathcal{S} = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within cluster similarity measure (equation 2.41):

$$\arg\min_{\mathcal{S}} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} ||\mathbf{x}_j - \mu_i|| \tag{2.41}$$

where μ_i is the mean (center of mass) of cluster S_i

It is show that the exact solution to this problem is **NP**-hard [156]. The following iterative refinement algorithm have been proposed by Lloyd [101]:

1. randomly choose k initial centers $C = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\}$
2. for each $i \in \{1, \ldots, k\}$ set $S_i = \{\mathbf{x} \in \mathcal{D} / ||\mathbf{x} - \mathbf{c}_i|| \leq ||\mathbf{x} - \mathbf{c}_j|| \forall j \neq i\}$
3. for each $i \in \{1, \ldots, k\}$, set $\mathbf{c}_i = \frac{1}{|S_i|} \sum_{\mathbf{x} \in \mathcal{S}_i} \mathbf{x}$
4. repeat 2 and 3 until convergence (*i.e.* $\mathcal{S}$ no longer changes)

This algorithm provides a simple and fast solution to the problem. Although it has no approximation guarantees at all, it is still one of the most employed method due to it's simplicity. Arthur and Vassilvitskii recently proposed an improvement of Lloyds algorithm by performing a careful choice on the initial centroids called kmeans++ [6]. Let us note $\delta(\mathbf{x})$ the shortest distance from a data point to the closest center already chosen.

1. randomly pick center $\mathbf{c}_1$
2. choose center $\mathbf{c}_i \in \mathcal{D}$ with probability $\frac{\delta(\mathbf{x})^2}{\sum_{\mathbf{x} \in \mathcal{D}} \delta(\mathbf{x})^2}$
3. repeat 2 until k centers have been chosen
4. proceed with the standard k-means

The intuition behind this initialisation scheme is to spread out the initial centers far away from each other. Although the seeding method add extra time for the computation of the initial centers it is shown experimentally that the convergence of the k-means algorithm is much faster, hence making the whole process more efficient. Moreover, is is guaranteed that k-means++ provides an $O(\log(k))$ optimal solution worst case [6].

2.4.2 Diday's dynamic cloud clustering

The dynamic cloud clustering algorithm of Diday [46] partition the data into $k < N$ non-empty sets $\mathcal{S} = \{S_1, S_2, \ldots, S_k\}$ and a set of prototypes $\mathcal{B}_i \subset \mathcal{S}_i, |\mathcal{B}_i| = T, T \leq |\mathcal{S}_i|$ representing the partitions minimizing criterion of adequacy between the partition and their prototypes. The k-means algorithm can be seen as a special case of the dynamic cloud clustering where the prototype for each class is the cluster centroid and the criterion to minimize is the within cluster similarity measure. Again, an iterative algorithm proceeding into successive steps of partitioning and refining prototypes is defined as follows:

1. randomly choose $\mathbf{b} \in \mathcal{B}_i$ from $\mathcal{D}$
2. for each $i \in \{1, \ldots, k\}$ set $\mathcal{S}_i = \{\mathbf{x} \in \mathcal{D} / \forall j \in [1, \ldots, k], j \neq i, \min_{\mathbf{b}_i \in \mathcal{B}_i} ||\mathbf{x} - \mathbf{b}_i|| \leq \min_{\mathbf{b}_j \in \mathcal{B}_j} ||\mathbf{x} - \mathbf{b}_j||\}$
3. for $i \in [1, \ldots, k]$ set $\mathcal{B}_i = \{\mathbf{x} \in \mathcal{S}_i / \Phi_i(\mathbf{x}_{\in \mathcal{B}_i}) < \Phi_i(\mathbf{x}_{\ni \mathcal{B}_i})\}$
4. repeat 2 and 3 until convergence (*i.e.* $\mathcal{B}$ no longer changes)

$\Phi(\mathbf{x})$ is the criterion to minimize such as the distance to the cluster centroid (equation 2.42).

$$\Phi_i(\mathbf{x}) = ||\mathbf{x} - \mathbf{c}_i|| \text{ with } \mathbf{c}_i = \frac{1}{|S_i|} \sum_{\mathbf{u} \in \mathcal{S}_i} \mathbf{u} \tag{2.42}$$

Diday's algorithm is a generalization of the k-means clustering where the cluster is not uniquely represented by it's centroid, but by the set of *prototypes*. Hence, while k-means cluster are represented by hyper-spheric clusters, the dynamic cloud clustering enable to have clusters with a more flexible shape.

2.4.3 DBScan

DBScan is a clustering algorithm originally proposed by Ester *et al.* for the clustering of large spatial databases [51]. DBScan belongs to the class of density-based clustering algorithms. The idea of density-based clustering is intuitively pictured in figure 2.17. When looking at the sample sets, we can easily and unambiguously detect clusters of points and noise points, *i.e.* those points that do not lie in any of the clusters. This is due to the fact that within each cluster, there is a typical density of points which is considerably higher than outside of the clusters.

The DBScan algorithm defines the notion of density in a neighbourhood of an object using 2 parameters: ϵ, which is a distance radius around the object and $MinPts$ the minimal number of points required to form a cluster. The algorithm proceeds as follows:

1. Choose an initial data point $\mathbf{x}_i \in \mathcal{D}$ that does not belong to any cluster.
2. $\mathbf{x}_i$ and it's neighbours are recursively added to the current cluster if they respect the *criterion of density*: there are at least $MinPts$ data points in the cluster and any data point in the cluster has at least 1 other data point in the cluster at a distance less than ϵ.

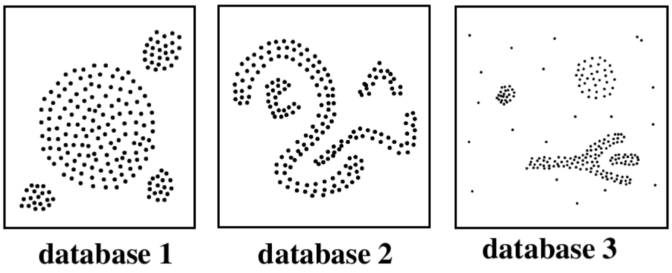


Figure 2.17: Sample Databases (picture from [51]).

The advantage of this algorithm is that it can recover clusters of arbitrary shape such as the one in figure 2.17. However, it's complexity is quadratic with respect to the number of data in the database.

2.5 Evaluation metrics

In this section, we introduce the main concepts and metrics used in the evaluation of the performances of systems. These metrics will be used in the following chapters to evaluate the tools that we have developed during this PhD. These metrics are related to the *effectiveness* of the system, that is the ability of the system to retrieve relevant documents while at the same time holding back non-relevant one [**?**]. During our evaluation, we assumed that the relevance of documents is not an issue[37] and can be obtained straightforwardly from the ground truth (definition below). Those metrics, coming from the Information Retrieval (IR) community, are directly applicable to CBIR and image classification systems. A *document* in an IRsystem correspond to a single image in a CBIR system. In the following we might refer to document or image equivalently.

Let us introduce the following concepts:

- The **ground truth** (**GT**) is a set of documents which have been manually annotated by a trustworthy human operator. The annotation can take different forms:
  - A *global categorization* of images, where each image is classified into one (or possibly more than one) predefined categories.
  - A *roughly localised* annotation of images, where parts of the image depicting specific categories (specific concepts) are localised, using regions-of-interest.
  - A *pixel-wise* annotation of images, where each image pixel is classified as belonging to one (or possibly more than one) of the predefined categories.

For a given classifier C and a given class λ we denote:

- **True Positive** (**TP**) the number of examples for which the true class is λ and that has been assigned class λ by the classifier.

- **True Negative** (**TN**) the number of examples for which the true class and the classifier answer is not λ.
- **False Positive** (**FP**) the number of examples for which the true class is not λ but the classifier answer is λ.
- **False Negative** (**FN**) the number of examples for which the true class is not λ but the classifier answer is λ.
- **Positive** (**P**) $=$ **TP** $+$ **FN** is the total number of positive examples.
- **Negative** (**N**) $=$ **FP** $+$ **TN** is the total number of negative examples.

Most of the evaluation metrics can be computed using the quantities just defined. The **Percentage of Correct Classification** (PCC, equation 2.43) [144] is the simplest way to assess the classification performances. However in the case of unbalanced dataset (*e.g.* $\mathrm{N} \gg \mathrm{P}$) this coefficient is not very representative: a classifier classifying all instances as negative (respectively as positive) will obtain very good (resp. very bad) PCC score.

$$\mathrm{PCC} = \frac{\mathbf{TP} + \mathbf{TN}}{\mathbf{TP} + \mathbf{TN} + \mathbf{FP} + \mathbf{FN}} \tag{2.43}$$

The **Jaccard Coefficient**(JC, equation 2.44)[154] and the **Yule Coefficient** (YC, equation 2.45)[154] behaves better in that case, minimising or eliminating the expected effect of a large amount of **TN** examples.

$$\mathrm{JC} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP} + \mathbf{FN}} \tag{2.44}$$

$$\mathrm{YC} = |(\frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}}) + (\frac{\mathbf{TN}}{\mathbf{TN} + \mathbf{FN}}) - 1| \tag{2.45}$$

Note that the Yule Coefficient cannot be computed when a classifiers gives only positive (or only negative) answers since the denominators would be 0.

Important evaluation metrics come from the IRcommunity such as the **Recall** (equation 2.46) and the **Precision** (equation 2.47). The precision can be interpreted as a measure of the *exactness* of the system, while the recall represents its *completeness*.

$$\mathrm{R} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}} \in [0, \ldots, 1] \tag{2.46}$$

$$\mathrm{P} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}} \in [0, \ldots, 1] \tag{2.47}$$

An ideal classifier yields $\mathrm{R} = \mathrm{P} = 1$. Precision increases when **FP** decreases. Recall increases when **FN** decreases. This usually but not necessarily means that when the recall increases the precision decreases, hence both measures are usually evaluated. An aggregated measure of Recall and precision is the F_β score (equation 2.48). In the case where $\beta = 1$, the F_1-Score is the harmonic mean of precision and recall. F_β score measures the effectiveness of retrieval with respect to a user who attaches β times as much importance to recall as precision [140].

$$\mathrm{F}_\beta = \frac{(1 + \beta^2)(\mathrm{P} \times \mathrm{R})}{\beta^2 \times \mathrm{P} + \mathrm{R}} \in [0, \ldots, 1] \tag{2.48}$$

N P	A	B	C
A	10	0	3
B	6	24	4
C	3	0	15

Table 2.1: Example of confusion matrix

For multi-class classification tasks, a common way of evaluating the performances of a system is to represent the results with a **confusion matrix**. Each row in the matrix represent the actual classes while each column represent the predicted classes. Hence, as an example on table 2.1, the 6 in the second row, first column means that 6 instances of class B have been predicted as class A by the system. A prefect classifier would lead to a confusion matrix where only the diagonal elements (**TP**) are non-zero. As it's name stands, this representation is convenient to see if the system is confusing some classes. Moreover, Recall and Precision figures for a class i can be computed efficiently from the matrix by dividing the diagonal value ($M(i,i)$ = **TP**) by, respectively, the sum of the i^{th} row (**TP** + **FN**) or the sum of the i^{th} column.

In a classic document retrieval task, the user wants the retrieved documents to be sorted according to their relevance to the query. A desirable feature for the system is the ability to rank the relevant documents among the first returned results. This can be captured by computing the precision at given cutoffs, the **Precision at** n (equation 2.49):

$$\mathrm{P}_{(\mathrm{n})} = \tfrac{1}{n}\textstyle\sum_{i=1}^{n}\delta(i) \quad \text{with} \quad \delta(i) = \begin{cases} 1 & \text{if the } i^{th} \text{ document retrieved is relevant} \\ 0 & \text{otherwise} \end{cases} \tag{2.49}$$

$\mathrm{P}_{(\mathrm{n})}$ does not measure Recall. The Average Precision (AP, equation 2.50) is a measure that combines Precision, Recall and relevance ranking.

$$\mathrm{AP} = \frac{\sum_{n=1}^{N}\mathrm{P}_{(\mathrm{n})} \times \delta(n)}{\mathbf{P}} \tag{2.50}$$

N is the number of documents retrieved by the system. To get $\mathrm{AP} = 1$, the system must retrieve all the relevant documents ($\mathrm{R} = 1$) and rank them perfectly ($\mathrm{P} = 1$). Note that the average precision does not quantify the cost of returning irrelevant document after all relevant documents in the query.

The AP is computed for a single query. The system performances for several queries can be computed using the Mean Average Precision (MAP, equation 2.51):

$$\mathrm{MAP} = \frac{1}{K}\sum_{k=1}^{K} AP^{(k)} \tag{2.51}$$

K is the number of queries issued to the system, $AP^{(k)}$ is the average precision of the k^{th} query. As it's name indicates, the MAP is the mean of the average precisions over several queries of the system. Usually, systems are evaluated by computing the MAP with all the available documents in the dataset. MAP scores can also be computed for specific categories of document, by querying the system with only documents belonging to this category. R, P and MAP are widely spread metrics which are used for the evaluation of systems in renown international challenges (TrecVideo [73], Pascal Visual Object Class [52], ImageCLEF [122]).

Finally, the performances of a system can be summarized graphically using Precision-Recall graphs [95] (see figure 2.18[1]).

The graph displays the precision(Y-axis) versus recall (X-axis) scores obtained after each relevant query, hence illustrating the trade-off between precision and recall. An ideal goal for a retrieval system is to increase both precision and recall, *i.e.* the curve must bend over the top right hand corner so that both recall and precision are higher at every point along the curve. Average precision is the area under the precision-recall graph. By moving the curve up and to the right, the area under the graph increases, thereby increasing the average precision.

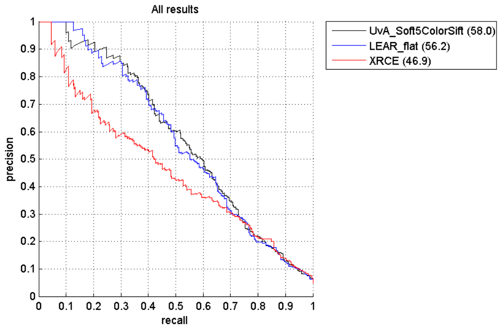


Figure 2.18: Example of Precision-Recall graph. Image taken from Pascal VOC2008 challenge results

[1] Available at `http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2008/results/cls/_voc2007_comp1_bicycle_pr.shtml`

Chapter 3

Indexing and organizing image database through global descriptors

3.1 Multidimensional Clustering Index

3.1.1 Introduction

In this section, we propose an indexing schema for image databases based on multiple clustering according to the available set of visual descriptors. We will start by describing the difficulties of multidimensional indexing and giving elements of the state-of-the-art in multidimensional indexing structures. We will then describe our approach and demonstrate experimentally it's efficiency.

3.1.2 Data structures for multidimensional Indexing

The use of image content as an alternative to keyword-based indexing is an intuitive and attractive approach. In chapter 2, we saw that most visual descriptors take the form of a d-dimensional vector, where d can be arbitrary large. Let us set the following notations:

- d the number of dimensions.
- N the number of data points.
- $\Omega = [0;1]^d$ the data space. We consider all descriptors normalised, that is with all coordinates in the interval $[0;1]$.
- $\mathcal{D} \subseteq \Omega$ the data set.

Considering a data space Ω and a dataset $\mathcal{D}$, such a task as finding *similar* images in a database can be expressed as finding the nearest or k-nearest neighbours (**k-NN**) in the chosen description space. Intensive research has been carried on the subject of **k-NN** search for multidimensional spaces.

Definition 3.1.1. *The Nearest Neighbour (**NN**) to a query point Q is the data point $nn(Q) \in \mathcal{D}$ which lies closest to Q in $\mathcal{D}$: $nn(Q) = \{P \in \mathcal{D} | \forall P' \in \mathcal{D}, ||P - Q|| \leq ||P' - Q||\}$ where $|| \bullet - \bullet ||$ is typically assumed to be an L_p distance (cf equation 2.27) [172].*

The k-nearest neighbours (**k-NN**) to a given point can be defined in a similar way.

An obvious solution to **k-NN** search is to sequentially compare each element to the query and keep the k most similar. However, this is feasible only for relatively small databases. High-dimensional indexing schemes are alternatives for accelerating the search which rely on data structures that simplifies the retrieval task. The vast literature on the topic was reviewed by Castelli [29]. Multidimensional indexing techniques can be classified into different categories:

- Space partitioning methods divide the data space along predefined or predetermined lines (hyperplanes) regardless of data clusters. Space partitioning method have been the first to be proposed. They include techniques like Quad-trees[56], k-d-b-trees [142] and the Grid-File [124]

- Data partitioning methods divide the space according to the distribution of data inserted into the structure. Such methods take advantage of the fact that in real datasets, the samples are usually clustered and occupy only a small portion of the full data space. Several variants have been proposed such as the TV-trees [99], the X-trees [18], the M-trees [34] or the SR-trees [90].

- Space filling curves have been proposed as an alternative approach for multidimensional indexing. Those methods have been inspired by the research in fractal geometry, which show that there exists continuous mappings from the unit interval $[0; 1]$ to any unit hypercube $[0; 1]^d$ [163]. Indexing using space filling curves is intuitive. The high dimensional data points are mapped into a curve, obtaining a one dimensional coordinate named the *extended key*. The extended keys are used to perform straightforward one dimensional similarity search. The assumption is that points which are close to each other on the curve are also close together in the multidimensional space. However, this is not always the case especially when the number of dimensions are increased. Most authors propose the usage of several space filling curves [111, 149, 98, 163].

Despite the active research for efficient multidimensional indexing structures, it has been shown that none of these methods can perform well in high dimensional spaces [24, 172]. Here, high dimensional means more than 20 dimensions. This is problematic for image similarity search where the description space can grow much larger. In such spaces, most of the indexing methods are outperformed by the straightforward sequential scan. This phenomenon is known as the **curse of dimensionality**. The curse of dimensionality refers to the exponential growth of hyper-volumes as a function of dimensionality. Our perception and the geometric properties that rule the 3-dimensional world do not hold in higher dimensional spaces. Data structures that rely on the properties of low dimensional spaces do not perform well in high dimensional spaces. Such data structures can even become intractable. As an example, imagine a space partitioning structure that sample the input space into regularly spaced intervals (cubes or hypercubes for higher dimensions) of size 0.1. Hence 10 such intervals are sufficient to cover the one dimensional unit interval. For d dimensional unit hypercube, the number of hypercube necessary to span the entire input space is 10^d. As the number of dimensions grows, it is more likely that this number will exceed the number of samples in the dataset. Moreover, this number makes the index intractable due to the huge amount of memory requirement, while most of the hypercubes contain no data.

Another difficulty arising in high dimensional spaces is the fact that randomly sampled points from the same distribution appear uniformly far from each other, and each point sees itself as an outlier [29]. On the other hand, the other points in the database appear to be at almost the same distance. This is illustrated by the following experiment, quoted from [29]:

> « *Consider the following simple example: let a database be composed of* $20,000$ *independent 100-dimensional vectors, with the features of each vector independently distributed as standard Normal random (i.e., Gaussian) variables. Normal distributions are very concentrated: the tails decay extremely fast, and the probability of sampling observations far from the mean is negligible. A large Gaussian sample in 3-dimensional space looks like a tight, well concentrated cloud, a nice* cluster. *Not so in 100 dimensions. In fact, sampling an independent query template according to the same 100-dimensional standard Normal, and computing the histogram of the distances between this query point and the points in the database, yields the result shown in Figure 3.1. In the data used for the figure, the minimum distance between the query and a database point is 10.1997, and the maximum distance is 18.3019. There are no* close *points to the query, and there are no* far *points from the query.* »

Recent research have proposed methods for overcoming the running time bottleneck by using approximation [121, 4, 2]. In this formulation the algorithm is allowed to return a point whose distance from the query is at most c times the distance from the query to its nearest points, $c > 1$ is called the approximation factor.

Despite the very active research in multidimensional indexing, it is very common that the dimension of the descriptor space for CBIR systems exceeds the current expectation for indexing structures. Griga [68] proposed a semi supervised categorisation of image database with a set of low level descriptors leading to a feature vector of more than 600 dimensions. Ferecatu and Boujemaa [54] proposed an interactive retrieval tool for remote sensing images where the concatenated feature vector also exceed 600 dimensions. In this case, one solution is to perform dimensionality reduction of the data space prior to the indexing schema. Griga and Feracatu both employed Principal Component Analysis (PCA) to achieve this goal. PCA seeks the best approximation of a given dataset of points using a linear combination of a set of vectors which retain maximum variance along their directions [48]. This set of vectors is

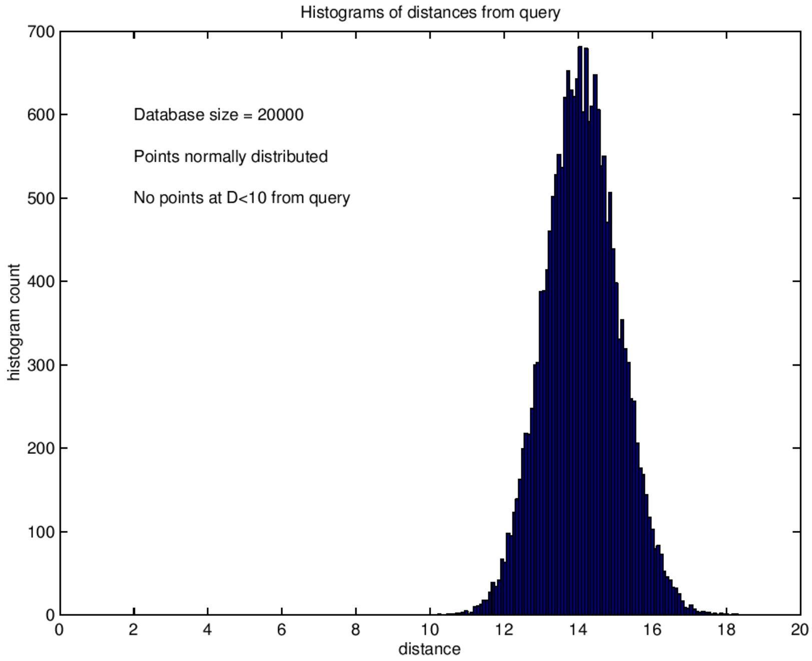


Figure 3.1: Distribution of the distances to a query in a 100-dimensional dataset independently distributed according to Gaussian distributions. Figure from [29].

called the set of Principal Components. One can fix the cardinality of the set of principal components to project into. Griga mapped the original input space into a 29 dimensional feature vector. Ferecatu and Boujemaa did not set this dimension but they discarded the dimensions which represented less than 1% of the total energy of the dataset. PCA was also used in other publications such as Ke and Sukthankar [92]. Based on the SIFT scale-space keypoint detection, they projected the normalised horizontal and vertical gradient patches (a 3042-dimensional feature vector) into a linear combination of 20 principal component axis, and showed improvement over the SIFT descriptors both in compactness of representation and in efficiency for image retrieval.

3.1.3 Proposed framework

Despite its successful application as a dimensionality reduction technique prior to indexing, PCA has some limitations. Using PCA, we assume that the dataset *can* be accurately represented by a *linear* combination of it's basis. A more application-dependant limitation to PCA is the fact that the the semantics of the description space is lost. Indeed, in the original description space, one has the control over what each dimension or set of dimensions represents. As an example, in the MPEG-7 EHD descriptor, each dimension encodes the proportion of a particular edge orientation in a particular sub-image. If we consider a feature vector as a concatenation of several (possibly multi-dimensional) descriptors, each descriptor yields a particular view of the content of the image. In the projected PCA space, the dimensions of the feature vectors are expressed as linear combinations of the original dimensions. One has no control over this combination, which is driven by the data. The dimensions of the projected space can represent a mixture of totally uncorrelated descriptors in terms of semantics.

Queries in CBIR are ill-defined. In a typical Query By Example engine, the same image could be used as query for users having different intentions in mind. Relevance feedback [159, 54] techniques involves interaction with the users to improve the set of returned results. Typically, after an initial query, the user mark the results as *relevant* or *irrelevant*. The system takes these information to refine the retrieval results based on this feedback. These two steps are carried out iteratively to improve the performance of the image retrieval system by gradually learning the user's preferences. A different approach trying to overcome the problem of user intent was developed by Naud *et al.* [123]. According to the user feedback, the system captures the allowed variability in the request (*e.g.* different color is allowed but same shape). The similarity measure between image is interactively adapted to better fit the user requirement.

The goal of our indexing technique is to structure the image database to enable efficient retrieval of a set of relevant images while filtering out large portions of non relevant images. Moreover, we want to keep the semantics of the descriptors used to characterise the images intact. To do so, we propose to compute a multidimensional index that encodes the different views of an image database yielded by different low level descriptors. The index is computed through multiple clustering of the database using the different low level descriptors.

Let $\mathcal{L} = \{\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_K\}$ be a set of descriptors. Each of them has the dimensionality d_k. Let us consider the *hyper space* $\Omega = [0;1]^{d=\sum_{k=1}^{K} d_k}$. Let us now consider subspaces Ω_k of Ω such that $\cup_{k=1}^{K}\Omega_k = \Omega$ and $\cap_{k=1}^{K}\Omega_k = \emptyset$. Let $\mathcal{D}$ be an image database (*i.e.* the dataset) and $\mathcal{D}_k$ its projection onto the space Ω_k. We note $\mathcal{S}_k = \{\mathcal{S}_k^1, \mathcal{S}_k^2, \dots, \mathcal{S}_k^{N_k}\}$ a partition of $\mathcal{D}_k$, with $\mathcal{S}_k^i$ the elements of this partition (clusters).

Let us consider a document X^l from $\mathcal{D}$ represented by a data point $\mathbf{y}_k^l$ in $\mathcal{D}_k$. We say that X^l is indexed in $\mathcal{D}_k$ by e_k^* if:

$$\mathbf{y}_k^l \in \mathcal{S}_k^* \tag{3.1}$$

Definition 3.1.2. *We call the* ***multidimensional clustering index*** *of a document X a vector $\mathbf{e} = (e_1^*, e_2^*, \dots, e_K^*)^T \subset \mathbb{N}^{+K}$ where e_k^* is the label of the cluster in partition $\mathcal{S}_k$ satisfying (3.1).*

Each visual descriptor encodes a specific information about the content of the image. The images which are grouped in the same cluster according to a descriptor share common characteristics with respect to this descriptor.

A typical query using the multidimensional clustering index is to perform a **within-distance** (or **range** query) query on the index with the hamming distance.

Definition 3.1.3. *The* ***within-distance*** *query returns all the data at a distance less than ϵ from a query X^q: $wd(X^q) = \{Y^l \in \mathcal{D} | Dist(X^q, Y^l) < \epsilon\}$*

The hamming distance gives the number of descriptors for which the result images shares common characteristics (*i.e.* belong to the same cluster) with the query. Moreover, the computation of the hamming distance can be tuned to adapt to the user query. We know the relation between the dimensions of the index and the underlying visual descriptors that were used during clustering. Descriptors irrelevant to the query can be discarded when computing the hamming distance.

Inserting new images in the database is a fast operation. The clustering index for the new images is computed by querying the already existing clustering models to get the cluster indexes for the new image. The multidimensional clustering index can also be enriched without modifying the already computed indexes: when a new low level descriptor is available, one can add a dimension to the index using the same procedure without requiring to modify the other dimensions. However, if one wants to update the clustering models, *e.g.* after several new images have been inserted into the database to better match the new distribution, the index has to be recomputed for all the existing images. The last point actually depends on the clustering algorithm employed. Some clustering algorithms called incremental clustering algorithms allow to insert new data and update the models (*i.e.* add/delete or merge clusters) on the fly [70, 105]. Only the part of the index that are concerned with those modifications could be updated. In our experiments, we did not use incremental clustering algorithms.

3.1.4 Experiments

3.1.4.1 Datasets and experimental protocol

We have experimented our approach with two image databases. The first database is the Amsterdam Library of Object Images database (ALOI). ALOI is a color image collection of one-thousand objects, recorded for scientific purposes. Each object was captured with systematically varied viewing angle, illumination angle and illumination color for each object. 96 shots of each object is taken, leading to a collection of 96000 images. Details on the ALOI database have been published by Geusebroek *et al.* [63]. Figure 3.2 shows examples of objects and their different capture conditions.

The second image database is the WANG database [171]. This database is a subset of the Corel database containing 10 categories with 100 images per category. Except for one category (dinosaurs), the images are natural outdoor images. As opposed to the ALOI database, the imaging conditions have not been controlled using a strict experimental protocol. Example of images are shown in table 3.1.

We compared the efficiency of the multidimensional clustering index in terms of retrieval accuracy compared to the use of the full content descriptor. For a document X^l in the database $\mathcal{D}$, we call *full content descriptor* a vector $\mathbf{y}^l \in \Omega = [0;1]^{d, d=\sum_{k=1}^{K} d_k}$. It is formed by the concatenation of descriptors of X^l in subspaces Ω_k. This is known as the *early fusion* of the descriptors. We also compare the performances with a dimensionality-reduced vector using PCA. The PCA is performed on the full content descriptor into K dimensions, with K the number of descriptors. Hence, the dimensionality of the PCA descriptors and the clustering index is the same.

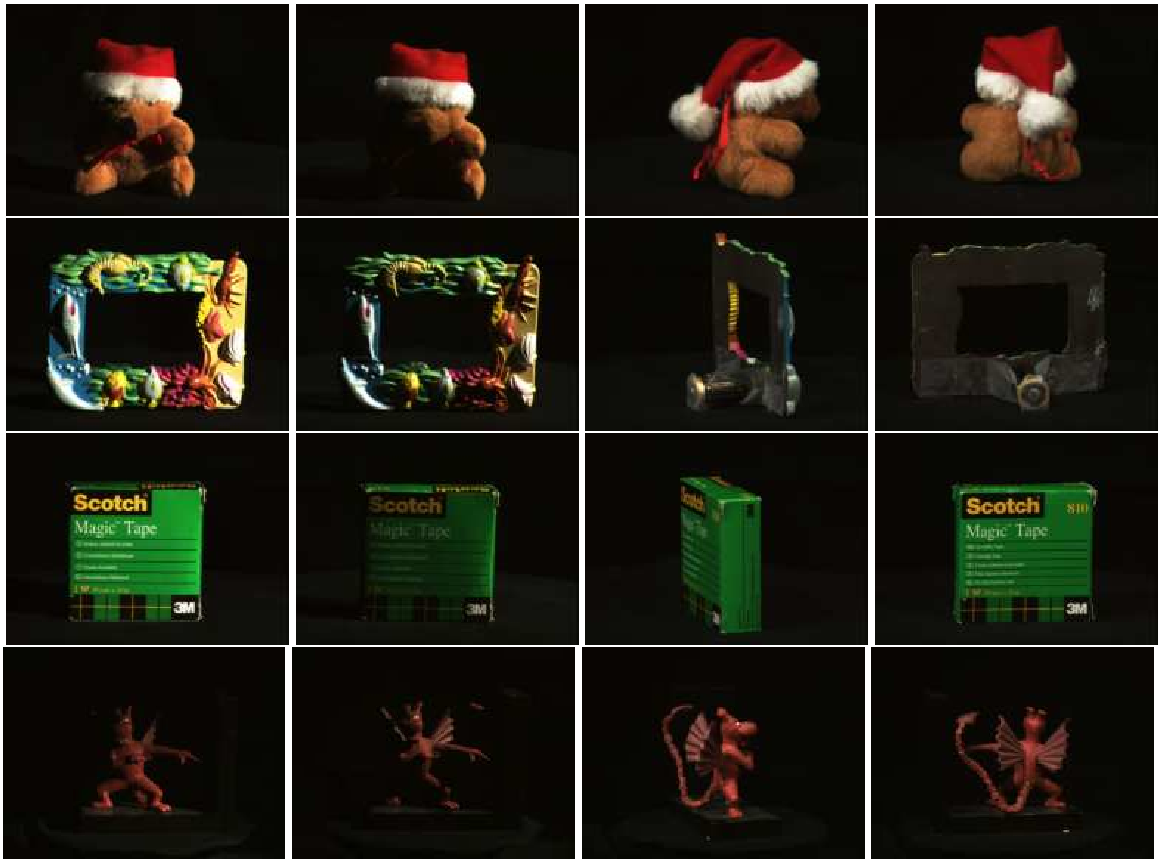


Figure 3.2: Example images from the ALOI database. Each line represents shots of the same object. First and second column show the object from the front with different illumination conditions. Third and fourth column show the object rotated with an angle of 105 and 185 degrees respectively.

africa		**beach**	
monuments		**buses**	
dinosaurs		**elephants**	
flowers		**horses**	
mountains		**food**	

Table 3.1: Samples from the WANG dataset: 100 images per category, 10 categories

Descriptors	MAP score using ALOI	MAP score using WANG
Full Descriptors	0.522	0.523
Clustering Index	0.407	0.517
PCA	0.266	0.458

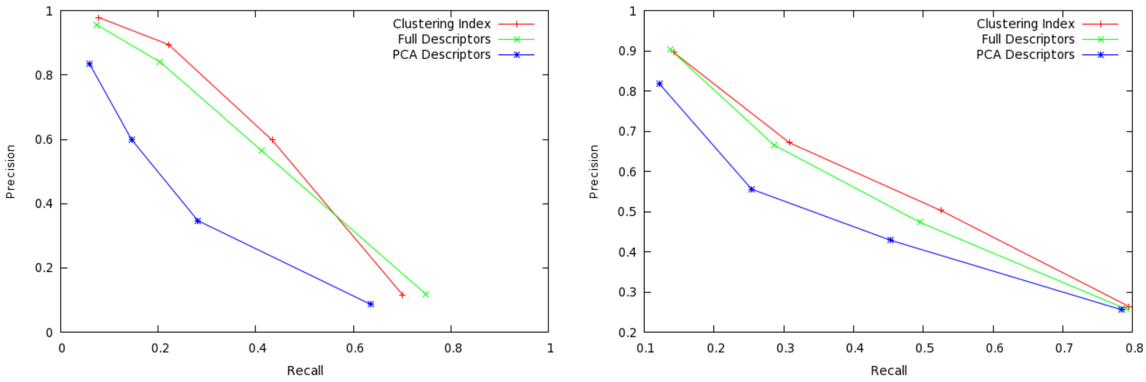
Table 3.2: Mean Average Precision scores on the ALOI database using the Full MPEG-7 feature vector, the proposed clustering index method and the PCA.

3.1.4.2 Results and discussion

To compute the multidimensional clustering index, we used k-means clustering into $k = 1000$ classes for ALOI database and $k = 10$ for WANG. The set of visual descriptors was set to EHD, CLD, CSD, SCD.

As a first experiment we compared the Mean Average Precision of the system using the multidimensional clustering index, the full descriptors and the PCA descriptors. For each image in the database, we compute the distance between the query and all the other images. The results are returned sorted by increasing distance from the query. With the multidimensional clustering index, the hamming distance between the indexes was considered. For the full feature vectors and PCA feature vectors, we computed the L1 distance between the vectors. The results of this experiment are reported in table 3.2. The best results are obtained when computing the distance with the full feature vector, with a MAP of 0.522 and 0.523 using ALOI and WANG database respectively. Note that the WANG database was used by Deselaers *et al.* for the evaluation of features for CBIR. The best MAP on the dataset was 0.505 using color histograms. Here, the early fusion of descriptors brought a slight improvement over the single descriptors. The second ranking approach is the clustering index, with respective MAP scores of 0.407 and 0.517. Finally, the PCA description space comes third with 0.266 and 0.458. Despite the high dimensionality of the concatenated feature vector for the full descriptors, this descriptor space still yields the best results. However, such a high dimensional space does not allow indexing the feature vectors as such to provide efficient retrieval performances. When working on the reduced dimensionality space, the multidimensional clustering index experimentally shows a better efficiency compared to the same dimension PCA feature vector. Let us recall that the results with the multidimensional clustering index are ranked according to the hamming distance while the other results are ranked with the L1 distance. Hence, a much coarser ranking is achieved with the multidimensional clustering index, where there are several ties in terms of hamming distance (which is an integer distance). Despite this coarse ranking, the MAP results shows decent performances and outperform the PCA based ranking. Moreover, as opposed to the PCA descriptors, it is possible with the clustering index to perform queries focused on particular descriptors. This information is lost in the reduced PCA space.

The first experiment showed the efficiency of the multidimensional clustering index with respect to a similar PCA-based feature vector. Nevertheless, the full feature vector yields the best results in terms of MAP. However, combined with the hamming distance, the primary



(a) ALOI

(b) WANG

Figure 3.3: Recall/Precision curves for range queries using the multidimensional clustering index and their equivalent **k-NN** queries using the full content descriptor and PCA descriptors

purpose of the multidimensional clustering index is not to perform such a task as ranked retrieval. Indeed, the hamming distance provides only coarse ordering of the set of results. The purpose of the index is rather to be used as a pre-filtering, which returns a set of candidate similar images. This set can be accurately ranked using the full descriptors in a second time. To demonstrate the efficiency of the clustering index for such a task, we performed a second experiment on the same datasets. The experiment consisted in performing range queries with the multidimensional clustering index, and computing Recall and Precision figures for this query. We varied the threshold on the hamming distance from 0 to $K - 1$. A hamming distance of 0 means that the query image and the results images are in the same clusters for all the visual features considered. A hamming distance of $K - 1$ means that the query and the images returned share at least one cluster according to one of the descriptors. Obviously, a hamming distance of K would return the whole dataset. Since it is not possible to use the same threshold with the other descriptors, we performed a **k-NN** search with k set to the cardinal of the set returned using the range query on the multidimensional clustering index. The results of these experiments are reported in figure 3.3.

Figure 3.3 shows the performances of the clustering index compared to the full content and PCA descriptors. Despite the lower MAP score, the multidimensional clustering index achieves better results than the full descriptors in terms of recall/precision for range query. Compared to the task of ranking using the coarse hamming distance, the clustering index is efficient by effectively grouping similar images into the same clusters. A low threshold (*e.g.* set to 0) will give a usually very small set of images which are visually close to the query, hence achieving a good precision but a low recall. The recall is increased as we become more tolerant in the threshold. Experiments on the ALOI database (figure 3.3(a)) show that when the threshold is too high (*i.e.* query image and returned results must share only 1 cluster) the full descriptors become more efficient. This is not the case on the WANG database where the clustering index stays better for all the tested threshold values (figure 3.3(b)). However as the threshold becomes high the capability of discarding irrelevant images is narrowed.

This experiment demonstrate that the multidimensional clustering index is effective as a pre-filtering, organisational tool of the image database into different views. Queries can be precisely defined by carefully choosing the set of descriptors we want to use when querying the index. For a finer ranking of the images by similarity, one should compute the accurate distance between the feature vectors for the images returned by the index.

3.2 Indexing via Semantic Clustering Model

We have seen in the industrial use case descriptions (chapter 1), that the typical job of a web crawler is to seek the web for interesting images and save them in a global image repository. Images in the repository are grouped into folders, representing particular, highly semantic, subjective categorization (such as car models, brands, etc.).

We propose to ease the work of such persons by constructing a model, based on global visual descriptors, that best approximates the user categorization. A good approximation of the user-defined categorization is more likely to be obtained using a combination of several visual descriptors. We propose to perform a hierarchical decomposition of the initial set of images until the user-defined number of group is reached. At each intermediate level of the hierarchy, the dataset (or sub-set) is split into several groups by k-means clustering. The clustering is performed using a single visual descriptor from a set of available descriptors. The process is illustrated in figure 3.4. We employ a brute force search of the possible hierarchies, and evaluate each solution with respect to the ground truth. The solution which yields the best assignment overall is retained. The hierarchical model is used to automatically assign new images to the existing folders. The objective of this task is very ambitious, and it would be an illusion to expect that there exist a perfect correspondence between visual and semantic similarities. However, we believe that an approximation of such a correspondence can be made. A problem is then to evaluate the quality of the automatic grouping and link the computed groups with the user-defined groups. In section 3.2.1, we describe the existing methods used to evaluate the quality of a clustering algorithm. In section 3.2.2, we propose to evaluate the quality of the partition as a best matching problem, for which efficient algorithm exists. We evaluate our method on datasets provided by our industrial partners in section 3.2.3.

3.2.1 Evaluating the quality of clustering

Evaluating the quality of a grouping of data is an open problem. The main difficulty comes from the fact that the task is highly subjective: different relevant clustering of the same data can coexists. Candilier identify four main methods to evaluate the quality of clustering [25].

1. Using artificial data: the grouping of the artificially generated data is known and can be compared with the output of the clustering algorithm. The problem lies in the fact

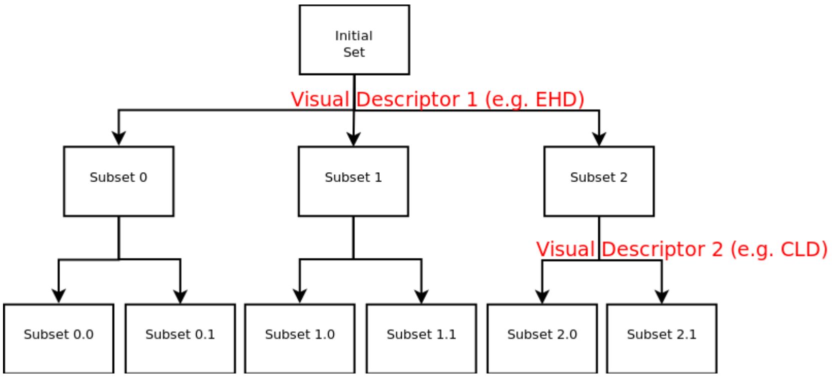


Figure 3.4: Hierarchical decomposition of the set of images into a user-defined number of clusters

that the clustering algorithm is confronted with specific distributions which does not necessarily reflect real data.

2. Using labeled data: observe if the method effectively groups data of the same label together. However the supervised labeled groupings does not always reflect the unsupervised grouping, where different clustering could be relevant.

3. Using numerical criterion: such as computing the intra/inter cluster energy. However these criterion are based on subjective notion of what a good clustering is. In example, well separated clusters is not always better than having overlapping clusters.

4. Using expert knowledge: a human expert can evaluate the relevance of a clustering for a given task. If the expert can say if a clustering is relevant, it is more difficult to quantitatively rate the clustering or saying if one clustering is better than the other.

In our context, we seek the clustering process that best approximates the human categorization of a set of picture. We have at our disposal a manually categorized set of pictures. Hence, we fall into the second approach for evaluating the quality of a clustering algorithm: using labeled data (**GT**). The drawback of this quality measurement is the fact that labeled groupings does not necessarily match unsupervised groupings which would still be relevant. However, it is our goal here to find the unsupervised grouping which best approximates the manual one. This will be achieved by customizing the set of descriptors used to characterize the images. When one wants to measure the quality of a partition according to the known classes of the data, the classical quality measurements of a clustering are the F-measure and Entropy. In section 2.5, we introduced the standard quality measures for classification tasks: recall (equation 2.46), precision (2.47) and F-measure (2.48). Those metrics are build for binary classification tasks, where the answer to a query is either relevant or irrelevant. In the case of clustering, we can generalize these metrics to sets.

Let us denote $\mathcal{S} = \{\mathcal{S}_\infty, \mathcal{S}_\in, \dots ,\mathcal{S}'_K\}$ the set of images categorized into K different categories. Those categories are given by a human expert and constitute the labeled data. Let us denote $\mathcal{S}' = \{\mathcal{S}'_1, \mathcal{S}'_2, \ldots, \mathcal{S}'_K\}$ the categories obtained by the algorithmic run. We note $|\mathcal{S}_i|$ the cardinal of the set. We note $N_{i,k}$ is the number of elements of category i that are found in the cluster k. The *generalized* recall for a category i in a cluster k is given by equation 3.2:

$$R(i,k) = \frac{N_{i,k}}{|\mathcal{S}_i|} \tag{3.2}$$

Likewise, the precision for category i in cluster k is given by equation 3.3:

$$P(i,k) = \frac{N_{i,k}}{|\mathcal{S}'_k|} \tag{3.3}$$

The *generalized* F-measure for sets is defined using the recall and precision measures as given by equation 3.4:

$$F(i,k) = \frac{(1+\beta^2) \times R(i,k) \times P(i,k)}{(1+\beta^2) \times R(i,k) + P(i,k)} \tag{3.4}$$

The entropy of a cluster is defined with respect to the precision by equation 3.5:

$$E_{(k)} = -\sum_{i=1}^{K} P(i,k) \times \log P(i,k) \tag{3.5}$$

An evaluation of the full partition is obtained by summing the max F-measure (equation 3.6) or the entropies (equation 3.7).

$$F(\mathcal{S}') = \sum_{i=1}^{K} \frac{|\mathcal{S}_i|}{|\mathcal{S}|} \max_{k=1}^{K} F(i,k) \tag{3.6}$$

$$E(\mathcal{S}') = \sum_{k=1}^{K} \frac{|\mathcal{S}'_k|}{|\mathcal{S}|} \times E(k) \tag{3.7}$$

Hence, the *best* clustering is the one that maximizes (3.6) or that minimizes (3.7).

3.2.2 The assignment problem

3.2.2.1 Problem formulation

The assignment problem is a classic combinatorial optimization problem [23]. Consider a set W (*e.g.* a set of *workers*) and a set T (*e.g.* tasks) of size N. Consider a *cost-function* $\mathrm{C} : W \times T \rightarrow \mathbb{R}^+$. The *best assignment* is a bijection B of W and T that minimizes the objective cost function: (3.8):

$$\text{Cost} = \sum_{i=1}^{N} \mathrm{C}(w_i, \mathrm{B}(w_i)) \tag{3.8}$$

The total number of possible assignments is $N!$, which is quickly intractable as N grows.

3.2.2.2 The Hungarian algorithm

A polynomial-time optimal solution to this problem was formulated by Khun [93]. The name of the algorithm comes from the fact that Khun was inspired by the earlier work of Hungarian mathematicians.

Let us consider the problem formulated in a $|W| \times |T|$ matrix $\mathbf{M}$ where $\mathbf{M}(w,t) = C(w,t)$ the cost of assigning task t to worker w. The best assignment problem can be solved using the following steps:

1. If necessary, add dummy variables to the matrix so that $|W| = |K|$. Set the cost to this dummy variables at least equal to the biggest assignement. Let $k = \min(|W|, |K|)$.

2. Substract the smallest element of each row of the matrix to all the elements in this row.

3. Find a zero element in the resulting matrix. *Star* this element if there is not an existing starred zero in it's row or column. Continue this procedure for all zeros.

4. Count the number of columns containing a starred zero. If it is equal to k, the starred zeros contain the complete set of assignements which solves the problem.

5. *Prime* an uncovered zero. If there is a starred zero in the row containing the primed zero, cover this row and uncover the column containing the starred zero. Continue until all zeros are covered. Save the smallest uncovered value and go to step 7. If there is not starred zero in the row containing the primed zero, go to next step.

6. Build a series of primed and starred zero: consider Z_0 the uncovered primed zero found in step 5. Z_1 is the starred zero in the column of Z_0(if it exists). Z_2 is the primed zero in the column of Z_1. Continue the serie until Zi is a primed zero with no starred zero in its column. Unstar the starred zero of the series, star the primed zeros of the series, erase all primes and uncover every line in the matrix. Go to step 4.

7. Add the smalled element found in step 5 to every element belonging to a covered row, and substract it to every element of an uncovered column. Return to step 5.

The assignments returned by the algorithm are identified by the starred zeros in the matrix.

3.2.3 Experiments

We carried experiments using 2 image datasets provided by our industrial partners. The first dataset, from FIAT, contained about 4000 images grouped into 4 different categories. Images from category **Front** (1019 images) contained pictures of car exterior taken from the front. Similarly are defined the pictures from the category **Profile** (1023 images). Images from category **Instrument Panel** (747 images) are pictures from the whole dashboard. Finally in the category **Cluster** (name given by FIAT, 795 images) are images from the speed and motor gauge. The dataset was split in two halves, one for training and one for test. The second dataset, provided by **RR**, consists in images extracted from multimedia documents produced during engine damage investigation process. This dataset is very heterogeneous, since no specific imaging condition is applied during the process. In this dataset, there was no human categorization of the pictures. In order to evaluate our method, we submitted the image dataset to **RR** experts so that they carry the grouping of images according to their own expertise. In the meantime, non experts users were asked to do the same thing. 509 training images were grouped into 36 categories by **RR** experts, and into 49 categories by non-experts. After the grouping of training images, 280 additional test images were assigned to the existing categories. Note that not all the defined categories were populated with images from the test set. Hence, 20 out of the 36 expert categories contained at least one image, and 31 out of the 49 non-expert categories.

We performed the search for an optimal hierarchical clustering model using the MPEG-7 descriptors EHD, CLD, CSD and SCD. We shorten down the exhaustive search by examining only the hierarchies that lead to *well balanced* trees. During the assignment problem, we set the weights to the F_1 between the ground truth and algorithmic clusters. The results for **FIAT** dataset are reported in table 3.3 and in table 3.4 for **RR** dataset. We report in the tables the percentage of correctly assigned images, after the link between the ground truth categories and the automatically obtained clusters has been achieved using the Hungarian algorithm.

Training Set	**Test set**
0.56	0.53

Table 3.3: Percentage of correct classification on FIAT dataset

Training Set	**Test set**
0.25	0.19
0.29	0.16

Table 3.4: Percentage of correct classification on **RR** dataset with expert categorization (1^{st} row), with non-expert categorization (2^{nd} row)

The results shown in tables 3.3 and 3.4 show the difficulty of the task. For the **FIAT** dataset, more than half of the images are correctly grouped together. The hierarchical model built using the training images is able to sort the test images with almost the same accuracy as the training images (56 and 53% respectively). For the **RR** dataset, we see that the task is even more complex. Considering the training images, the model built according to expert ground truth correctly classifies 25% of the images, and 29% according to the non-expert ground truth. Those results drop to 19% and 16% respectively for the test images. The discrepency of the results between obtained using the training and test images can be explained by the fact that, unlike **FIAT** dataset, training and test images were quite different. Indeed, a lot of groups defined in the expert or non expert ground truth with the training set remain empty with the test set.

3.2.4 Conclusion

The correspondence between a user-defined categorization and an algorithmically defined grouping supposes a good corresponding separability in the description space. To achieve this correspondance, we proposed a hierarchical decomposition of the dataset based on the selection of different description spaces for splitting the different levels of the hierarchy. We proposed to model the matching between the ground truth and the algorithmically obtained clusters as an assignement problem, and explored the set of possible hierarchical decompositions of the dataset in order to retain the best one. Experiments have shown that it is still difficult to approximate the user-defined ground truth. The background expertise and analysis of experts reminds hard to approximate using purely visual descriptors.

However, we would like to push forward the method in future research. One interesting path to investigate is to avoid the exhaustive search for the possible hierarchical decompositions. Indeed, the number of possible hierarchies becomes very large as the number of user-defined categories and the number of visual descriptors increases. We will investigate a solution taking the form of a greedy binary decomposition algorithm. Starting from the initial dataset, we compute the 2-means clusterings using each of the available visual descriptors. We evaluate these groupings and keep the one which achieves the best results. We can then select the cluster to split and apply the same procedure, until the target number of clusters is reached. This algorithm takes fully advantage of the proposed best-matching evaluation method. Indeed, the algorithm is independant of the cost that are set up between the groups. We can imagine to compute a cost function which varies with the current number of leaves in the hierarchical tree. Clearly, when the number of leaves is low, images from different ground truth category will be in the same groups. Hence, the precision of the clustering is also low but the evaluation should not penalize this. However, recall is important, as we do not want images from the same category to be split over different clusters especially when there are just a few of them. As the number of clusters in the hierarchy grows, precision becomes important. Hence, we could use the F_β metric and adjust the β coefficient with the number of clusters in the current decomposition to reflect those properties.

3.3 CBIR for specific imaging modalities

Methods for computing image similarities relying on general purpose visual descriptors such as color and texture might not be accurate for specific imaging modalities. Indeed, in some industrial processes, images are acquired by specific devices and computed from measurements on these devices. This is not limited to industrial processes, but is also the case for example in biomedical images, where there exists several imaging acquisition modalities.

The traditionnal CBIR approaches based on low level descriptors comparison might not be relevant for this kind of modalities. Hence, particular descriptors and particular similarities measures can be used to improve the retrieval quality. In the context of this Phd, vibration images produced by the Rolls Royce Experimental Vibration Use Case are special kind of images called Electronic Speckle Pattern Interferometry (ESPI) images. In this section, we will detail the work that has been accomplished for this use case. First, we will detail the acquisition process and acquire some insights about the particular nature of ESPI images (section 3.3.1). The notion of *similarity* in the context of vibration testing will be defined. Due to the specific nature of ESPI, we propose a visual descriptor based on the Radon transform (section 3.3.2). We introduce the mathematical background of the Radon transform, the motivations underlying it's usage in this context and the matching procedure that we adopted for the proposed descriptor. In section 3.3.3, we describe a method for accurate localization of the vibrating component in the ESPI image. Localizing the object is necessary to ensure more robustness to affine transformations of the Radon Transform, and is a challenging task for ESPI images. Finally, in section 3.3.4 we experiment our descriptor and compare the results with an approach driven by local descriptors (SURF).

3.3.1 Electronic Speckle Pattern Interferometry

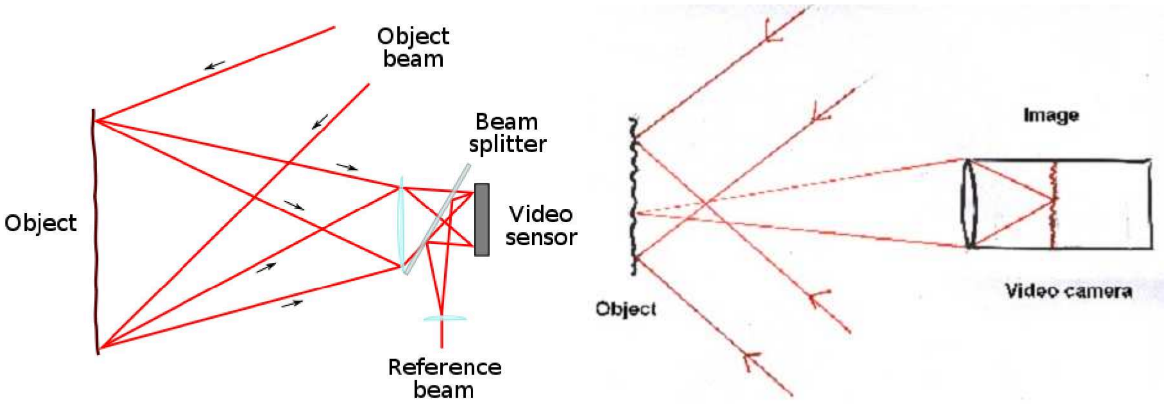
3.3.1.1 Principles

Electronic Speckle Pattern Interferometry is technique for recording and visualizing static and dynamic displacement of components with optically rough surfaces. The main advantages of this technique is it's effectiveness in performing non desctructive tests of the surface deformation of objects. This technique is commonly used for vibration, stress and strain measurements in various industrial field. The optical arrangement for ESPI images acquisition is shown in figure 3.5[1]. When illuminated with a laser beam, objects with optically rough surface are reflected as *speckle patterns*, a succession of dark and bright spots. The speckle patterns are inherently induced by the internal structure of the object. They can be seen as the fingerprint of the surface material of the object of study. When the object is submitted to a workload, it's internal structure (or part of it's internal structure) will move. By substracting the speckle pattern of the initial object configuration (reference speckle) and the current speckle pattern, correlation fringes appears that indicates the area of displacement of the object (figure 3.6[1]). The displacement can be measured in all 3-D directions. For the interested readers, a book on the theory and practice of ESPI imaging was written by Jones and Wykes [86]

3.3.1.2 Vibration testing using ESPI imaging

A vibration test consists in exciting the object of study at different frequencies in order to discover its vibration *modes*. A vibration mode is characterized by a modal frequency and a mode shape. A mode shape is a description of the expected curvature of the surface vibrating at this particular mode. As opposed to correlation fringes which are obtain by substracting speckle patterns, vibration mode shapes are directly observable in the ESPI images, as parts of the object which vibrate will appear with a higher contrast speckle than part of the object which do not vibrate (figure 3.7[1])

[1] Figures extracted from Wikipedia: `http://en.wikipedia.org/wiki/Electronic_speckle_pattern_interferometry`, March 2010



(a) Out-of-plane displacement

(b) In-plane displacement

Figure 3.5: Optical arrangement to obtain ESPI fringes

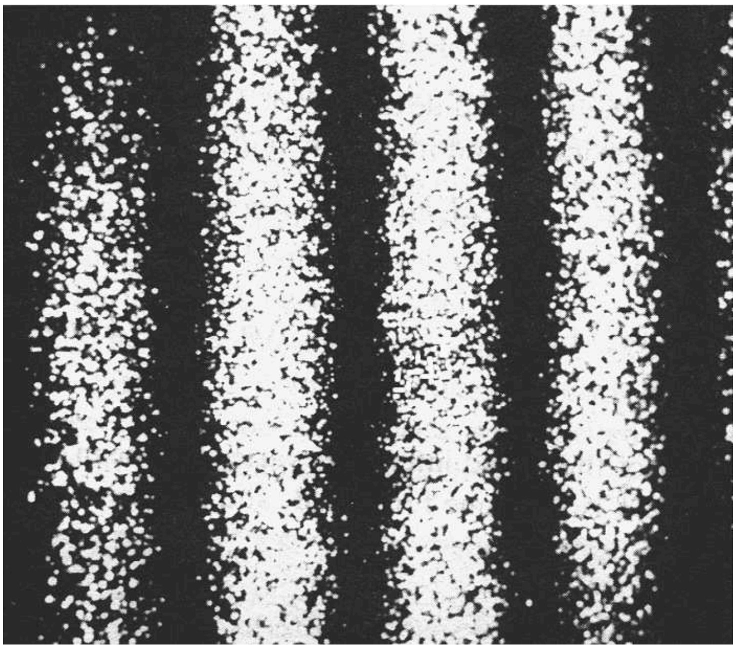


Figure 3.6: ESPI fringes - flat plate rotated about a vertical axis - the fringes represents displacement in the viewing direction; the difference in displacement between fringes is about 0.3 μm.

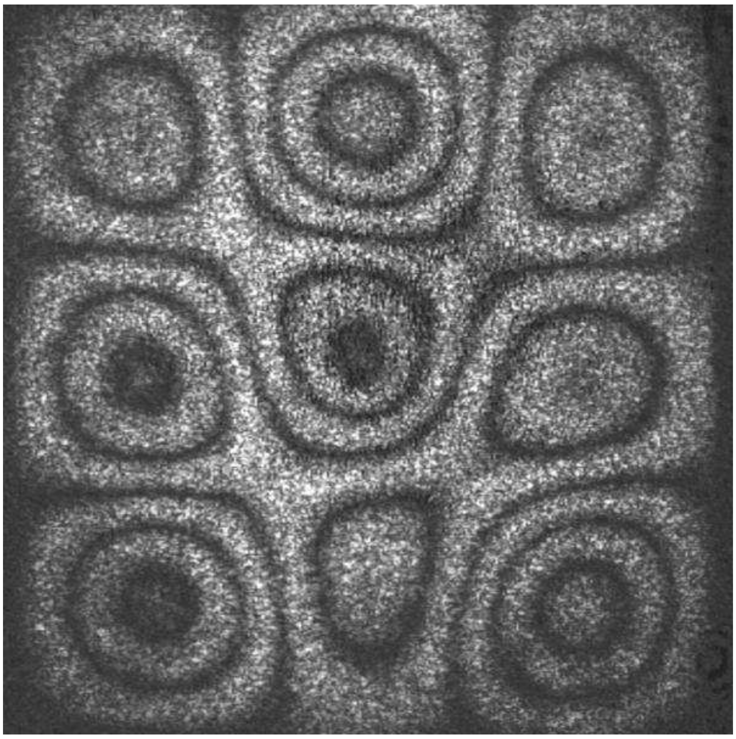


Figure 3.7: ESPI fringes showing one of the vibration modes of a clamped square plate

3.3.1.3 Semantic of *similarity* for ESPI vibration images

During vibration tests, specific modeshape will appear at some frequency of excitation. The appearance of some modeshape patterns gives the vibration engineers a hint about the reliability of the component. Some specific modeshape patterns appear only in component showing a weakness in their internal structure. Usually, the recognition of such problematic modeshapes relies on the expertise of skilled, senior engineers.

For vibration testing, the object is vibrated at a specific frequency. The parts of the object which do not move will continue to be speckled. Depending on the vibration frequency, parts of the object will speckle with high or low contrast. The particular configuration between the high contrast and low contrast speckles are visible on the image as fringes showing specific mode. A mode is a particular configuration of the location of the fringes. Hence, the semantic similarity between modeshape images as seen through the user point of view is the similarity between the modeshape configurations, that is the number and relative location of the fringes. Several challenges arise for solving this problem: first, the large amount of noise due to the imaging process makes preprocessing steps such as image segmentation very inaccurate. Second, traditionnal low level descriptors such as texture and colors are useless for such kind of images. Third, it is important to localize, among the noisy image, the patterns formed by the vibrating part of the component. Thus, we propose an alternative low level description of the image content based on the Radon transform, and an associated similarity measure.

3.3.2 ESPI image descriptor based on the Radon Transform

3.3.2.1 The Radon Transform

The Radon Transform [160] is a technique traditionally used in Computer Tomography for the reconstruction of images from their integral projections. We are not concerned with reconstruction of images but with the direct application of the Radon transform, for image description purpose. The Radon transform is the integral of a two-dimensional function $f(x, y)$ over straight lines. Equation 3.9 gives the expression of the Radon transform over a line positioned by ρ –the smallest distance from the line to the origin– and θ –the angle between the horizontal axis and the perpendicular to the line–.

$$R_f(\rho, \theta) = \iint\limits_{-\infty}^{+\infty} f(x, y)\delta(x\cos(\theta) + y\sin(\theta) - \rho)dxdy \tag{3.9}$$

where δ is the Dirac delta function.

From the definition of the Radon transform, we can derive the following mathematical properties [41, 158] that link transformations of the image plane with transformations in the Radon domain:

1. Translation: a translation of $f(x, y)$ by a vector $\vec{u} = (x_0, y_0)$ leads to a the modified Radon Transform $R(\rho - x_0 cos(\theta) - y_0\sin(\theta), \theta)$
2. Rotation: Rotation of f by an angle θ_0 leads to $R(\rho, \theta + \theta_0)$
3. Symmetry: $R(\rho, \theta) = R(-\rho, \theta \pm \pi)$
4. Periodicity: $R(\rho, \theta) = R(\rho, \theta \pm 2\pi)$
5. Scale: a scaling of f by a factor $\alpha \neq 0$ leads to $\frac{1}{|\alpha|}R(\alpha\rho, \theta)$

In a CBIR application, it is necessary to ensure robustness to affine transformation of the image, illumination changes, etc. An appropriate similarity measure has to be found that could cope with the aforementionned properties. In the following sections, we present the advantages of the Radon transform when dealing with ESPI images. Then, we propose a computation of a similarity score between images in the Radon domain. We will show the effectiveness of our approach in real data.

3.3.2.2 Motivations

The main domain of application of the Radon transform since its formalization by Johan Radon in 1917 [137] is computer tomography. Tomography images such as CT-Scans are directly acquired in the Radon domain, and the inverse Radon Transform needs to be applied to reconstruct 2D images and 3D volumes [61]. Nevertheless, the Radon Transform has also been studied and successfully applied in other contexts. Benois-Pineau *et al.* [17] have proposed an approach to video structuring and indexing based on 1D image representation in videos. Based on the discrete *Mojette* Radon Transform [69], they propose an estimation of the motion in the 1D domain and a color signature of video shots. Classical video indexing operations such as shot change detection and scene grouping can be performed in the 1D domain, reducing drastically the computationnal burden. In addition, they proposed a concept of 1D spatio-temporal mosaicing which allows the definition of a spatio-temporal color signature. An application of the Radon Transform for CBIR was proposed by Tabbone and Wendling [158]. In this method, the author characterize objects using the R-signature, which is computed from the Radon Transform according to equation 3.10.

$$R_s(\theta) = R^2(\rho, \theta) d\rho \tag{3.10}$$

The R-signature can be thought intuitively as the repartition of the energy of the shape for the different orientations. Based on the mathematical properties of the Radon Transform, the R-signature is theoretically invariant to affine transformations. Experimental validation has been provided by the authors in their publication. A method to extend the R-signature to grayscale and color images is also given. Seo *et al.* have proposed robust fingerprinting system based on the Radon transform [148]. Fingerprinting is the process of identifying multimedia content using a fixed small number of bits. This process is similar to cryptographic hashing. However, cryptographic hashing functions cannot be applied directly to multimedia content: a single bit change in the content of a file will lead to completely different cryptographic key. However, it is common that multimedia objects can be modified without changing their perceptual meaning, *e.g.* due to compression, affine transformations, etc. Hence, fingerprinting functions must invariant under perceptual similarities, while very different for perceptually different objects. Seo *et al.* rely on the mathematical properties of the Radon transform to propose one such function, which is invariant to affine transformations.

In the context of ESPI image similarity search, the use of the Radon transform yields several advantages. First, it is obvious that descriptors based on colour are not applicable to this modality. Similarly, descriptors based on texture are not relevant, since they will more likely capture the speckled aspect of the surface. Moreover, the important feature for modeshape similarity is the location and geometric arrangement of the fringes. Descriptors based on the shape of the fringes could provide the most useful information. However, shape descriptors requires an initial segmentation, which is not trivial with such images. The Radon Transform enables to capture such geometric shape information, without the need of a segmentation. This is illustrated in figure 3.8. On each line, we see two example images of component which vibration exhibits a specific modeshape pattern. The white rectangle on the top left of the images hides the vibration frequency that is displayed in the original image. On the left of each image, we display two curves. The green curve is the 1D cut profile of the image extracted along the vertical line in the middle of the image, that is the grey level values of the pixels of the image that go through this line. The line itself is highlighted in white. The red curve correspond to the integral projection of the image on the vertical axis, *i.e.* a single projection of the Radon Transform on the vertical axis. Similarly, we display below each image the 1D horizontal cut profile and 1D integral projection along the horizontal axis. The shape of the object is characterised by the intensity distribution over the different projections. As illustrated in figure 3.8, the vertical integral projection is quite similar for all the images with a large hill in the center of the blade and a slowly decreasing slope with some minor bumps on each side of the main hill, toward the edges of the component. However, the horizontal integral projection is quite different for the two modeshapes. It looks like a large plateau starting from the left border of the component, followed by a slowly increasing slope and a small plateau on the right of the component (figures 3.8(a) and 3.8(b)) for the images of the first line. For the images of the second line, it looks like a large valley delimited by two clear hills on the left and the right of the component. These profiles indeed reflects well the modeshape pattern, where two vibrating regions appear as darker oscillations in the first modeshape, one above the other, while there are 4 such oscillations in the modeshape of the images of the second line.

An important feature of the Radon Transform for ESPI images is that the integration of the 2D signal over 1D projections performs a strong low pass filtering, hence reducing the effect of speckle noise. This is clearly depicted in figure 3.8, where the green lines corresponding to real pixel values are very noisy with a sawtooth aspect while the red lines are smooth.

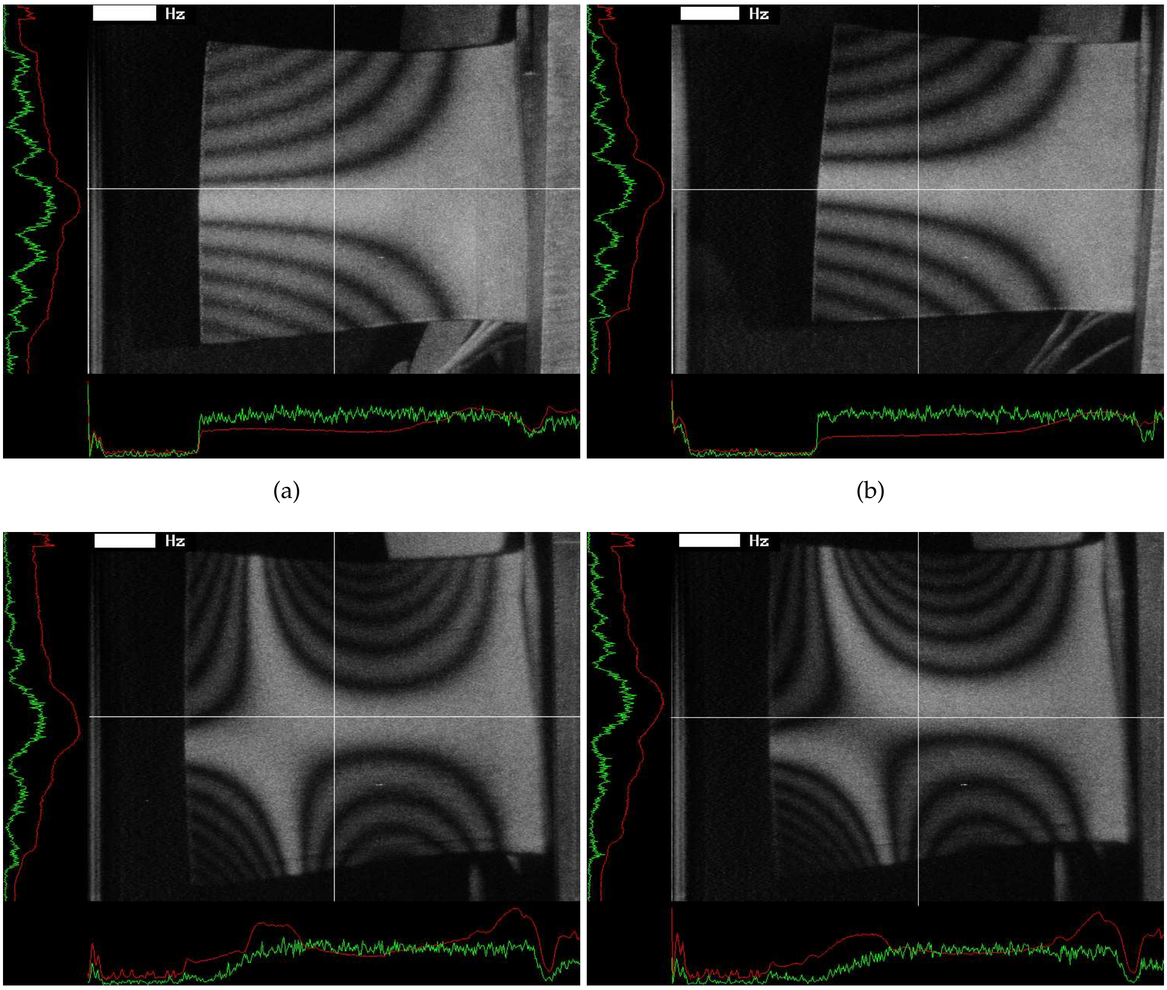
3.3.2.3 Similarity measure

The previous section have shown the advantages of applying the Radon transform in the context of ESPI images. In order to achieve the goal of ESPI image similarity search, we must define a similarity measure in the Radon domain. Let us consider two images $f(x, y)$ and $g(x, y)$, $R_f(\rho, \theta)$, $R_g(\rho, \theta)$ their Radon transform with respect to the parameters ρ, θ. We propose to measure the similarity between two projections with their normalized cross correlation coefficient (equation 3.11):

$$\gamma_u(R_f, R_g) = \frac{\sum_x [R_f(x) - \bar{R_f}][R_g(x-u) - \bar{R_g}]}{\sqrt{\sum_x [R_f(x) - \bar{R_f}]^2 [R_g(x-u) - \bar{R_g}]^2}} \tag{3.11}$$

The correlation coefficient is a statistical measure of the similarity between two signals. The experimental setup to acquire the ESPI images lead to a controlled environment for capturing the image. Each image contain a single, centered object (see figure 3.8). All the images have the same resolution (640x480 pixels), and we set an equal size for each radial projection, equal to the size of the diagonal of the image, with zero padding if necessary. Hence, we can compute a single value of the correlation coefficient $\gamma_{u=0}$, without shifting the signals. Let us consider the discretization of the (ρ, θ) plane with δ_ρ and δ_θ steps, $\rho \in [0, \rho_{max}], \theta \in [0, \pi]$. If we consider that the Radon transform is computed along N sampled directions of projection, $\theta_n = \theta_0 + n\Delta\theta, n = 0, \dots, N-1$ the similarity between two images $f(x, y), g(x, y)$ is given by equation 3.12:

$$S(f, g) = \frac{1}{N} \sum_{n=0}^{N-1} \gamma(R_f(\rho, \theta_n), R_g(\rho, \theta_n)) \tag{3.12}$$



(a) (b)

(c) (d)

Figure 3.8: Examples of ESPI images. The first line shows two different components which vibrate at a specific modeshape, the second line shows another modeshape. The green line on the left (resp. below) each image is the 1D vertical (resp. horizontal) cut profile extracted along the white vertical (resp. horizontal) line. The red line is the integral projection of the image along the vertical (resp. horizontal) axis.

Let us examine the robustness of the similarity measure proposed in equation 3.12 with respect to different transformations.

- **Rotation:** The formula given in equation 3.12 does not consider rotations of the object. In the dataset, due to the constraints inherent to the image acquisition technique, there is no need to compensate for rotations of the object. Nevertheless, according to the mathematical properties of the Radon transform, a rotation of angle θ_0 in the image plane leads to a shift of the radial projections in the Radon domain. Hence, if we consider the set of possible permutations Π of the projections, we can reformulate equation 3.12 as:

$$S(f,g) = \max_{\Pi} \left\{ \frac{1}{N} \sum_{n=0}^{N-1} \gamma(R_f^{\Pi}(\rho,\theta_n), R_g(\rho,\theta_n)) \right\} \tag{3.13}$$

  The maximal similarity measure is obtained for the permutation which compensate the rotation of the image. The coarseness of rotation compensation is then proportionnal to the number of radial projections computed.

- **Brightness:** when computing the normalized cross correlation coefficient (cf equation 3.11), the images are normalized, by substracting their mean and dividing by their standard deviation. This ensure robustness with respect to a uniform change in the brightness due to varying illumination conditions.

- **Translation:** The effects of translation on the radial projections is a shift of the signal by a quantity proportional to the translation vector. The normalized cross correlation function is obtained by applying equation 3.11 with a shift of one of the signals. Again, translations can be compensated by retaining the global maximum of the γ_u value, as was proposed by Benois-Pineau to estimate the motion of video sequences from the Radon domain [17]. However as we said earlier on, we do not compute γ for the full signal range, but we consider that the signals are already registered. As the object is not always perfectly aligned with the center of the image, we propose a method to localize the center of the component. The Radon transform is computed using the object center as the reference point by which the radial projections go through, hence registering the signals with respect to translations.

3.3.3 Object center localisation

Due to the specific nature of ESPI images, it is difficult to obtain a reliable segmentation from the image. The high quantity of speckle noise, the low contrast in particular in the most interesting regions (*e.g.* the vibrating regions), the fact that some structures which are not part of the component of interest appear in the image are examples of challenges for image segmentation. However, as stated above, it would be interesting to compute the Radon transform with respect to the center of the component as a reference point. An easy way to obtain such information would be that the operator manually localize the component center in the image when acquiring the image. Compared to the set up of the experimental protocol to obtain ESPI images, the additionnal burden is insignificant. These information could be written in metadata about the experiment, that would be efficiently processed within the context of the XMedia project. However, this is only possible for future experiments. One important aspect of the RR EV use case is to take advantage of the processing of historical knowledge, which is currently stored as sets of images from experiments, and based on the experience of skilled engineers. We propose an algorithm for localizing the component of interest from the images that take advantage of the redundancy of information we can get from historical data. Indeed, the set up for the lab experiments for vibration testing is the following:

1. Choose a component to test
2. Set up the ESPI acquisition system
3. Get the component to vibrate to an increasing frequency
4. Save the images obtained for every reasonance frequency.

Hence, there exists several images of the same component vibrating at different frequencies. It is important to recall that the component stay at the same position during it's whole testing. We take advantage of all these images to compute an *average* image of the component. Inspired by the moving object segmentation techniques in videos [31], we propose a foreground/background mask of the image by computing the difference between a particular instance image of the component and it's average image. Since for each image, only a portion of the component correspond to its vibrating part (*i.e.* it's foreground), we obtain a more complete mask of the whole component by adding (*i.e.* bitwise AND operation) each separate mask. Hence, we obtain a foreground/background mask for each component, where the foreground part correspond to the component of interest. The algorithm for computing such a mask is detailed in algorithm 1.

Algorithm 1 Algorithm for Component Localization from ESPI images

```
 1: BinaryImage : COMPONENT_LOCALIZATION(images $\mathcal{I}$)
 2: Image averageImage = $\frac{1}{\text{sizeof(Images)}} \sum_I Images(i)$ (figure 3.9(b))
 3: BinaryImage mask = 0
 4: for all Image Img $\in$ Images do
 5:     BinaryImage thisMask = $\begin{cases} 0 \text{ if } |Img - AverageImage| < T \\ 1 \text{ otherwise} \end{cases}$ (figure 3.9(c))
 6:     MORPHO_OPEN(thisMask)
 7:     mask = mask $\vee$ thisMask
 8: end for
 9: MORPHO_OPEN(MORPHO_CLOSE(mask)) (figure 3.9(e))
10: return mask
```

Let us detail algorithm 1 with the illustrations of figure 3.9. Line 1 is the name of the algorithm. It takes as parameters a set of images $\mathcal{I}$ from the same component. It returns a binary image corresponding to a foreground/background segmentation of this component (see figure 3.9(e)). In figure 3.9(a) we show one of the image of the set $\mathcal{I}$. As we can see, a large part of the image is occupied by the supporting structure which is not part of the component itself. In line 2 of the algorithm, we compute the *average* image by simply computing the average pixel value over the set $\mathcal{I}$ for each pixel. The average image for the component of figure 3.9(a) is given in figure 3.9(b). It is possible from this image to localize the component border more clearly than in the original image where the vibrating parts on the left edge of the component appear with verly low contrast. In line 3, we initialize an empty binary image mask. Then, for all images in the set (line 4), we proceed to compute a binary mask (line 5). This is done by computing the pixelwise absolute difference between the original image and the average image. If this difference is larger than a threshold T, the pixel is considered as foreground, otherwise, it is set to background. The result of this operation is illustrated in figure 3.9(c). We set the threshold to $T = 10$. As we can see, this segmentation is very noisy, with a lot of speckles appearing in particular in the visible structures which are not part of the component. The vibrating part of the component are highlighted by much more dense, white area. To get rid of these isolated white pixels, we perform a simple morphological opening with a rectangular structuring element of size 3×3. The result of the morphological filtering is shown in figure 3.9(d). Figure 3.9(d) hence corresponds to the fg/bg mask computed with a single instance image of this specific component. The global mask is obtained by computing the binary OR operation with all the masks hence obtained. We perform a morphological closing of the final mask in order to reconnect the possibly disconnected part of the component and a morphological opening to get rid of small isolated regions (figure 3.9(e)). Once the mask is obtained, there can exist several connected components. Typically, in figure 3.9(e) there is a white rectangle in the top left hand corner corresponding to the area where the vibration frequency is written. We assume that the area corresponding to the actual component localization is the one occupied by the biggest connected component in the mask. In figure 3.9(f), we outline in green the localization of the component hence obtained as the convex hull of the biggest foreground region.

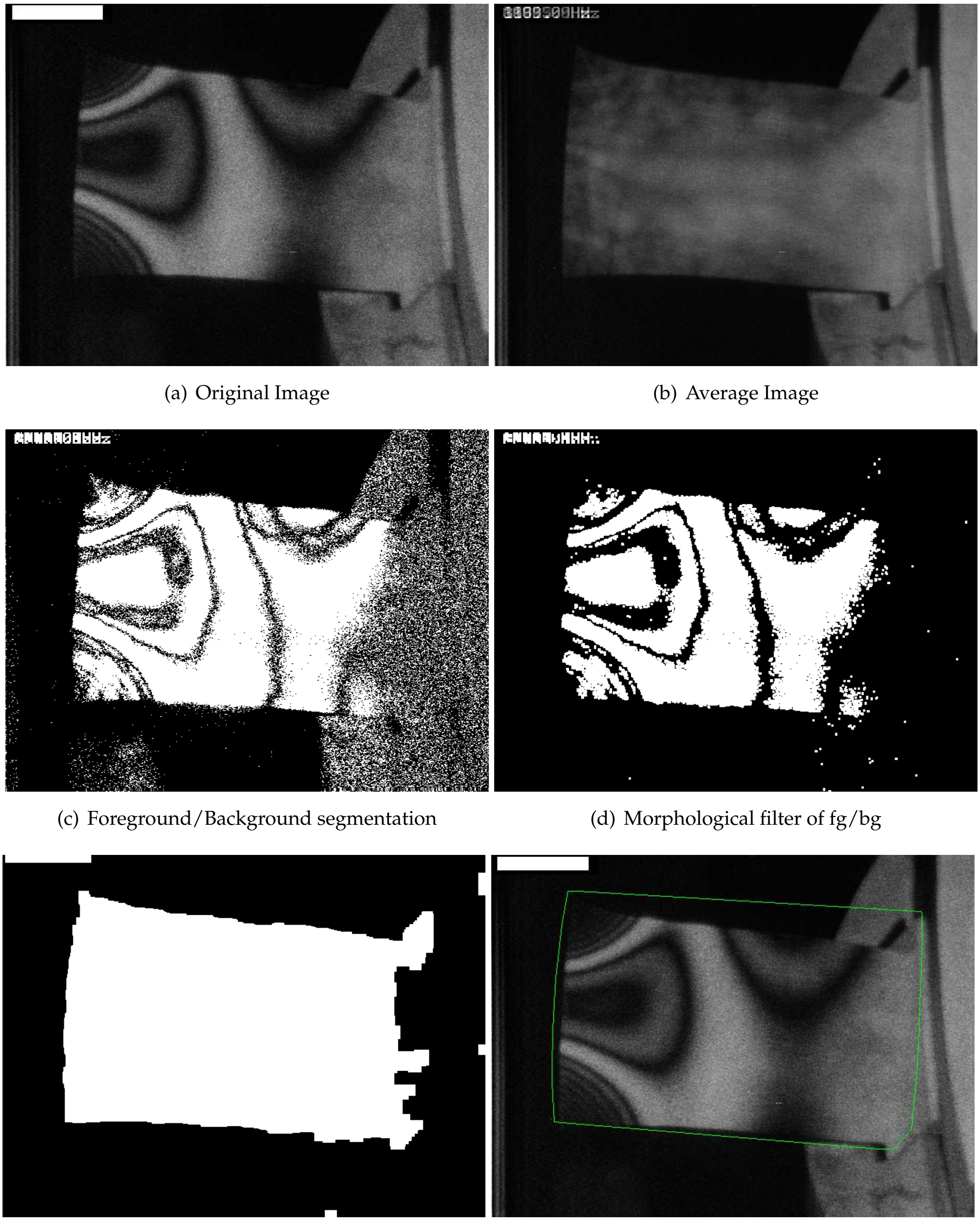
3.3.4 Experimentations

3.3.4.1 Keypoint-based similarity score

In order to assess to performance of our descriptor with respect to keypoint based descriptors, we compared it with a baseline interest-point based image matching comparison using SURF points [15]. We set up the SURF descriptors to 64 dimensions, and the distance between SURF descriptors is the euclidean distance between the feature vectors.

We compute a similarity score between image I_a and I_b as follows: let us note $\mathcal{K}_a$ and $\mathcal{K}_b$ the respective keypoints of I_a and I_b, and $\mathcal{K}_{a,b}$ the set of matching pair of keypoints whose definition will be given below. Then, the similarity score $S(I_a, I_b)$ is computed as the ratio of the number of keypoints in I_a that can be matched with keypoints from I_b by the total number of keypoints of I_a. This is given by equation 3.14:

$$S(I_a, I_b) = \frac{|\mathcal{K}_{a,b}|}{|\mathcal{K}_a|} \tag{3.14}$$



(a) Original Image

(b) Average Image

(c) Foreground/Background segmentation

(d) Morphological filter of fg/bg

(e) Final Fg/Bg mask

(f) Component Localization

Figure 3.9: Illustration of ESPI images component localization

Note that the similarity score is not symmetric, *i.e.* $S(I_a, I_b) \neq S(I_b, I_b)$ if $|setK_b| \neq |setK_b|$. The set of matching keypoints $\mathcal{K}_{a,b}$ is computed according to equation 3.15:

$$\begin{aligned}\mathcal{K}_{a,b} = \{(k_i \in \mathcal{K}_a, k_j \in \mathcal{K}_b) | L_2(k_i, k_j) < 0.6 \times L_2(k_i, k_l) \in \mathcal{K}_b\} \text{ with} \\ L_2(k_i, k_j) \leq L_2(k_i, k_m) \forall k_m \in \mathcal{K}_b, \text{ and} \\ L_2(k_i, k_l) \leq L_2(k_i, k_m) \forall k_m \in \mathcal{K}_b, k_m \neq k_j\}\end{aligned} \tag{3.15}$$

Equation 3.15 state that a pair of keypoint (k_i, k_j) is a match if the distance between k_i and k_j is sufficiently smaller than the distance to the second closest neighboor of k_i. This condition was formulated by Lowe to identify robust matches among SIFT keypoints [103], and is also applicable to SURF keypoints.

3.3.4.2 Dataset and experimentation protocol

Our dataset consists of 1309 ESPI images of vibration tests given by our industrial partners. The database has been split into 213 folders containing images of component showing a specific modeshape. This classification has been done under the supervision of skilled vibration engineers from RR. The categories are heterogeneous in terms of the number of representative per category, the most populated folders having 102 images while 52 out of 213 categories had only 1 image. In all the experiments, we computed the MAP score by querying the sytem with every image in the database. For a specific query, the system computes the similarity score of all the images with respect to the query and returns the list of images sorted by decreasing similarity. We tested four different systems:

1. SURF: Keypoint-based similarity score as described above

2. Radon Full: the descriptor is the Radon transform of the full image, with respect to the image center.

3. Radon Blade Center: the descriptor is the Radon transform of the full image, with respect to the automatically computed component center.

4. Radon Segmented: the descriptor is the Radon transform of the segmented image, that is only the pixels which are part of the automatically extracted component mask are taken into account, the other pixels beeing set to zero.

For all Radon system, the similarity score is the one given in equation 3.12. We set the number of radial projections in the Radon transform to 10, and the angles of projections are uniformally sampled starting with an angle of 0.

3.3.4.3 Results and discussion

The MAP scores obtained by the different systems are reported in table 3.5. We reported the total MAP scores and the MAP score for the 50 most populated categories. Figure 3.10 shows the average precision per category for the 50 most populated categories in the dataset. The

System Name	Total MAP score	MAP score for the 50 most populated categories
SURF	0.225	0.139
Radon Full	0.468	0.365
Radon Blade Center	0.428	0.327
Radon Segmented	0.469	0.372

Table 3.5: MAP scores for the different CBIR systems run on the ESPI image dataset

first noticeable fact from figure 3.10 is the fact that any of the systems based on the Radon transform outperform the results obtained with SURF keypoints. As expected, although keypoint-based descriptors are pushing the state-of-the-art in a lot of computer vision related applications [83], they are not applicable in the case of this dataset. The low contrast and the high quantity of noise makes the localization of interest points difficult, and leads to poor matching results. This confirms our conviction that specific features have to be chosen for specific image type.

Concerning the differences between the systems based on the Radon transform, it seems that the Radon Transform computed with respect to the center of the blade performs slightly lower than the other systems. The Radon transform computed on the segmented components performs almost the same than the Radon computed on the whole image, but the benefits of segmentation are slightly higher when comparing the scores for the most populated categories (0.372 to 0.365). However, we note that the average precision scores with segmentation seems to be more sensitive than without segmentation, in the sense that for some categories the increase in performances is substantial while it performs the worst of all Radon-Based methods for some others. A closer look at the results per category shows that the decrease in results is most often due to an incorrect segmentation, where the object of interest was misdetected. This can happen when the assumption that the biggest connected component of the mask is the object is false. On the countrary, the biggest increase in results are obtained for categories in which the images show different component with a substantial part of background visible in the image. When segmented, this background structure is discarded leading to more robust results. In our publication about this method [170], we led experiments using a reduced dataset of 593 images with a manual segmentation. The results showed that the Radon transform computed with respect to the object center performed slightly better than the Radon transform computed with respect to the image center. The Radon transform computed using the (manually) segmented components showed a substantial improvement over the other methods. Hence, we assume here that the improvement brought by the segmentation could be much more beneficial on this dataset with a better automatic segmentation. As stated earlier on, a robust segmentation of the blade could be acquired manually when performing vibration tests and used in our algorithm, without much additionnal burden for the user. Another benefits of the segmentation is the fact that the computationnal burden is lowered, both when computing the Radon transform and when computing the similarity measure.

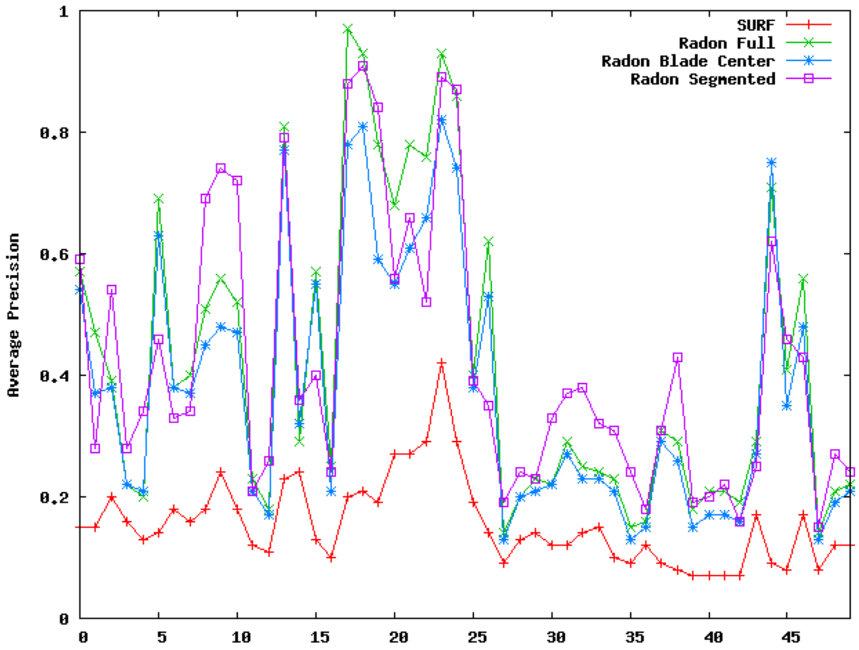


Figure 3.10: Average Precision Scores for the 50 most populated categories for the different CBIR systems.

3.4 Conclusion

In this chapter, we have proposed three approaches whose common background is in the application of global image descriptors for different image management task in the context of industrial processes.

In the first section, we proposed an indexing scheme of an image database that handles a part of the inherent problem of the definition of similarity for a user, which is always context-dependant. A *multidimensional* clustering index is computed using multiple clustering of the image database with different visual descriptors. Experiments showed that ranked retrieval results on this index, despite the coarse ranking of images using the hamming distance, outperforms a PCA based descriptor of the same dimensionality. Moreover, we demonstrated that this index is suitable for range queries, where a set of image potentially relevant to the query can be quickly retrieved. A finer ranking of the set can be performed using a more appropriate distance measure.

In the second section, we investigated the ambitious problem of approximating a user-defined grouping of images using clustering algorithms. We model this process using a hierarchical decomposition of the dataset, where the splitting decisions at the different levels of the hierarchy are taken according to different sets of visual descriptors. We performed a brute force scanning of the possible hierarchies, and evaluated those models using an evaluation method based on solving the assignment problem.

Finally, in the third section, we proposed an approach that was closely linked to the

RR Experimental Vibration use case of the **X-MEDIA** project. The very specific nature of the images handled in this use case makes traditionnal color and texture visual descriptors useless. We proposed a descriptor based on the Radon transform of the images that filter out the high quantity of noise present in the images while keeping information about their main structures. We proposed a similarity matching measurement using this descriptor and showed experimentally that the results were better than using a keypoint-based matching approach.

Chapter 4

Local Image Analysis

4.1 Introduction

In this chapter, we will focus on another problematic formulated the context of this PhD, that is the accurate localization (or extraction) and recognition (or categorization) of specific object classes within images. The object detection and categorization aims to extract two kinds of information from the images: which objects are present in the image and where exactly are those objects located. The detection is closely linked to the task of object segmentation. One of the difficulties of object segmentation is to extract complex, highly structured objects composed of regions visually dissimilar. On the other hand, the recognition of objects is affected by several problems: low quality of images, partial occlusions, visually heterogeneous object categories, cluttered background, . . . Many research efforts have been carried in the task of global image recognition such as scene classification [150, 64, 131], but object recognition remains a challenge. This task is even more ambitious when multiple objects are present in the images.

One of the earliest work closely linked to the object detection and recognition topic was proposed by Duygulu *et al.* [49]. In this work, the authors address the recognition problem as the process of attaching words to image segments considering the task as a translation between one language (English words) to another (visual words, or *blobs*). A mapping between the keywords and the visual blobs is performed using a method based on Expectation Maximization. The rest of the literature [72, 151, 168, 62, 9] noticeably differs from the original work by Duygulu *et al.* in the sense that the models built try to exploit the maximum of information that can be extracted from the image: not only low level features (color, texture, *etc.*), but also local contextual relationships between pixels or image segments, location and even global relevance estimates. He *et al.* [72] proposed a pixel-wise labelling into a finite set of labels using a multiscale conditional random field formulation(mCRF). The mCRF combines the output of a local pixel-wise classification, a representation of the local geometric relationships between the objects and global label features into a probabilistic framework, trained using supervised labeled image data. Shotton *et al.* [151] introduced a new approach for learning a discriminative model of object classes based on texture, layout and context information. Their model is based on *texture-layout filters*, a feature which jointly model patterns of texture and their spatial layout. Texture layout filters are combined with lower level image features (color, location) into a Conditional Random Field to provide a high-level discriminative model. They also discuss an efficient learning method of the CRF parameters based on boosting and piecewise training method. Verbeek and Triggs [168] point out that Markov Random Field and aspect models such as Probabilistic Latent Semantic Analysis and Latent Dirichlet Allocation are complementary methods which attempt to improve the coherence of the labellin of image patches. MRF provides crisper local labelling by exploiting neighbourhood-level couplings while PLSA and LDA use global relevance estimates. They studied two spatial extensions of the aspect models, one based on a forest of minimal spanning trees and the other one on a regular 8-neighbor MRF. Galleguillos *et al.* [62] have shown that introducing contextual information about the co-occurrences and the relative location of image regions with local appearance-based features improves the global labelling. Athanasiadis *et al.* [9] define a framework for simultaneous image segmentation and object labelling operating at the semantic level. They represent the contextual information as an ontological taxonomy of the set of possible semantic labels and employ fuzzy algebra to adjust the labelling of the regions given by region growing segmentation algorithms.

In this chapter, in order to build models for the recognition of object classes under different conditions, we have relied on supervised classification and more precisely Support Vector Machine (SVM) classifiers. In section 4.2, we will introduce elements of the theory underlying this classification model. Then, we propose two different approaches of the problem. In section 4.3, we adopt a region-of-insterest (ROI) based framework. ROI are identified in the image as potential localization of an instance of the object. Then, a classifier is used to determine the presence or absence of the object in the ROI. We applied this procedure for the concrete task of air-duct detection within car images.

In section 4.4, we face the problem of multi-class object detection and localization using image segmentation. Unlike the ROI approach, the image is fully segmented and each region is submitted to a classifier. Several approaches are proposed to improve the region classification results based on the global coherence of the labelling and the late fusion of multiple segmentation/detection results.

4.2 Introduction to Support Vector Machines

4.2.1 Supervised Learning Theory

Since the emergence of computer science, man has been interested in transfering it's ability to *learn* to machine. Machine learning has been considered as a central topic in Artificial Intelligence since it's creation [113, 114, 42]. However, although one could agree on the common understanding of the concept of learning, it's formal content remained fuzzy. A consensual definition of learning usually involves a *learner* having some computationnal capabilities, an object to learn, a protocol defining how the object to learn is presented to the learner and a criterion of success which state when the said object is learnt [28]. It is then necessary to answer the question: what is learning? Learning is the process of adapting some parameters to obtain the desired answer to an input or a stimulus.

4.2.1.1 Inductive Inference Principle

Learning methods aim to identify object classes using object specific characteristics (features), and a chosen protocol. Learning methods are applicable to a high number of human activities and are suitable for automatic decision making processes [38]. A straightforward approach consists in building *expert systems*, where the expert knowledge is described as a set of rules applied to classify new data [125, 161]. The efficiency of the learning system is highly depedent on the ability to correctly extract and describe expert rules. In this section, we consider that the learning/classification procedure is extracted automatically from a set of training examples. A training example consists in the description of a specific case, along with it's corresponding classification (ground truth). Given a set of training examples, a learning system must be able to compute a classification procedure able to classify some given test examples. The process of computing a generic classification model according to a set of training examples is known as *inductive inference*. Given the training samples, the aim is to infer a classification procedure that minimizes the classification error. Not only the classification procedure must classify correctly the training samples, but also it should predict correctly previously unknown samples. Hence, it is essential that the training process is accurate enough to be able to learn, but should prevent *over*-learning.

4.2.1.2 Statistical Learning Theory of Vladimir Vapnik

We present here the theory of statistical learning that was formalized by Vapnik [164, 165] and it's concrete application to classification using Support Vector Machines.

In supervised learning, we assume that there exists a finite training base $\mathcal{D} = \{\mathbf{x}_i, y_i\}$, where $\mathbf{x}_i \in \mathbb{R}^n$ are independantly drawn training vectors according to a fixed but unknown probability density $p(x)$. Each training vector $\mathbf{x}$ is assigned a label y according to the law $p(x, y)$ again fixed but unknown. The training base $\mathcal{D}$ of size m constituted of m couples of observations that are independant and identically distributed (i.i.d.) according to the joint probability law $p(x, y) = p(x)p(y|x)$ can hence be formalized as in equation 4.1.

$$\mathcal{D} = (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m) \in \mathbb{R}^n \times \{\pm 1\} \tag{4.1}$$

The dimensionnality of the training vectors $\mathbf{x}$ corresponds to the dimension of the feature space that will be used to capture the object characteristics. The choice of the labels $\{\pm 1\}$ is arbitrary but refers to the binary classfication problem. The goal of the learning process is to find a function f belonging to the set of functions $\mathrm{f}(\mathbf{x}, \alpha), \alpha \in \Lambda$ where Λ represent a set of abstract parameters, that classify correctly unknow examples $(\mathbf{x}, y)$ that is $\mathrm{f}(\mathbf{x}) = y$ for $(\mathbf{x}, y)$ drawn from the same probability distribution function $p(x, y)$. As already stated, a *good* training function, *i.e.* a function that correctly classify the training set S does not necessarily implies that the function *generalizes* well, *i.e.* a function that classifies correctly test samples that were not used for training. In order to find the best function that approximates the supervisor answer, a cost function is introduced between the superviser answer y and the learning system answer $\mathrm{f}(x, \alpha)$. Hence, the functionnal risk $\mathrm{R}(\alpha)$ is defined as in equation 4.2.

$$\mathrm{R}(\alpha) = \int \mathrm{L}(y, \mathrm{f}(\mathbf{x}, \alpha)) dP(\mathbf{x}, y) \tag{4.2}$$

$P(\mathbf{x}, y)$ is the probability of $(\mathbf{x}, y)$ Hence, the problem can be formulated as finding the function $\mathrm{f}(\mathbf{x}, \alpha_0)$ that minimizes the functionnal risk $\mathrm{R}(\alpha)$. In the case of binary classification where the supervisor answer can only take two values $y = \{\pm 1\}$ or $y = \{0, 1\}$, $\{\mathrm{f}(\mathbf{x}, \alpha), \alpha \in \Lambda\}$ is the set of indicator functions, and the associated cost function is defined as equation 4.3

$$\mathrm{L}(y, \mathrm{f}(\mathbf{x}, \alpha)) = \begin{cases} 0 & \text{if } y = f(\mathbf{x}, \alpha) \\ 1 & \text{otherwise} \end{cases} \tag{4.3}$$

The function expressed in equation 4.2 determines the probability of the answers given by the supervisor and the function $\mathrm{f}(\mathbf{x}, \alpha)$, and is called the classification error. Since one only knows the training base, while the functionnal risk $\mathrm{R}(\alpha)$ is expressed as a function of the unknown probability distribution $P(\mathbf{x}, y)$, the following inductive principle is applied:

- The functionnal risk $\mathrm{R}(\alpha)$ is replaced by the empirical risk $\mathrm{R}_{emp}(\alpha)$(equation 4.4), which can be computed from the training set.

$$\mathrm{R}_{emp}(\alpha) = \frac{1}{m} \sum_{i=1}^{m} \mathrm{L}(y, f(\mathbf{x}_i, \alpha)) \tag{4.4}$$

- To estimate the function $\mathrm{f}(\mathbf{x}, \alpha_0)$ which minimizes the empirical risk (4.4), we seek to minimize the number of training errors — *minimisation of empirical risk principle* —.

In *The Nature of Statistical Learning Theory* [164], Vapnik studies the necessary conditions for a learning process based on the minimization of the empirical risk to be relevant, and establish a theory to build machine learning algorithms. As a direct application of the theory, he establishes a new statistical learning algorithm, the Support Vector Machines.

4.2.1.3 Support Vector Machines

Support Vector Machines is a statistical learning algorithm that rely on a simple geometrical approach: finding the hyperplane that best separates the positive and negative training samples. Support Vector machines have first appeared in publications related to vision such as optical character recognition [164, 22], and have soon been extended to a high number of other domains such as face detection [27, 128], text categorization [85], tracking [26, 10], etc.

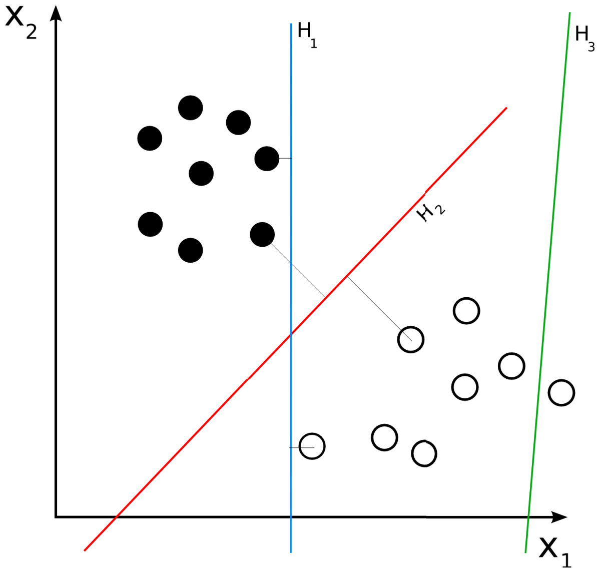


Figure 4.1: Illustration of the concept of optimal separating hyperplanes: H_3 is not a separating hyperplane (it does not separate correctly the two classes). H_1 and H_2 are separating hyperplanes, H_1 separate the classes with a small margin, H_2 separate the classes with the largest margin, hence beeing the optimal separating hyperplane.

Linearly separable data Consider the training set $\mathcal{S} = \{(\mathbf{x}_i, y_i)\} \in \mathbb{R}^n \times \{\pm 1\}, i \in \{1, \ldots, m\}$ of linearly separable elements. A separating hyperplane is an hyperplane satisfying the following condition:

$$y_i(\mathbf{w}.\mathbf{x}_i + b) \geq 1, \forall i \in \{1, \ldots, m\} \tag{4.5}$$

The distance $d(\mathbf{w}, b, \mathbf{x})$ from a point $\mathbf{x}$ and the hyperplane $(\mathbf{w}, b)$ is given by:

$$d(\mathbf{w}, b, \mathbf{x}) = \frac{|\mathbf{w}.\mathbf{x} + b|}{\|\mathbf{w}\|} \tag{4.6}$$

Definition 4.2.1. *The optimal hyperplane is the separating hyperplane located at the maximal distance from the closest training vectors* $\mathbf{x}_i$ *belonging to two different subspaces. It is said that the optimal hyperplane maximizes the margin.*

The optimal hyperplane is given by maximizing the margin $\mathrm{M}(\mathbf{w}, b)$, under the constraints of equation 4.5. This is illustrated by figure 4.1, where several hyperplanes are shown, but $H2$ is the maximum margin separating hyperplane. The margin $\mathrm{M}(\mathbf{w}, b)$ is given by:

$$\begin{aligned}
\mathrm{M}(\mathbf{w}, b) &= \min_{\mathbf{x}_i, y_i = -1} d(\mathbf{w}, b, \mathbf{x}_i) + \min_{\mathbf{x}_i, y_i = 1} d(\mathbf{w}, b, \mathbf{x}_i) \\
&= \min_{\mathbf{x}_i, y_i = -1} \frac{|\mathbf{w}.\mathbf{x}_i + b|}{\|\mathbf{w}\|} + \min_{\mathbf{x}_i, y_i = 1} \frac{|\mathbf{w}.\mathbf{x}_i + b|}{\|\mathbf{w}\|} \\
&= \frac{\min_{\mathbf{x}_i, y_i = -1} |\mathbf{w}.\mathbf{x}_i + b| + \min_{\mathbf{x}_i, y_i = 1} |\mathbf{w}.\mathbf{x}_i + b|}{\|\mathbf{w}\|} \\
&= \frac{2}{\|\mathbf{w}\|}
\end{aligned}$$

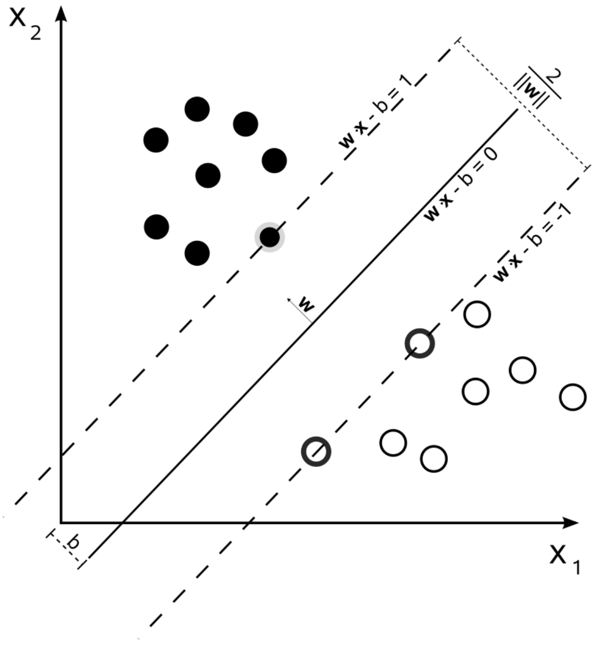


Figure 4.2: The support vectors are those samples which are located on the margin of the optimal separating hyperplane

The larger the margin is, the smaller the expected error. Maximizing the margin is then equivalent to minimise $\|\mathbf{w}\|^2$ under the constraints of equation 4.5:

$$\begin{array}{cc} \text{Minimise} & \Phi(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{with} & y_i(\mathbf{w}.\mathbf{x}_i + b) \geq 1, \forall i \in \{1,\ldots,m\} \end{array} \tag{4.7}$$

The problem of constrained optimisation can be solved using the Lagrangian formulation (equation 4.8).

$$\mathrm{L}(\mathbf{w}, b, \lambda) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \lambda_i(y_i(\mathbf{w}.\mathbf{x}_i + b) - 1) \tag{4.8}$$

The solution $(\bar{\mathbf{w}}, \bar{b})$ is located at a saddle point of the Lagrangian, which must be maximised with respect to the multipliers λ_i, and minimised with respect to $\mathbf{w}$ and b.

Definition 4.2.2. *The* support vectors *are the vectors $\mathbf{x}_i$ of the training set $\mathcal{S}$ that verify the equality $\lambda i[y_i(\bar{\mathbf{w}}.\mathbf{x}_i + \bar{b} - 1] = 0$, that is the points which lie the closest to the optimal hyperplane.*

An illustration of definition 4.2.2 is given in figure 4.2. The examples that do not lie on the hyperplane boundary are useless in the computation of the solution. Their constraints are automatically satisfied since their corresponding coefficients $\lambda_i = 0$. This implies an enhanced generalization capability of the optimal hyperplanes.

Hence, the dual of the Lagrangian can be expressed as [22]:

$$\begin{array}{cc} \text{Maximize} & W(\lambda) = \sum_{i=1}^{m} \lambda_i - \frac{1}{2}\sum_{i,j=1} \lambda_i \lambda_j y_i y_j (\mathbf{x}_i.\mathbf{x}_j) \\ \text{respecting} & \lambda_i \geq 0 \forall i = 1,\ldots,m \\ \text{and} & \sum_{i=1}^{m} \lambda_i y_i = 0 \end{array} \tag{4.9}$$

The saddle point solution is found by maximizing the quadratic optimisation problem $W(\lambda)$. The solution gives the equation of the optimal hyperplan:

$$\sum_{i=1}^{m} \bar{\lambda}_i y_i(\bar{\mathbf{w}}.\mathbf{x}_i) + \bar{b}, \text{ with } \bar{b} = -\frac{1}{2}[(\bar{\mathbf{w}}.sv_{+1}) + (\bar{\mathbf{w}}.sv_{-1})] \tag{4.10}$$

where sv_{+1} and sv_{-1} are the support vectors respectively on the positive and negative side of the hyperplane.

Non linearly separable data When the input data are not linearly separable, it is possible to reformulate equation 4.5 introducing positive slack variables ξ_i:

$$y_i(\mathbf{w}.\mathbf{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0 \forall i \in \{1, \ldots, m\} \tag{4.11}$$

The *soft-margin* SVM [39] allows misclassified examples by choosing a hyperplane that splits the training examples as cleanly as possible while still maximising the marging between cleanly separated examples. The ξ_i give a bound for the allowable error. Following the new constraints, we seek to minimize:

$$\begin{array}{cc} \text{Minimise} & \Phi(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i \\ \text{with} & y_i(\mathbf{w}.\mathbf{x}_i + b) \geq 1 - \xi_i, \forall i \in \{1, \ldots, m\} \\ \text{and} & \xi_i \geq 0 \forall i \in \{1, \ldots, m\} \end{array} \tag{4.12}$$

The solution can be found using the same process as with the linear case, by maximising the dual of the Lagrangian of function 4.9. Only the constraints changes, for that $0 \leq \lambda_i \leq C, \forall i \in \{1, \ldots, m\}$. C is a user defined parameter that allow to adjust the fitting of the decision function.

The Kernel Trick The support vector machines yield accurate results when the data are linearly separable and the soft maring formulation allows more robustness with respect to the presence of outliers. However, it is more likely that real data is not linearly separable in practice. It is hence desirable to find a non linear hypersurface that would separate the positive and negative examples. However, parameterizing such an hypersurface quickly becomes intractable.

Guyon *et al.* [71] propose to map the original description space into a higher dimensionnal feature space, also called the descriptor space. Let us note $\Phi : \mathbb{R}^n \rightarrow \mathcal{H}$ the function that achieves such a mapping, with $\mathcal{H}$ the higher dimensional space. It is then possible to find the optimal hypersurface in the augmented descriptor space, using the standard linear algorithm. Since the mapping function might not be linear, the projected hyperplane hence obtained might not be linear in the initial feature space. However, computing the dot product in the descriptor space can become quickly inefficient. However, the explicit computation can be avoided by the use of the *kernel trick*. The idea originally proposed by Aizerman *et al.* [3] is to replace every dot product by a non-linear kernel function $K(\mathbf{x}, \mathbf{y})$:

$$\Phi(\mathbf{x}).\Phi(\mathbf{y}) = K(\mathbf{x}, \mathbf{y}) \tag{4.13}$$

$K(\mathbf{x}, \mathbf{y})$ is a symetric function satisfying the Mercer conditions:

$$\int K(\mathbf{x}, \mathbf{x}')\psi(\mathbf{x})\psi(\mathbf{x}')dxdx' \geq 0 \text{ for any function } \psi \text{ such that } \int \psi^2(\mathbf{x})dx < \infty \tag{4.14}$$

In the case of non linearly separable data, the optimal separating hyperplane is obtained by replacing the dot products of the input vectors $(\mathbf{x}.\mathbf{y})$ by their equivalent in the high dimensional descriptor space $K(\mathbf{x}, \mathbf{y})$. The quadratic form to minimize becomes

$$W(\lambda) = \sum_{i=1}^{m} \lambda_i - \frac{1}{2} \sum_{i,j=1} \lambda_i \lambda_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j), 0 \leq \lambda_i \leq C \forall i \in 1, \ldots, m \tag{4.15}$$

Finally, the non linear decision function can be expressed as:

$$f(\mathbf{x}) = sign\Big[\sum_{i=1}^{N} \lambda_i y_i K(\mathbf{x}, \mathbf{x}_i) + b\Big] \tag{4.16}$$

where N is the number of support vectors, $\mathbf{x}_i$ and y_i are respectively the i^{th} support vector and it's label and λ_i is a constant learned during the training.

Several functions satify the Mercer conditions. Here are the kernel functions commonly used with SVM:

- **Polynomial:**
$$K_{P^d}(\mathbf{x}, \mathbf{y}) = [(x.y) + 1]^d, d \geq 2 \tag{4.17}$$
- **Gaussian:**
$$K_G(\mathbf{x}, \mathbf{y}) = \exp^{-\frac{\|\mathbf{x}-\mathbf{y}\|}{2\sigma^2}} \tag{4.18}$$
- **Sigmoid:**
$$K_{Th}(\mathbf{x}, \mathbf{y}) = \tanh(C\mathbf{x}.\mathbf{y} - \sigma) \tag{4.19}$$
with C a constant to ensure the Mercer conditions are satisfied.
- **Triangular:**
$$K_T(\mathbf{x}, \mathbf{y}) = -\|\mathbf{x} - \mathbf{y}\| \tag{4.20}$$

4.3 Regions-of-Interest Based Classification

4.3.1 Introduction

We will focus in this section on the detection and localisation of objects within regions-of-interest. This particular work finds it's root in the **FIAT** CSF use case that was described in section 1.1.1.1. One of the requirements in this use case is the ability to detect and classify specific car components in the images. Unless specifically shot in close captions, such components usually occupy only a small part of the image. It is then necessary not only to detect those images that contain the object of interest, but also to localize which part of the image the object lies in.

4.3.2 Generic Methodology

This work was accomplished in the **X-MEDIA** project as part of the broad project framework, which collect collect evidences accross different media: images, text and raw data. Within the **X-MEDIA** framework, the so-called *knowledge extraction tools* use a common infrastructure to enable the sharing and reuse of knowledge among different components. The user **ontology** defines the view of the industrial process in terms of object entities and the relationships between them. Other partners of the project were directly involved in the creation and use of the ontology for complex knowledge management. We limit our view of the ontology as a list of the object classes that can be recognized and localized in images. Hence, our problem is:

1. To propose algorithms for the candidate localisation of objects of interest (ROI detection).
2. To propose efficient models for the recognition of objects inside the ROI (supervised learning).

The generic methodology proposed for image analysis in **X-MEDIA** is depicted in figure 4.3 [11]. This methodology defines the different steps of image analysis in terms of processes and potential applications. The main pipeline for image analysis is illustrated by the arrows going from the input image to the knowledge base. Every blue box represents a processing module. The straightforward processing pipeline is composed of the following modules:

- **Preprocessing:** this step could include any kind (and combination) of pre-processing algorithm such as filtering, denoising, etc. Image segmentation and ROI detection are also considered as pre-processing steps, which enables local image analysis.
- **Feature Extraction:** this step corresponds to the computation of image features (*i.e.* visual descriptors such as the one described in chapter 2). It follows preprocessing since features could be extracted on image regions, on the global image, on filtered images, *etc*.
- **Classification:** this step corresponds to the computation of models for the recognition of specific object classes. The models are computed using the features extracted in the previous step.
- **Concept detection:** In this step, the models built from the previous steps are used to extract knowledge from unknown images. As illustrated in figure 4.3, this step is linked with a double arrow to the knowledge base: the concept detection module populate the knowledge base with instances of object classes found in the images, while the knowlege base defines the particular object classes that have to be recognized (the one defined in the user ontology).

Two additionnal processing modules are depicted as side of the main pipeline:

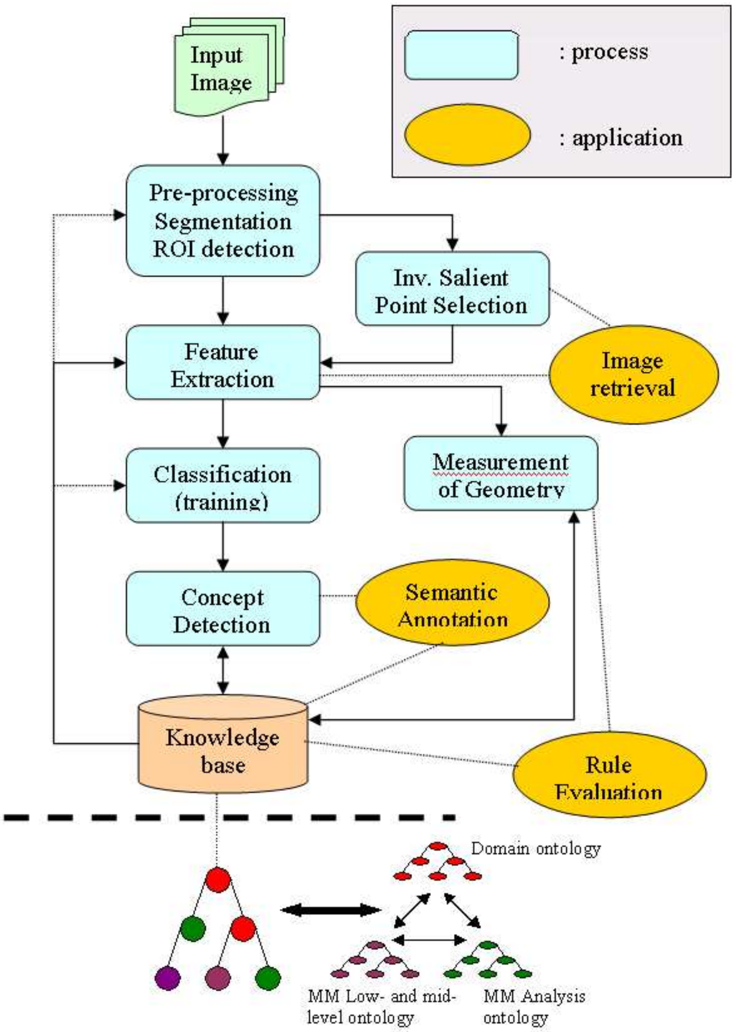


Figure 4.3: Diagram of the XMedia Generic Methodology for Image Analysis

- **Invariant Salient Point Detection:** this step corresponds to the localization of salient point in the images, used for the computation of local interest point descriptors such as SIFT and SURF.

- **Measurement of geometry:** this steps correspond to the computation of particular features linked to the geometry of the segmented regions.

In a generic scenario, there is a feedback loop between concept detection and the previous tasks with a view to refining initial decisions or, complementarily, to repeating the same process at finer granularity. For instance, after detecting that an image illustrates the interior of a car, the system may proceed in detecting more detailed objects such as steering wheel, pedals, seats, etc. Orthogonally, in case of low confidence in the initial labelling, additional features may be taken into account, or different classifiers may be applied, or the segmentation may be revised.

In figure 4.3, some applications relying on the output of the processing modules are shown as orange-filled circles. This includes the applications we have described in chapter 3 and the one we will decribe in chapter 5. **Semantic annotation** is another application, where the semantics of images is extracted using the concept detection module. Semantic queries such as *give me all pictures representing dashboards* can be performed. Since semantics can be extracted from compound documents using other technologies (such as knowledge extraction from text), queries that go beyond the semantics extracted from images can be performed. **Rule evaluation** is another possible application where the geometry of particular components can be used to apply rules and infer more knowledge. However this was out of the scope of this PhD.

In this section, we will focuse on the instantiation of the framework for the task of air duct detection in car images. Air duct detection is achieved using a straightforward instantiation of the framework, where the different process to apply are sequantially: ROI detection, feature extraction, model training and SVM classification. The contributions in this section span several of the process of the methodology.

- An original ROI extraction algorithm has been proposed, where detection of lines is achieved using the Hough Transform and extraction of candidate regions in an unsupervised manner is carried using DBScan clustering algorithm.
- We proposed to rely on the MPEG-7 Edge Histogram descriptor as the underlying feature space for air-duct classification and have experimentally validated this choice.
- Concerning the classification algorithm, we have used SVM classifier and proposed to enhance the classification results relying on the boosting paradigm.
- Finally, we provide an evaluation of our framework. In this evaluation, we test the efficiency of the system as a whole but also the influence of the ROI extraction method on the overall efficiency.

4.3.3 ROI extraction

The goal of the ROI extraction is to provide a set of candidate regions that are likely to contain the object of interest. As such, the frontier between the ROI extraction and the classification algorithm becomes shallow, since localizing ROI could be considered as *recognizing* instances of the object. On the countrary, regions of the image not detected by the ROI extraction algorithm will never been considered for further processing, hence resulting in a missed detection. The ROI detection algorithm has to favor a high recall, while the classification algorithm will be in charge of discarding non relevant regions hence improving the precision of the whole chain. In the context of airduct detection, we propose to rely on a specific ROI detection algorithm based on the visual appearance of this component. Generic ROI detection algorithms exists such as the work of Kapsalas *et al.* [88]

Air duct in cars can have all kind of shapes and sizes. The most common characteristics between different air ducts is that they are covered by a grid. We propose to localize the candidate regions as the one which have a *high density* of parallel lines. This is achieved in three steps: first, image contours are extracted using Canny-Deriche edge detection algorithm [44]. Then, we perform the detection of lines in the image using the Hough Transform algorithm [47]. Finally, we extract regions of high density of parallel lines using the DB-SCan clustering algorithm [51]. In the next sections, we describe in more details the Hough transform. DBScan clustering as already been presented in section 2.4. We will then give the detailed procedure of our ROI detection algorithm.

4.3.3.1 Line Detection using Hough Transform

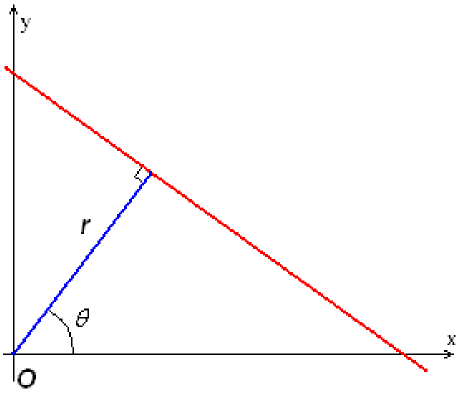


Figure 4.4: Parametrization of straight lines using polar coordinates

The Hough transform was originally formulated by Paul Hough [76, 77]. Several publications on the topic have been proposed and even compared [80].

A binary image can be considered as a set of *characteristic points*. Taking into acount only the information about the point localizations, one lacks information about their geometric organisation, possibly into meaningful shapes. The Hough transform enables the task of finding geometrical primitives (such as lines and curves) into a problem of finding a maxima in a parameterized space. A generalized version of the Hough Transform was formulated by Duda and Hart for line detection [47].

Consider the equation of a line as expressed by it's polar coordinates $\rho = x \cos\theta + y \sin(\theta)$, where ρ represents the distance between the line and the origin and θ is the angle from the origin to the closest point (figure 4.4). This parameter space is better suited than the traditionnal $y = ax + b$ since the parameters a, b can be unbounded while ρ, θ are finite parameters. In this parameter space ρ, θ, a point $p(x, y)$ is regarded as a constraint of the straight lines on which the point may be located. In the parameter space, we can draw a point for every couple ρ, θ that yields a line which go through p. The set of points hence obtained draws a sinusoid in the ρ, θ space, as illustrated in figure 4.5. Every point yields a vote for several possible straight lines. When the points are aligned in the image space, the sinusoids corresponding to each point in the parameter space cross at a common point $\bar{\rho}, \bar{\theta}$, which correspond to the parameters of the straight line containing all the points. Hence, the detec-

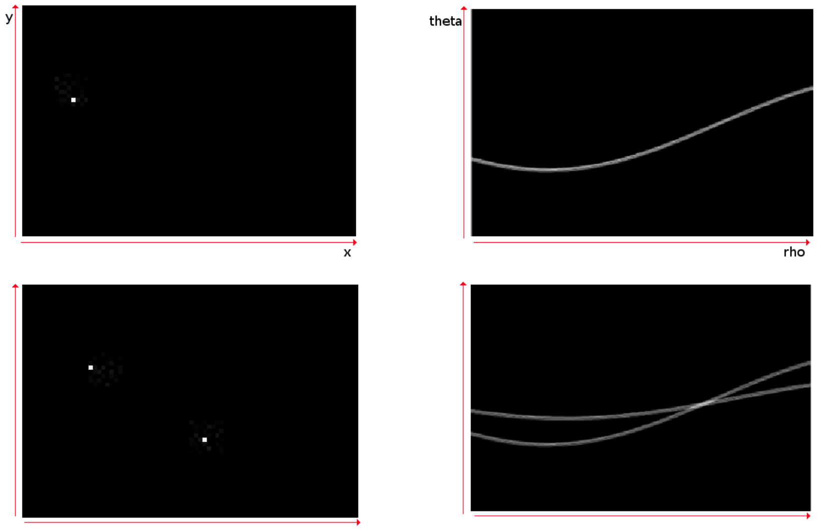


Figure 4.5: Correspondance between points in the image and sinusoids in the accumulator space. Each point potentially holds a number of straight lines going through it, with varying ρ, θ. The line going through both points is caracterized by the equation found at the location of the crossing sinusoids in the parameter space.

tion of straight lines using the Hough transform is achieved in two stages: first, the candidate straight lines are collected into the parameter space, also called the accumulator space. Then, the accumulator space is examined in order to find local maxima, corresponding to straight lines in the image.

4.3.3.2 ROI extraction algorithm in details

The main steps of the ROI extraction algorithms have been given in paragraph 4.3.3. In this section, we introduce the detailed step of the whole ROI extraction algorithm. The pseudo-code for the algorithm is given in algorithm 2. To illustrate the algorithm, we give in figure 4.6 the detailed output of each step of the algorithm applied on the image given in figure 4.6(a).

- In line 2, the procedure CANNY extracts the edges of the image. This is illustrated in figure 4.6(b) where the edges are outlined in white.
- In line 3, line segments are computed in the Hough parameter space. The extracted segments are drawn in red in figure 4.6(b).
- In line 4, we start a processing loop for all segment orientations. We uniformally quantize the orientations $[0, \pi]$ into 12 different bins. We process all the segments that fall in the same quantization level (condition line 6).
- In line 7, we call the subroutine SAMPLE_POINTS_FROM_SEGMENT to initialize a list of points. The subroutine compute points regularly sampled on the segment at a

Algorithm 2 Algorithm for ROI Extraction

```
 1: ROI_EXTRACTION(Image img, double $\epsilon$, int $T$)
 2: Edge_img = CANNY(img)
 3: Segments = HOUGH(Edge_img);
 4: for all Orientation $O \in [0, \pi]$ do
 5:   for all Segment S ∈ Segments do
 6:     if ORIENTATION(S) == O then
 7:       Points ← SAMPLE_POINTS_FROM_SEGMENT(S, $\epsilon$);
 8:     end if
 9:   end for
10:   Clusters = DBSCAN(Points, $\epsilon$, $T$)
11:   for all Clusters C do
12:     ROIS ← COMPUTE_BOUNDING_BOX(C)
13:   end for
14: end for
```

distance slightly lower than the threshold ϵ. This is illustrated in figure 4.6(c), where the green points are the points sampled on the segment lines extracted previously. Hence, we initialize a dataset of 2D points to be clustered at the sampled points extracted from (almost)parallel segment lines.

- In line 8, we run the DBScan algorithm to automatically discover the zones of high density of points.

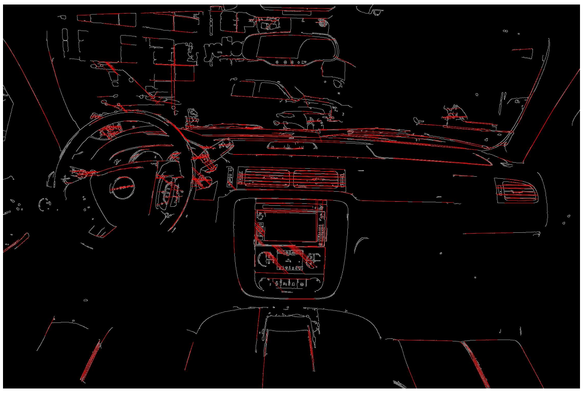
- In lines 11/12, we compute the bounding box of each cluster discovered by DBScan, that is the rectangle which contains the whole set of points beeing member of a same cluster according to DBScan. We illustrate this step in figures 4.6(d) and 4.6(e) for different orientations. The points belonging to a common cluster are drawn with the same color, and the bounding boxes aroung each cluster are depicted as red rectangles. These bounding rectangles will form the candidate Regions-Of-Interest that will be submitted to the SVM classifiers.

4.3.4 Classification

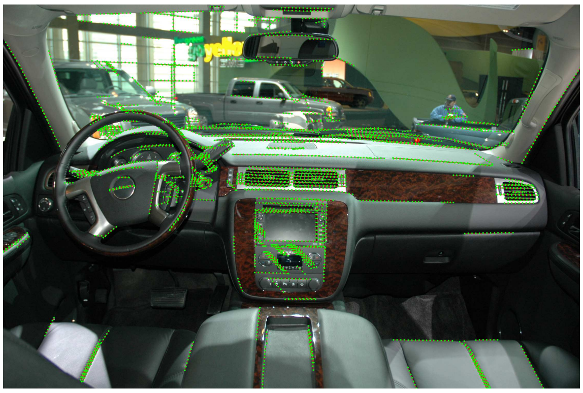
4.3.4.1 Feature Extraction

The choice of the descriptor space is based on the knowledge of the appearance of the object of interest. In the case of air duct, we should use a descriptor able to capture the characteristics of a grid, while beeing quite robust to the changes in shapes. We have used the MPEG-7 EHD for it's inherent capability of capturing the orientation of edges, foreseen as an interesting discriminant feature. Then, following the processing pipeline, the EHD is computed for every ROI given by the ROI extraction algorithm. The discrimination between positive (*e.g.* the ROI actually containing air ducts) and negative examples will be achieved by the SVM. As a preliminary study of the EHD, we computed a training and test set of ROI. The positive

(a) Original Image



(b) Edge Detection and Hough Transform



(c) Sampling points on detected lines

```
ERROR: ioerror
OFFENDING COMMAND: image

STACK:

(
PPOOPPPPPQRRQPPPPPQRSSSTRRSSSSRRTVVTRTXZXWVVVWXYWZ[ZYZ[[]]]]^]\[_^_a___‘a‘aaabaaa‘_‘abb
)
-filestream-
-mark-
-savelevel-
```