Dipartimento di Informatica e
Scienze dell'Informazione

Local image descriptors for matching and classification

by

Elisabetta Delponte

Theses Series **DISI-TH-2007-01**

DISI, Università di Genova
v. Dodecaneso 35, 16146 Genova, Italy http://www.disi.unige.it/

Università degli Studi di Genova

Dipartimento di Informatica e
Scienze dell'Informazione

Dottorato di Ricerca in Informatica

Ph.D. Thesis in Computer Science

Local image descriptors for matching and classification

by

Elisabetta Delponte

April, 2007

Dottorato in Scienze e Tecnologie dell'Informazione e della Comunicazione
Indirizzo Informatica
Dipartimento di Informatica e Scienze dell'Informazione
Università degli Studi di Genova

DISI, Univ. di Genova
via Dodecaneso 35
I-16146 Genova, Italy
http://www.disi.unige.it/

Ph.D. Thesis in Computer Science (S.S.D. INF/01)

Submitted by Elisabetta Delponte
DISI, Univ. di Genova
delponte@disi.unige.it

Date of submission: April 2007

Title: Local image descriptors
for matching and classification

Advisor: Alessandro Verri
Dipartimento di Informatica e Scienze dell'Informazione
Università di Genova
verri@disi.unige.it

Ext. Reviewers: Andrea Fusiello
Università di Verona
andrea.fusiello@univr.it

Carlo Colombo
Università di Firenze
colombo@dsi.unifi.it

Abstract

This thesis considers view-based object recognition in images, a problem that is still lacking an effective solution despite decades of research in computer vision. It is worth remembering that recognising an object from its appearance is considered as a keystone for image understanding, one of the most challenging problems in computer vision.

The study and the analysis of the visual information coming from an image can be tackled with different approaches: to global image description we preferred the local approach since recent research has demonstrated that it leads to a more compact and robust representation of the image even when there are major changes in the object appearance. Thus in our work we concentrated on the use of local features and interest points to determine a representation of the image content by means of its most informative elements.

We model 3D objects using a visual vocabulary whose words represent the most meaningful component of the object: the description obtained is complete and compact and is capable to describe the object when it is seen from different points of view. The robustness of this approach is remarkable also when the object is in a very cluttered scene and it is partially occluded. In respect to local approaches to object recognition, we focused on the following problems:

- detection and description of local image features
- estimation of the similarities between feature descriptors and matching points between images
- formalisation of the concept of visual vocabulary and setting up of a classifier capable to compare several models of different objects.

Within this framework, the contributions of this thesis are in the following areas:

Matching techniques. We propose a matching strategy based on the use of a spectral method in association with local descriptors: since the representation we use is robust to scale and illumination changes, we obtain a compact and simple algorithm that can be used for severe scene variations.

Object recognition. We present a method for 3D object modelling and recognition which is robust to scale and illumination changes, and to viewpoint variations. The object vocabulary is derived from a training image sequence of the object and the recognition phase is based on a SVM classifier.

We introduced another adaptive strategy for object recognition in image sequences which is strongly based on the use of spatio-temporal coherence of local descriptors. In this case our work is motivated by the fact that an image sequence does not just carry multiple instances of the same scene, but also information on how the appearance of objects evolves when the observation point changes smoothly.

a tutte le persone che mi hanno insegnato qualcosa

La regola del signor Palomar a poco a poco era andata cambiando: adesso gli ci voleva una gran varietà di modelli, magari trasformabili l'uno nell'altro secondo un processo combinatorio, per trovare quello che calzasse meglio su una realtà che a sua volta era sempre fatta di tante realtà diverse, nel tempo e nello spazio.

(I. Calvino)

Acknowledgements

Il mio portatile si chiama `whimper`. No. Non Whymper come il famoso alpinista inglese, primo ad aver scalato la splendida cima delle Grandes Jorasses che gli è stata per questo dedicata. Non Whymper, bensì `whimper`. Credo che si possa immaginare facilmente il mio disappunto, quando, accorgendomi di aver scambiato la `i` con la `y` ho scoperto che whimper, in inglese, significa lamento. Per cui, il mio portatile si chiama mugugno.
Non pochi avranno pensato che il nome mi calzasse a pennello: spesso l'ho pensato anche io. Quindi, ora, per smentire le voci secondo cui sono sempre lamentosa, griderò al mondo il mio

esultante e barbarico grazie.

Grazie Francesca! Grazie per avermi aiutata, spronata e sgridata. Grazie per tutto il lavoro insieme, le risate e le pazienti correzioni. Insomma: grazie! Un grazie con tutto il cuore ad Alessandro, per avermi dato la possibilità di fare il dottorato, per aver sopportato il mio peregrinare a Trieste e per avermi insegnato così tanto.

Quando penso a tutte le persone fantastiche che ho incontrato in questi tre anni di dottorato mi rendo conto di essere una privilegiata: grazie per aver condiviso con me i momenti bellissimi, i momenti tristissimi e quelli medi. Insomma grazie per essere stati lì: Augu, Anna, Ema, Fro, Lorenzo, Laura, Sofia, Nicola, Elise, Curzio, Matteo, Gianlu e tutti gli altri dello SlipGURU e della 210. Un grazie specialissimo a Nicoletta, a *Dewey* e a *Goofy* (See Figures 4.1 and 4.1): senza di voi non sarei mai arrivata a scrivere i ringraziamenti di questa tesi!!!
Non posso scordarmi dei miei amati amici di Bertinoro: Paola, Roberto, Matteo, Valentina e Giuseppe! E ancora un grazie di cuore alle *women of valor* per aver sopportato la mia bifasicità! Grazie, perché noi valiamo! Grazie al Cicci per avermi spiegato la proprietà caratteristica dell'estremo superiore, e non solo quella...

Il mio ringraziamento più grande va alla mia mamma e al mio papà, a Cris, Marco, Marti e tutti i nipotastri. E a Matteo: grazie perché ci sei.

Table of Contents

Introduction

Scope and motivations of the work

Finding correspondences between images and recognising objects from their visual appearance are two important competencies of the human visual system and in the course of this thesis we will describe how they have been faced from the viewpoint of a *machine*. Both of these issues may be addressed globally – that is, using the information contained in images considered as a whole – or locally – that is, first computing the most reliable or more informative image features, and then applying further computations only to these points.

Global approaches are usually based on colour or texture distributions [SB91, OBV05, PP00] but unfortunately, they are not robust against occlusions, background clutter, and other variations of the content, which are due to arbitrary changes of the imaging conditions. Thus global features are simple representations but they hardly cope with background/foreground segmentation or varying image attributes.

Local features are meaningful and detectable parts of an image [TV98]. Usually, points where the image content locally changes in two directions are called interest points [SMB00, Mik02]. These points convey more information due to signal changes, therefore they are considered more representative for the image. Thus the idea behind local approaches is to find a more compact representation of the image content, by localising the main areas carrying the most useful information, while discarding poor or noisy data.

First approaches to 3D object recognition have been tackled from the geometric point of view: the information used to characterise an object is organised in the form of a 3D model focused on geometric measurements [BJ85, Gri90, Hut89, Pol00]. This approach is usually called model-based recognition but it is not the approach that we are interested in this thesis. In fact our aim is to discuss about view-based object recognition since there are several reason motivating the introduction of visual-based approaches to object recognition, among which we remember biological inspiration and computational efficiency [ECT99, CWT00].

The use of local descriptors is of particular interest for the study of view-based object recognition: in fact some recent research in neuroscience have shown that, for recognising objects, primates use simple features which are invariant to changes in scale, location and illumination [Tan97]. Edelman, Intrator and Poggio in [EIP] makes use of a model of biological vision for object recognition. Their approach is based on complex neurons in primary visual cortex which respond to a gradient at a particular orientation but allows for shift over a small receptive field rather than being precisely localise. The authors hypothesised that the function of these complex neurons was to allow for matching and recognition of 3D objects from a range of viewpoints.

In the last decades many local features have been proposed in the computer vision literature [Mor77, HS88, Can86, ST94, Lin93, MCUP02, LZ03, MS04b] and appropriate descriptions for them have been discussed [SB91, KvD87, SM97, BMP02, Low04, MS04a]. The take-home message of all these studies is that a good descriptor should be invariant or at least tolerant to small variations of illumination and pose, and it should tackle scale variations. In this thesis we focus on those descriptors which have demonstrated high performances with respect to stability and robustness such as for instance SIFT, firstly devised by Lowe in [Low99]. We take particular care in analysing the relationship between features and their own scales, since it has been demonstrated that dealing with image structures at different scales is of utmost importance when one aims at understanding image content.

Our work can be divided in two complementary parts: the first involves local approaches to matching, while the second deals with view-based object recognition. For what concerns *matching* we present a method for determining the correspondences between sparse feature points in images of the same scene based on the SVD-matching paradigm and on SIFT descriptors [DIOV06]. We show that applying SVD-matching to SIFT descriptors instead than to image patches improves its especially in the presence of scale changes, severe zoom and image plane rotations, and large view-point variations.

Our approach to view-based *3D object recognition* aims at integrating spatial information and temporal coherence of local image descriptors extracted from a video sequence. It is worth noticing that in the course of the thesis we face object recognition but we do not explore the connected problem of object categorisation. If we think about a robot grasping an object and observing it from different points of view it is natural to assume that there are local descriptors which can be spatially and temporally combined. Moreover we can also build a description which is evolving incrementally as the time goes by. Thus, our work is motivated by the fact that an image sequence does not just carry multiple instances of the same scene, but also information on how the appearance of objects evolves when the observation point changes smoothly. Since in many applications image sequences are available and often under exploited, our aim is to fill this gap. To do this, we start from an image sequence of the object and find a compressed description of it that starts with the extraction of local keypoints. Then we track the keypoints over the sequence and combine

this information to obtain time-invariant features that will represent all the information we retain about the object.

This is the visual model at the basis of two different approaches that we design for recognising three dimensional objects from their visual appearance. The first method falls in the class of learning from examples approaches and it is based on Support Vector Machines for classification: training and test images are both compared with respect to the visual vocabulary and the similarities are computed by an *ad-hoc kernel* [OBV05, ADOV06]. For the second approach we exploit a novel technique for matching models and test sequences [DNOV07]. The main constraints used in this case are determined by temporal cooccurences of visual words thus we emphasise matching between groups of spatially close and coeval features.

Outline of the Thesis

The thesis is structured as follows:

Chapter 1 gives an overview of local methods for describing images. After a brief introduction to the problems related with visual perception we focus our attention on the detection, the location and the representation of special parts of images that are defined as *image features*. These elements correspond to the most interesting areas of images. This chapter also includes a brief introduction to the concept of *scale space* and to the main aspects related to features detection and description in the scale space framework.

Chapter 2 introduces one of the central problems in theory of vision which is that of establishing correspondences between points of two related images. Since matching points is a very common framework with a great variety of applications, in the first part of the chapter, we briefly review some of the most commonly used techniques for matching. We mainly focus our attention on spectral methods for features matching. Then we propose a novel version of the SVD-matching proposed by Scott and Longuet-Higgins [SLH91] and later modified by Pilu [Pil97], that exploits the robustness of SIFT descriptions.

Chapter 3 deals with the problem of object recognition, one of the most difficult and outstanding problems in computer vision that has been studied since the birth of this discipline. We focus on view-based object recognition methods and, in particular, on those methods based on local descriptions since it has been shown that they are able to cope with object occlusions, changes in background and foreground and they are also biologically inspired. Thus, after a brief introduction to the best known methods for 3D object recognition we describe some novel ideas on the use of local descriptors in sequences of images: we obtain view-based models of objects exploiting spatial and temporal coherence of image sequences.

Chapter 4 contains an extensive analysis of the view-based model presented in Chapter 3. The experiments are devised to investigate some of the typical object recognition problems. Thus, this chapter shows the results obtained using different methods and approaches.

In the **Conclusions** we discuss some qualities of the proposed approaches, and lay down the path for future work in the attempt to overcome the limitations of our approach.

Chapter 1

Describing images

This chapter is devoted to a review of the basic methods for the description of images. First of all we will consider the detection, the location and the representation of special parts of the image, called image features, *usually corresponding to interesting elements of the scene. This chapter describes also the theory of the* scale space *which is a framework for multi-scale signal representation of signals. We will Analise how the problem of detection casts in the scale space framework.*

1.1 Introduction

Visual perception is the end product of vision, consisting of the ability to detect light and interpret the consequences of light stimuli. The major problem in visual perception is that what people see is not simply a translation of retinal stimuli (i.e. the image on the retina). Thus, people interested in perception have long struggled to explain what visual processing does to create what we actually see. Visual perception is one of the oldest fields within scientific psychology, and there are correspondingly many theories about its underlying processes.

Hermann von Helmholtz is often credited with the founding of the scientific study of visual perception. Helmholtz held vision to be a form of unconscious inference, in other words a matter of deriving a probable interpretation from incomplete data. Thus we can say that inference requires prior assumptions about the world.

Among the oldest theories is the one developed by the **Gestalt** psychologists from the 1930s to the 1940s (notice that *gestalt* is the German word for *shape*). The *Gestalt Laws of Organisation* have guided the study of how people perceive visual components as organised patterns or wholes, instead of many separate parts. One of the basic principles stated in

Gestalt theory of vision is that the whole is different from the sum of its parts. According to this theory, there are main factors that determine how we group elements according to visual perception:

- Symmetry: symmetrical items are more likely to group together.
- Similarity: objects similar in size or shape are more likely to form a group.
- Proximity: closer objects are more likely to form a group.
- Closure: interpretations which produce *closed* rather than *open* figures are favoured.
- Continuity or smoothness: contours based on smooth continuity are preferred to abrupt changes of direction.
- Smallness: smaller areas tend to be seen as figures against a larger background.
- Surroundedness: areas which can be seen as surrounded by others tend to be perceived as figures.

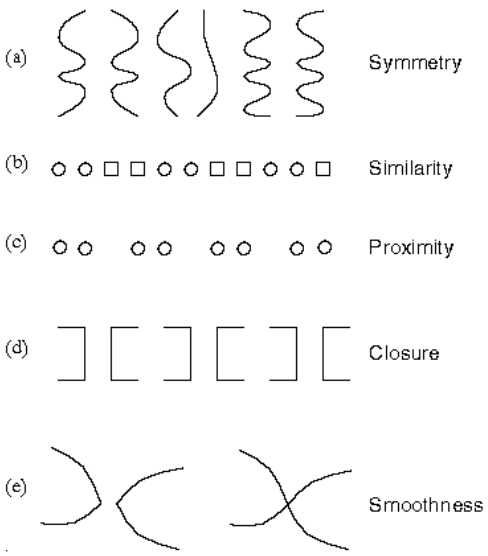


Figure 1.1: The "Laws of perceptual organisation": some examples describing five of the principles of Gestalt [Tuc].

The first five principles are represented in Figure 1.1, while the last two are illustrated in Figure 1.2 and Figure 1.3. All of these principles of perceptual organisation serve the

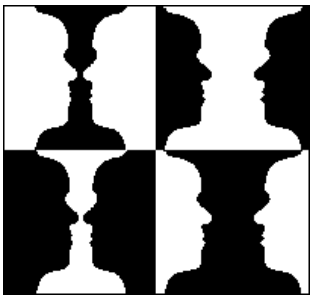


Figure 1.2: This ambiguous illustration, devised by the Danish psychologist Edgar Rubin, describes the principle of *smallness*: smaller areas tend to be seen as figures against a larger background [Cha01].

Figure 1.3: This figure shows the *surroundedness* principle. This image is confusing since observers assume that the white area is the background rather than the figure. We should now be able to read the word "TIE" [Cha01].

overarching principle of *pragnänz*, which is that the simplest and most stable interpretations are favoured.

Nevertheless there is also the hypothesis that human vision works in a **hierarchical way** in the sense that we first extract low level features such as local orientation and colour, and then look for higher level features [BCP96]. No one knows for sure what these high level features are, but there are some some indications that curvature, circle patterns, spiral patterns and star patterns are among them [GBE93]. Indeed, perceptual experiments indicate that corners and curvature are very important features in the process of recognition and one can often recognise an object form just a few local curvature fragments [Att54, Bie87].

For instance Biederman in [Bie87] defines a theory for object recognition which is based on the use of *geons*, which are simple 3-dimensional shapes such as spheres, cubes, cylinders, cones or wedges. Figure 1.4 shows a set of these volumetric primitives. Biederman suggests that visual input is matched against structural representations of objects in the brain, and these structural representations consist of geons and their interrelations. Geons can be used to represent a large number of possible objects with very few components; for instance 24 geons can be recombined to create over 10 million different two-geon objects.

Biederman's ideas inspired the fundamental observation that an observer, when looking at a complex scene, concentrates and focuses on certain points of the image more than others. For instance when we are looking at a person we tend to pay more attention to the face than the rest of the body and within the face we concentrate on the eyes and the mouth more than the cheeks or forehead [LZ03]. People process visual information selectively, because some points contain more interesting information than others. In computer vision

Geon	Edge Straight S Curved C	Symmetry Rot & Ref ++ Ref + Asymm –	Size Constant ++ Expanded – Exp & Cont – –	Axis Straight + Curved –
	S	+ +	+ +	+
	C	+ +	+ +	+
	S	+	–	+
	S	+ +	+	–
	C	+ +	–	+
	S	+	+	+

Figure 1.4: Some examples of Biederman's geons classified on the base of curviness, symmetry, size and curviness of axis [Bie87].

these are called *points of interest*.

The study of *points of interest* is of utmost importance for research in computer vision, since many of the algorithms are based on an initialisation step for extracting the most detectable and in some sense stable information for images: matching, mosaicing, object recognition, pose estimation, and tracking are only some of the problems that can be faced using a set of interesting points in the image. Moreover, the approach which considers only the most meaningful parts of an image has positive effects from engineering and computational points of view. The use of *points of interest* allows us to manage compact image representations, and this is an advantage in several cases. Usually for *points of interest* we mean every point in the image in which the signal changes at least two-dimensionally [SMB00, MC02, LZ03, Joh04], which are the only points of an image that are not affected by the typical problem of *aperture* [MU81].

It is worth noticing that an interesting attribute of an image feature is its relationship with the other part of the image, that we can consider as its scale with respect to the whole image. Thus, before we give other details on feature detection and description, in the next section we review the concept of scale space: indeed this framework is at the basis of many algorithms for detection of interest points and image features at different levels of detail.

1.2 Scale space

An inherent property of real-world objects is that they only exist as meaningful entities over certain ranges of scale. This fact, that objects in the world appear in different ways depending on the scale of observation, has important implications if one aims at describing or recognising them.

The problem of scale is well-known in physics, where phenomena are modelled at several levels of scale ranging from particle physics and quantum mechanics at fine scales, through thermodynamics and solid mechanics dealing with every-day phenomena, to astronomy and relativity theory at scales much larger than those we are usually dealing with. Notably, the type of physical description that is obtained may be strongly dependent on the scale at which the world is modelled, and this is in clear contrast to certain idealised mathematical entities, such as "point" or "line", which are independent of the scale of observation [Lin98a].

Also, when we are analysing digital images the notion of scale is of utmost importance. For this reason the computer vision, image processing and signal processing communities have developed the scale space theory since the early days of these disciplines [TR71, Wit83, Koe84, YP86]. Scale space is a formal theory for handling image structures at different scales in such a way that fine-scale features can be successively suppressed and

a scale parameter t can be associated with each level in the scale space representation [Lin96]. Figure 1.5 on the left shows a one dimensional signal which has been successively convolved with a 1D Gaussian filter. Increasing the parameter t the high resolution details are suppressed to give more evidence to the characteristic trend of lower scales. The next sections are devoted to a review of the basic notions in scale space theory.

1.2.1 Scale space representation: definition and basic ideas

One of the motivations at the basis of scale space construction is that, if no prior information is available about what are the appropriate scales for a given data set, then the only reasonable approach for an uncommitted vision system is to represent the input data at multiple scales. The essential idea of this approach is to embed the original signal in

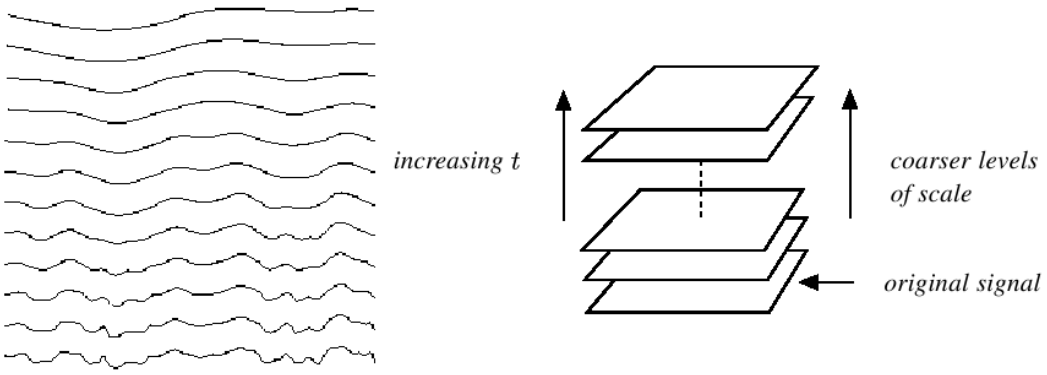


Figure 1.5: (On the left) A one dimensional signal has been successively convoluted with Gaussian kernels of increasing width. (On the right) A multi-scale representation of a signal is an ordered set of derived signals intended to represent the original signal at different levels of scales [Lin96].

a family of derived signals in which fine scale structures are successively suppressed. In Figure 1.5 on the right it is possible to see a representation of the family of images obtained by the convolution of the original image with different 2D Gaussian masks. Notice that the original signal is at the bottom of the pyramid.

Let us consider a continuous signal $f : \mathbb{R}^D \to \mathbb{R}$, then the linear scale space representation $L : \mathbb{R}^D \times \mathbb{R}_+ \to \mathbb{R}$ of f is given by the convolution of the original signal with Gaussian kernels of various width t

$$L(\cdot\,; t) = g(\cdot\,; t) \;*\; f(\cdot)\,, \tag{1.1}$$

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

$$g(x;t) = \frac{1}{(2\pi t)^{N/2}}\, e^{-(x_1^2+\ldots+x_D^2)/(2t)}\,, \tag{1.2}$$

and $x = (x_1, \ldots, x_D)^T$. An equivalent way to derive this family of signals [Koe84] is to solve the heat conduction, or diffusion, equation

$$\partial_t L = \frac{1}{2}\nabla^2 L = \frac{1}{2}\sum_{i=1}^{D} \partial_{x_i x_i} L \tag{1.3}$$

with initial condition $L(\cdot\,;t) = f(\cdot)$.

There are several mathematical results [Koe84, YP86, Flo93, tHR94] stating that within the class of linear transformations the Gaussian kernel is the unique kernel for generating a scale space. The conditions that determine the uniqueness are essentially linearity and shift invariance combined with different ways of formalising the notion that new structures should not be created in the transformation from a finer to a coarser scale.

Figure 1.6 shows the images belonging to the scale space obtained by repetitively convolving the "cameraman" image with a Gaussian kernel with variance 1.5. After the first convolution, we repeat the convolution of the output with another Gaussian with the same variance. It is apparent that the minute features, visible at finer levels, disappear as the scale increases. Even though a complete description of this matter is not in the scope of this thesis, among the extensions of the theory of scale space, it is worth remembering the formulation of a non-linear scale space theory which is based on the use of specific ad-hoc kernels [Wei98].

In the next section we describe some important properties required for the creation of a linear scale space representation.

1.2.2 Axiomatic scale space formulation

Why one should make use of Gaussian smoothing, and why not just carry out any type of smoothing kernel? To describe the special properties that have lead computer vision researchers to consider linear scale space representation as a natural model for an uncommitted visual front-end, in this section we shall give a very brief review of some of the most important axioms for scale space formulations. More extensive reviews can be found in [Wit83, tHR94, Lin94].

The first property of a smoothing operation is that it should preserve *causality*: any feature at a coarser level of resolution is required to possess a (not necessarily unique) "cause" at a finer level of resolution although the reverse need not be true. In other words, no spurious

Figure 1.6: The scale space built on the image of the "cameraman".

detail should be generated when the resolution is decreased. The first researcher who introduced the concept of *causality* in scale space was Koenderink in [Koe84], where he also demonstrated that the one parameter family of derived images (or signals) may equivalently be viewed as the solution of the heat conduction, or diffusion, equation. Koenderink formalised the idea of *causality* saying that new level surfaces

$$\{(x, y, t) \in \mathbb{R}^2 \times \mathbb{R} : L(x, y, t) = L_0\} \tag{1.4}$$

must not be created in the scale space representation when the scale parameter is increased.

The second criterion that guides the choice of the filtering kernel is based on the notions of *homogeneity* and *isotropy*, which means that the blurring is required to be space invariant [PM90]. By combining causality with the notions of *isotropy* and *homogeneity*, which essentially mean that all spatial positions and all scale levels must be treated in a similar manner, Koenderink showed that the scale space representation must satisfy the diffusion equation.

It is interesting to notice that *isotropy* is not a fundamental axiom to the scale space, indeed in [PM90] it has been proven that modifying this request it is possible to achieve a better detection of edges. Another remarkable fact about the construction of the scale space using Gaussian derivatives is that there are interesting relations between this formulation and the biological vision. Indeed, there are neurophysiological studies [You87, EIP97] showing that there are receptive field profiles in the mammalian retina and visual cortex which can be well modelled by superpositions of Gaussian filters.

The scale space framework is often useful to detect image features at different scales. The next section reviews some of the better known feature detectors while, later in this chapter, we will describe points of interest detection algorithms with automatic scale selection.

1.3 Image features

In computer vision and image processing the concept of feature is very general: we could say that there is no universal or exact definition of what constitutes a feature. However, usually with the term feature we denote a piece of information which is relevant for solving the computational task related to a certain application, in other words we can say that it refers to an interesting part of the image. Generally features can refer to

- the result of a general neighbourhood operation (feature extractor or feature detector) applied to the image
- specific structures in the image itself, ranging from simple structures such as points or edges to more complex structures such as objects.

A possible definition and classification of the term feature can be found in [TV98]. Here the world *feature* refers to a representation of special parts of an image, usually corresponding to some interesting elements of the scene. It can refer to two possible entities:

Global feature is a global property of an image, as the average gray-level intensity or a histogram of direction of the gradient of the image.

Local feature is local part of the image with special properties as a circle, a line or a change in illumination.

In most of the cases we can assume that a feature is a local, meaningful and detectable part of an image [TV98]. By meaningful we mean that the feature has to be associated to an interesting element of the scene as for instance to the contour of an object. When we say that a feature has to be detectable we mean that some localisation algorithm must exist: in other words we need to find a mathematical definition of the image structure to localise it using some automatic detection technique.

Features can be considered as the core of many problems in computer vision, therefore there are many different structures which have been studied since the beginning of this discipline. It is worth noticing that the problem of feature detection has to be considered as an elusive problem. For instance, in the case of a 3D object whose colour is identical to the background of the image, edges that are important from a 3D point of view cannot be found at all, while in the case of a highly textured image there will be a multitude of noisy edges [Mar82].

Anyway it is worth remembering that feature detection often is not the final result by itself, but it is an intermediate step of a more complex process. For this reason defining a complete state of the art for image feature is a strenuous and tough job. Our aim in this chapter is giving a perspective on local features with an emphasis on those which have some good invariance properties.

In the remainder of this section we focus the attention on the use of local features, and we will give a short description of some of the better known detection algorithm pertaining the work developed in this thesis.

1.3.1 Local feature detectors

Many computer vision tasks rely on low-level features: for this reason there are several algorithms for feature detection. Schmid et al. propose a classification of interest point detectors and give an experimental analysis of some of them [SMB00]. They group detectors in three categories:

contour based methods : first extract contours and then search for maximal curvature or inflexion points along the contour chains;

intensity based methods : compute a measure that indicate the presence of an interest point directly from the gray-values;

parametric model based methods : fit parametric intensity model to the signal.

Most of the feature detectors reviewed in this chapter fall in the second category, while for details on parametric model and contour based methods we refer to the original paper. In the next sections we will describe edge [Can86, TP86], corner [Mor77, HS88], and blob detectors [Lin93]. Then we will briefly introduce other kind of feature detectors which have been recently proposed in literature [MCUP02, LZ03].

Edge and corner detectors

Let us consider the definitions of edges and corners. Edge points, or edges, consist of pixels at or around which the image values undergo a sharp variation [TV98]. The difficulties of edge detection are due to the higher number of spurious edges generated by noise. Some of the best known algorithms for edge detection have been designed by Marr and Poggio in [MP79], by Canny in [Can86] and by Torre and Poggio in [TP86]. Figure 1.9 shows the edges extracted by the original image in Figure 1.7 using Canny algorithm. An edge can be considered as the projection of an object boundary or a surface mark or another interesting element of a scene. Instead it is not easy to interpret **corners** as geometric entities in the scene. They are not only intersections of image lines since they capture corner structure in patterns of intensities [TV98]. Figure 1.8 shows the corners extracted by the original image in Figure 1.7 with a Harris corner detector.

In other words we can say that in correspondence to an edge, the signal changes in one direction while a corner consists in a variation of the signal at least in two directions.

Probably the first point of interest detector was developed by Moravec in [Mor77]. His detector is based on the auto-correlation function of the signal: it measures the gray-values differences between a window and windows shifted in several directions. Four discrete shifts in directions parallel to the rows and columns of the image are considered and if the minimum of these four differences is superior to a threshold, an interest point is detected.

Harris corner detector [HS88] is one of the best known feature detector in the literature. It is based on the same auto-correlation matrix proposed in [Mor77] but it improves Moravec's approach since the estimation of the image gradient, $\nabla I = (I_x, I_y)$, avoids the use of discrete direction. In the original paper the gradient is computed with small differential filters, for instance $[-1\ 0\ 1]$, while today differentiated Gaussian filters are probably more

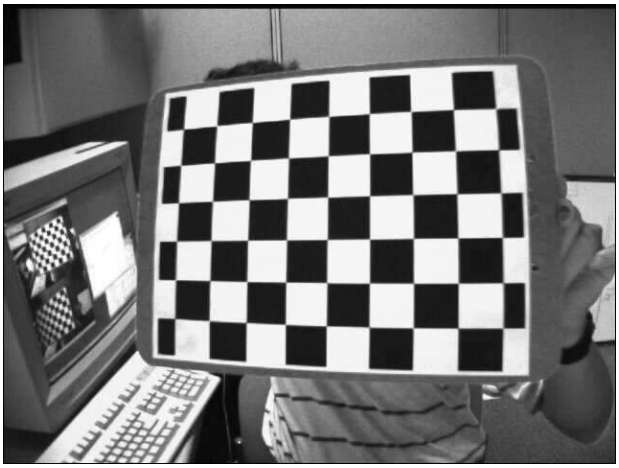


Figure 1.7: The original image

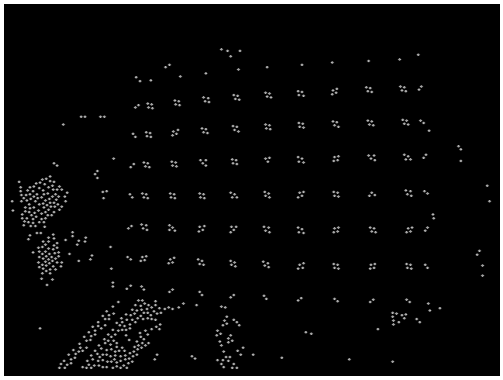


Figure 1.8: Corners extracted by the original image

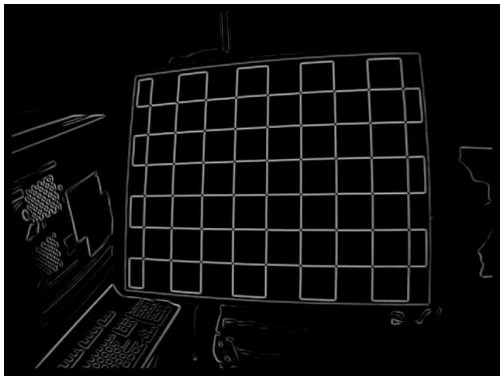


Figure 1.9: Edges extracted by the original image

used [ST94]. Thus the auto-correlation matrix computed on a window W is defined as follows:

$$C = \begin{bmatrix} \sum_W I_x^2 & \sum_W I_x I_y \\ \sum_W I_x I_y & \sum_W I_y^2 \end{bmatrix} \tag{1.5}$$

This matrix captures the structure of the neighbourhood. If C is of rank two, then we have found a corner, if it has rank one we have found an edge while a matrix of rank zero indicates a homogeneous region. Thus Harris detector can be used as an algorithm to detect edges and corners.

Blob features

Blob-like features are points belonging to regions in the image which are brighter or darker than the surrounding, or in other words we can say that a blob is a homogeneous and bounded region [Lin93]. Some of the principal methods for detecting blobs are based on derivative expressions, others are based on the location of extrema in the image.

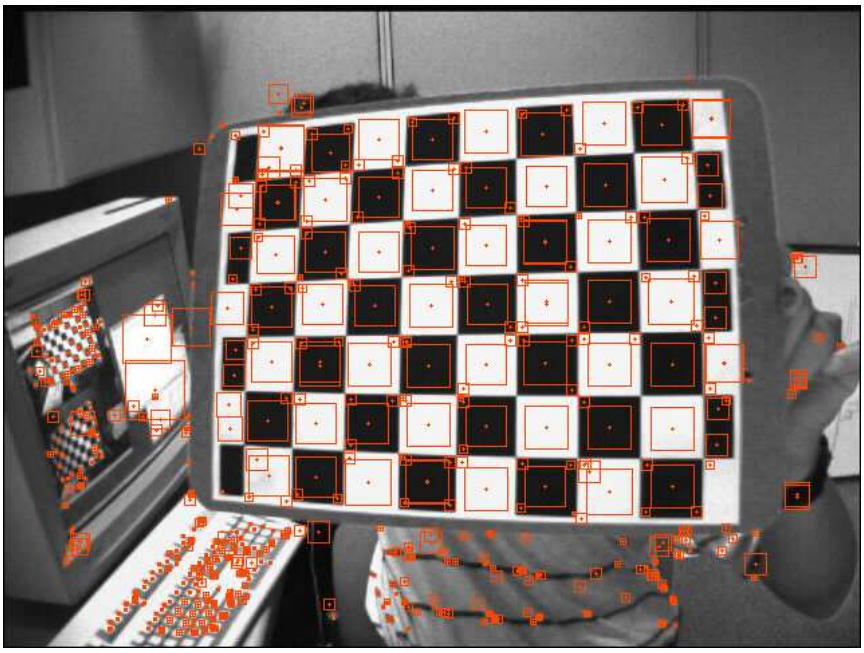


Figure 1.10: The blobs extracted by original image localised as the extrema of DoG where The squares are proportional to the dimension of blobs because a scale-space detection is performed.

It is worth noticing that the shape of *blob-like* features gives rise to the concept of blobs dimension,or, in other words, the blobs scale. The scale of a feature is an important concept that will be deeply analysed in Section 1.2.

One of the first and also most common blob detectors is based on the Laplacian of the Gaussian [Lin93, MC04] and can be briefly summarised as follows. After convolving the image with a Gaussian kernel g,

$$L(x\,,y\,;\sigma) = g(x\,,y\,;\sigma)\ *\ I(\cdot)\,,$$

the Laplacian operator is computed

$$\nabla^2 L = L_{xx} + L_{yy}$$

where L_{xx} and L_{yy} are the operators computing second derivatives. It can be shown that this operator has a strongly positive response for dark blobs of dimension $\sqrt{\sigma}$ and a strongly negative response for bright blobs of similar size [Lin93]. Therefore it is commonly used to detect blob-like regions in images.

A similar approach to blob detection is based on a pyramid of *difference of Gaussians* (DoG): the extrema of each level of the pyramid constitute the centres of the blobs. Figure 1.10 shows the centres of the blobs extracted from the image of Figure 1.7 using the DoG method for location. The squares around the centres, represent the size of the blob. More details on the detection of DoG of features are given in the Section 1.4.3.

Maximally stable extremal region

Another type of blob-like features are the Maximally Stable Extremal Regions (MSER). These regions are defined solely by an extremal property of the intensity function in the region and on its outer boundary.

The concept can be explained following the introduction given in [MCUP02]. Imagine all possible thresholdings of a gray-level image I. We will refer to the pixels below a threshold as *black* and to those above or equal as *white*. If we were shown a movie of thresholded images I_t, with frame t corresponding to threshold t, we would see first a white image. Subsequently black spots corresponding to local intensity minima will appear and grow. At some point regions corresponding to two local minima will merge. Finally, the last image will be black. The set of all connected components of all frames of the movie is the set of all maximal regions; minimal regions could be obtained by inverting the intensity of I and running the same process.

In Figure 1.11 it is shown an example of detection of maximally stable extremal region[1].

[1]MSER are obtained using the software developed by K. Mikolajczyk. The software is available for download at http://www.robots.ox.ac.uk/vgg/research/affine/index.html

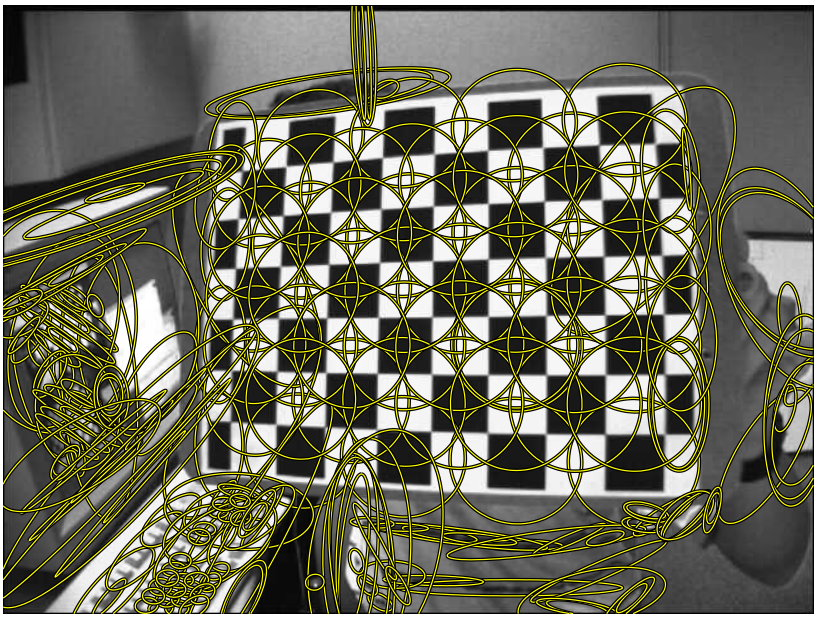


Figure 1.11: The MSER extracted by original image: they are visualised using ellipses proportional to dimension and with the same orientation of the region.

Fast radial symmetry for interest points

It has been observed that visual fixation tends to concentrate along lines of symmetry [LN87]: for this reason there are several operators capable to extract features with a radial symmetry.

We will briefly describe the operator proposed by Loy and Zelinsky in [LZ03]: the authors introduce an operator defined at each point for a certain radius. The value assumed by the operator at a given distance indicates the contribution to radial symmetry of the gradients at that distance. Even if the transform can be calculated for a continuous set of radii, this is not necessary because a finite subset of distances is usually sufficient to find meaningful results.

In Figure 1.12 it can be seen that this transform detects points of images which are characterised by a circular symmetry: choosing the radius one can detect features of different size. This is due to the fact that the transform analyses a pixel neighbourhood of dimension strictly proportional to the radius of the transform.

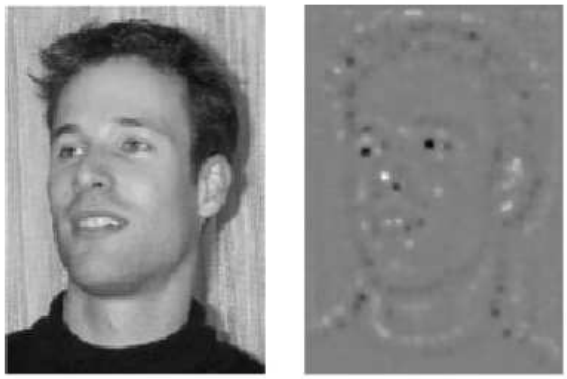


Figure 1.12: The input image and the output obtained with the transform for detection of fast radial symmetries [LZ03].

1.3.2 Evaluation of feature detectors

Along with the growth of feature selection algorithms a number of evaluation criteria have been proposed. Thus, in [SMB00], there is a brief review of criteria that can be used to evaluate the goodness of a keypoint detector:

Localisation accuracy measures if an interest point is accurately located at a specific location.

Ground-truth verification determines the undetected features (false negatives) and the false positives. Ground-truth is in general created by a human, thus it relies on symbolic interpretation of the image and is therefore subjective.

Visual inspection visually compares a set of detectors, but this method is even more subjective than using a ground-truth.

Specific context evaluation aims to describe detectors behaviour for a particular task, since the feature detection is not the final result by itself. Thus performances are evaluates how well the algorithm prepares the input for the following step.

Then, Schmid et al use two criteria for evaluating interest point detectors: repeatability and information content. They claim that the use of these criteria improves the evaluation since they are a direct measure of the quality of the feature for tasks like image matching, object recognition and 3D reconstruction. In the following sections we describe with more details how to compute repeatability and information content for feature detectors. Finally we will give some details about the localisation accuracy of keypoints.

Repeatability

Repeatability means that detection is independent of changes of the imaging conditions including illumination, and camera intrinsic and extrinsic parameters. In other words, 3D points detected in one image should also be detected at approximately corresponding positions in the other ones (see Figure 1.13). Given the 3D point X and the projection

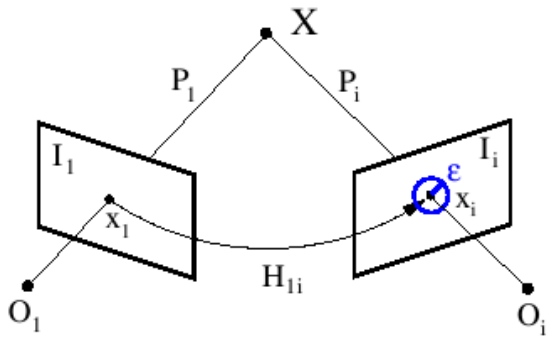


Figure 1.13: The points x_1 and x_i are the projection of 3D point X into images I_1 and I_i: $x_1 = P_1X$ and $x_i = P_iX$, where P_1 and P_i are the projection matrices. A detected point x_1 is repeated if x_i is detected. The ϵ-neighbourhood defines a tolerance.

matrices P_1 and P_i, the projection of X into the images I_1 and I_i are defined as $x_1 = P_1X$ and $x_i = P_iX$. A point x_1 detected in image I_1 is repeated in image I_i if the corresponding point x_i is detected in image I_i.

The repeatability rate is defined as the number of points repeated between two images with respect to the total number of detected points [SMB00]. To measure the repeatability we have to find a unique relation between x_1 and x_i. Repeatability has to take into account the uncertainty of detection: this means that a repeated point, in general, is not detected exactly at the corresponding position. Thus we consider an ϵ-neighbourhood for the detection of correspondences.

It is important to notice that this process is difficult for general 3D scenes whose epipolar geometry is unknown, while in the case of a planar scene this relation is defined by a homography.

Information content

Information content is a measure of the distinctiveness of an interest point. Distinctiveness is based on the likelihood of a local gray-value descriptor computed at the point within the population of all observed point descriptors [SMB00]. In other words this means that the

information content of an image is lower if all the descriptors associated with its keypoints are very similar. For instance, if all the descriptors lie close together, matching fails as any point can be matched to any other.

It is worth noticing that the information content of the descriptors is measured using the entropy, which means that the more spread the descriptors are, the higher is the entropy. In other words, entropy measures average information content. In information theory, the information content I of a message i is defined as

$$I_i = log(\frac{1}{p_i}) = -log(p_i)$$

where p_i is the probability of a partition $\mathcal{A} = \{A_i\}$. The information content of message is inversely related to its probability, thus when an event is certain (probability $p_i = 1$) then the information that it contains is none.

Evaluating the robustness of region detectors

The criteria of repeatability and information content have been used to evaluate and compare several interest point detectors: the results and the comparison can be found in [SMB00]. Another attempt to evaluate detectors is described in [MTS$^+$06] where affine covariant region detectors are reviewed and their performances are compared on a set of test images under varying imaging conditions.

The repeatability of detected regions plays again a fundamental role in the process of evaluation, since [MTS$^+$06] considers the amount of overlapping of regions using homographies. Another parameter which is at the bases of this analysis is the accuracy of localisation of the region. The comparison developed in the paper shows that the performances of all the detectors decreased slowly as the change of viewpoint increases. The conclusion is that there does not exist a detector which outperforms other detectors for all the scene types and all types of transformations. In other words we can say that the detectors can be considered as complementary since they extract regions with different properties. Therefore several detectors should be used to improve performances, even if this could increase the amount of matches and the expense of processing time.

Other approaches to image features detection require the evaluation of keypoints with respect to their scale: this can be done in the framework of scale space that we have introduced in Section 1.2. This means that when an image is given, one can perform the detection of keypoints at different levels of details of the image, so that every keypoint is located in space and in scale. Section 1.3.3 aims at giving details about features detection with automatic scale selection.

1.3.3 Feature detection in the scale space framework

In Section 1.2 we have seen that, given an image, multi-scale Gaussian filtering is the initial step for building a scale space on this 2D signal. Once the scale space is built, it can be interesting to look for features in this framework: there will be details which belong to many different levels of the scale space while other features may be present only at certain levels.

This can be helpful when one aims at analysing the structures of the image with respect to their resolution in the image itself. Usually, the approach to feature detection in scale space consists of extracting features from each level of the scale space representation. As we have already mentioned it will easily happen that a feature is appearing in many levels of the scale space representation, therefore we need a criterion to associate the right scale to each feature.

Let us first analyse the approach proposed by Lindeberg in [Lin98b], based on the following consideration. Considering a sinusoidal signal it is possible to demonstrate that the scale at which its normalised derivative is maximum over scales is proportional to the wavelength of the signal.

In this respect, maxima over scales of normalised derivatives reflect the scales over which spatial variations take place in the signal. This property is not restricted to sinusoidal signal, furthermore it can be used as a basic idea in algorithms for automatic scale selection, which automatically adapt the local scales of processing to image data.

We can generalise the above observation into the definition of the following principle:

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

In other words this means that it is possible to follow the evolution along the scale of an operator based on some normalised derivatives. This operator is computed over the same feature at different levels of resolution: its maximum along the scale, will select the level of scale characteristic of that particular feature. The remainder of this section is devoted to describe which kind of operator can be used when we are interested in detection of edge, corners and blob-like features.

Feature type	Normalised strength measure for scale selection	Value of γ
Edge	$t^{\gamma/2}\, L_v$	1/2
Corner	$t^{2\gamma}\, L_v^2\, L_{uu}$	1
Blob	$t^{\gamma}\, \nabla^2\, L$	1

Table 1.1: Measures of feature strength used for feature detection with automatic scale selection [Lin98b]. L_v and L_{vv} denote the first and second derivatives of the gradient direction.

Automatic scale selection: edges, corners and blobs

As mentioned before, in scale space framework it is possible to automatically select the scale at which an image feature can be better detected. The approach described in [Lin94, Lin98a, Lin98b] is based on the use of operators which are designed for different kinds of features: they consist of a combination of normalised derivatives, defined by

$$\partial_{\xi_i} = t_{x_i}^{\gamma/2}$$

where γ is a free parameter to be tuned to the task at hand. See Table 1.1 for a detailed description of the operators for the various kinds of features. The basic idea proposed in the above mentioned sources is to apply the feature detector at all scales, and then select scale levels from the scales at which normalised measures of feature strength assume local maxima with respect to scale. Intuitively, this approach corresponds to the selection of the scales at which the operator response is as strongest. We refer to the original paper for further examples and details.

Another blob detector in scale space

Another approach to automatically detect blob-like features in scale space is described in [Low99]. This approach is based on a pyramid of filtered images, which is a pyramid of DoG obtained by consecutively convolving the original image with a Gaussian mask with $\sigma = \sqrt{2}$. Figure 1.14 shows the pyramid of DoG as it is obtained in [Low04]. We can sketch the basic steps of the procedure:

1. The original image I is convolved with a Gaussian function with $\sigma = \sqrt{2}$. The result $I_{\sigma 1}$ is convolved again with the same Gaussian to obtain a double-filtered image $I_{\sigma 2}$.

2. The DoG is computed as a difference of the two filtered images:

   $$D_{12} = I_{\sigma 1} - I_{\sigma 2}.$$

   This will be the first level of the pyramid.

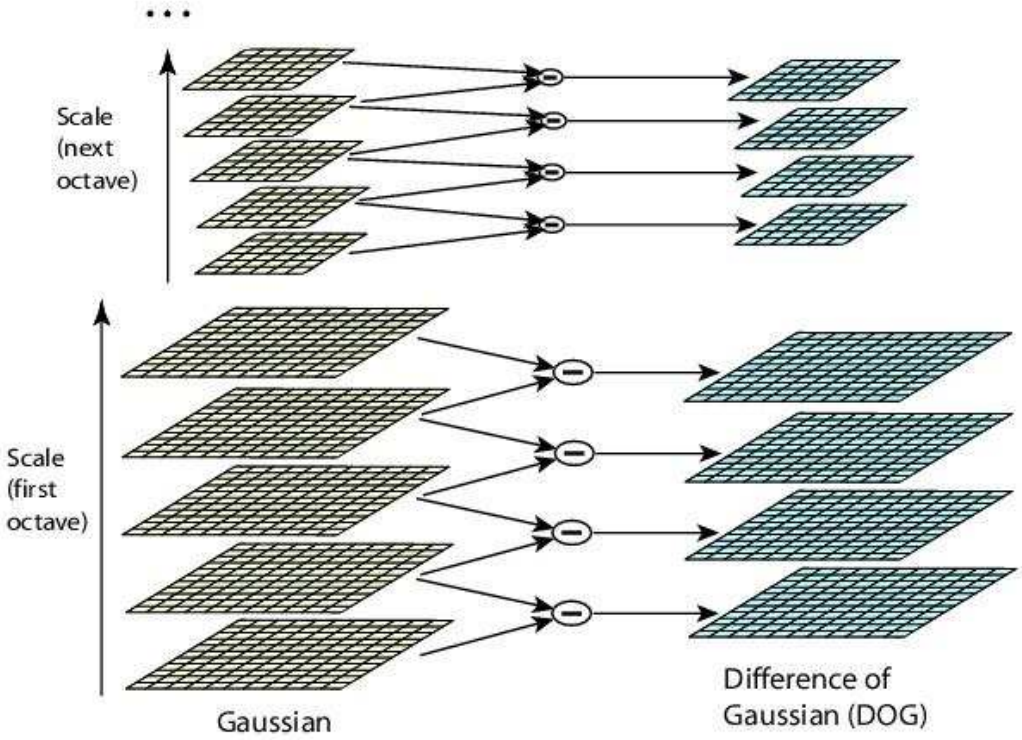


Figure 1.14: The pyramid of Difference of Gaussian [Low04].

3. The image is subsampled and the Gaussian convolution is repeated to obtain the following levels of the pyramid of DoG.

Once the pyramid of DoG is completed, the structure is scanned to detect extrema in a neighbourhood both in space and in scale. The maxima and the minima of the pyramid coincide with the centres of blob-like regions. It can be proven [Lin94] that the Difference of Gaussian function provides a close approximation to the scale-normalised Laplacian of Gaussian operator.

As we will see in Section 1.4.3, the detection of these keypoints will be the first step in the algorithm described by Lowe to detect features invariant for scale, illumination and orientation changes.

1.4 Describing image features

Once the keypoints are extracted from an image, the consequent step is that of characterising them in order to obtain a meaningful representation of image content. In other words we have to associate a description to the point detected in the image. It is worth noticing that recent approaches to image description associate descriptors to extended areas of image: [FTG06, FFJS06], for instance, consider a grid on the image and for each part of the grid compute a description. In this case we face a description which cannot be defined as global, but it can not be thought of as local since it is not based on keypoints detection.

Of course, features detection and description are closely related: both of them are not to be intended as the final goal of the system and they have to be evaluated with respect to the context of any specific task. Let us now overview the state of the art on local descriptions

1.4.1 Local descriptors: a brief state of the art

As in the case of feature detector, there exist several different techniques for describing local image regions. Perhaps the simplest local descriptor is the raw image patch. Cross correlation can then be used to compute a similarity score between two descriptors.

The advantages of such a representation are that it is easy to implement and it allows for a direct and intuitive visualisation. The main disadvantage is the fact that it strongly depends on illumination changes, rotations of the image and view point variations.

According to the classification described in [MS03], we can consider *distribution based descriptors* which use histograms to represent different characteristic appearance or shape. These kind of descriptors can be considered as more global information on the image: for instance gray-value or colour histograms [SB91] describe images by counting the *colour* of each pixel. The main drawback of histograms is that the representation is solely dependent of the colour of the object being studied and it looses any geometric information. Combining information coming from local keypoints with different kind of histograms is an approach recently investigated in [Low99, Low04, RSSP06]. SIFT descriptors [Low99, Low04] are defined as localised histograms of gradient orientations computed at multiple scales and they will be described in detail in the next section.

Shape context presented in [BMP02] implements a method to compute a 3D histogram of location and orientation for edge points where all the edge points have equal contribution in the histogram. These descriptors were successfully used, for example, for shape recognition of drawings for which edges are reliable features. It is worth noticing that these descriptors may be used for describing a complete 2D shape (as for instance a silhouette) but they can also be used to describe a neighbourhood of the interest point.

Other description techniques are based on the analysis of spatial frequencies of the content of an image as for instance the Gabor transform [Gab46] or the wavelets [Vet95] which have been frequently used in the context of texture classification.

Other descriptors are those based on derivatives, usually called *differential descriptors*. The properties of local derivatives, or *local jets*, were investigated by Koenderink [KvD87] and they consist in higher order derivatives of the image computed at interest point. Freeman and Adelson [FA91] developed steerable filters, which steer derivatives in a particular direction given the components of the local jet. Steering derivatives in the direction of the gradient makes them invariant to rotation. A stable estimation of the derivatives is obtained by convolution with Gaussian derivatives.

Schmid introduces a descriptor based on derivative in [SM97]. The local characteristics used in this work are based on differential gray-value invariants to ensure invariance under the group of displacements within an image and a multi-scale approach makes this characterisation robust to scale changes. The application described in this work is image retrieval.

Gradient Location and Orientation Histogram (GLOH) is an extension of the SIFT descriptor designed to increase its robustness and distinctiveness. Mikolajczyk and Schmid introduced GLOH in [MS04a]: they compute the SIFT descriptor for a log-polar location grid with 3 bins in radial direction (the radius set to 6, 11 and 15) and 8 in angular direction which makes 17 location bins. The descriptor obtained is a vector of 272 whose dimension is reduced via PCA. In their work, they compare GLOH descriptor with many other different feature descriptors.

1.4.2 Evaluation of local descriptors

In [MS03, MS04a] there is a comparison of the performances of different descriptors computed for local interest regions extracted with a given method as the one described in Section 1.3.1. Many different descriptors have been proposed in the literature. However, it is unclear which descriptors are more appropriate and how their performance depends on the interest region detector. The descriptors should be distinctive and at the same time robust to changes in viewing conditions as well as to errors of the detector.

In [MS03] the evaluation of descriptors is performed in the context of matching and recognition of the same scene or object observed under different viewing conditions. The evaluation criterion is based on recall and precision: in other words it is based on the number of correct and false matches between two images.

In [MS04a] the evaluation uses as criterion recall with respect to precision and is carried out for different image transformations. The analysis takes into account several descriptors

and there are some evidences that the best performances are obtained by SIFT based descriptors. Another interesting conclusion is that the ranking of the descriptors is mostly independent of the point detector.

As a consequence of these studies we based our work on SIFT descriptors. The next section is devoted to a more detailed overview of such descriptors.

1.4.3 SIFT

In this section we describe in detail the algorithm for computing SIFT, or *Scale Invariant Feature Transform*, since it has been widely used in the development of this thesis. Lowe [Low99, Low04] introduced a local feature descriptor inspired by the response properties of complex neurons in the human visual cortex. An intrinsic quality of SIFT description is that of being robust to scale changes since the construction of this representation is heavily inspired by scale space framework. SIFTs are defined as 4×4 grids of localised histograms of gradient orientations computed at multiple scales.

SIFT are invariant to image scale and rotation and they provide robust matching across a substantial range of affine distortion, change in 3D viewpoint and change in illumination. We will now briefly describe the algorithm which is used in [Low04] to locate and recognise objects in images. The algorithm can be essentially sketched in the following steps:

Keypoints detection in the scale-space pyramid. We have already described in detail pyramid in Section 1.3.3.

Refinement of detection. The feature detection phase ends with a cleaning procedure, where all the pixels belonging to edges or that have low contrast are discarded.

Kepoints description in the scale space pyramid. We compute local histograms of gradient direction in each level of the scale space: the final description vector is composed of a combination of these histograms.

Since the descriptions are obtained for each level of the scale space, the various histograms contain information belonging to different levels of resolution. Let us give more details about the creation of the descriptor vectors.

Keypoints description

Once keypoints are selected and refined, the next step is that of choosing a suitable representation. The pixel position is defined in the scale space to gain invariance to scale change. To achieve also invariance with respect to image rotation another feature is attributed to

the keypoint: its orientation. The orientation of a keypoint is defined using a histogram for the gradient direction in a fixed circular neighbourhood of the pixel. The peak in the histogram corresponds to keypoint principal direction. In case of multiple peaks we obtain several keypoints with different principal directions. Figures 1.15 shows an example of a histogram direction computed on a patch. In the case of Figure 1.16 there is a peak which is 80% high the highest peak: this is the case in which a keypoint is given a double principal direction. Thus this keypoint will have two description vectors.

After the computation of the keypoint principal direction, a keypoint is defined by

$$p_k = (x_k, y_k, s_k, d_k) \ ,$$

where x_k and y_k define a position in the space of the image, s_k is the scale level at which the keypoint is found and d_k is its principal direction. Now we have all the necessary information to compute the proper descriptor. The description chosen for the keypoints [Low04] is a vector with 128 elements obtained with a direction histogram computed at the level selected by s_k. The directions in the neighbourhood are normalised on the basis of the principal direction previously computed for each keypoint. Since there can be many s_k for each keypoint, there can be many descriptors associated to each keypoint.

For most of the experiments accomplished in [Low04], the window is 4×4 square region around the keypoint. In other words, the region around the keypoint is divided in 16 square sections. For each square sub-region we compute an 8-bins histogram and we normalise it with respect to the canonical direction of the keypoint. The 8-bin histogram entries will be the components of the long descriptor, since the process is repeated for every square zone until the 128 dimension vector is completely filled. This approach has been suggested to the author by a model of biological vision [EIP97]. Figure 1.17 shows a descriptor computed on a squared window with a 2×2 grid.

Figure 1.18 shows the results of detection of DoG extrema drawn at the appropriate scale. The figure also shows the principal direction and the coloured square represent the area where a SIFT descriptor have been computed. It is worth noticing that in the regions with high texture there are many little features at small scale and edges are correctly rejected by the detection algorithm. Lowe exploits SIFT invariance to obtain a clustering technique [Low01] and a robust 3D object recognition algorithm[Low04]. As we will see in Chapter 2, we will use SIFT descriptors to develop a novel matching method, while in Chapter 3 we will put them at the basis of a 3D object recognition approach for video sequences.

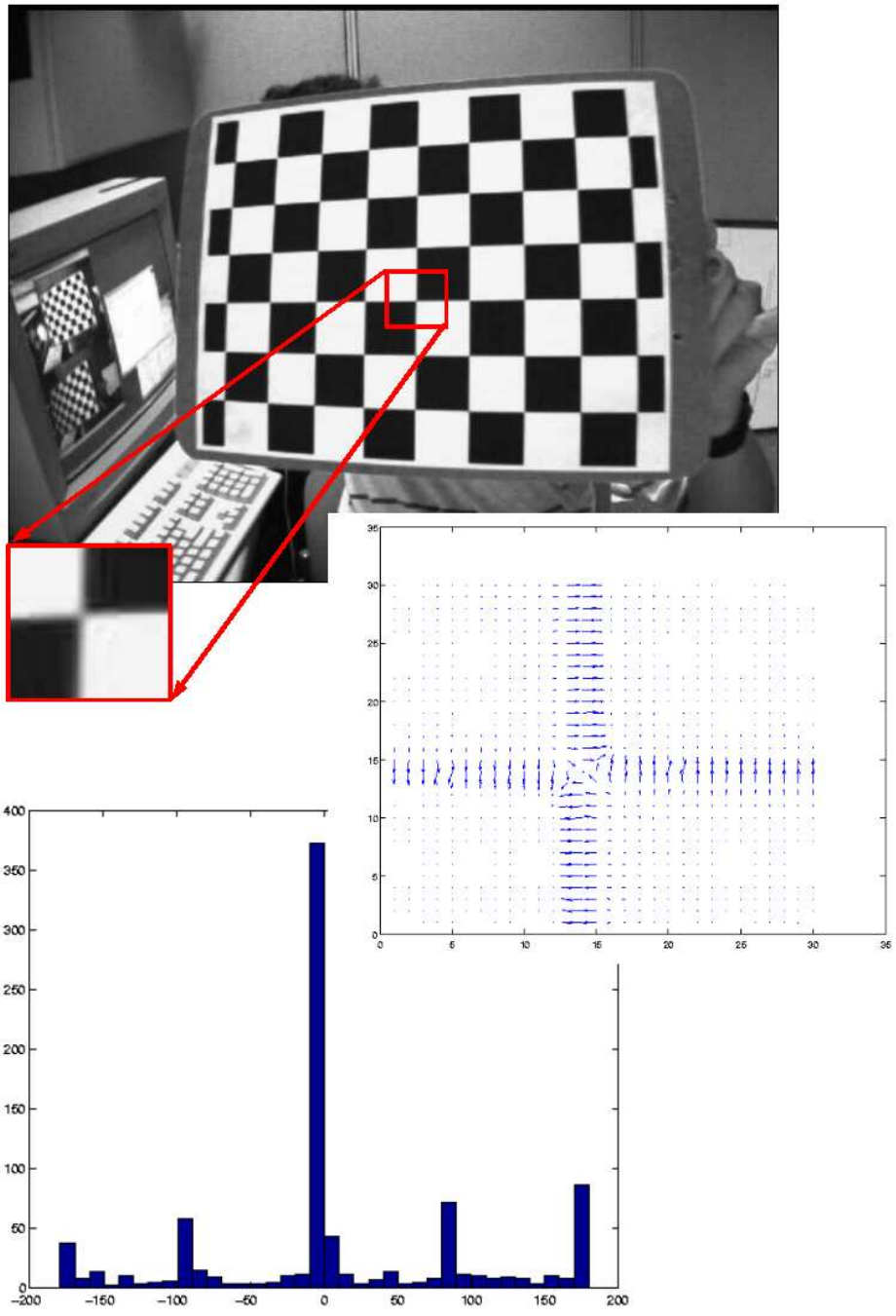


Figure 1.15: An example of a histogram direction computed on a square patch. In the middle there is a vector representation of the gradient of the image patch.The great peaks on $0°$ is due to the high number of points with null gradient, while multiple peaks in the histogram are originated by the fact that directions $-180°$, $180°$ and $0°$ are the same.

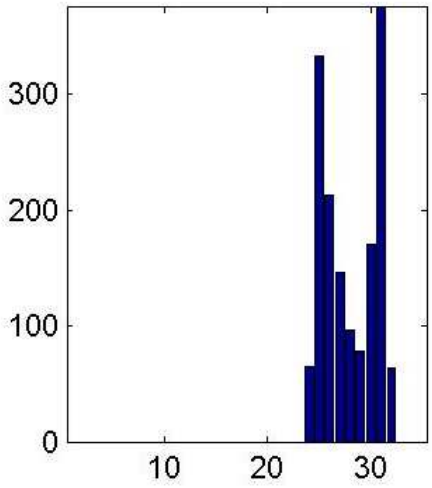


Figure 1.16: This figure shows a direction histogram which has a double peak: since there is one peak which is 80% high the highest peak the keypoint corresponding to this histogram is given a second principal direction. Each bin is 10°.

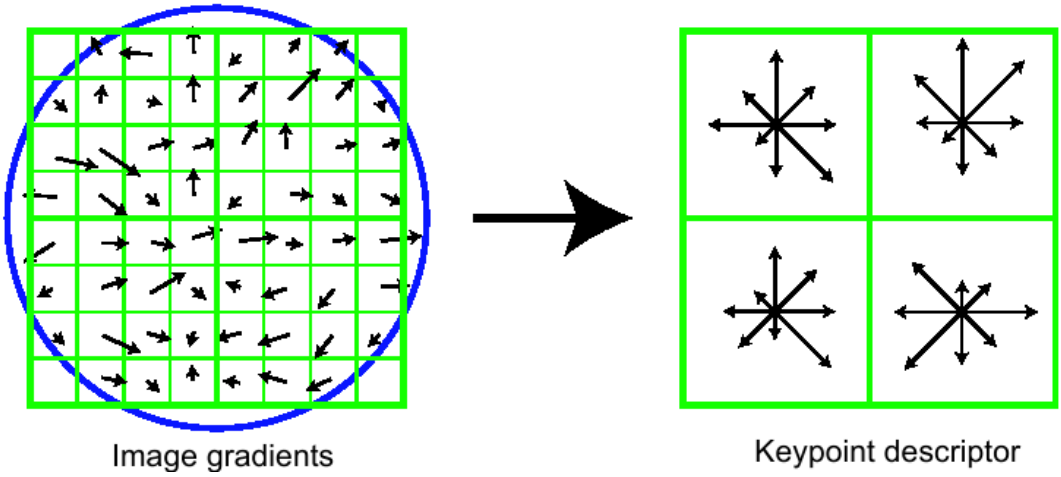


Figure 1.17: Illustration of the SIFT descriptor of Lowe [Low04]. Image gradients within a patch (left) are accumulated into a coarse 4×4 spatial grid (on the right, only a 2×2 grid is shown). A histogram of gradient orientations is formed in each grid cell. 8 orientation bins are used in each grid cell giving a descriptor of dimension $128 = (4 \times 4 \times 8)$.

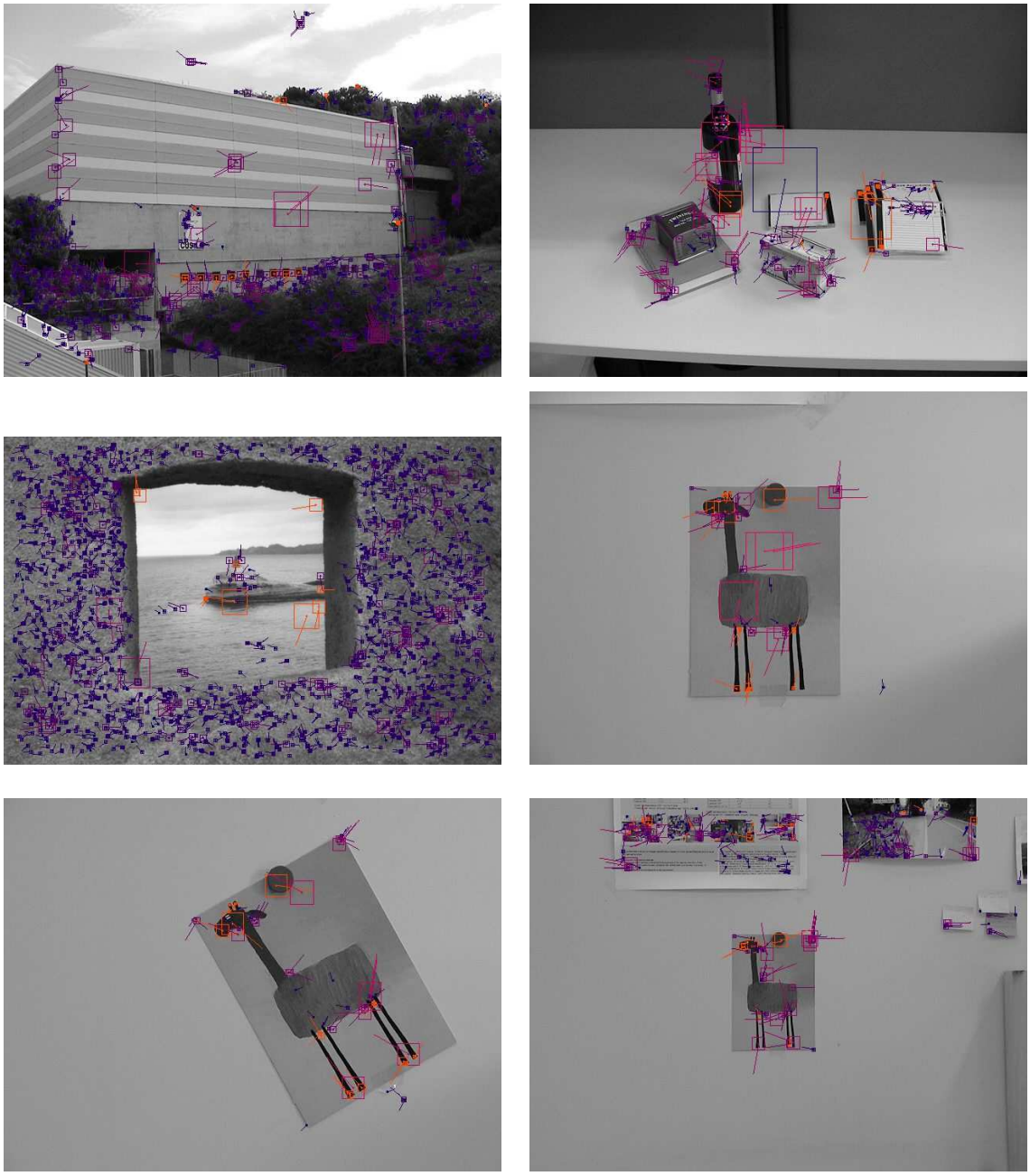


Figure 1.18: Results of DoG detection on a set of different images. The coloured squares represent the area for the computation of SIFT descriptors (different colours and dimensions refer to different levels of scale).

Chapter 2

Matching points

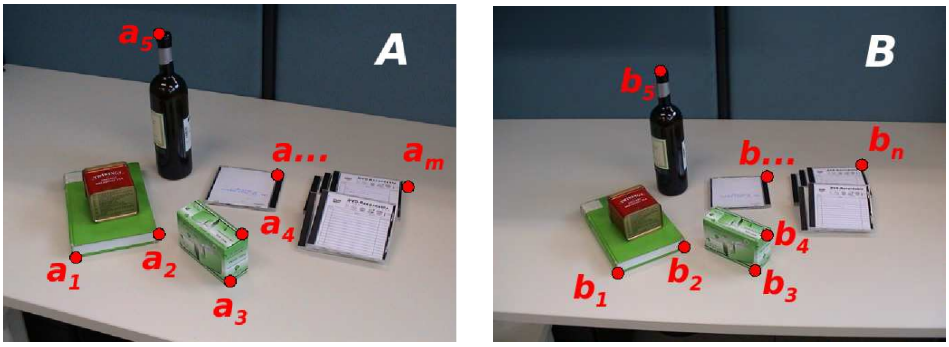
Finding correspondences between feature points is one of the keystones of computer vision, with application to a variety of problems. In this chapter we first describe some of the most commonly used methods for image matching, focusing our attention on spectral approaches to feature matching. Then we propose a novel version of the SVD-matching proposed by Scott and Longuet-Higgins [SLH91] and later modified by Pilu [Pil97], that exploits the robustness of SIFT descriptions.

2.1 Introduction

A central problem in theory of vision [Wer12, Mar76, Ull79] is that of establishing a correspondence between the features of two stereo pair or successive frames in motion sequence. If the two images are temporally consecutive, then computing correspondence determines motion. If the two images are spatially separated but simultaneous, then computing correspondence determines stereo depth. When the two images are acquired by different sensors the problem is called *sensor fusion*. Finally, the case of two images of the same object acquired from different viewpoints is considered as the problem of building a rich description of the object, thus this task points at object recognition. All of these tasks can be seen under the common framework of matching.

Let A and B be two images (see Figure 2.1) which contain m and n features respectively (A_i, $i = 1, \ldots, m$, and B_j, $j = 1, \ldots, n$): for each feature a_i detected in image A, we look for the best matching feature b_i in image B. The criterion to decide the best matching is usually based on a measure of similarity computed on a descriptor such as for instance gray level patches or direction histograms.

Automatic feature matching is often an initialisation procedure for more complex tasks,



(a) (b)

Figure 2.1: Matching image A with image B.

such as fundamental matrix estimation, image mosaicing, object recognition, and three-dimensional point clouds registration. Thus the state of the art on algorithms for image matching is vast. Hence in the remainder of this section we sketch a taxonomy of matching techniques. It is common practice to distinguish between *feature-based methods* and *direct methods (or area-based)*. The former rely on first acquiring image meaningful features and then matching them, producing sparse matches. Direct methods try to find matches over all image positions. The results are dense disparity maps, less reliable in flat areas because of the aperture problem.

Direct methods usually assume a small view-point change, and they are often applied to stereo and motion estimation. Feature-based methods assume a keypoints detection step so that the number of points to match is smaller with respect to area-based methods. When a priori information on the image is not available, feature-based methods seem to be computational more convenient, even if sparse methods can be often considered as an useful initialisation for dense methods. When epipolar geometry is known, area-based method algorithms compare a window in one image with several windows on the corresponding epipolar line in the other image.

The comparison between points from different images can be performed with many different techniques: Section 2.2 is devoted to extend the description of some of these methods, focusing on spectral methods and on the problem of matching with a large baseline.

2.2 State of the art on sparse matching

In this section we focus our attention on feature-based methods: we review the main contribution to this topic, afterwards we will give some details on methods for large-baseline matching. The section ends with a brief overview of spectral methods applied to

matching.

2.2.1 Matching interest points

It is important to notice that the state of the art for matching is strongly related to the literature on local features. Indeed early works on matching images with salient features were based on using keypoints such as corners [Mor81, HS88]. Since in Chapter 1 we have described in depth different algorithms for detecting and describing image features, now we will briefly mention some algorithms which use interest points for matching.

Classical approaches to point matching with unknown geometry assume a short baseline, and they are usually based on correlation (see, for instance, [DZLF94]). It is well known that correlation-based approaches suffer from view-point changes and do not take into account the global structure of the image. On this respect, in [CM95], is reported a set of experiments aiming at comparing different forms of SSD and normalised correlation of image neighbourhoods. Correlation methods suffer also from illumination changes, thus Zabih and Woodfill in [ZW98] propose a way to circumvent this problem: they compute visual correspondence on the basis of local ordering of intensities. Their approach is based on non-parametric measures of association but also accounts for the spatial variation of disparities. This method is used both for video-rate stereo and motion.

One limit of the above methods is that they only consider a simple image transformation, as for instance a translation. Among methods allowing different kind of transformation between images, it is worth to remember the technique proposed by Baumberg [Bau00] which uses Harris corners and a description based on the Fourier-Mellin transform to achieve invariance to rotation. Harris corners are also used in [ADJ00], where rotation invariance is obtained by a hierarchical sampling that starts from the direction of the gradient.

Recently, Grauman and Darrell [GD05], introduced an algorithm for matching images which is based on local invariant features. They assume that local feature invariant to common image transformations are an effective representation to use when comparing images. They present a method for efficiently comparing images based on their discrete distributions of distinctive local invariant features without clustering descriptors. The similarity they use is an approximation of the Earth Mover's Distance. An approach considering a non rigid transformation is proposed by Chui and Rangarajan in [CR00]. They introduce the algorithm for a *non rigid point matching* using a *thin-plate spline* to parametrise the non rigid mapping. TPS is used since it can be considered as a natural non rigid extension of the affine map.

(a) (b) (c)

Figure 2.2: The same scene is seen from three different points of view: (a) and (b) can be considered a pair of short baseline matching images while (a) and (c) are a pair of wide baseline images.

2.2.2 Matching with large or wide baselines

It is well known that a major source of appearance variation is view-point change. This variation becomes more challenging to model as the distance between observation points (i.e., the baseline) grows: see Figure 2.2 for an example. This section reviews some methods considering this issue. Early applications to local image matching were stereo and short-range motion tracking. Zhang *et al.* showed that it was possible to match Harris corners over a large image range, with an outlier removal technique based on a robust computation of the fundamental matrix and the elimination of the feature pairs that did not agree with the solution obtained [ZDFL95]. Later on, the invariant features described above were extensively studied as they guaranteed some degree of flexibility with respect to view-point change. Recently, many works on extending local features to be invariant to affine transformations have been proposed, including a variant of SIFT [BL02]. On this respect, Matas et al. [MCUP02] introduce the concept of maximally stable extremal region (see Section 1.3.1) to be used for robust wide baseline matching.

Tuytelaars and Van Gool [TG00] deal with wide-baseline matching extracting image regions around corners, where edges provide orientation and skew information. They also address scale variation by computing the extrema of a 2D affine invariant function; as a descriptor they use generalised colour moments. The actual matching is done using the Mahalanobis distance. In a more recent work [FTG03] they establish wide-baseline correspondences among unordered multiple images, by first computing pairwise matches, and then integrating them into feature tracks each representing a local patch of the scene. They exploit the interplay between the tracks to extend matching to multiple views. A method based on automatic determination of local neighbourhood shapes is presented in [GSBB03], but it only works for image areas where stationary texture occurs.

An alternative approach for determining feature correspondences relies on prior knowledge

on the observed scene, for instance in knowing the epipolar geometry of two or more views [SZ97]. Georgis et al. [GPK98] assume that projections of four corresponding non coplanar points at arbitrary positions are known. Pritchett and Zissermann [PZ98] use local homographies determined by parallelogram structures or from motion pyramids. Lourakis et al [LTAO03] present a method based on the assumption that the viewed scene contains two planar surfaces and exploits the geometric constraints derived by this assumption. The spatial relation between the features in each images, together with appearance, is used in [TC02].

Recently a simple ordering constraint that can reduce the computational complexity for wide baseline matching, for the only case of approximately parallel epipolar lines, has been proposed in [LM04].

2.2.3 Spectral analysis for point matching

Spectral graph analysis aims at characterising the global properties of a graph using the eigenvalues and the eigenvectors of the graph adjacency matrix [Chu97]. Recently this subject has found a number of applications to classical computer vision problems, including point matching, segmentation, line grouping, shape matching [Ume88, SB92, CH03, SP95, Wei99]. In this section we review some works on point matching with spectral analysis.

Most of these contributions are based on the so called *proximity* or *affinity* matrix, that is a continuous representation of the adjacency matrix: instead than being set to 0 or 1, the matrix elements have weights that reflect the strength of a pair relation (in terms of proximity or sometimes similarity). Usually the proximity matrix is defined as:

$$G_{ij} = e^{-r_{ij}/2\sigma^2} \tag{2.1}$$

with σ a free parameter, and r_{ij} a distance between points x_i and x_j computed with an appropriate affinity.

Scott and Longuet-Higgins [SLH91] give one of the most interesting and elegant contributions to this topic, that we will describe in Section 2.3. One of the first applications of spectral analysis to point matching is due to Umeyama [Ume88]. The author presents an SVD method for finding permutations between the adjacency matrices of two graphs. If the graphs have the same size and structure of the edges the method is able to find correspondences between the nodes of the graph.

Shapiro and Brady [SB92] propose a method that models the content of each image by means of an intra-image point proximity matrix, and then evaluates the similarity between images by comparing the matrices. The proximity matrices are built using a Gaussian weighting function, as in Equation 2.1. For each proximity matrix, a modal matrix (a matrix the columns of which are eigenvectors of the original matrix) is built. Each row of

the modal matrix represents one point of the corresponding image. The authors find the correspondences by comparing the rows of the two modal matrices, using a binary decision function based on the Euclidean distance.

Carcassoni and Hancock [CH03] propose a variant of this approach that changes the original method in three different ways. First, the evaluation of proximity matrices are based on other weighting functions, including a sigmoidal and an Euclidean weighting function; second, the use of robust methods for comparing the modal matrices; third, an embedding of the correspondence process within a graph matching EM algorithm. Experiments reported in the paper show that the latter contribution is useful to overcome structural errors, including the deletion or insertion of points. The authors also show that the Gaussian weighting function performs worst than the other weighting functions evaluated.

2.3 SVD-matching

In this section we present a method for determining the correspondences between sparse feature points in images of the same scene based on the SVD-matching paradigm, that has been used by different authors in the past, and on SIFT. We show that including SIFT point descriptors in the SVD-matching improves the performance with respect to the past versions of this algorithm [DIOV06]. In particular it returns good results for scale changes, severe zoom and image plane rotations, and large view-point variations. Section 2.4 presents an extensive experimental evaluation on different image data.

As for many spectral methods, the SVD-matching algorithm is based on the choice of an appropriate weighting function for building a proximity matrix. The previous SVD-matching algorithms were using a Gaussian function. We compare its performance against other functions borrowed from the robust statistics literature: the Lorentzian and the double-exponential function.

2.3.1 Previous work

Here we summarise the algorithms proposed in [SLH91] and [Pil97] upon which we base our matching technique. Scott and Longuet-Higgins [SLH91], getting some inspiration from structural chemistry, were among the first to use spectral methods for image matching. They show that, in spite of the well-known combinatorics complexity of finding feature correspondences, a reasonably good solution can be achieved through the singular value decomposition of the proximity matrix of Equation (2.1) followed by a simple manipulation of the eigenvalues. As pointed out in [Pil97] their algorithm is rooted into the solution of the subspace rotation problem known as orthogonal Procrustes problem (see [GL83] for

details).

Consider again Figure 2.1. A and B are two images containing m and n features respectively (A_i, $i = 1, \ldots, m$, and B_j, $j = 1, \ldots, n$). The goal is to determine two subsets of the two sets of points that can be put in a one to one correspondence. In the original algorithm proposed by Longuet-Higgins, the main assumption was that the two images were taken from close points of view, so that the corresponding points had similar image coordinates.

The algorithm consists of three steps:

1. Build a proximity matrix $\mathbf{G}$, where each element is computed according to Equation (2.1). Let $r_{ij} = ||A_i - B_j||$ be the Euclidean distance between the two points, when considering them in the same reference plane. The parameter σ controls the degree of interactions between features, where a small σ enforces local correspondences, while a bigger σ allows for more distant interactions. The elements of $\mathbf{G}$ range from 0 to 1, with higher values for closer points.
2. Compute the Singular Value Decomposition for $\mathbf{G}$: $\mathbf{G} = \mathbf{VDU}^\top$.
3. Compute a new correspondence matrix $\mathbf{P}$ by converting diagonal matrix $\mathbf{D}$ to a diagonal matrix $\mathbf{E}$ where each element D_{ii} is replaced with a 1: $\mathbf{P} = \mathbf{VEU}^\top$.

The algorithm is based on the two principles of proximity and exclusion, that is, corresponding points must be close, and each point can have one corresponding point at most. The idea is to obtain from the similarity matrix $\mathbf{G}$ a matrix $\mathbf{L}$ such that the entry ij is 1 if i and j are corresponding points, 0 otherwise. The matrix $\mathbf{P}$ computed by the algorithm is orthogonal (in the sense that the rows are mutually orthogonal), as all the singular values are 1, and it is the orthogonal matrix closest to the proximity $\mathbf{G}$. Because of the orthogonality, if the parameter σ is chosen properly, $\mathbf{P}$ enhances good pairings, as its entries have properties close to those of the ideal matrix $\mathbf{L}$. Following this idea the algorithms establishes a correspondence between the points i and j if the entry P_{ij} is the largest element in row i and the largest element in row j.

In the case of real images, point localisation is affected by noise and keypoint detection is unstable — keypoints may be detected or not depending on the viewing angle. The algorithm presented in [SLH91] was working well on synthetic data, but performance started to fall down when moving to real images. Pilu [Pil97] argues that this behaviour could be taken care of by evaluating local image similarities. He adapts the proximity matrix in order to take into account image intensity as well as geometric properties. The modified matrix appears as follows

$$G_{ij} = \frac{C_{ij} + 1}{2} e^{-r_{ij}^2 / 2\sigma^2} \tag{2.2}$$

where the term C_{ij} is the normalised correlation between image patches centred in the feature points.

In [Pil97] experimental evidence is given that the proposed algorithm performs well on short baseline stereo pairs. In fact the performance falls when the baseline increases. It is our target to show that the reason for this behaviour is in the feature descriptor chosen and is not an intrinsic limit of the algorithm.

2.3.2 SVD-matching using SIFT

In this section we discuss the use of the SIFT descriptor in the SVD matching algorithm. As mentioned in the previous section SVD-matching presented in [Pil97] does not perform well when the baseline starts to increase. The reason for this behaviour is in the feature descriptor adopted. The original algorithm uses the grey level values in a neighbourhood of the keypoint. As pointed out in Section 2.2 this description is too sensitive to changes in the view-point, and more robust descriptor have been introduced so far.

The comparative study of the performance of various feature descriptors [MS03], described in Section 1.4.1, showed that the SIFT descriptor is more robust than others with respect to rotation, scale changes, view-point change, and local affine transformations. The quality of the results decrease in the case of changes in the illumination. In the same work, cross-correlation between the image grey levels returned unstable performance, depending on the kind of transformation considered. The considerations above suggested the use of a SIFT descriptor, instead of grey levels. The descriptor is associated to scale and affine invariant interest points [MS04b].

In a previous version of this work [DIOV05] we left the matrix $\mathbf{G}$ in equation (2.2) unchanged in its form, but C_{ij} was the cross-correlation between SIFT descriptors. This straightforward modification improves the performance of the SVD-matching, and also gives better results, in terms of number of points correctly matched, with respect to the SIFT distance used for the experiments reported in [MS03]. However the matrix terms are still strongly dependent on the distance on the image plane between feature points, causing a large number of mismatches when the distance between points increases. For this reason we decided to switch back to the original form of the $\mathbf{G}$ matrix, with

$$G_{ij} = e^{-r_{ij}^2/2\sigma^2} \tag{2.3}$$

where r_{ij} is now the distance between the feature descriptors in the SIFT space.

In order to reduce the number of mismatches even further we also added a constraint on the entry P_{ij} for determining the correspondence between points i and j. Let a_{ij_1} and a_{ij_2} being respectively the largest and second largest elements in row i, and $b_{i_1 j}$ and $b_{i_2 j}$ the largest and second largest elements in column j. We say that i and j are corresponding points if

1. $j_1 = j$ and $i_1 = i$

2. $0.6a_{ij_1} \geq a_{ij_2}$ and $0.6b_{ij_1} \geq b_{ij_2}$

In plain words it still needs to be the largest element in row i and column j, but also the largest by far.

2.4 Experimental results

In this section we report some experiments carried out on different image pairs and sequences. First we show some of the matches returned by our algorithm on few image pairs. Then we attempt a more quantitative analysis of the performance of our algorithm on short image sequences.

2.4.1 Experiments on image pairs

The first lot of experiments that we show refers to results on image pairs of two different scenes returned by the algorithm proposed in this chapter.

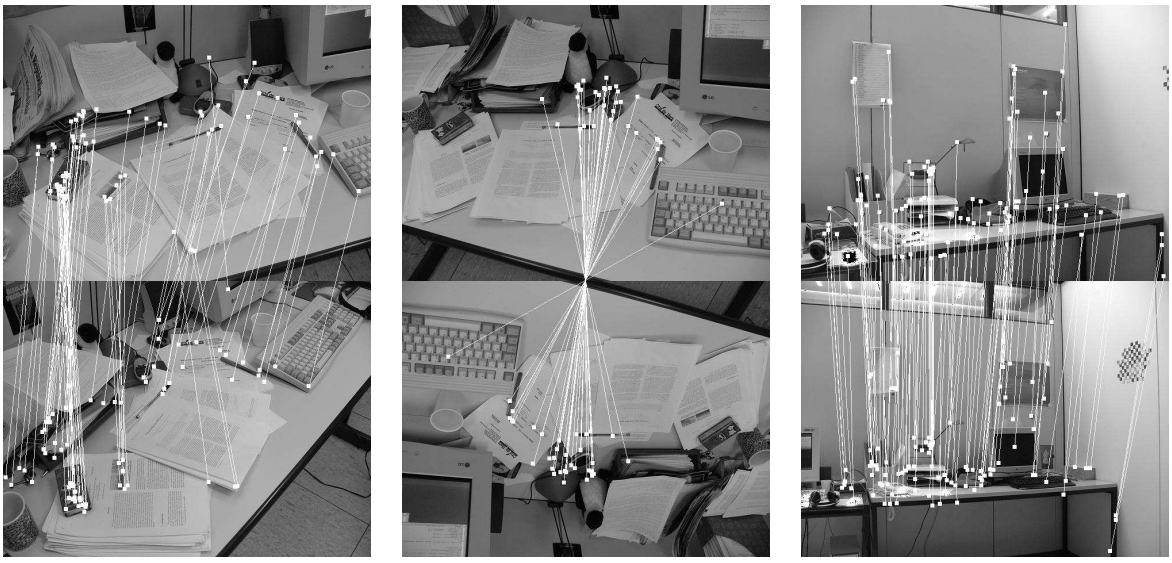
In Figures 2.3 (a) and 2.3 (b) we show all the matches determined on two pairs of images of a desk scene. The first one presents a reasonable level of scene variation, whereas the latter is a synthetic rotation of the first image. We spotted only a wrong match in Figure 2.3 (a). The last image pair is relative to a studio scene with scale variation. The result is shown in Figures 2.3 (c). Our visual inspection of the results determined only few wrong matches between points on the border of the table.

In Figure 2.4 we show the matches determined on a large baseline stereo pair. A visual inspection could spot no more than three wrong matches.

2.4.2 Comparative experiments

We performed different comparative experiments. The first group of experiments focuses on proximity matrices built in the descriptor space as for the one given in (2.3), that uses a Gaussian weighting function. Following [CH03] we test against the Gaussian the performance of two other weighting functions, drawn from the literature on robust statistics.

The second group of experiments tests the performance of the algorithm using the proximity matrix proposed in (2.3) against two other matrices proposed in previous works [Pil97, DIOV05], and a SIFT based point matcher, based on the Euclidean distance between SIFTs, proposed by Lowe in [Low99], and used in [MS03] for measuring the SIFT performance.



(a) (b) (c)

Figure 2.3: Matches determined for stereo pairs of a desk. a) A reasonable level of scene variation. We could notice only one wrong match between the wall and the corner of the screen. b) The second image is a synthetic rotation of the first one. No wrong matches have been determined. c) Scale variation, wrong matches on the edge of the table.

For evaluating the performance of the three point matching methods used for this work we computed: a) the total number of matches detected; b) the number of correct matches; c) the accuracy, defined as the ratio between number of correct matches and the total number of matches detected.

The data used are of different nature. We considered a stereo image sequence taken with a stereo system with relatively large baseline, and in particular we focused our experiments on input sequences for an immersive video-conferencing system [ITKS04]. Then we used short sequences with large variations with respect to the first frame: the kind of variations considered are viewpoint changes and zoom plus rotation[2]. Some of these last sequences were used in [MS03]. The experiments performed on the video sequences compare the first frame of the sequence with all the others. Sample frames from the sequence used are shown in Figure 2.5.

The method used for determining the correct matches depends on what geometric information on the camera geometry is available. For sets of data consisting of fixed camera sequences or sequences of planar scenes for which the homographies between the different

[2]Sequences available from http://www.robots.ox.ac.uk/∼vgg/research/affine/index.html

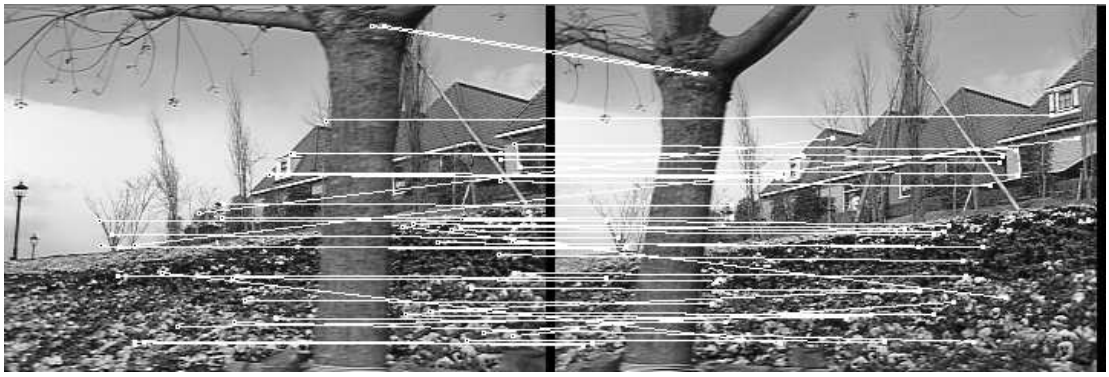


Figure 2.4: Matches determined for a large baseline stereo pairs. Only 2-3 wrong matches are determined.

views were available, we say that a pair of corresponding points (p, p') is a correct match if

$$||p' - Hp|| < 5$$

where H is the homography between the two images.

For the stereo sequence with a fixed baseline the correspondence were computed between images of each stereo frame. In this case, because the scene is not planar, we compute the fundamental matrix F from the calibrated camera matrices, and a pair of corresponding points (p, p') is a correct match if

$$(d(p', Fp) + d(p, F^t p'))/2 < 5$$

where $d(p', Fp)$ is the distance between point p' and the epipolar line corresponding to point p [TV98].

For all the experiments we set the parameter σ to 1000.

Comparison of different weighting functions

The weighting function models the probability of the similarity between the feature points. In previous works it was used the Gaussian weighting function. The reason for trying functions different from the Gaussian is that the distance between feature descriptors of corresponding points increases with the baseline. In this case a function with more prominent tails than the Gaussian can give the chance to detect some more matches. This, as we will see, at the price of a sometimes lower accuracy.

In this section we considered a small sample of different weighting functions borrowed from the literature on robust statics, in particular from the literature on M-estimators

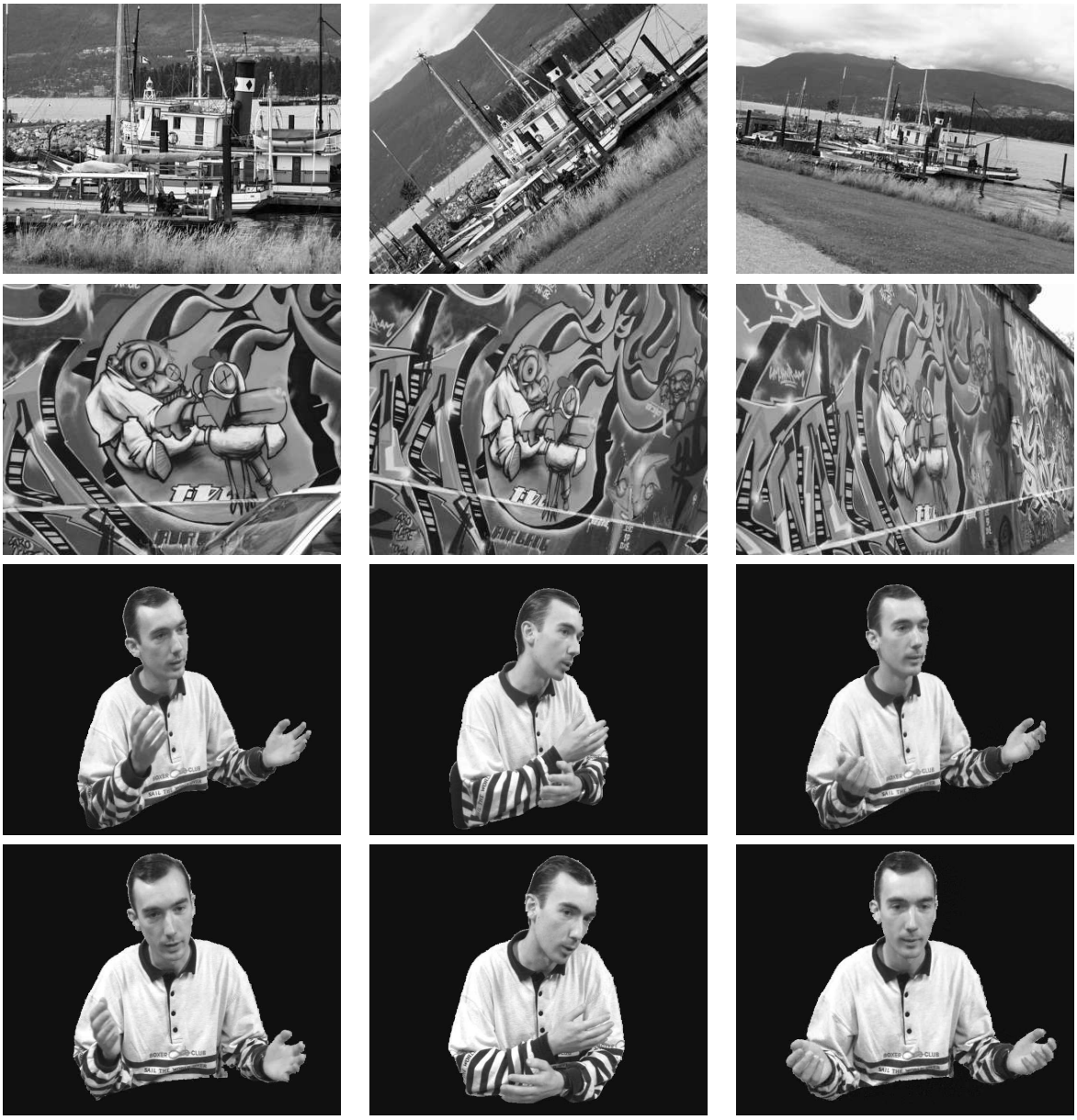


Figure 2.5: Sample from the sequences used for the experiments presented in this chapter. First row: 1st, 3rd and 5th frame of the *Boat* sequence. Second row: 1st, 3rd and 5th frame of the *Graf* sequence. Third and fourth rows: left and right views respectively of the 1st, 16th and 30th frame of the stereo sequence.

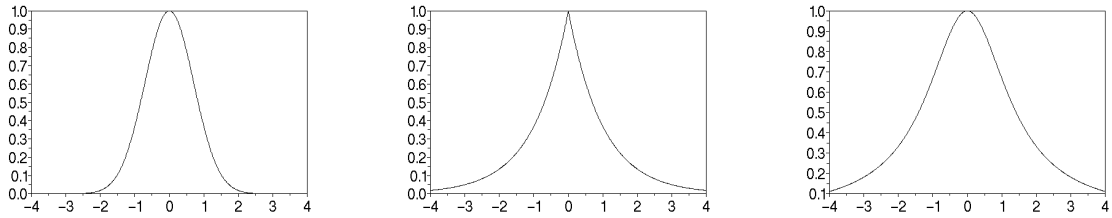


Figure 2.6: The different weighting functions used. Left: Gaussian. Middle: Double-exponential. Right: Lorentzian.

[HRRS86]. The comparative evaluation on the performance of different matching methods, whose description is given in 2.3.2, are based on the following different weighting functions:

- **S-SVD**: a Gaussian weighting function as it has been used all along the experiments described in this chapter;
- **D-SVD**:a double exponential weighting function

$$G_{ij} = e^{(-|r_{ij}/\sigma|)} \tag{2.4}$$

- **L-SVD**: a Lorentzian weighting function, defined as

$$G_{ij} = \frac{1}{1 + \frac{1}{2}\frac{r_{ij}^2}{\sigma^2}} \tag{2.5}$$

The different weighting functions are shown in Figure 2.6. We choose

In Figure 2.7 and 2.8 we show the results for the video-conferencing stereo sequence. We see that in terms of number of matches and correct matches the double exponential function returns the best results, while the Gaussian and the Lorentzian have similar performance. These last two report an average accuracy of .6. The accuracy returned by the double-exponential is lower, but on average above .5, that means that at most 50% of the matches detected are wrong matches, and this is the largest amount of wrong matches that standard robust statistics tools can tolerate.

The results for the *Boat* sequence are shown in Figures 2.9 and 2.10. Even in this case the D-SVD returns the highest number of correct matches, and, except for the last frame, the accuracy reported is above .7. The other two functions reported an accuracy always well above .5.

For the *Graf* sequence the results are similar to what seen in the previous section (see Figure 2.11). The double-exponential still returns the largest number of correct matches,

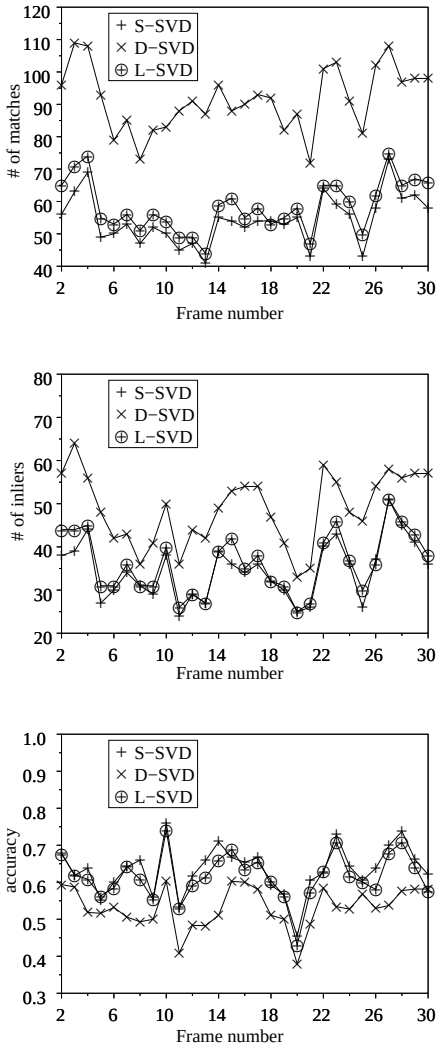


Figure 2.7: Comparison with other weighting functions: results for the 30-frames stereo sequence. The baseline is fixed for all the stereo pairs, and the correspondences are computed for each stereo frame of the sequence. Top: total number of matches detected. Middle: number of correct matches. Bottom: accuracy of the method.

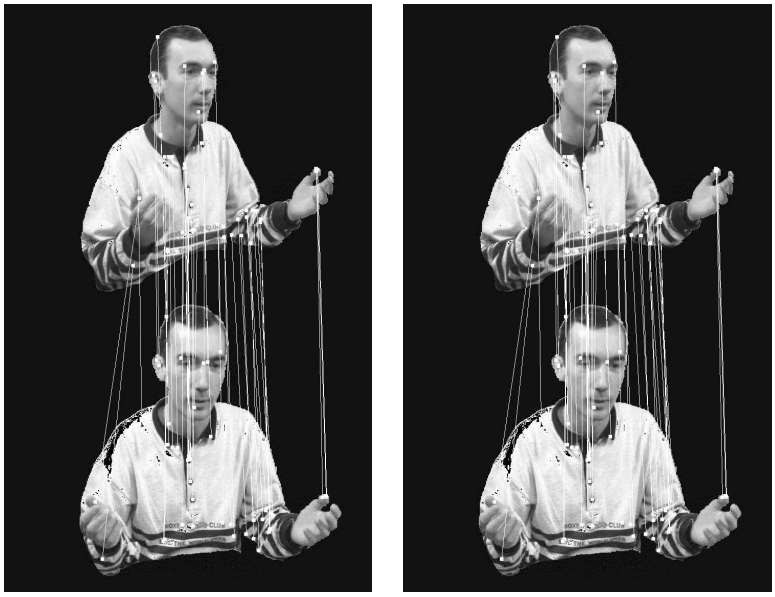


Figure 2.8: Comparison of different weighting functions: results for the 30-frames stereo sequence. Correct matches between the left (top) and right (bottom) 27th frames. Left; S-SVD Right: L-SVD. The results for D-SVD are in Figure 2.13.

but again the performance drops for the last two frames of the sequence when the change in the point of view is too large.

We can conclude this evaluation of the weighting functions saying that the double-exponential performs slightly better than the other two functions considered, but it does not seem that the use of any of these function dramatically changes the performance of the algorithm.

The double-exponential weighting function will be used in the following analysis.

Comparison with other matching algorithms

The comparative evaluation on the performance of different matching methods considers the following techniques:

- **D-SVD**: point matches are established following the SVD-matching algorithm of Section 2.3 with the proximity matrix $\mathbf{G}$ given in (2.4);
- **C-SVD**: point matches are established following the algorithm discussed in [DIOV05]

$$C_{ij} = \sum_t \frac{(S_t^i - mean(S^i))(S_t^j - mean(S^j))}{stdv(S^i)stdv(S^j)}$$

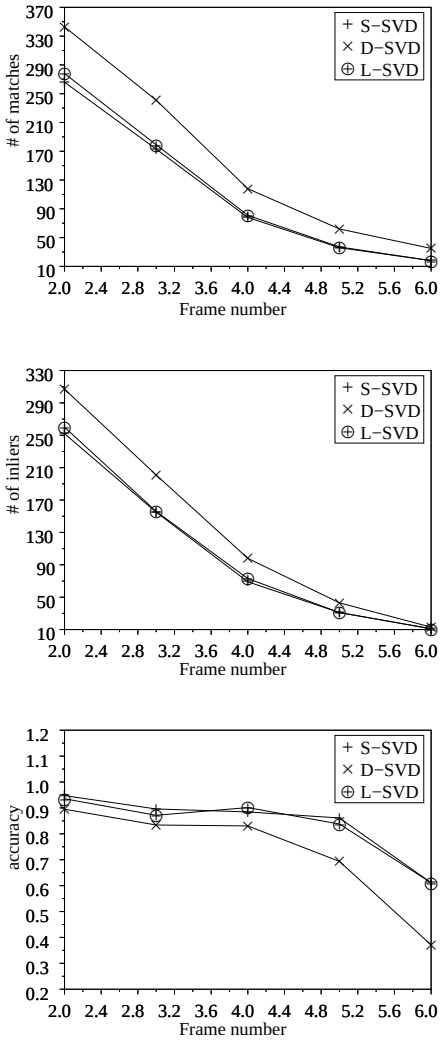


Figure 2.9: Comparison of different weighting functions: results for the *Boat* sequence. The images are zoomed and rotated respect to the first frame. Matches are computed between the first frame and each other frame of the sequence. Top: total number of matches detected. Middle number of correct matches. Bottom: accuracy of the method.

Figure 2.10: Comparison of different weighting functions: results for *Boat* sequence. Correct matches between the left (top) and right (bottom) last. Left; S-SVD Right: L-SVD. The results for D-SVD are in Figure 2.15.

where S^i and S^j are the SIFT descriptors;

- **P-SVD**: point matches are determined as for C-SVD but with

$$c_{ij} = \sum_t \frac{(I_t^i - mean(I^i))(I_t^j - mean(I^j))}{stdv(I^i)stdv(I^j)}$$

where I^i and I^j are the two grey-levels neighbour;

- **S-DIST**: point matches are established following the method proposed in [Low99], that is two features i and j matches if

$$d_{ij} = \min(D_i) < 0.6 \min(D_i - \{d_{ij}\})$$

and

$$d_{ji} = \min(D_j) < 0.6 \min(D_j - \{d_{ji}\})$$

where $D_i = \left\{ d_{ih} = ||S^i - S^h|| \right\}$.

In Figures 2.12 and 2.13 we show the results for the video-conferencing stereo sequence. The S-SVD returns the largest number of matches and of correct matches (an average of

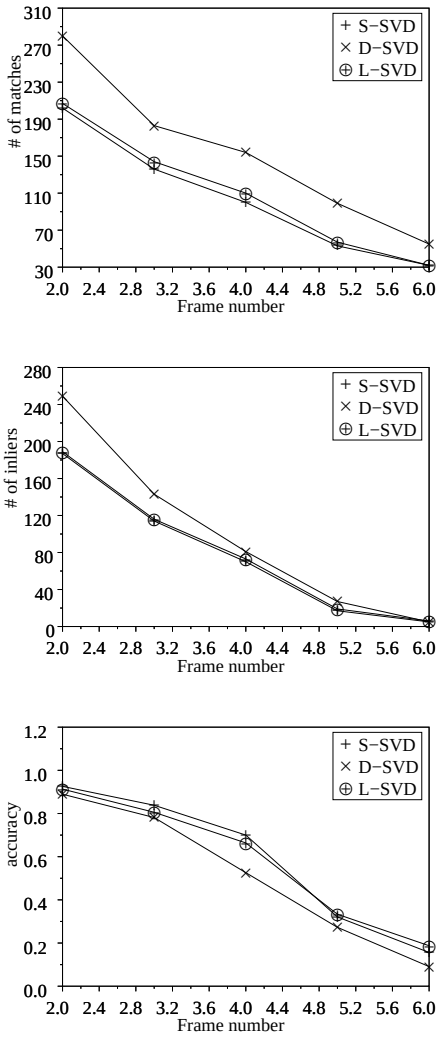


Figure 2.11: Comparison of different weighting functions: results for the *Graf* sequence. The images present a change in the view point respect to the first frame. Top: total number of matches detected. Middle: number of correct matches. Bottom: accuracy of the method.

50 and 40 respectively for each stereo frame) with respect to the other three: the C-SVD presents an average of 30 and 20 per stereo frame, while the values returned by the other two methods are much lower.

S-DIST returns the highest accuracy (almost always 1), but a very small number of matches. The accuracy obtained with D-SVD and C-SVD is slightly lower (ranging from 0.7 yo 0.5) but it is high enough to use standard robust statistics tools for identifying and discarding wrong matches. As for P-SVD we notice that accuracy drops down to 0.4 that is too low for treating outliers with robust statistics.

The results shown in Figure 2.14 and 2.15 are relative to a six frames sequence where the fixed camera zooms and rotates around the optical centre. In this case D-SVD is still giving the larger amount of correct matches. The number of matches goes down sensibly, because of the large zoom effect between the first and the last frame, so that the points detected at a finer scale in the first frame cannot be matched. The C-SVD still has acceptable performance while the other two methods perform poorly on this sequence. In particular P-SVD can only find matches between the first two frames. This is because this method uses correlation between image patches, that are very sensitive to rotation and scale changes.

The performance of the algorithms starts to go down with severe changes in the view point, as shown in Figures 2.16 and 2.17. In fact for the last 2 frames the amount of matches and the accuracy obtained are too low. The results returned by the S-DIST algorithm, that has been designed for the SIFT descriptor, are even worse, implying that the descriptor cannot cope with too large viewpoint changes. Similar results have been reported in [MS03] for several descriptors.

In conclusion we can state that the use of the SIFT descriptors in combination with a SVD-matching algorithm improves the performance with respect to older versions of the algorithm, as already shown in [DIOV05]. Moreover the experiments show that replacing the distance between feature points with the distance between point descriptors in the weighting function used to build the proximity matrix gives better results when large changes in the scene occur. This is particularly noticeable in the case of severe zoom/rotation changes.

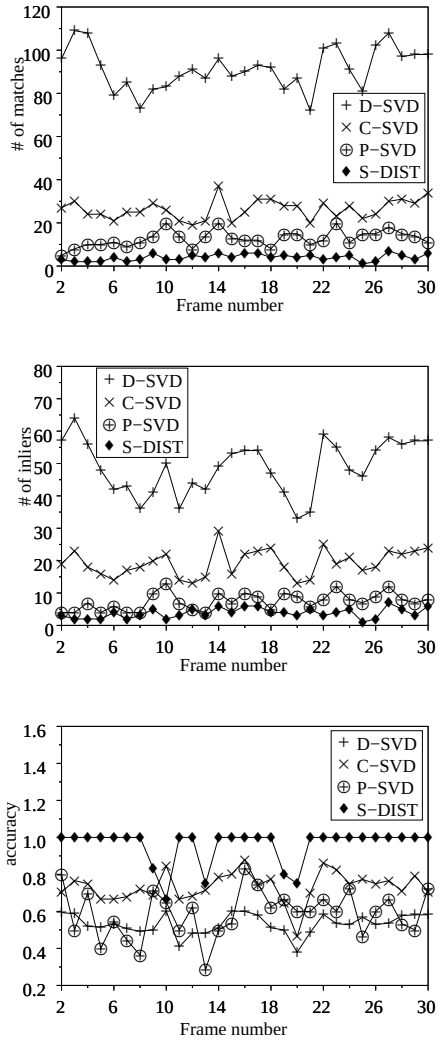


Figure 2.12: Comparison with other algorithms: results for the 30-frames stereo sequence. The baseline is fixed for all the stereo pairs, and the correspondences are computed for each stereo frame of the sequence. Top: total number of matches detected. Middle: number of correct matches. Bottom: accuracy of the method.

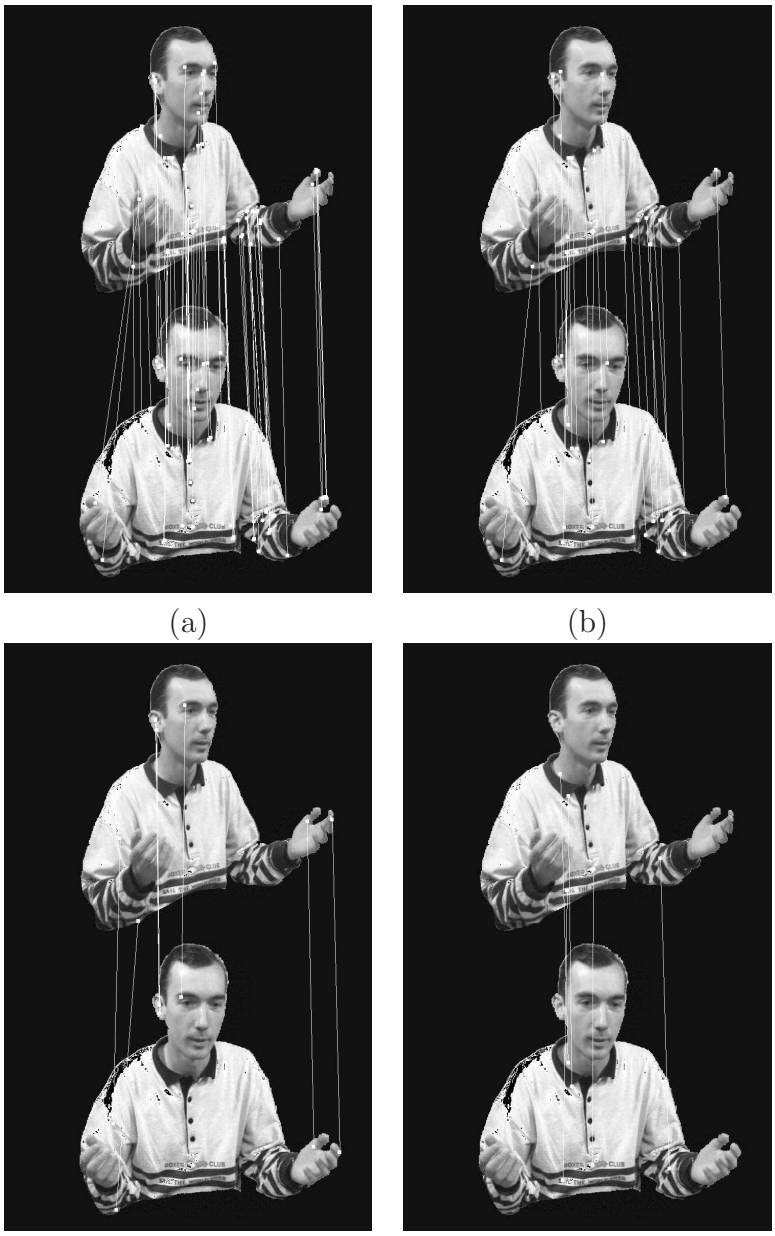


Figure 2.13: Comparison with other algorithms: results for the 30-frames stereo sequence. Correct matches between the left (top) and right (bottom) 27th frames. a) D-SVD. b) C-SVD. c) P-SVD. d) S-DIST.

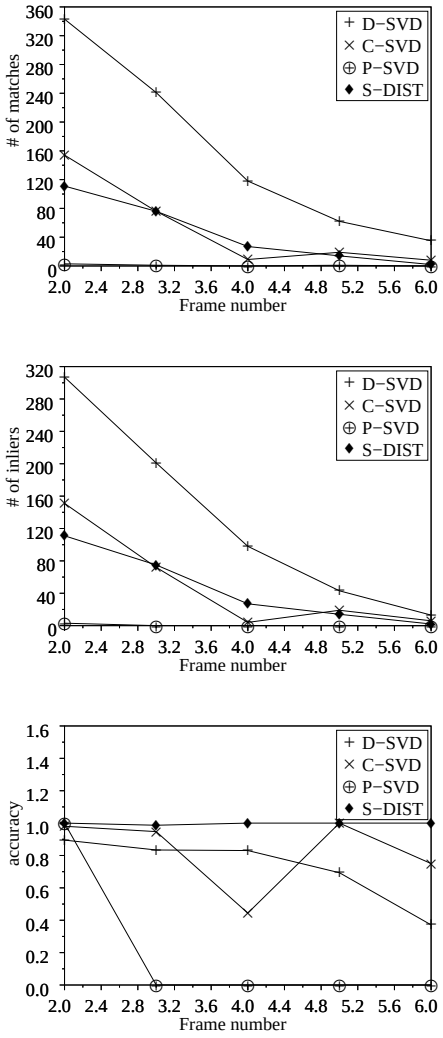


Figure 2.14: Comparison with other algorithms: results for the *Boat* sequence. The images are zoomed and rotated respect to the first frame. Matches computed between the first frame and each other frame in the sequence. Top: total number of matches detected. Middle: number of correct matches. Bottom: accuracy of the method.

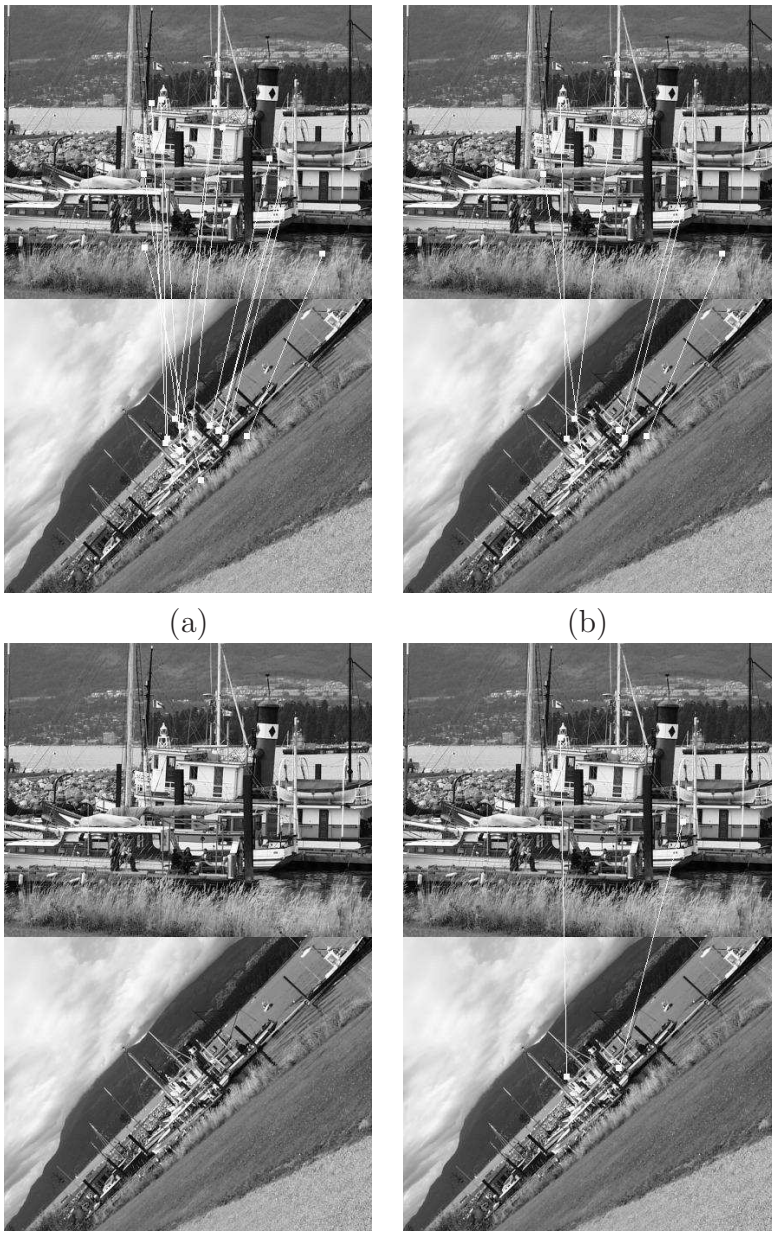


Figure 2.15: Comparison with other algorithms: results for the *Boat* sequence. Correct matches between the left (top) and right (bottom) last frames. a) D-SVD. b) C-SVD. c) P-SVD. d) S-DIST.

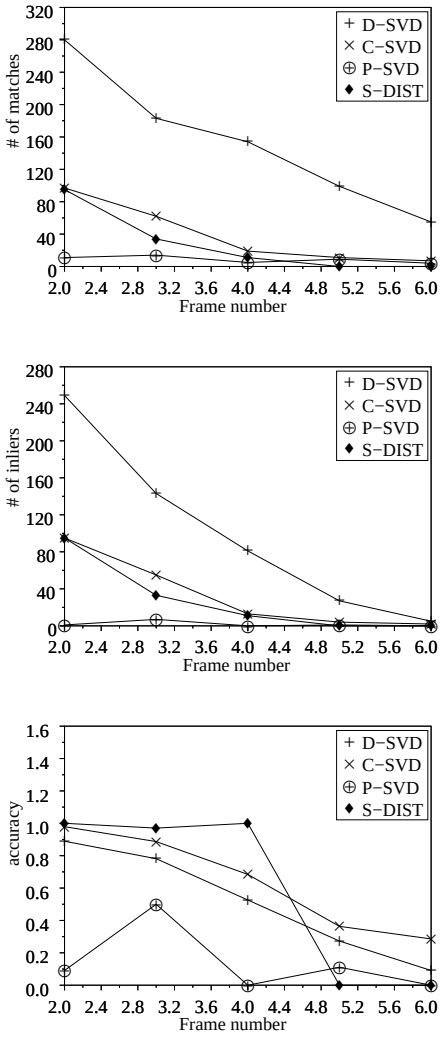


Figure 2.16: Comparison with other algorithms: results for the *Graf* sequence. The images present a change in the view point respect to the first frame. Top: total number of matches detected. Middle: number of correct matches. Bottom: accuracy of the method.

Figure 2.17: Comparison with other algorithms: results for the *Graf* sequence. Correct matches between the left (top) and right (bottom) last frames. a) D-SVD. b) C-SVD. P-SVD. and S-DIST did not return any correct match.

Chapter 3

Recognising 3D objects

In the first part of this chapter we concentrate on methods for 3D objects recognition based on appearance and we review some of the better known methods for object recognition. In the second part of the chapter we introduce novel ideas on the use of local descriptors in image sequences deriving view-based models of objects exploiting spatial and temporal information.

3.1 Introduction

One of the classical problems in computer vision is that of determining whether or not the image data contain some specific object, feature, or activity. This task can be solved robustly and without effort by a human, but it is still not solved in computer vision for the general case in which there is an arbitrary object in an arbitrary situation. The existing methods for dealing with this problem can at best solve it only for specific objects, such as simple geometric objects, human faces, printed or hand-written characters, vehicles, and in specific situations, typically described in terms of well-defined illumination, background, and pose of the object relative to the camera.

According to [Gri90] we provide a definition of the problem of recognising objects in images. The goal of *recognition* systems is to identify which objects are present in a scene, while *localisation* means to determine the pose of each object relative to a sensor. In the case the objects to be recognised are rigid (the distance between any two points on the object is constant), the problem of the estimation of the pose is constrained to a small number of parameters. The problem is harder to solve for articulated, flexible or deformable objects.

Before we give more details on object recognition it is worth distinguishing it from another common problem of computer vision which is multi-class *object categorisation*. In this

case the aim is recognising the class to which an object belongs instead of recognising that particular object. As an example, recognition would distinguish between images of two structurally distinct cups, while categorisation would place them in the same class. It is important to notice that the notion and abstraction level of object classes is far from being uniquely and clearly defined. Thus, recognising the object category is another challenging issue that has already been tackled with several approaches. While some very good results have been achieved for the detection of individual categories such as faces, cars and pedestrian [PP00, RBK98, VJ01] little progress has so far been made on the discrimination of multiple categories with some notable exceptions [TMF04, Lei04].

Object recognition and categorisation are also tightly coupled with other classical problems in computer vision:

Content based image retrieval which is the process of retrieving images in a database or an image collection on the basis of the similarity between low-level image features, given a query image or manually constructed description of these low-level image features.

Object detection that refers to deciding whether or not an object is present in a given image.

When speaking about 3D object recognition one may refer to a recognition process based on 3D geometric model of the object or consider an appearance-based approach. In the remainder of this dissertation we will consider only those methods based on the visual appearance of the object. It is worth to notice that there is a difference with those view-based recognition methods that represent objects only from one particular point of view [SC96, Low99]. Indeed, an effective 3D object recognition is based on the use of many views of the object to obtain a model which is capable to describe it from different viewpoints [Low01, CDFB04, GB05, FTG06].

Next section is devoted to present the state of the art in object recognition: we review some methods for recognising objects and we focus our attention on those techniques which are considered local since they use interest points to obtain general information on the whole image.

3.2 State of the art on object recognition

The first approaches to 3D object recognition have been tackled from the geometric point of view: the information used to characterise an object is organised in the form of a 3D model focused on geometric measurements. This approach is usually called model-based recognition. Several model-based approaches to recognition have been studied and details

on possible classifications can be found in [BJ85, Hut89, Gri90] in which the authors presented their taxonomy for model-based object recognition. A more specific work that addresses 3D object construction from image sequence, is reported in [Pol00]. A first way to distinguish among the different approaches is dividing the recognition method on the basis whether the pose is inferred from global properties or from mappings of local model features. Usually such models are sophisticated, difficult to build, and often hard to use. Aspect graphs [KvD79, BJ85], instead, are one of the first attempts to represent the 3D appearance of objects in terms of their 2D projections. With aspect graphs, though, there is no computational gain relative to computing the full 3D model of the object. Nevertheless the idea of representing objects using 2D rather than 3D models has been supported by recent advances in biological vision [BET95, Sin95], and has been adopted by various researchers.

For understanding the necessity of view-based approaches to object recognition, now we need to focus on a classical observation reported by Edelman in [Ede97] and first suggested by Wittgenstein [Wit73] who discussed the difference between seeing a shape and seeing it as something. Unlike "merely" perceiving a shape, recognising it involves memory, that is accessing at representations of shapes seen in the past. The form of these representations is constrained by the various factors such as orientation and illumination that affect the appearance of objects. This implies the necessity to store information coming from different views of the objects.

There are several reasons motivating the introduction of view-based approaches to object recognition, among which we recall biological inspiration. The research in neuroscience has shown that, for recognising objects, primates use simple features which are invariant to changes in scale, location and illumination [Tan97]. The approach proposed by Edelman, Intrator and Poggio in [EIP] makes use of a model of biological vision based on complex neurons in primary visual cortex which respond to a gradient at a particular orientation but allows for shift over a small receptive field rather than being precisely localise. They hypothesised that the function of these complex neurons was to allow for matching and recognition of 3D objects from a range of viewpoints.

Among view-based techniques, it is worth mentioning 2D morphable models [PV92, BSP93, VP97, JP98] and view-based active appearance models [ECT99, CWT00], which use a selection of 2D views for the purpose of modelling a 3D complex object. The idea behind both methods (which achieve comparable results in two rather different ways) is that of synthesising a model for each object or class of objects, and then matching it with a novel image to check for consistency. Both morphable models and active appearance models are firmly based on registering a selection of images belonging to the same object or the same class, and compute linear combination of the examples in terms of their 2D shape and their texture, which represent a model of the object.

It is important to notice that techniques based on the appearance follow a considerable

body of previous research in model-based vision. In fact also appearance-based methods can be divided in two categories, global or local, in case they are building a representation of the object by integrating information over the entire image [SB91, MN95, CK01, PV98] or over a set of local interest points respectively [Low04, MC02, RSSP06, TG00].

As we have already said, global methods build an object representation by integrating information over the whole image and therefore they are very sensitive to background clutter and partial occlusions. Hence, global methods only consider test images without background, or necessitate a prior segmentation, a task which has proven extremely difficult or else they rely on the fact that context is as important to recognition as the object is [BOV02, Tor03]. Additionally, robustness to large viewpoint changes is hard to achieve, because the global object appearance varies in complex and unpredictable ways.

Local methods counter problems due to clutter and occlusions by representing images as a collection of features extracted on local information only, thus now we focus on these approaches. These are the reasons why we explore appearance based 3D object recognition using local information for recognising 3D objects. Therefore in the next section we give a historical perspective on local approach methods to object recognition.

3.2.1 Local approach to object recognition

The methods reviewed in this section are all based on describing objects, no matter how complex, by means of local image structures. Starting from this local information they try to find descriptors of the objects that are characteristic and meaningful. Sometimes global information may be used as an add on to strengthen or disambiguate a decision. The second part of the section describes contributions based on this idea. Most of the local descriptors used have been reviewed in Sections 1.4.1.

Schiele and Crowley, in [SC00], proposed one approach to 3D object recognition based on a framework for the statistical representation of the appearance of arbitrary 3D objects. The representation consists of a probability density function or joint statistics of local appearance as measured by a vector of robust local appearance as measured by a vector of robust local shape descriptors. The object representations are acquired automatically from sample images. Multidimensional histograms are introduced as a practical and reliable means for the approximation of the probability density function for local appearance. The method proposed in [SC00] combines global techniques as those used in [SB91, BCGM98] with object recognition methods based on local characteristics (similar to point matching tasks) [Gri90]. The set of objects used for testing the method proposed in [SC00] is quite large (over 100 objects). The results reported demonstrate that it is possible to recognise objects in cluttered scenes using local approaches, but it is worth to notice that most of the objects used in the test have a planar structure, thus they cannot be considered as real

3D objects.

Lowe describes a method for 3D object recognition which is based on the use of SIFT descriptors (see Section 1.4.3). After the extraction and the description of local features of the image, they perform best-bin-first search, which is a modification of the $k-d$ tree algorithm, to identify the nearest neighbours [Low99]. In [Low01] a clustering step is added and a probability model is used to achieve more robustness also in cases of little non-rigid deformations. The performances for object recognition are increased since the model can be generalised across multiple viewpoints by linking features between different views. It is also possible to combine features obtained under multiple imaging conditions into a single model view. The object are recognised even when there are few matches between the test image and the model.

In [BMP02], Belongie, Malik and Puzicha introduced a new shape descriptor, the *shape context*, for measuring shape similarity and recovering point correspondence with the aim of recognising some particular categories of objects (silhouette, handwritten digits). We have already described the shape context as a local descriptor in Section 1.4.1, to which we refer for further details. The main drawback of the method presented in [BMP02] is that it assumes that the image has been segmented.

There are techniques, based on the use of models, that look for relationships between the object and its projection to the image and usually these methods recognise objects by visual similarity without attempting any high level analysis of the image. For instance, Obdrzalek and Matas [OM02] describe an approach which aims at combining good qualities of model-based and view-based object recognition techniques: their method is based on the extraction of visual local features of the object and on the construction of a model built upon these features. This approach assumes that the deformation of the image should be at most affine. The features used in [OM02] are the MSER (Maximally Stable Extremal Region) that we have described in Section 1.3.1. Several affine-invariant construction of local affine frames are used for determining correspondence among local image patches. The robustness of the matching procedure is accomplished by assigning multiple frames to each detected region and not requiring all the frames to match. The objects used in the experiments are mainly planar objects coming from the COIL-100 and the SOIL-47 databases.

A Text retrieval approach to object recognition

One of the typical problems of local methods for object recognition is that of combining the different viewpoints and obtaining a global description as a *patchwork* of local descriptions. A possible solution in inspired by text retrieval approach. On this respect, one of the first works exploring the analogy between text retrieval approaches and object recognition is [SZ03] by Sivic and Matas. The authors require a visual analogy of a word and this paper

uses a vector quantifying the descriptor vectors.

Before introducing the analogy to object recognition, let us first briefly recall the main features of many text retrieval methods. Usually text retrieval is based on the following standard steps:

- the document is parsed into words
- the words are represented by their stems, for example *walk, walking* and *walks* would be represented by the same stem *walk*
- a stop list is used to reject very common words.

The remaining words are then assigned a unique identifier, and each document is represented by a vector with components given by the frequency of occurrence of the words the document contains [STCW02]. It is also possible to assign weights to the various words, for instance in the case of Google, the weighting of a web page depends on the number of web pages linking to that particular web page. Then these information are organised in a sort of table of content which will facilitate efficient retrieval. A test is retrieved by computing its vector of words frequencies and returning the documents with the closest vectors.

This scheme is the basis for the analogy between text retrieval and visual descriptors matching described in [SZ03]: the regions which are used in the paper are of two kinds (Shape Adapted and Maximally Stable) to have a complete and complementary description of each frame. The descriptors used are the SIFT. The regions extracted are tracked in the video sequence, and those which do not survive for more than three frames are rejected. Then, after a clustering procedure, the retrieval is performed on the basis of an index created by the set of these descriptors. The experimental evaluation is performed on a group of frame extracted by a movie with the aim of matching scene using the descriptors referred to as *visual words* [SZ03].

Bags of keypoints for object categorisation

Another approach to image classification strongly inspired by *text classification* is the so called *bags of keypoints* [CDFB04]. This approach has been devised to solve image categorisation problem. Let us briefly describe the scheme of this approach: after the detection and the description of image patches, a vector quantisation algorithm runs with the aim of assigning patch descriptors to a set of clusters called *the visual vocabulary*. Then the bags of keypoints are built counting the number of patches assigned to each cluster. The last step is classification of the image: the bag of keypoints is used as a feature vector and all the data are represented with respect to it. A learning from examples procedure is used

to determine which category or categories to assign to a new image. In the categorisation step two different algorithm are compared: Naïve Bayes and Support Vector Machine. The experiments performed in the paper use images belonging to a database of seven classes: faces, trees, cars, buildings, phones, bikes and books. The results demonstrate that SVM are clearly superior to Naïve Bayes classifier.

In [FSMST05] the authors propose a method for improving the bags of keypoints approach using Gaussian Generative Mixture Models. The aim of this work is that of include the keypoint identification in the learning process to ensure that discriminative distinctions are preserved while irrelevant ones are suppressed. The second idea at the basis of this improvement is that of considering the histogram representing the set of input feature vectors as an estimate of their density distribution. Thus improving this density representation to represent better the input feature set will improve classifiers performances. A novel idea in [FSMST05] is to model each set of vectors by a probability density function (PDF) and then in the SVM use a kernel defined over the PDFs.

3.2.2 Geometric, spatial and temporal constraints

The main problem with local methods is that, while observing minute details, the overall appearance of the object may be lost. Also, small details are more likely to be common to many different objects (this feature has actually been exploited in [TMF] to design efficient multi-class systems). For this reason local information is usually summarised in global descriptions of the object, for instance in codebooks [CDFB04, LMS06]. Alternatively, closeness constraints can be used to increase the quality of matches [FFJS06]. Thus in this section we will briefly introduce some methods which combine local descriptors with spatial, geometric or temporal constraints.

In the real world, there exists a strong relationship between the environment and the objects that can be found in it: for instance in a street we are looking for cars and pedestrians while in a room we look for chairs but not for trees. Human visual system uses these relationships for facilitating object detection and recognition [OTSM03, Tor03]: thus we can say that adding geometric and global information and using the context can improve object detection and recognition performances. Then, in this section we briefly review some of the methods which add geometric and global information to the recognition based on the use of keypoints.

In [FPZ05] the authors present a "parts and structure" model for object categorisation. The model is learnt from example images containing category instances, without requiring segmentation from background. The model obtained is a sparse representation of the object and consists of a star topology configuration of parts modelling the output of a variety of feature detectors. This star model has the good qualities of being translational and scale

invariant and it is used both in learning and in recognition steps.

Another approach to object categorisation based on the use of a model is introduced by [LLS04]. The authors present a technique based on the use of an Implicit Shape Model, computed for each class of object, based on a *codebook* or vocabulary of local appearance that are prototypical for the object category. The model uses also a spatial distribution which specifies where each *codebook* entry may be found on the object. In other words, this approach is reminiscent of the visual vocabulary, but it has the add that it keep also spatial information. The experiments are carried out on a database of cars and cows images and the results show good performances.

In [RSSP06] the authors associate geometric constraints to different views of the same patches under affine projection. This lead to a true 3D affine and Euclidean model from multiple unregistered images without the need of any segmentation. This paper is based on the idea that the surface of a solid can be represented by a collection of small patches, their geometric and photometric invariants and a description of their 3D spatial relationships. While the invariants provide an effective appearance filter for selecting promising match candidates in modelling and recognition tasks, the spatial relationships achieve an efficient matching algorithms for discarding geometrically inconsistent candidate matches. It is interesting to notice that for the appearance-based selection of potential matches [RSSP06] exploits also the use of colour descriptor in association with SIFT: it is shown that when two very similar patches have different colours their SIFT descriptors appear almost identical. Thus colour histograms can be used to discriminate among patches with a similar pattern but different colours.

The method proposed in [FTG06] is based on a matching procedure that no longer relies solely on local keypoints. The first step is to produce an initial large set of unreliable region correspondences. The aim, here, is that of maximising the number of correct matches, at the cost of introducing many mismatches. Then the method explores the surrounding image areas, recursively constructing more and more matching regions. This process covers the object with matches, and simultaneously separates the correct matches from the wrong ones: during an *expansion phase* surrounding regions are explored and during a *contraction phase* incorrect matches are removed. In other words once a number of initial correct matches are found, this methods *looks around* them trying to construct more. This method succeeds in covering also image areas which are not interesting for feature extractor thus deciding the object identity is based on information densely distributed over the entire portion of the object visible in the test image. To integrate the information coming from multiple views of the object, the authors introduce the concept of *group of aggregated matches* or GAM, which is defined as a set of region matches between two images, which are distributed over a smooth surface of the object. The use of GAM increases the discriminative power of the technique, but, as a drawback, the method proposed in [FTG06] is computational expensive and the good results obtained for textured areas are

not possible for uniform objects, for which is better to combine the extraction of region with the detection of edges.

In their work [Gra04, GB05], Grabner and Bischof propose an approach for extracting discriminative descriptions of 3D objects using spatio-temporal information. Similarly to [SZ03] they extract local features that are tracked in image sequences leading to local trajectories containing dynamic information about the evolution of the descriptors. These trajectories are evaluated with respect to their quality and robustness and finally each of them is assigned a single local descriptor from a key-frame in order to obtain an object description.

Their descriptors are based on the assumption that local features are always distinctive patches of an object and that this object is made of piecewise planar patches. According to this idea, the descriptor is selected as a candidate to represent all the patches belonging to the trajectory. The selection of the candidate is based on the assumption that the patches change according an affine transformation, thus the patch falling in the middle of each trajectory is the best compromise to describe all the other patches, from the head to the tail of the trajectory. Another possible choice of the descriptor is to compute the average of all the descriptors in the trajectory. To test the models obtained using these descriptors they use a voting procedure based on nearest neighbour. The dataset[3] of objects that they use for their experiments contains several planar objects with a textured pattern, as for instance different types of boxes. Their results demonstrate that the robustness of the descriptor is effective to recognise objects, even if the background is highly textured and the objects are partially occluded. But it is worth to notice that for general objects the heuristics that they use to locate the most stable part of a trajectory is not necessarily met.

3.3 Spatio-temporal features

In this section we introduce the spatio-temporal features around which we base our object recognition method. We start recalling the main drawbacks of local approaches. When looking at minute details, an observer may loose the overall object appearance, therefore small details are more likely to be common to many different objects. Another problem that affects local approaches is that when building a model considering many minute details the size of the model can be huge. These observation lead to some of the methods discussed in Section 3.2.2.

Let us consider the case in which an object is described by a dense set of views capturing smoothly the appearance variations. Then we can use temporal information to integrate

[3]The image dataset is available at http://conex.icg.tu-graz.ac.at

the appearance and spatial description provided by local descriptors. Thus, our aim is to exploit the information hidden in an image sequence to obtain a complete description of the object represented in the sequence itself. Our work is motivated by the fact that an image sequence does not just carry multiple instances of the same scene, but also information on how the appearance of objects evolves when the observation point changes smoothly. Since in many applications image sequences are available and often under exploited, our aim is to fill this gap.

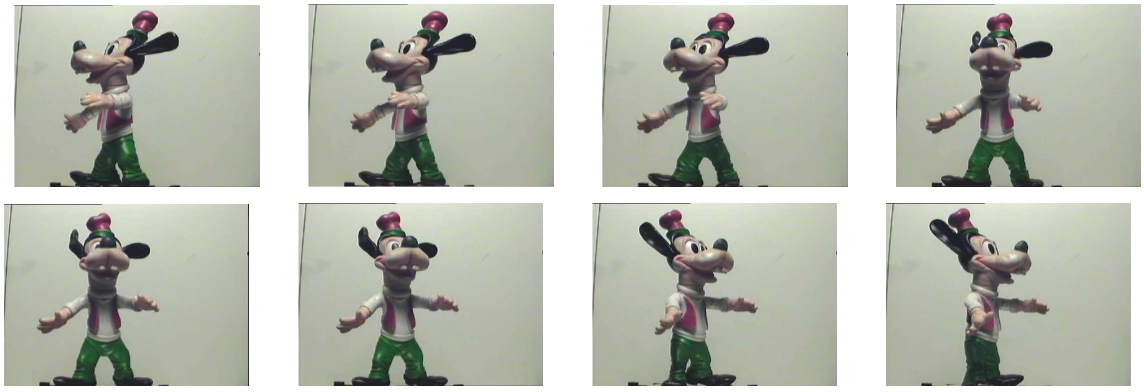


Figure 3.1: The object *goofy* seen in a sequence capturing its variation in space and time.

If we think about a robot grasping an object and observing it from different points of view it is natural to assume that there are local descriptors which can be spatially and temporally combined. Moreover we can also build a description which is evolving incrementally as the time goes by. Another example in which it is possible to use temporal description, is when the camera moves and observes an object from different points of view. Figure 3.1 shows an example in which the object *goofy* is seen from different points of view. These are the applications that we have in mind while building our approaches: in such cases we need to model the spatio-temporal changes of object appearance.

Our approach can be briefly described as follows:

- identification of a set of keypoints for each image of the sequence
- description and tracking of the keypoints to capture their variation in time and space
- creation of a model to describe the variation of keypoints appearance
- object recognition based on these view-based models.

Thus, next two sections are devoted to explain and give details about the first three steps which are related to the process of building a model for an image sequence. Finally Section 3.4 will describe two different approaches to object recognition.

3.3.1 Features invariants in space and time

We base our recognition approach on the use of local descriptors of images: for each image of the sequence we extract keypoints and describe them using SIFT by Lowe.

As for the choice of the feature detection algorithm, we first considered the original work by Lowe and used DoG features. Then we approached the studies by Schmid and her co-workers [SMB00, MS04b, MS04a] (see Sections 1.3.2 and 1.4.2) were it is shown that Harris corners are more reliable that other features in terms of stability. For more comments on this we refer the reader to Chapter 4.

Despite these studies, it is not yet clear if one keypoint is best suited for object recognition or for other tasks [Lei04]. Our experience lead us to the conclusion that the exact choice of keypoint detectors is not crucial: each keypoint can be more suitable to describe a given image in relation with the qualities of the image itself. Instead it is worth to remember the importance of robustness of descriptors to image variations such as scale, illumination and orientation changes. On this respect, there is a general belief that SIFT descriptors are a good choice (see Section 1.4.3).

Keypoints detection and description

After the detection and the description of keypoints, each of our descriptors contains the following information about the keypoint k:

$$(\mathbf{p}_k\,,\mathbf{s}_k\,,\mathbf{d}_k\,,\mathbf{H}_k)$$

where $\mathbf{p}_k$ is the position in the space, $\mathbf{s}_k$ refers to the level of scale and $\mathbf{d}_k$ to the principal direction of the keypoint. $\mathbf{H}_k$ contains local orientation histograms around the keypoint. We will use the first three elements for tracking the keypoints while the orientation histograms $\mathbf{H}_k$ will be used for computing the similarities. It is important to remember that scale and main orientation are also implicitly used for computing the similarities, as $\mathbf{H}_k$ is built on an image patch centred at the keypoint position, and scaled and rotated according to scale and main orientation.

Tracking keypoints

Since we want to model the variation of keypoints in time and space, we track them and look for a descriptor capable to express the evolution of each keypoint along a sequence. In Section 3.2.2 we have briefly described the approaches proposed by Grabner and Bischof in [Gra04] and by Sivic and Zisserman in [SZ03]: they are both technique for object recognition based on tracking of keypoints. Instead of using a correlation based tracker as the KLT adopted in [Gra04] we choose dynamic filters with prediction capabilities, in particular we start considering the Unscented Kalman Filtering which falls in the class of Particle filters and it is designed for dealing with non linearity of the system. In our case, the unknown state is defined by the position in space and scale and the principal direction

of each keypoint $\mathbf{x}_k = \{\mathbf{p}_k, \mathbf{s}_k, \mathbf{d}_k\}$.

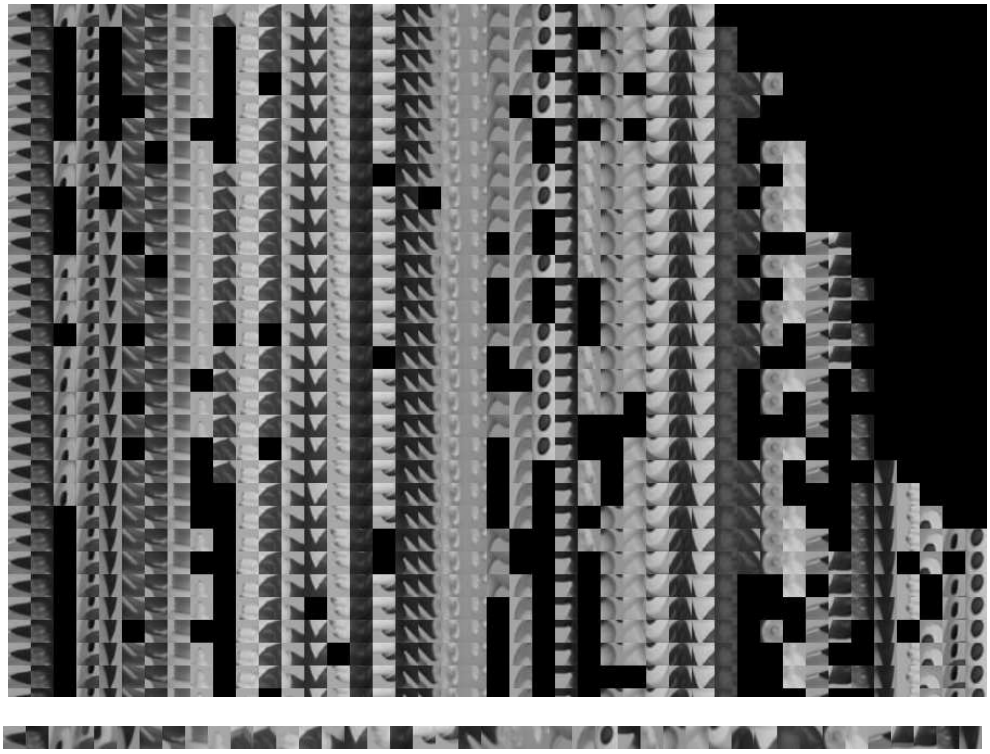


Figure 3.2: Several *trains of keypoints* extracted from a sequence of *dewey*. To obtain this representation we extract a 21×21 patch around each keypoint of each trajectory, then we compute an average of the grey levels of each patch. The average patch, computed for each trajectory, is shown in the line of 21×21 squares at the bottom.

The choice of the algorithm depends on the characteristics of the model, for instance it can be linear or not, Gaussian or not. As the dynamic equation of the keypoint locations is non linear, the Kalman filter can be not appropriate for the estimation of the filtering distribution. Recently, Particle Filters have been extensively used to deal with the non linearity of a system [AMCF05]. These methods are very interesting because they enable an accurate approximation of the distribution of keypoints even if this latter is highly multimodal. However, their interest can decrease if the system into consideration is weakly non linear as it is in the case that we consider here. In this case, the use of an algorithm that supposes a Gaussian approximation of the filtering density can be both sufficient and efficient. The simplest approach consists in linearising the model equations, that leads to the Extended Kalman filter [JU97]. However, the accumulation of the errors due to successive liberalisations may cause the divergence of the tracker. In order to avoid this problem, the Unscented Kalman filter [WvdM00] proposes to describe the Gaussian approximation of the posterior density by a carefully chosen weighted sample points. These

points capture the mean and covariance of the approximation accurately to the third order and are propagated through the non linear system. Even if in our first works we have used an Unscented Kalman Filtering [ADOV06], we have seen experimentally in our recent work that, when the sequence is dense enough, a Kalman filtering accomplishes the need of a stable tracking and it has an easier implementation [DNOV07].

At the end of this step we have a set of trajectories or *trains of keypoints*. Figure 3.2 shows the patches corresponding to the extracted keypoints and the averages of the grey levels of the patches computed along each *train of keypoint*. This representation is used to show the robustness of the trajectories obtained with the tracker (in this case we have used Unscented Kalman). This figure is meant to give a visual impression of the keypoints evolving in time. In practice we will not use image patches directly.

Cleaning trajectories

At the end of the tracking procedure, we have many trajectories that can be of variable quality: we may have robust and stable but also noisy and wrong trajectories. Thus there is the need of applying a cleaning procedure that eliminates noisy or wrong trajectories: first, we compute the variance of the scale and of the principal direction of each trajectory. Then, trajectories with a high variance are further analysed in order to check whether they contain abrupt changes that could be caused by tracking errors. To do so, we perform a SSD correlation test between the first gray-level patch of the trajectory and the subsequent ones. In the presence of an abrupt change, the trajectory is discarded. Figure 3.3 compares two trajectories: the dashed one refers to a good feature, whose appearance varies slowly and smoothly in time; the solid one refers to an unstable trajectory, containing tracking errors (a visual inspection of the gray-level patches confirms this hypothesis).

3.3.2 The visual vocabulary

The content of an image sequence is redundant both in space and time. Consequently we obtain compressed descriptions for the purpose of recognition, extracting a collection of trains of features and discarding all the other information. We call this collection *model* or *vocabulary* of the sequence. We do not keep any information on the relative motion between the camera and the object, as it is not informative for the recognition purpose.

Description

More in detail, all the keypoints linked by a tracking trajectory, or train, belong to the same equivalence class. We call *time-invariant feature* $\mathcal{V}_i$ the average of all local orientation histograms $\mathbf{H}_k$, with k running through the train of keypoints, and we use the time-invariant feature as a delegate for the train. Average values are good representatives of the original keypoints as the tracking procedure is robust and leads to a class of keypoints with

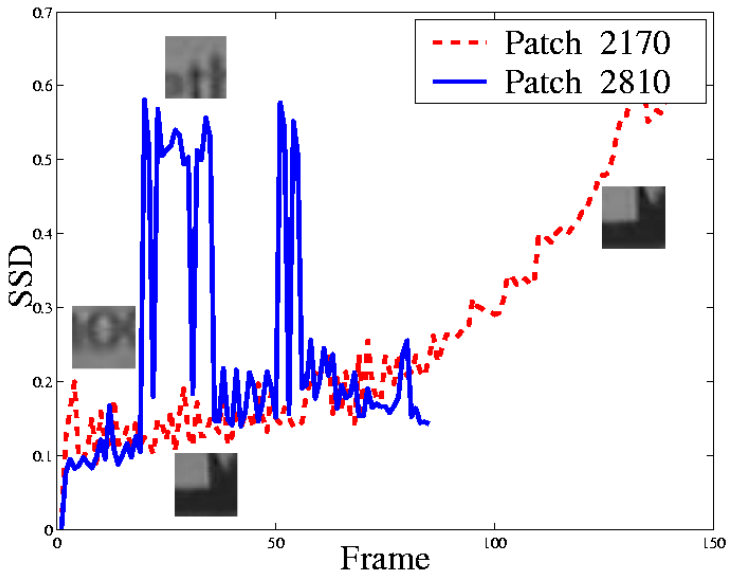


Figure 3.3: The SSD computed between the patch extracted in the first frame and the following ones. The red line shows the SSD trend for a stable patch while the blue one shows a patch that have an abrupt change due to an error in the tracking.

a small variance (see Figure 3.2. [Gra04] uses trajectory centroid to select the delegate for each trajectory. This choice is not convenient for our case since it gives the best results with planar object, while our aim is that of representing also keypoints of 3D objects. Thus we decide to use the average as a delegate for the trajectory. Being an average, some histogram peculiarities are smoothed or suppressed, but we will discuss this issue in the next section. It is worth noticing that the descriptors which are too different from the average are discarded to improve the robustness of the descriptor and to eliminate errors due to the tracker.

At the end of this process we have obtained a time-invariant feature that is described by

- a spatial appearance descriptor, that is the average of all SIFT vectors of its trajectory (the *time-invariant feature*);
- a temporal descriptor, that contains information on when the feature first appeared in the sequence and on when it was last observed.

The set of time-invariant features form a *vocabulary for the object*: $\mathcal{V} = \{\mathcal{V}_1, \ldots, \mathcal{V}_N\}$. The vocabulary $\mathcal{V}$ is also the model for the object. A visual representation of the information contained in one object vocabulary is shown in Figure 3.4.

Figure 3.4: A visual representation of *goofy* model.

We have shown that the tracking phase allows us to obtain a compressed description for the purpose of recognising objects. But there are some cases in which, as we have mentioned above, the *time-invariant feature* tends to oversmooth the description. Thus in the next section we will analyse the effects of changing the length of the train of keypoints.

The importance of the temporal range size

In the feature extraction phase, features robust to viewpoint variations are preferred, because we think that a keypoint present in most part of the sequence is more representative than others which are noisy and unstable. Therefore we extract long trajectories. But, if we use the entire long trajectory to compute a descriptor [ADOV06] then we risk to oversmooth the information contained in the trajectory.

On this respect it is important to notice that choosing to average a long sequence of descriptors the model is more general, but it can happen that we loose information: since long trajectories are the result of observing a feature on a long period of time (and possibly a high range of views). Thus descriptions generated from these long trajectories tend to oversmooth the appearance information, and may lead to a high number of false positives. This is especially true if the object is deeply 3D and its appearance changes dramatically over the sequence.

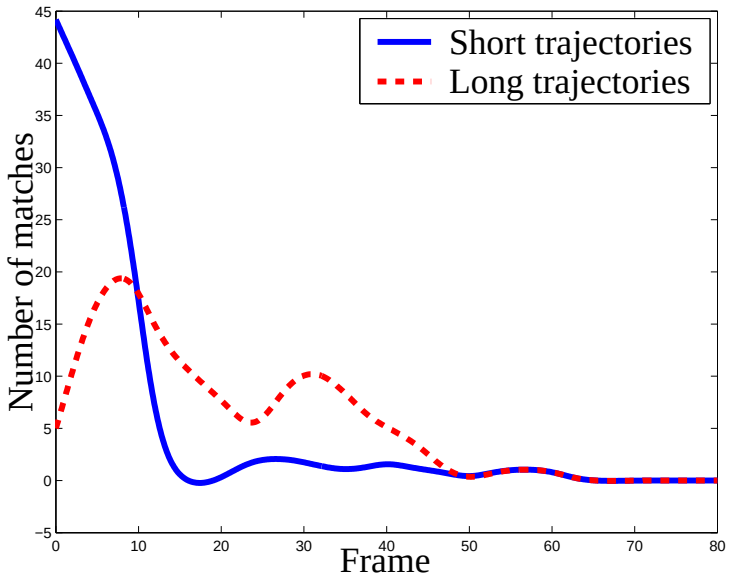


Figure 3.5: Matches between training and test sequences for different choices of the length of the train N: solid line is for $N = 10$, dashed line is for $N = 50$.

To this purpose we apply, before computing the time-invariant feature descriptions, a cutting phase that cuts a trajectory into many sub-trajectories of length N. The choice of N is not crucial, and common sense rules may be applied: if N is too small we loose efficiency and compactness, as the model becomes very big. If N is too big, the information within a feature is oversmoothed. Figure 3.5 shows the effect of changing N: a small training sequence containing a small range of viewpoints is compared with a test sequence of the same object. A similar viewpoint of the training appears in the first part of the test video. The solid line shows the matches between training and test models obtained with N=10, the dashed line is for N=50. The smoothing effect as N grows is apparent. To one extreme $N = 1$ is equivalent to discard temporal information and perform image matching, $N = L$ where L is the length of the trajectory is equivalent to avoid cutting.

Space occupancy issue

This approach allows us to obtain a compact representation of the object contained in the sequence. Indeed the representation can be extremely compact in the case we decide to have smoother descriptors or it can contain some more details if we decide to have shorter trajectories for the creation of the descriptors. The advantage of this representation is apparent because the space occupancy of our model is limited. Suppose that the keypoints detected in each frame are approximately 300. Then, if every keypoint of the sequence participates to build the model, in the average case (a sequence of length 250) the model

is made of approximately 75000 features. Our approach grants to obtain a compact representation of typically 1500 time-invariant features. A compact model for the object is fundamental when recognising many different objects, even if another problem that we have to consider is that of representing the image with respect to a model. Next section is devoted to describe a possible representation of images on the base of our model.

3.4 Time-invariant features for object recognition

Representing images with respect to a model is a problem which depends on the classification method to use. In this section we discuss two different approaches to recognition that we devised and analysed. The first one is based on using similarities computed between novel images and visual vocabulary. The choice was guided by a first assumption we made on the classifier to use (a binary SVM classifier in this case): training and test images are both compared with respect to the visual vocabulary. The second approach avoids the complexity of the representation at the basis of the first approach and explore a novel technique for matching models of sequences. The main constraints exploited in this case are determined by temporal cooccurences of visual words.

3.4.1 Bags of time-invariant features

Our approach is motivated by the choice of using SVM for recognising 3D objects and thus is based on the concept of visual vocabulary. Indeed the visual vocabulary is one of the classical ways to overcome the problem of variable-length descriptions in the learning from examples framework. In several studies [CDFB04, SZ03], images are represented using frequency histograms or vectors indicating the number of occurrences of features with respect to vocabulary entries. These representations tend to be sparse and in our case, since we aim at keeping temporal coherence distinguishing features originated by different frames, we end to obtain very sparse representation. Thus we keep similarity values to strengthen our representation: indeed when we compute similarities of visual words with respect to the visual vocabulary we compute an explicit mapping in a simpler feature space.

Thus next paragraphs will be devoted to the description of our approach based on computing similarities among a novel keypoint descriptor and the time-invariant feature of the visual vocabulary. The temporal information is used mainly during the phase in which we built the model. It is worth to notice that at this stage the similarity is computed between two slightly different kind of data: the visual vocabulary with its time-invariant features and novel features extracted from test frame. Therefore we will discuss about:

- the choice of the similarity measure that will be used for the comparison of descriptors

with time-invariant features

- the creation of the representation based on this similarity measure.

Finally we will give some details on the use of Support Vector Machines as classification method.

The choice of the similarity measure

For our choice of representation of images, a crucial point is to decide how to compare the local orientation histogram of a keypoint with the average orientation histogram of a time-invariant feature. Then, a comparison criterion for histograms seems to be appropriate. We consider

1. Euclidean distance D,
2. Chi-square distance χ^2
3. Kullback-Leibler divergence $\mathcal{K}$,
4. Histogram intersection $\cap$ [SB91].

Figure 3.6: Example images used for the comparative analysis of similarity measures.

Since the first three are distance measures we will use the exponent version:

$$D_{exp} = \exp(-D)\,,$$

$$\chi^2_{exp} = \exp(-\chi^2)\,,$$

$$\mathcal{K}_{exp} = \exp(-\mathcal{K})\,.$$

Also, since the keypoint descriptions may not be normalised, instead than measure 4 we will use

$$\cap_{norm}(H, H') = \frac{\bigcap(H, H')}{\bigcup(H, H')} = \frac{\sum_{i=1}^{n}(\min(H_i, H'_i))}{\sum_{i=1}^{n}(\max(H_i, H'_i))}. \quad (3.1)$$

If the histograms are normalised, this similarity measure is equivalent to histogram intersection.

Let us reason on what we would ask to a similarity measure: high scores on similar keypoints, low scores on different keypoints. Figure 3.7 shows the results of comparing two similar images (Figure 3.6, left and centre) and two very different images (Figure 3.6, centre and right), with the four similarity measures. The plots are obtained as follows: for each keypoint of the first image we compute the highest match value with respect to keypoints of the other image. The results show that Chi-square returns uniformly high scores in both cases. The best compromise between intraclass and interclass keypoints is obtained with normalised histogram intersection $\bigcap_{norm}$, which will be used in the rest of the experiments.

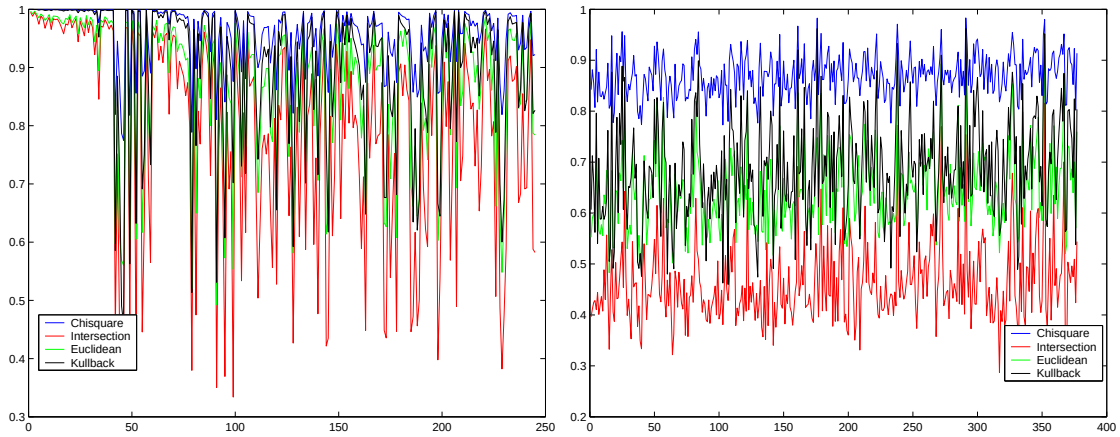


Figure 3.7: Match values obtained comparing 2 images with the 4 similarity measures. Top: results from two similar images (Fig. 3.6 left and centre). Bottom: results from two different images (Fig. 3.6 centre and right). On the x axis are the indices of keypoints in the first image, on the y axis the corresponding match values with the most similar keypoints of the second image.

Building the representation

An image F_i, after we extract local interest points, can be seen as a collection of keypoints $F_i = \{\mathcal{F}^i{}_1, \ldots, \mathcal{F}^i_M\}$, where M will vary. The vocabulary helps us to avoid the problem of variable length representations: each image F_i is represented with a vector R_i of length N.

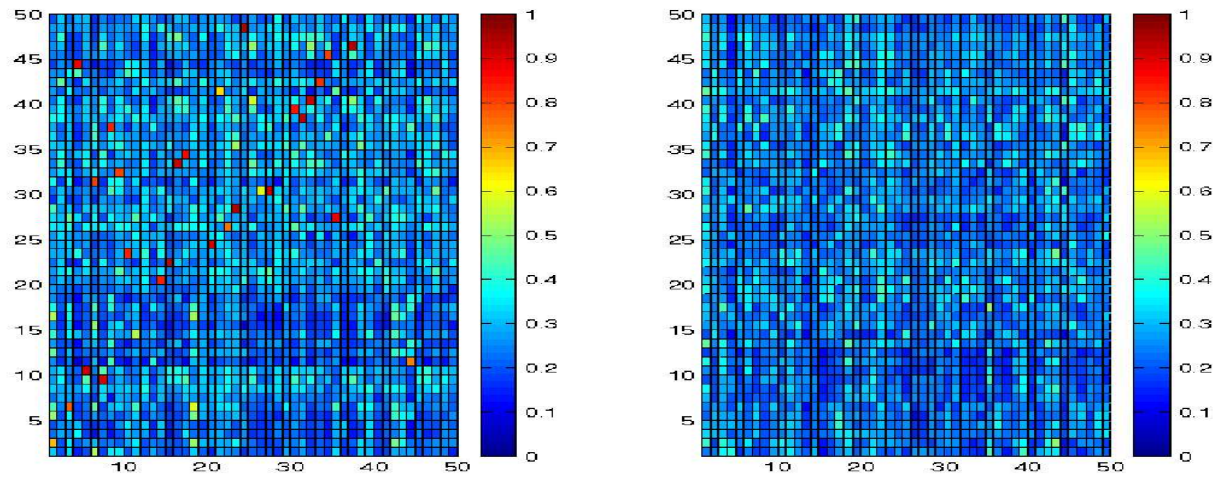


Figure 3.8: Similarity matrices obtained comparing the vocabulary of object $\mathcal{O}$ with one image of object $\mathcal{O}$ (on the left) and one image of another object (on the right). On the rows: the image keypoints, on the columns: the time-invariant features (see text).

Each entry k of R_i carries the contribution of the keypoint $\mathcal{F}_j^l$ most similar to $\mathcal{V}_k$, if there is one. Possible choices on how to build R_i include:

1. **Binary entries**, with $R_i^k = 1$ if there exist a keypoint $\mathcal{F}_j^l$ closer to $\mathcal{V}_k$ than a threshold.
2. **Real value entries** describing the degree of similarity between $\mathcal{V}_k$ and the most similar keypoint $\mathcal{F}_j^l$.
3. **SIFT entries**, with $R_i^k = \mathcal{F}_j^l$, where $\mathcal{F}_j^l$ is the most similar keypoint to $\mathcal{V}_k$.

Our image representations will be based on choice 2, as it is the best compromise between effectiveness and simplicity. It is worth mentioning that choice 3 corresponds to an explicit mapping of the intermediate matching kernel [BTB05].

We compute the similarity values between all keypoints of image F_i and all time-invariant features of the vocabulary $\mathcal{V}_k$. An explicit computation would lead to a similarity matrix as the ones shown in Figure 3.8. The final description is obtained by taking the maximum values column-wise. While finding the association between $\mathcal{V}_k$ and $\mathcal{F}_l^j$, keypoint that appear similar to more than one time-invariant feature are penalised. Figure 3.8 considers a vocabulary for object $\mathcal{O}$ and includes the comparison with one image of object $\mathcal{O}$ (on the left) and one image of another object (on the right). On the left matrix are clearly visible the high match values corresponding to the most similar keypoint.

SVM classification

The process of classification is usually divided in two steps: the first one is the *training* phase which serves to the classifier to learn the model of the object from a set of images containing the object, the second step is the *test* phase during which the classifier is asked to recognise if an unknown image contains the model. Classification problems can be faced with techniques coming from the area of *learning from examples*: it can be seen as a problem in which it is necessary to estimate an unknown functional dependency given only a finite (possibly small) number of instances [Vap98].

Among the methods for classification based on learning from examples, we focus our attention on Support Vector Machine for binary classification, whose basic idea is that of finding a hyperplane in the feature space associated with the kernel classifying correctly most of the training points. The hyperplane is found as the optimal trade off between making the smallest number of misclassifications and maximising the margin, or the distance of the closest points from the hyperplane. For further details we refer to [Vap98].

In the set of our experiments, for each object we build the vocabulary and then represent the training data (both positive and negative) with respect to it. We then train a binary SVM classifier with a histogram intersection kernel [BOV02], that was proved effective on a number of applications and does not depend on any parameter. Preliminary results showed that its performances are superior to standard kernels for the problem at hand.

Since we are using binary classifiers it is required a training phase during which the classifier learns to distinguish images of the object from images not containing the object. This requires that we decide which are the images representing negative examples for each object. In our case we decided to use as a group of images containing all the other objects of our database and a set of images of different kind: buildings, rooms, landscapes, cityscape, people. The test set is made of a collection of images: we acquire novel sequences and we use every frame as a test image. Thus each test image has to be represented with respect to the visual vocabulary exactly as we have done for training set.

To deal with the multi-class nature of the problem we use a one-against-all approach. Thus we train as many classifier as the objects and we count how many times the classifier recognise the object. Therefore we obtain confusion matrices which show the recognition rates for each object and put in evidence which are the most similar objects that can be confused. It is worth noticing that each model is object-centred, in other words each classifier depends on its visual vocabulary.

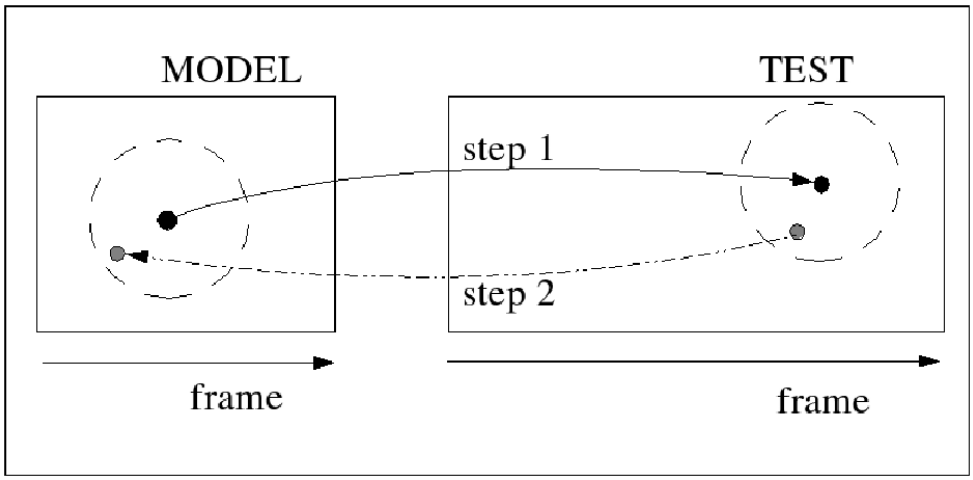


Figure 3.9: Description of the two-stage matching for comparison of sequences models.

3.4.2 Matching of sequences models

We now move on describing a direct recognition method based on a multi-stage matching that exploits spatial and temporal constraints. The main motivation of this approach with respect to the SVM-based is to obtain a simpler method that exploits at length the complexity of our description. More detailed comments and comparisons will be made in Chapter 4.

We devise a two-stage matching which exploits spatial and temporal information belonging to each sequence model. An important novelty of this approach with respect to the SVM-based one, is that now we use sequences both in training and in test. Thus, we avoid the asymmetry due to comparing a descriptor computed as an average of many descriptors with one raw feature descriptor.

Let us consider a test sequence represented by a model T, and a training sequence, represented by an object model M. The problem we address is to see whether the test model T contains the object M. Our two-steps matching strategy, visually described in Figure 3.9, is performed as follows:

1. We perform a nearest-neighbour between each training model and the test model. This procedure gives us initial hypotheses for the presence of known objects in the sequence. For each feature in M, we use histogram intersection [DAOV06] to check whether T contains a similar feature. We set a minimum similarity threshold (usually equal to 0.6) to obtain a collection of robust matches (f_M, f_T). This is the `step 1` shown in Figure 3.9.

2. This stage, based on enhancing spatio-temporally coherent matches, helps us to confirm or to reject each hypothesis. We use spatio-temporal constraints and perform a reverse matching procedure from the test to the training model. This is the `step 2` in Figure 3.9.

    - We detect the subsets of training and test models containing most matches: I_M and I_T. In this way we get a raw temporal localisation of the object in the test sequence, but also hints about its appearance, since I_M marks a particular range of views. In order to refine this detection, we reject matches which are spatially isolated.
    - We increase the number of matches around the matches obtained previously. Let us consider a match (f_M, f_T): we can specify two sets, F_M and F_T, containing features (of M and T respectively) appearing together with f_M and f_T for a chosen period of time (in our experiments 5 frames). A new matching step is then performed between F_M and F_T considering a lower similarity threshold: a pair $(\tilde{f}_M, \tilde{f}_T)$ is a new match if the features are spatially close to f_M and f_T, respectively. This new stage is repeated for each pair (f_M, f_T) belonging to I_M and I_T intervals. The incorrect recognition hypotheses are rejected while the correct ones are enriched by further matches in the surrounding areas. In Figure 3.9 the dotted circles bound the regions in which this second search is performed.

This procedure increases the number of high scores when groups of features appear both in training and test. This is more likely to happen if training and test models contain features coming from the same object. Section 4.4 will present experiments that show that this two-stages strategy is especially useful when the acquisition setting for the training sequence is very different from the test sequences. Thanks to the compact representation of visual information the matching phase is efficient even when the sequence is long and there are several detected keypoints.

3.5 Discussion

In this chapter we first introduced the state of the art of object recognition methods which exploit local descriptors for images. Then we proposed an approach to model sequences which is based on the tracking of keypoints and we devised two different way to use the model obtained with the aim of recognising 3D objects. The former of these methods is based on SVMs, while the latter exploits temporal co-occurrences to perform a two-stage matching process. We have already noticed that the SVM-based technique performs an asymmetric comparison between raw descriptors and time-invariant features.

Then it is worth noticing that each classifier is object-centred and when the number of objects increases each novel test need to be represented with respect to all the training models. This makes this approach complex and lead us to design an approach based on a simpler matching. Both these reasons motivate our choice to study and propose a simpler matching procedure that exploits spatio-temporal constraints.

The first of our approaches has been inspired by the bags of keypoints technique [CDFB04] for what concerns the classification aspects. Having said so, the differences between our method and bags of keypoints are many, since we had to devise a different representation that allowed us to keep the image sequences. The vocabulary approaches are a way of dealing with *variable-length descriptions* in the statistical framework. Another approach, computationally more expensive at run time but more elegant from the theoretical viewpoint, is to adopt *local kernels* [WCG03, BTB05]. In particular, as shown in [VO07], our representation based on similarity measure is an explicit mapping of the intermediate matching kernel [BTB05].

It is worth to notice that our time-invariant features are reminiscent of a method presented by Laptev and Lindeberg for detecting spatio-temporal events in video sequences [LL03]. Their approach is based on the detection of local structures in space-time where the image values have significant local variations in both space and time. Then the approach consists of building descriptors on these events and classifying novel events on the base of these descriptions. Thus the local spatio-temporal features for action recognition are based on finding sharp changes in the temporal direction that are distinctive of a given motion and this is a strong difference with respect to our time-invariant features.

With respect to approaches based on tracking of features, it is worth to remember the method proposed by Grabner and Bischof in [GB05]. The main difference among their approach and our method is that our modelling and matching method exploits temporal coherence *both in training and test*. For this reason it fits naturally in the video analysis framework. Moreover the descriptors used in Grabner and Bischof's work cope only with affine transformations of the patches associated to keypoints: they are capable to recognise mainly planar objects. Instead, as we will show in Chapter 4, our models are able to describe also 3D objects with a complex three dimensional structure. In spite of the redundancy of the data that we use, the descriptions we obtain are compact since they only keep information which is very distinctive across time.

Chapter 4

Experiments

In this chapter we report an extensive analysis of the view-based model presented in Chapter 3. The experiments are devised to investigate some of the typical object recognition problems. We will show results using different methods and approaches.

4.1 Introduction

In the previous chapter, we have described how to obtain a model for object recognition. As we have previously said, our aim is to recognise a three dimensional object using a dense set of views describing smooth changes in the appearance of the object. Thus, our idea is to exploit temporal information for integrating spatial and temporal description provided by local descriptors. For the sake of discussion we will briefly resume the main underlying concepts regarding the creation of the view-based models described in Sections 3.4 and 3.3.1.

Given a sequence of images showing an object seen from different points of view, the process to obtain a view-based model is described by the following procedure:

Keypoints detection The first step is the identification of a set of keypoints for each image of the sequence. Among the many keypoints described in literature, we have used Harris corners and DoG extrema: our experience lead us to understand that the choice of the keypoint is not crucial for the aim of recognition. Indeed a first set of experiments have been developed using DoG extrema [ADOV06], while a second group is based on the use of Harris corners [DNOV07]: in some cases this second group of keypoints have guaranteed better performances since they are more stable and robust when they are tracked.

Keypoints description Following [MS03], we have chosen SIFT as descriptor of the keypoints. Indeed SIFT have been compared [MS03] with several descriptors of local patches, and the results show that they are the more robust to changes in scale, illumination, orientation and viewpoint variations.

Keypoints tracking To capture smooth variations of the descriptors we analyse their temporal evolution, thus we track keypoints in the image sequence. The tracking algorithms that we considered for our experiments are those belonging to the family of dynamic filters [Kal60, JU97], since they are robust to occlusions and they are capable to cope with temporal detection failures. Even if in our first works we have used an Unscented Kalman Filtering [ADOV06], we have shown in our recent work that, when the sequence is dense enough, a Kalman filtering accomplishes the need of a stable tracking and is easier to implement [DNOV07].

Creation of the visual vocabulary Tracking returns a collection of trajectories, which comprehends keypoints and descriptors, that we call *trains of keypoints*. Then, after the cleaning procedure described in Section 3.3.1, we sum up all the acquired information in form of a view-based model. Thus, our model can be briefly described as composed by

- a *virtual feature*, which is a spatial appearance descriptor, that is the average of all SIFT descriptors of a clean trajectory
- a *temporal descriptor*, that contains information on when the feature first appeared in the sequence, on when it was last observed and which are its coeval features.

For further details on the creation of the model we refer to Section 3.3.2.

3D object recognition The last step of this procedure consists in the recognition phase. On the basis of view-based models computed in previous steps, we can understand if a novel sequence contains or not one of the objects that we have modelled. As we will show in the experiments, we have analysed two different approaches that we have deeply discussed in Sections 3.4.1 and 3.4.2.

This chapter is devoted to an extensive analysis of this general model. In particular we will consider three different approaches to description, modelling, and recognition, based on the same general view-based principle. All these approaches have in common the fact that the object of interest is modelled by a set of continuous views (stored in an image sequence). In the first two cases, at run time, we consider a frame at a time, as for the third method we use continuous views both in training and test. As we will see along the chapter, using similar information both in training and test allows us to achieve a neater recognition procedure and better results.

4.1.1 The set of objects

We have gathered a selection of 3D objects that we will use for all our experiments. We chose coloured matte objects of different complexity in terms of texture and 3D shape. Our selection includes objects which are similar (plastic toys, books, a ring binder, boxes, a coffee-maker, a teapot, a boiler, a telephone, a plastic box of pastels, a teddy bear and a teddy rabbit), but at the same time they are variable enough to represent a possible selection of things of a real indoor environment (or at least typical of a *computer vision laboratory*). It is worth to notice that our objects include textures of different complexity: in general few features are enough to obtain satisfactory results. Figures 4.1 and 4.2 show the set of objects which have been used in the following experiments.

Regardless the recognition method used, care is taken on the construction of the training model. The training sequence is acquired in a neutral but real environment, no segmentation is applied. The background is smooth and homogeneous so that few outlier features will appear in the model. Illumination is natural, when possible. Objects are analysed having them rotating on a turntable while a still camera acquires a video sequence, that will be, on average of a few hundred frames.

This procedure makes the process of acquiring models of objects easier, since the tracking procedure is more robust when the sequence is continuous. Furthermore the evolution of the descriptor is better captured with this continuous observation than from a sample of viewpoint images [Gra04].

We now move on describing our experiments starting from a simple matching between virtual features and test features.

4.2 Preliminary matching

This set of experiments aims at verifying the robustness of virtual features: we simply compare visual descriptors of novel images with the visual words of the models. We base our comparison on the use of a measure of similarity and, according with the results obtained in Section 3.4.1, we use the normalised histogram intersection. Once that the model for an object have been computed, we extract and describe visual words from a novel test image and then we compute the similarity among the set of descriptors. If the similarity among descriptors is higher than a threshold then we consider the match as a good one. The choice of the values for the threshold are based on empirical evaluations.

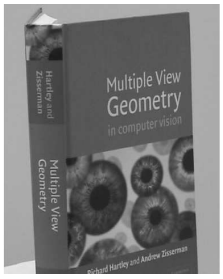
It is worth to notice that this matching process is asymmetric because we are comparing virtual features, obtained averaging many features, with original features extracted from test images.

(a) bambi

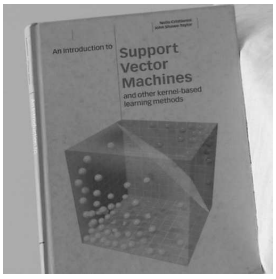
(b) box

(c) dewey

(d) biscuit



(e) bookGeo



(f) bookSvm

(g) dino

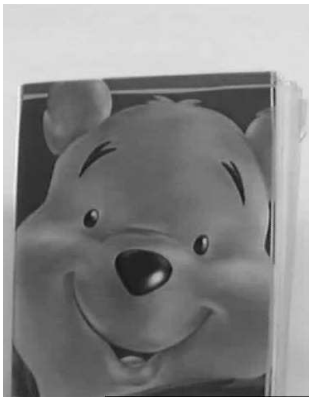
(h) teddy

(i) pino

(j) telephone

(k) goofy

(l) tommy



(m) winnie

Figure 4.1: Some of the objects used in the experiments.

(a) coffee

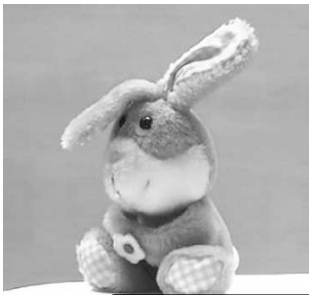
(b) delfina

(c) kermit

(d) eye

(e) donald

(f) scrooge



(g) rabbit

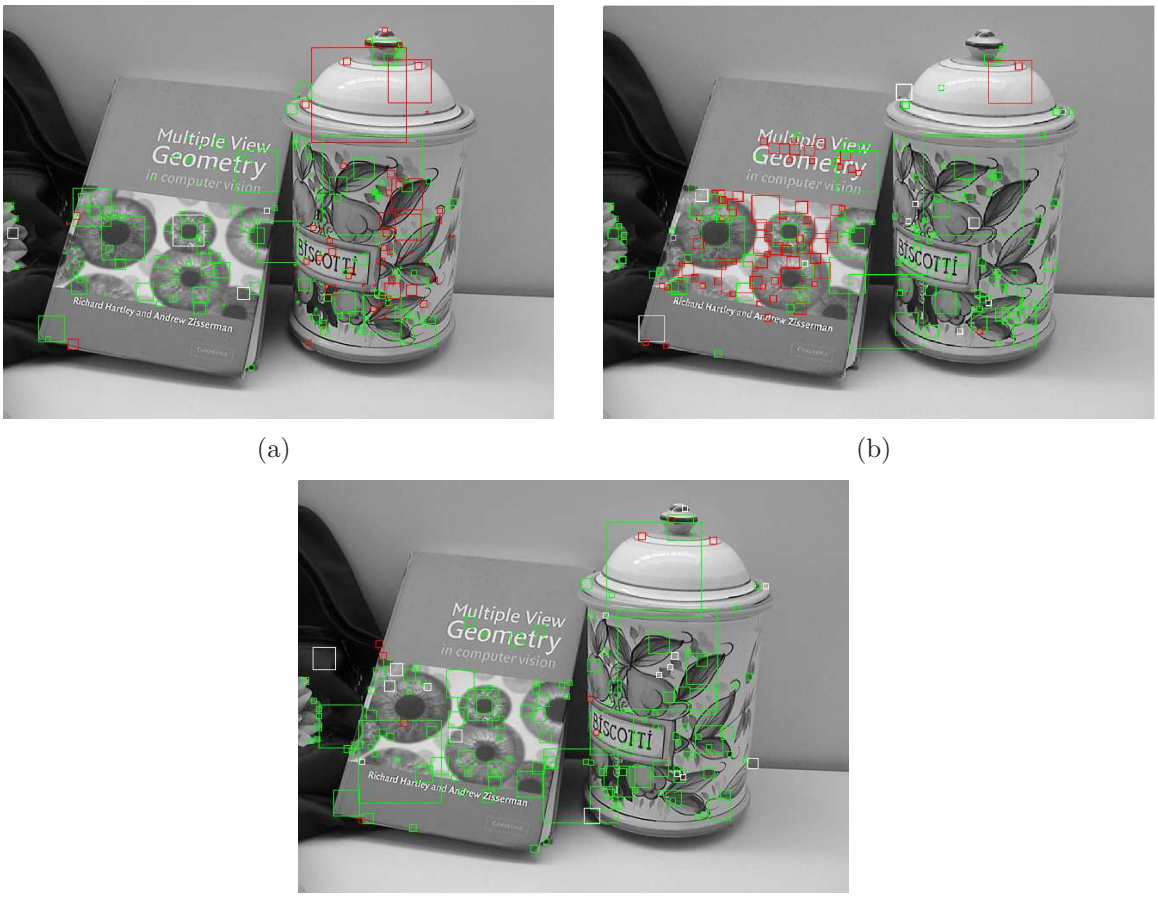
(h) sully

(i) pastel

(j) easyBox

(k) teapot

Figure 4.2: Some other objects used in the experiments.



(a) (b)

(c)

Figure 4.3: A test image is compared with the vocabularies of different objects: (a) biscuits, (c) book and (b) duck. The red matches correspond to the values of similarity higher than 0.6 while the green ones have similarity higher than 0.4.

4.2.1 Similarity of virtual features

We investigate the case of test images with multiple objects. Figure 4.3 shows a test image with two objects and the features that match 3 different object vocabularies. The matches with the higher similarity values are shown in red and it is apparent that they are positioned on the correct object.

Figure 4.4 shows the results obtained when looking for the vocabulary of *bookGeo* on images containing multiple objects. The red matches correspond to the values of similarity higher than 0.6 while the green ones have similarity higher than 0.4. It is apparent that there are many matches appearing on the wrong object, but the matches with similarity values higher than 0.6 appear on the object described by the vocabulary. It is worth to remember

Figure 4.4: Different test images are matched with the vocabulary of *bookGeo*. The red matches correspond to the values of similarity higher than 0.6 while the green ones have similarity higher than 0.4.

that planar object, as for instance *bookGeo*, are easier to recognise since they do not suffer from the parallax effect [TV98].

These images show that the descriptor that we call time-invariant feature has good properties of robustness in association with histogram intersection as similarity measure. Nevertheless it is important to notice that there are many matches with a similarity score higher than 0.4.

We test our model descriptors also for images of the dataset[4] used in [Gra04]: this dataset is relatively simple compared to the one we build, being constituted by many planar objects (boxes and containers with different textures and patterns). Also this dataset does not contain dense video sequences (each object is described by a sample of views from finite viewpoints taken at a distance of about 5 degrees), thus it does not fit exactly with our

[4]The image dataset is available at http://conex.icg.tu-graz.ac.at

approach. The results on this set of planar objects demonstrate that our descriptor achieves results in line with the ones reported in [Gra04]. Figure 4.5 reports examples of our matching on such a dataset.

4.2.2 Similarity of virtual features with colour histograms

The results shown in the previous section point out one of the main problems of local methods for object recognition: minute details are more likely to be common to many different objects. One possible solution to this problem is to reinforce the descriptor with other kind of information. In this section we briefly explore this choice, adding colour information to the feature description: we will show that adding colour information is not informative enough for the recognition purpose.

Since the virtual features consist essentially of a direction histogram, we have built a *coloured version of virtual features* by adding a colour histogram [SB91] to the descriptors obtained in Section 3.3.1. Figures 4.7 and 4.6 show the results of the comparison using histogram intersection as similarity measure on the local descriptor extended with the use of colour histogram. The results obtained on different test images show that adding colour information does not overcome the most important drawback of local approaches which is that minute details are common to many objects.

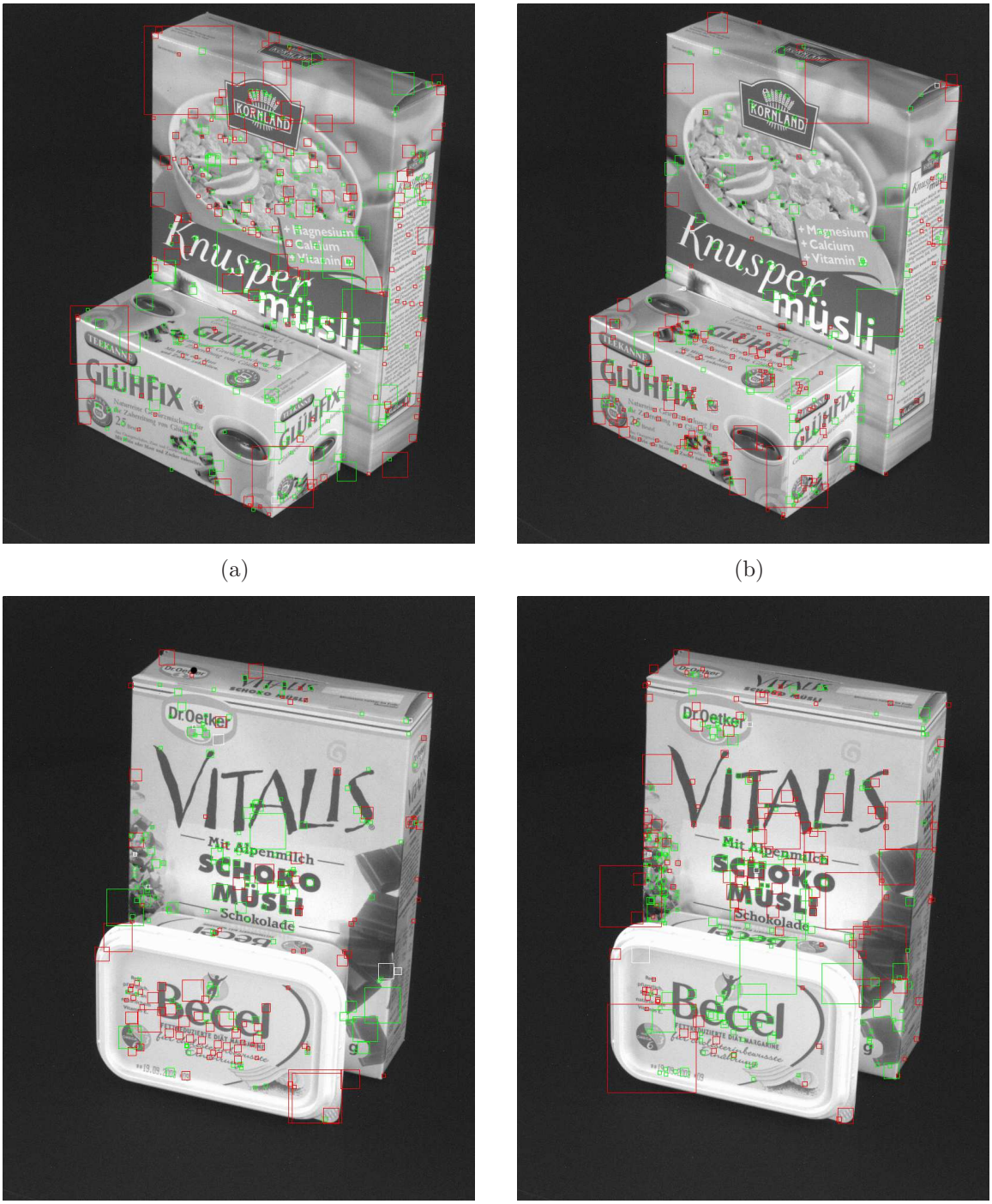
Then, to overcome this limit it is possible to plan two different solutions:

- devising a more complex approach to the recognition process
- adding global information as it is proposed in some of the approaches described in 3.2.

Thus in the remainder of this chapter we describe our approaches based on these ideas: first we use SVM classification to recognise 3D objects seen under different viewpoints and with different background, then we design a two-stage matching procedure exploiting spatial and temporal information.

4.3 Classification with SVM

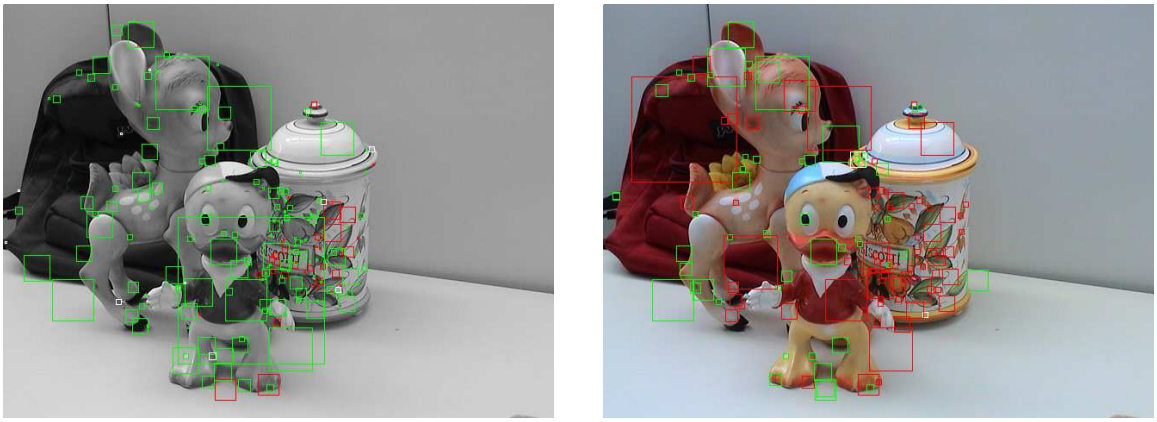
The simple matching among visual words and keypoint descriptors described in previous section is not sufficient for our aim: indeed we are not exploiting temporal relation between features and we are loosing more global information that can be obtained by the model. A possible choice to exploit the richness of our view-based models is to resort to learning from examples. Thus, in this section we consider a subset of the objects shown in Figures



(a) (b)

(c) (d)

Figure 4.5: In (a) and (b) the image is compared with the vocabulary of the object *kornland* and *gluhfix* respectively. In (c) and (d) the image is compared with the vocabulary of the object *becel* and *oetker* respectively.



(a) (b)

Figure 4.6: Matches based on similarity values computed on an extended version of visual features using colour histograms. In (a) the matching is performed using virtual features without colour information while in (b) the similarity is computed on the descriptor extended with a 8 bin size colour histogram. The red matches correspond to the values of similarity higher than 0.6 while the green ones have similarity higher than 0.4.

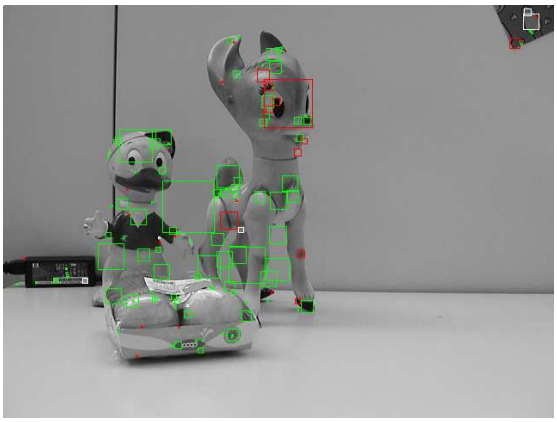
4.1 and 4.2 and we describe the set of experiments performed with the use of Support Vector Machines as classifiers.

Given an object of interest the same image sequences used to build the object vocabulary can be used as training data. We compute a visual vocabulary for every object on the basis of its training set images. Then we need to represent both training images and test images with respect to visual vocabularies of objects (see Section 3.3.2). This is a fundamental step, since SVMs assume that data belong to a given space and therefore they are capable to compare feature vectors of equal length. So, we represent each image of the training and test sets as a vector of similarities computed with the visual vocabulary (see Section 3.3.2). It is worth to notice that in this case, the training images are the same images used to create the visual models.

Then we can train a set of SVM binary classifiers to recognise each object: the positive examples are given by training set images, while the negative examples are given by background and other objects images. Thus for each object we use about 200 positive training examples and 300 negative training examples.

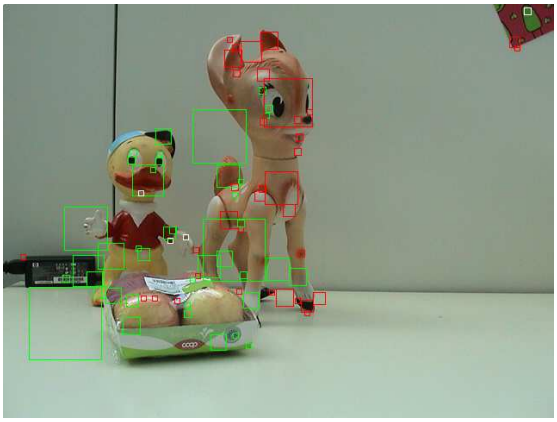
We evaluate the performance of this SVM-based object recognition in six different scenarios of increasing complexity. To do so we acquire six different test sets for each object of interest:

1. similar conditions to the training



(a)

(b)



(c)

Figure 4.7: Matches based on similarity values computed on an extended version of visual features using colour histograms. In `(a)` the matching is performed using virtual features without colour information while in `(b)` the similarity is computed on a descriptor extended with a 8 bin size colour histogram and in `(c)` the similarity is computed on the descriptor extended with a 64 bin size colour histogram. The red matches correspond to the values of similarity higher than 0.6 while the green ones have similarity higher than 0.4.

objects	simple	scale	occlusions
bambi	99.50	92.34	100.00
box	100.00	100.00	100.00
duck	100.00	98.90	100.00
biscuit	100.00	100.00	100.00
bookGeo	100.00	100.00	100.00
bookSvm	100.00	95.53	77.78
dino	100.00	100.00	100.00
teddy	100.00	98.86	100.00
pino	100.00	99.59	92.96
tele	100.00	100.00	100.00
tommy	100.00	100.00	100.00

Table 4.1: Hit percentages of the 11 classifiers against the simple test set, the test set with a scale change and the test set in which the object is partially occluded.

2. moderated illumination changes
3. different scale
4. allowing for severe occlusions of the object
5. placing the object against a plain, but different background
6. placing the object against a complex and highly textured background.

All these settings are shown in Figure 4.8 for an example object. At run time we test an image considering it as a whole (no multiple search is performed). Since each sequence is made of approximately 300 frames, each object has about 18 000 images of test examples.

The recognition rates obtained over test sets (1-4) are summarised in Table 4.1. The column **simple** refers to the results obtained on test sets (1) and (2), the column **scale** refers to test set (3), while the column **occlusions** refers to test set (4). The good results confirm how SIFT descriptors combined with a robust feature tracking produce a model which is robust to illumination and scale changes, and to occlusions. The description proposed captures the peculiarity of objects, and allows us to recognise them correctly even if the possible classes contain many similar objects. In the column about the results obtained in case of occlusions, the drop obtained for object "bookSvm" is due to the fact that in many test images the object was entirely hidden.

In the case of more complex backgrounds, instead, it is worth showing the confusion matrices (Tables 4.2 and 4.3). They show how, if the amount of clutter is small, the recognition

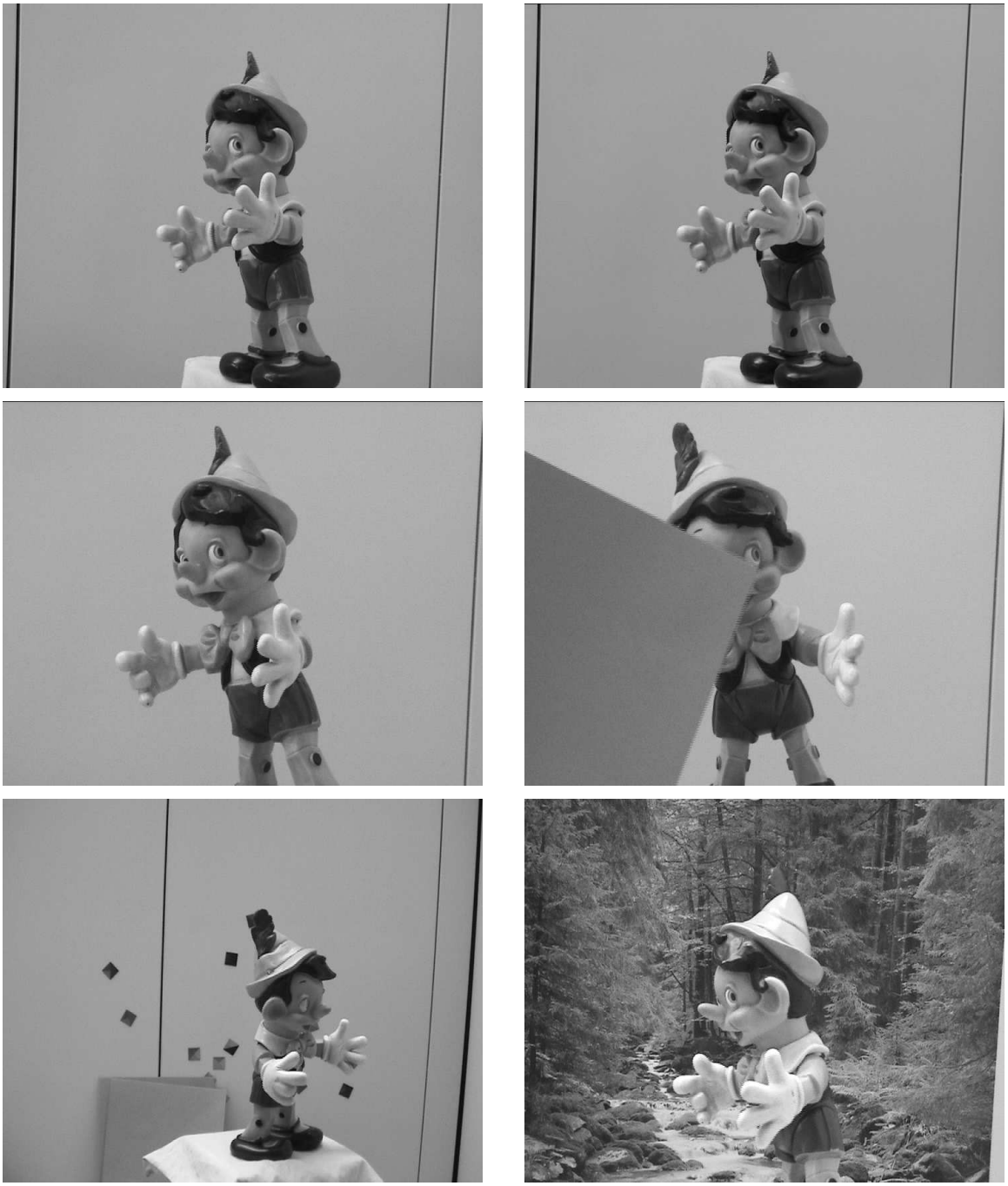


Figure 4.8: The different conditions under which the test data have been acquired (see text).

rates are still very satisfactory. In the case of very complex and textured backgrounds the performance drops because of the high number of keypoints detected (of which only a small number belong to the object).

	bambi	**box**	**duck**	**biscuit**	**bookGeo**	**bookSvm**	**dino**	**teddy**	**pino**	**tele**	**tommy**
bambi	71.46	0.00	13.69	0.00	0.00	0.00	0.00	3.48	2.55	0.23	8.58
box	0.34	98.65	1.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
duck	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
biscuit	0.00	0.00	0.22	99.78	0.00	0.00	0.00	0.00	0.00	0.00	0.00
bookGeo	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00
bookSvm	5.22	0.00	0.37	0.00	0.00	91.04	0.00	1.49	0.00	1.49	0.37
dino	9.48	0.00	13.73	0.00	0.00	0.00	58.17	0.33	0.00	0.00	18.30
teddy	0.00	0.00	3.13	0.00	0.00	0.00	0.00	96.87	0.00	0.00	0.00
pino	15.66	0.00	15.93	0.00	0.00	0.00	7.42	1.92	41.48	0.00	17.58
tele	0.93	0.93	6.48	0.00	0.00	0.00	0.00	0.93	0.00	90.28	0.46
tommy	4.86	0.00	2.86	0.00	0.00	0.00	4.29	0.57	2.29	0.00	85.14

Table 4.2: Confusion matrix for test set (5), with moderate quantities of clutter on the background.

	bambi	**box**	**duck**	**biscuit**	**bookGeo**	**bookSvm**	**dino**	**teddy**	**pino**	**tele**	**tommy**
bambi	2.11	0.00	5.15	19.67	2.11	11.01	0.00	11.94	0.00	8.90	39.11
box	0.00	85.81	0.65	0.65	0.00	8.06	0.65	0.00	0.00	0.65	3.55
duck	0.53	0.00	40.74	9.52	1.06	0.53	0.53	4.23	0.53	4.23	38.10
biscuit	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
bookGeo	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00
bookSvm	0.37	0.00	0.74	0.00	0.37	96.68	0.00	0.37	0.00	0.74	0.74
dino	1.08	0.00	0.65	16.85	33.69	3.46	2.38	11.45	1.08	2.81	26.57
teddy	1.24	0.00	3.96	0.25	1.73	4.70	0.50	36.14	7.43	14.60	29.46
pino	0.00	0.63	8.15	25.08	13.48	7.52	0.00	4.70	0.63	10.34	29.47
tele	0.00	0.47	0.47	0.00	0.94	12.21	0.00	0.00	0.00	81.22	4.69
tommy	2.07	0.00	0.00	1.38	7.24	6.55	0.00	33.79	1.72	6.55	40.69

Table 4.3: Confusion matrix for test set (6), with a very complex background.

4.3.1 Discussion

The approach based on the use of SVM classifiers described in this section has demonstrated good performances and allows us to obtain good results even in cases of strong illumination and scale variations and occlusions. Table 4.3 helps us to understand some of the limits intrinsic of our approach: when the amount of clutter in the background increases, the recognition rates decrease dramatically. This is due to the great number of keypoints detected in the background, which are statistically confused with those in the model. This problem can be overcome by introducing a search procedure that evaluates image regions at different scales. The main drawback to sliding windows usually consists of their high computational cost but are very useful to overcome background problems.

Another intrinsic limit of this approach is that every training model is object-centred, thus each time a new object is added there is the need to repeat a high number of operations. Every time we are adding a new object to the dataset, then we have to repeat all the process from the beginning: compute the visual vocabulary, represent training and test sets with respect to the vocabulary, train a SVM classifier for the new object and eventually compute the recognition rate with respect to all the other objects. At run time we need to represent each test image with respect to all the training models. As they grow in number the approach becomes impractical. It is worth to notice that, to obtain a confusion matrix as the one in Table 4.3, we need to compute again all the SVM classifiers considering the new object. Therefore, all this procedure makes the approach more complicated and limits also the number of classes that can be considered with this method.

Finally, it is worth observing another characteristic of this approach which is its asymmetry: in fact when we compute similarities with respect to the visual vocabulary, we compare a descriptor computed in a unique point in space and time with the visual word of the view-based model, which consists of an average of descriptors spread in an interval of time.

All these observations lead us to devise a new recognition method based on the same underlying principle but explicitly taking into account the following issues:

- obtaining a system that can easily scale as the number of modelled objects grows
- use similar descriptions both in training and test.

We abandon the SVM architecture resorting to a much simpler matching procedure that stands on spatio-temporal features co-occurrences. Thus in the next section we describe the new approach based on local and temporal matching.

4.4 Local and temporal matching

In this section we report some experiments carried out using the approach based on the matching method described in Section 3.4.2. It is worth remembering that this approach is based on a two-stage matching which exploits spatial and temporal information belonging to each sequence model. In other words the first step gives us a set of hypothesis matches that are confirmed or rejected by the second step on the basis of spatial and temporal constraints.

With this method we introduce something which reminds of a spatial search in sub-parts of the image, but it is not computational expensive since this particular search is performed only in the few regions selected by a first positive response to the matching process. Then, it is important to notice that with this approach we compare visual words from the model and the test computed as averages of descriptors on identical intervals of time: this allows us to recover the symmetry in the comparison.

Finally we would like to emphasise that the two-stage matching procedure has a computational cost which depends only on the dimension of the models to compare: indeed our test models are bigger than those of training but they are at most 6000 visual words (in the case the object is highly textured and the trajectories very robust).

Thus we test our models with different video sequences that have been acquired under various conditions. A first set of experiments is accomplished to assess the method and evaluate its robustness with respect to time delays between training and test, scene changes, variations in the camera motion. This first set of experiments is accomplished on a smaller group of objects. In a second set of experiments the objects are placed in a very complex scene; we acquired *tracking shots* recording all the objects in one sequence. These sequences are meant to reproduce the scene observed by a robot moving in a real environment. Finally we test our approach performances in the case the number of objects increases, thus we acquire an *incremental dataset* of 20 very similar objects and we compare results when the number of objects goes from 5 to 10 and finally to 20 objects.

4.4.1 Method assessment

In this section we report some experiments carried out on different video sequences acquired under various conditions. We trained the system to model 4 objects of those appearing in Figures 4.1 and 4.2: 2 planar objects (*bookSvm* and *winnie*) and 2 complex 3D objects with many concavities (*goofy* and *dewey*). Each object is represented by a training sequence of 250 frames acquired by a still camera observing the object on a turntable. Each training model has about 1000 entries.

Figure 4.9: Matching results in the presence of illumination and scale changes, clutter, multiple objects. Circles: *dewey*'s features; squares: *book*, crosses: *winnie*; X's: *goofy*.

Changes in scale, light and scene background. Preliminary tests confirmed that our modelling and matching techniques do not suffer from scale and illumination changes and clutter: in Figure 4.9 it is shown that even if there are some changes in scale, illumination and variations in the background, the objects are correctly recognised. The matches are visualised as white X's for *goofy*, green circles for *dewey*, red squares for *bookSvm* and blue crosses for *winnie*

Acquisition of training and test in different days. Table 4.4 shows the results of comparing training models with test models acquired in a different room, with a different illumination. Also, training and test sequences are acquired in different days and daytime. The table shows how, using our two-stage matching strategy, noticeably increases the matches: columns M1 report nearest neighbour matches, while columns M2 are obtained with our matching strategy.

	Book		Goofy		Dewey		Winnie	
	M1	M2	M1	M2	M1	M2	M1	M2
Book	**85**	**97**	4	0	11	7	8	1
Goofy	5	1	**80**	**97**	23	0	3	0
Dewey	3	1	11	2	**63**	**93**	2	0
Winnie	7	1	5	1	3	0	**87**	**99**

Table 4.4: Hit percentages between training and test videos acquired in different conditions with and without the use of spatio-temporal constraints (M2 and M1 respectively).

	Book		Goofy		Dewey		Winnie	
	M1	M2	M1	M2	M1	M2	M1	M2
Book	42	**69**	14	19	11	0	9	3
Goofy	28	7	52	**81**	29	0	21	17
Dewey	19	17	17	0	41	**100**	12	5
Winnie	11	7	17	0	19	0	58	**75**

Table 4.5: Hit percentages between training and test videos acquired with a hand-held camcorder, with and without spatio-temporal constraints (M2 and M1 respectively).

Acquisition with a hand-held camcorder. Table 4.5 presents results obtained while checking the method's robustness when the camera moves of a more complex motion, in this case it is hand-held by the user: since the ego-motion is shaky, features are more difficult to track, and, as a result, trajectories tend to be shorter and more unstable. Again, columns M1 show the hits obtained by simple nearest neighbour appearance matching, while columns M2 show how the results have been increased with our matching strategy. The advantage of this approach is apparent: our matching increases the number of positive matches, while limiting the false positive matches.

4.4.2 Real environment scene experiments

We apply our method to the localisation of objects placed in very complex scenes, containing many other objects of the same sort. The motion of the camera is almost rectilinear, displaying the objects as in a *tracking shot*. We collected various long videos about 500 frames each. The videos have been acquired at a variable speed, slowing the camera down in the presence of possible objects of interest. When the speed is higher, the video quality decreases and very few time-invariant features are detected. The objective of these experiments is to test the videos against the presence of our 4 objects of interest.

Figure 4.10 shows the results obtained on a *tracking shot* video: the plot above shows the number of matches obtained per frame, the intermediate horizontal coloured bars are the ground truth, the plot below shows the total number of trajectories detected along the sequence. When the camera speed grows the number of detected features decreases and so does the number of matches. All the objects are detected correctly in the high quality sections of this video with the exception of the book, because it appeared only briefly in a low quality section of the video.

Figure 4.11 shows matching examples on sample frames from our video shots. Matches are kept when the similarity value is higher than 0.6. White X's (*goofy*), blue crosses (*winnie*), red squares (*bookGeo*) and green circles (*dewey*) represent the features matching the objects models with temporal constraints. The big circles indicate the regions where the spatial constraints are explored: yellow X's, light blue crosses, orange squares, and light green circles are the matches obtained expanding the search on the above mentioned regions.

4.4.3 Increasing number of objects

Since one of the main reasons to adopt this matching method relies in the fact that it is more scalable than the SVM approach, in this group of experiments we test its performances when the number of objects increases. We show the results of the two-stage matching procedure computed on an *incremental dataset* that we can describe as follows:

The first group contains 5 objects: *bambi, biscuit, bookGeo, dewey* and *winnie*, which are variable and different enough.

The second group contains other 5 objects which are added to the first group. Choosing the objects in this second group we take care of considering objects similar but different to those of the first. In other words for each planar object in the first group we have chosen a planar object in the second and so for the 3D objects with many concavities. Thus the objects in this set are *bookSvm, easyBox, eye, pino* and *tommy*.

The third group is made of 10 different objects, which have been selected on the basis of the same principle described for second group objects. Their name are *coffee, delfina, kermit, donald, scrooge, pastel, goofy, rabbit, sully* and *teapot*.

You met all of them in Figures 4.1 and 4.2. For each object we acquired at least two test sequences in which the object is at a different scale and there are other objects in the scene.

First of all it is important to notice that for the experiments described in this section, we acquired images with a higher resolution camera, so the trajectories obtained for this last set of experiments are much longer and stable than those obtained in previous sections.

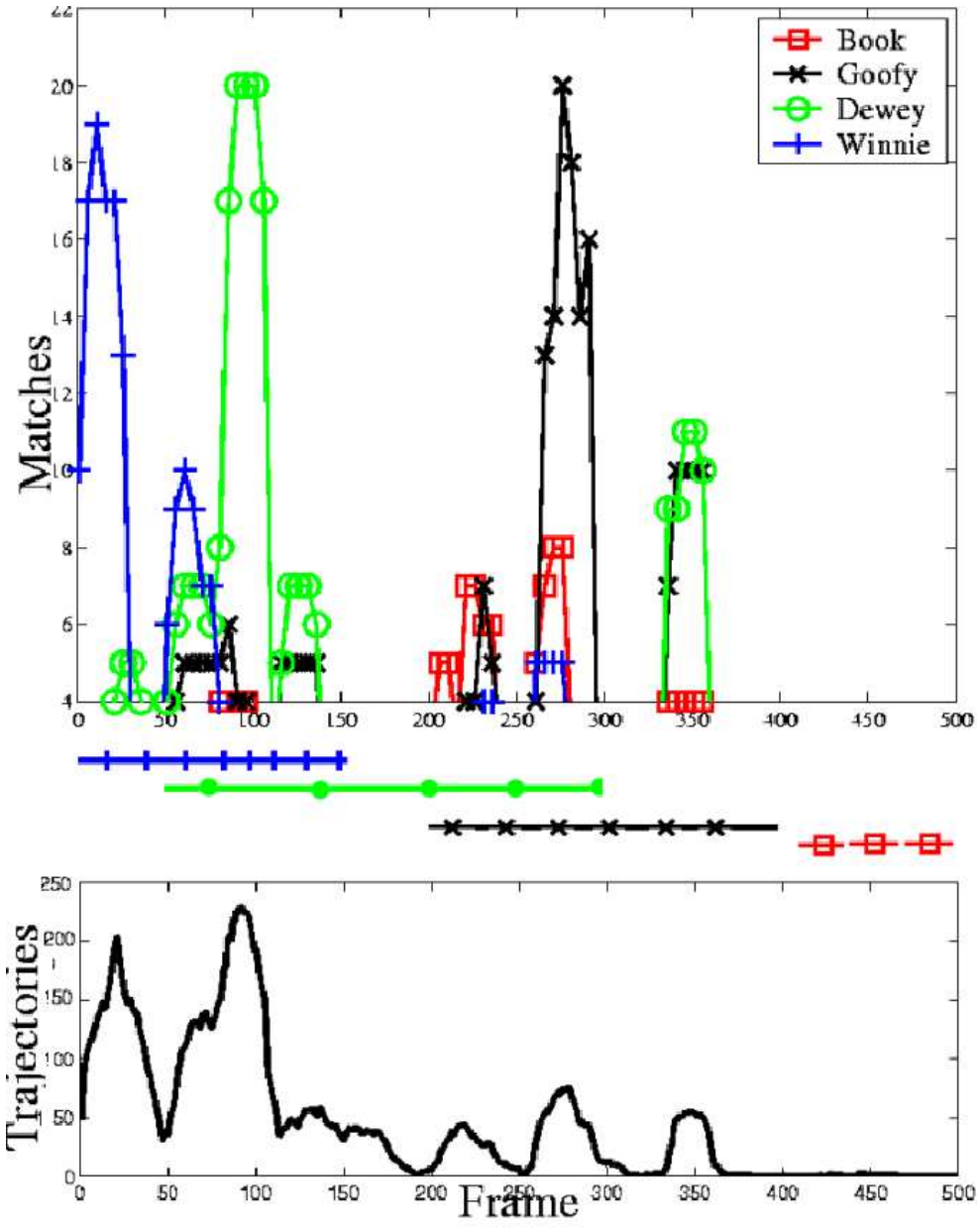


Figure 4.10: Number of matches visualised per frame on one *tracking shot*. Below, a plot of the number of time-invariant features detected per frame (see text).

Figure 4.11: Matches on sample frames. Green circles: *dewey*. Red squares: *book*. Blue crosses:*winnie*. White X's: *goofy*. Only similarity scores above 0.5 are shown.

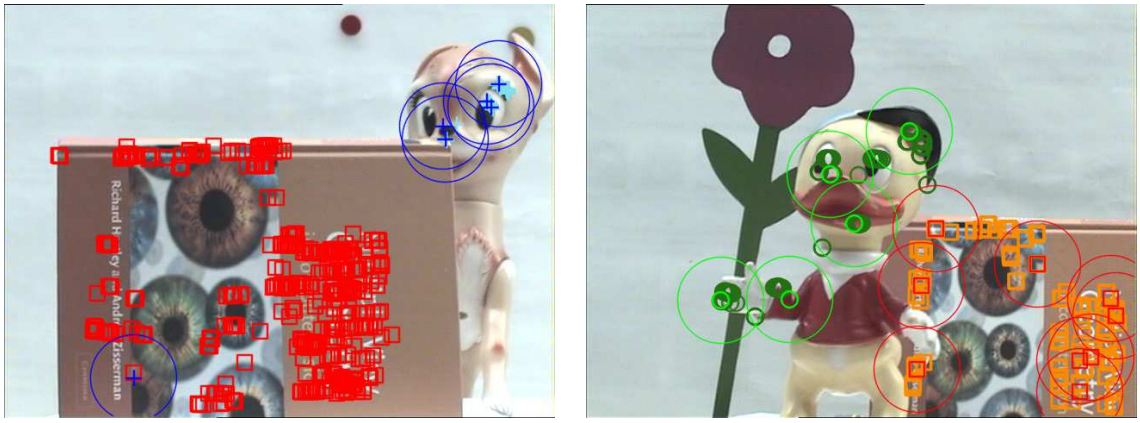
	Bambi	Biscuit	BookGeo	Dewey	Winnie
Bambi 1	172 – 596	0 – 0	0 – 0	0 – 0	0 – 0
Bambi 2	93 – 448	0 – 0	0 – 0	0 – 0	1 – 0
Biscuit 1	0 – 0	660 – 3521	0 – 0	0 – 0	1 – 0
Biscuit 2	0 – 0	560 – 4576	0 – 0	0 – 0	0 – 0
BookGeo 1	1 – 0	1 – 0	581 – 3333	1 – 0	1 – 0
BookGeo 2	1 – 0	0 – 0	599 – 3141	1 – 0	0 – 0
BookGeo 3	5 – 152	0 – 0	697 – 4271	0 – 0	1 – 0
Dewey 1	1 – 0	0 – 0	1 – 0	222 – 629	1 – 0
Dewey 2	0 – 0	0 – 0	0 – 0	857 – 706	1 – 0
Dewey 3	1 – 0	0 – 0	108 – 1662	107 – 561	1 – 0
Winnie 1	2 – 0	0 – 0	0 – 0	1 – 0	255 – 988
Winnie 2	0 – 0	3 – 6	0 – 0	0 – 0	144 – 700

Table 4.6: Matches obtained with the two-stage matching procedure computed on the first group of 5 objects. The slots corresponding to the models appearing in tests sequences are emphasized with a darker gray.

The increasing robustness of trajectories lead us to obtain bigger models for objects and therefore each visual word of each model is obtained as an average on a bigger number of descriptors. Thus when the matches are visualised on the object it is worth reminding that the position displayed is not precise, because also the position is an average of the positions of each descriptor.

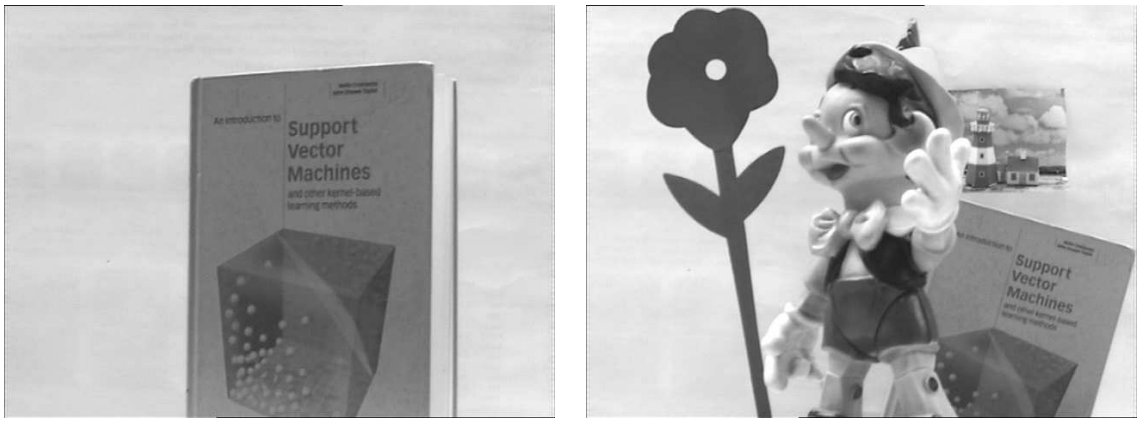
First let us consider a set of five objects which are two plastic toys, a ceramic highly textured box, a book and a ring binder. Since each of these object is distinctive with respect to the others, then the recognition procedure performs very well: the object are recognised on the basis of the higher number of matches computed in the second step of our two-stage matching. In the following tables the models are always displayed on the columns while the test sequences are on the rows, therefore the darker entries correspond to the position in which a model appear in a test sequence.

Table 4.6 shows the results obtained with this first data set of objects. It is worth noticing that there are no false positives: in the third test sequence for *bookGeo*, we have also *bambi* which is correctly recognised even if the number of matches is lower and in the third test sequence for *dewey* there is also *bookGeo*. Figure 4.12 shows a sample of frames extracted from these test sequences, the light colour matches are those detected in the second step of our matching procedure (notice that for the book we do not show the second-step matches to ensure the image readability).



(a) (b)

Figure 4.12: Subfigure (a) shows one frame from the test of *bookGeo* in which *bambi* is correctly recognised. (b) shows one frame from the test of *dewey* in which *bookGeo* is correctly recognised. Red squares (*bookGeo*) and green circles (*dewey*) represent the features matching the objects models without temporal constraints. The big circles indicate the regions where the spatial constraints are explored: orange squares, and light green circles are the matches obtained expanding the search on the above mentioned regions. For *bookGeo* we show only the matches of the first stage of matching to improve the readability of the image.



(a) (b)

Figure 4.13: Subfigure (a) shows one frame of the sequence used for the model of *bookSvm* while in (b) there is one frame of the test of *pino* in which *bookSvm* is not recognised. This is mainly due to the fact that the book has a strong scale difference between model and test and is almost totally occluded by other objects.

Let us consider what happen when we add the second group of objects. If we look at Table 4.7 we can notice that the first part is identical to Table 4.6, but then we have reported here the comparisons between models of the first group with tests of the second and viceversa. Looking at the results in the table it is apparent that now there are some false positives, indeed our matching procedure confuses similar planar objects such as for instance *bookGeo*, *bookSvm*, *easyBox* and *eye* (which is another book). But it is still robust enough with similar 3D objects such as for instance *dewey*, *tommy*, *pino* and *bambi*. Notice that in the first test sequence for *pino* is also appearing *bookSvm* which is not recognised. This is due to the fact that the most meaningful part of *bookSvm* is not visible, therefore the model have been acquired with a very strong different in illumination with respect to the test. Look at Figure 4.13 to see the differences between model and test sequences.

It is interesting to notice that the number of matches detected for an object in foreground is noticeably different from the number of matches obtained for the same object partially occluded or in the background of the images. Figure 4.14 shows the matches on the object *bambi* when he appears in the foreground for the test `Bambi 1` and in the background for the sequence of test `Tommy 1`.

Finally we add the objects of the third group, thus we compute the matching between all the possible models and all the possible test sequences. We report the results of this set of matching in three tables: Table 4.8 contains the results of matching the test sequences of the third group objects with their own models, Table 4.9 contains the results of matching the test sequences of the third group objects with models of the first and second groups

(a) (b)

Figure 4.14: In subfigure (a) it is shown on frame from the test of *bambi* to show the high number of correct matches while in subfigure (b) the number of matches on *bambi* decreases. White X's (*tommy*), blue crosses (*bambi*) represent the features matching the objects models without temporal constraints. The big circles indicate the regions where the spatial constraints are explored: yellow X's and light blue crosses are the matches obtained expanding the search on the above mentioned regions.

of objects and eventually Table 4.10 contains the results of matching the tests of the first and second groups of objects with the models of the third group.

Looking at Table 4.8 it is possible to see that almost all the objects are correctly recognised with some exceptions which are for the model of *kermit* in `Delfina 2`, the model of *goofy* in `Rabbit 1` and the model of *scrooge* in `Donald 2`. This is due to the fact the the objects in the background are partially occluded, as we have already said. Indeed `Donald 2` sequence is really difficult since it contains all the members of Duck family (at least the ones in my dataset), *scrooge*, *donald* and *dewey*, and at least two of them have been recognised (see `Donald 2` in Table 4.8 and Table 4.9). Figure 4.15 shows the results of matching obtained for this sequence. It is worth noticing that the matches for *scrooge* model are wrong since they fall on *donald*, but those of *dewey* are in the correct positions.

Finally, Table 4.10 shows the results of the matching between the models of the last group with the test of the first and the second group: it is worth noticing that in these test sequences there is no one of the models of the third group of object, thus there are no darker entries in the table.

20 objects and tracking shot videos. We look for these 20 objects models in video sequences in which several objects appear, thus we show resulting matches for each frame of the video sequence. We show here one test video which is a *tracking shot* sequence

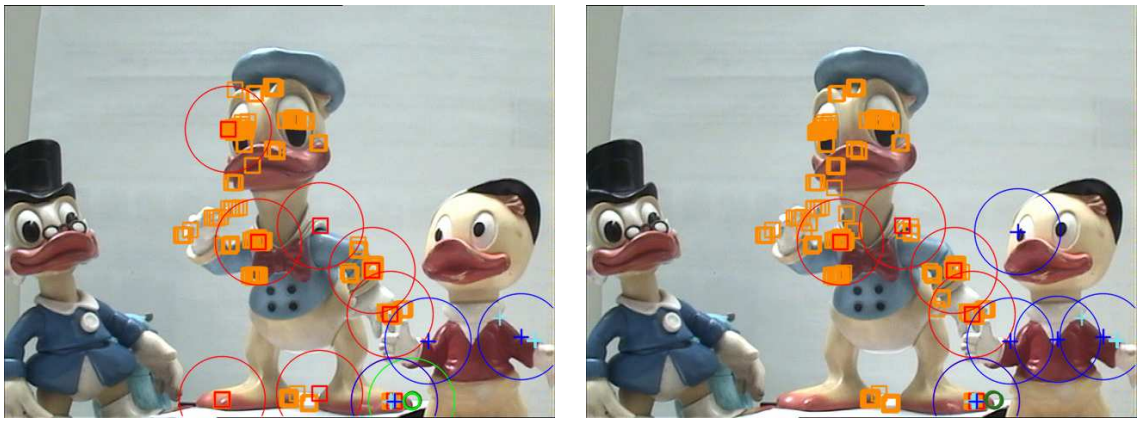


Figure 4.15: Matches obtained on the sequence `Donald 2`. The matches for *scrooge* model are wrong since they fall on *donald*, but those of *dewey* are in the correct positions. Red squares (*donald*) and blue crosses (*dewey*) represent the features matching the objects models without temporal constraints. There is also one green circle corresponding to *scrooge* but it is detected on *donald* foot.

similar to the one that we used for the experiments in Section 4.4.2. In this sequence *dewey* appears first, while *sully* appears at frame 130.

Figure 4.16 shows a sample of frames extracted from the sequence: red squares are used for matches of *dewey* model (the orange squares are given by the second step of matching), white X's are for *sully* (yellow X's for matches of step2), green circles are for the model of *donald* while blue crosses are for the model of *pastels*. The localisation is not precise on keypoints, since we show average positions of matches. Figure 4.17 shows the results obtained on this *tracking shot* video: the plot shows the number of matches obtained for each frame. It is apparent that *pastels* and *donald* are confused with the object present in the sequence: the main source of errors is the black box on which *sully* stand on. This is due to the complex texture of the box which makes fail the matching procedure.

4.4.4 Discussion

The approach based on the two-stage matching procedure have shown good performances even in cases of highly complex background or great changes in scale and illumination. The set of experiments performed on the *incremental dataset* of objects have demonstrated the feasibility of the method and the robustness of our approach in the case the number of object increases. There are some false positives in the recognition since some of the objects are similar and especially planar objects tend to be confused.

Figure 4.16: A sample of frames of the *tracking shot* video in which *dewey* and *sully* appear. Red squares are used for matches of *dewey* (the orange squares are given by the second step of matching), white X's are for *sully* (yellow X's for matches of step2), green circles are for the model of *donald* and blue crosses are for the model of *pastels*.

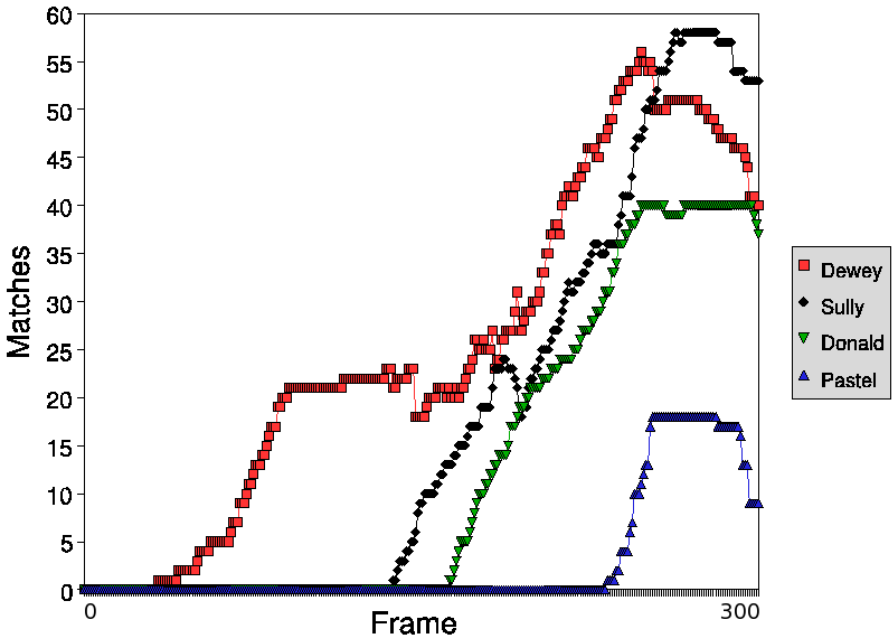


Figure 4.17: Number of matches for each frame of a video sequence in which *dewey* appears first and *sully* appears at frame 130. Dewey disappears almost at the end of the sequence.

This limit may be overcome by devising a tournament strategy in the recognition stage, in which the first step of the recognition aims at distinguishing between categories of objects and then a second step is recognising the particular object. This approach would make the first step to object recognition more similar to an object categorisation problem.

Another improvement to our object recognition may be devising an on line acquisition of models that can be updated meanwhile the camera is observing a new object. Actually we are working on the on line version of the matching procedure for object recognition that will be discussed in the **Conclusion** section.

	Bambi	Biscuit	BookGeo	Dewey	Winnie	BookSvm	EasyBox	Eye	Pino	Tommy
Bambi 1	172–596	0–0	0–0	0–0	0–0	1–0	2–0	0–0	0–0	0–0
Bambi 2	93–448	0–0	0–0	0–0	1–0	1–0	3–3	0–0	0–0	0–0
Biscuit 1	0–0	660–3521	0–0	0–0	1–0	1–0	2–0	0–0	1–0	0–0
Biscuit 2	0–0	560–4576	0–0	0–0	0–0	0–0	0–0	2–3	0–0	0–0
BookGeo 1	1–0	1–0	581–3333	1–0	1–0	4–112	3–57	12–395	0–0	0–0
BookGeo 2	1–0	0–0	599–3141	1–0	0–0	2–203	3–2	7–133	0–0	0–0
BookGeo 3	5–152	0–0	697–4271	0–0	1–0	1–0	3–0	13–433	0–0	0–0
Dewey 1	1–0	0–0	1–0	222–629	1–0	2–0	5–7	1–0	0–0	0–0
Dewey 2	0–0	0–0	0–0	857–706	1–0	1–0	0–0	0–0	0–0	1–0
Dewey 3	1–0	0–0	108–1662	107–561	1–0	1–0	0–0	3–3	0–0	0–0
Winnie 1	2–0	0–0	0–0	1–0	255–988	1–0	2–0	0–0	1–0	0–0
Winnie 2	0–0	3–6	0–0	0–0	144–700	1–0	2–0	0–0	0–0	0–0
BookSvm 1	0–0	1–0	15–343	0–0	1–0	177–879	0–0	0–0	0–0	0–0
BookSvm 2	0–0	1–0	31–524	0–0	1–0	159–1239	0–0	0–0	0–0	0–0
EasyBox 1	2–0	0–0	6–26	0–0	6–22	4–2	317–1498	1–0	0–0	2–38
EasyBox 2	0–0	2–22	2–0	1–0	1–0	4–2	160–1039	4–7	0–0	0–0
Eye 1	1–0	0–0	15–421	0–0	0–0	0–0	0–0	211–2231	0–0	0–0
Eye 2	1–0	1–0	30–437	0–0	0–0	1–0	2–21	219–1665	3–66	0–0
Pino 1	0–0	0–0	0–0	0–0	0–0	0–0	0–0	0–0	129–517	0–0
Pino 2	0–0	0–0	0–0	0–0	0–0	0–0	0–0	0–0	90–513	0–0
Tommy 1	9–121	0–0	0–0	0–0	0–0	0–0	0–0	0–0	0–0	221–1371
Tommy 2	0–0	1–0	0–0	0–0	0–0	0–0	1–0	0–0	0–0	45–604

Table 4.7: Matches obtained with the two-stage matching procedure computed on the first and the second groups of objects. The slots corresponding to the models appearing in tests sequences are emphasized with a darker gray.

	Coffee	Delfina	Kermit	Donald	Scrooge	Pastel	Goofy	Rabbit	Sully	Teapot
Coffee 1	266–1724	0–0	0–0	1–0	0–0	3–23	2–14	0–0	0–0	2–11
Coffee 2	97–946	3–0	1–0	0–0	4–0	0–0	4–19	2–0	0–0	3–3
Delfina 1	1–0	207–841	3–0	1–0	1–0	1–0	2–0	2–0	1–0	3–0
Delfina 2	0–0	80–300	1–0	0–0	2–0	0–0	2–0	0–0	3–2	0–0
Kermit 1	0–0	5–12	428–1518	2–0	1–0	0–0	2–0	1–0	0–0	2–4
Kermit 2	2–0	4–13	258–1621	0–0	1–0	0–0	2–0	2–0	1–0	2–0
Donald 1	6–11	2–0	3–0	106–830	1–0	0–0	1–0	1–0	3–0	7–62
Donald 2	2–0	0–0	2–0	138–1174	6–13	0–0	1–0	1–0	1–0	4–0
Scrooge 1	1–0	0–0	0–0	0–0	642–2426	0–0	6–44	0–0	1–0	3–0
Scrooge 2	2–0	2–0	1–0	0–0	98–1259	0–0	6–13	3–16	1–0	1–0
Pastel 1	0–0	0–0	0–0	0–0	0–0	950–7934	1–0	1–0	0–0	0–0
Pastel 2	3–761	1–0	0–0	0–0	3–0	419–7159	0–0	0–0	1–0	1–0
Goofy 1	2–0	2–0	1–0	1–0	0–0	0–0	434–2022	2–0	4–2	4–0
Goofy 2	1–0	0–0	0–0	1–0	0–0	0–0	236–1168	1–0	0–0	1–0
Rabbit 1	2–0	2–2	0–0	3–0	3–0	0–0	7–0	356–1522	0–0	3–8
Rabbit 2	0–0	2–0	0–0	1–0	2–0	0–0	13–86	96–840	0–0	1–0
Sully 1	1–0	5–5	1–0	3–11	0–0	1–0	4–0	2–0	237–1488	0–0
Sully 2	4–0	7–12	0–0	0–0	1–0	0–0	12–275	2–0	92–1330	4–2
Teapot 1	0–0	3–2	1–0	2–12	0–0	0–0	1–0	1–0	4–14	296–1326
Teapot 2	4–0	3–0	0–0	1–0	0–0	0–0	3–14	3–0	1–0	37–405

Table 4.8: Matches obtained with the two-stage matching procedure computed on the third group of objects (models of the third vs. test of the third group). The slots corresponding to the models appearing in tests sequences are emphasized with a darker gray.

	Bambi	Biscuit	BookGeo	Dewey	Winnie	BookSvm	EasyBox	Eye	Pino	Tommy
Coffee 1	1–0	0–0	0–0	3–2	1–0	0–0	1–0	0–0	2–63	3–45
Coffee 2	2–0	1–0	0–0	0–0	3–0	4–5	6–0	0–0	0–0	0–0
Delfina 1	5–9	4–7	1–0	0–0	3–0	3–0	9–12	3–0	1–0	0–0
Delfina 2	0–0	1–0	2–0	1–0	2–14	2–0	3–3	0–0	0–0	0–0
Kermit 1	3–0	1–0	2–20	1–0	3–3	5–7	8–45	2–3	0–0	3–8
Kermit 2	1–0	2–0	2–3	0–0	3–0	2–0	5–9	1–0	0–0	0–0
Donald 1	8–6	0–0	0–0	6–6	2–0	5–5	10–25	3–107	0–0	0–0
Donald 2	4–0	0–0	0–0	14–113	1–0	2–0	3–2	0–0	1–0	0–0
Scrooge 1	0–0	1–0	0–0	0–0	2–0	4–0	2–0	0–0	0–0	0–0
Scrooge 2	0–0	2–0	0–0	0–0	3–0	3–3	2–0	0–0	1–0	0–0
Pastel 1	0–0	0–0	0–0	1–0	1–0	2–2	1–0	0–0	3–65	0–0
Pastel 2	0–0	13–204	2–131	1–0	0–0	0–0	1–0	0–0	1–0	1–0
Goofy 1	3–4	0–0	0–0	3–2	5–0	6–5	12–19	1–0	0–0	0–0
Goofy 2	1–0	1–0	2–3	0–0	2–0	1–0	1–0	0–0	0–0	0–0
Rabbit 1	5–15	2–55	0–0	0–0	0–0	0–0	3–0	1–0	0–0	0–0
Rabbit 2	0–0	1–0	0–0	0–0	0–0	3–0	2–0	1–0	1–0	0–0
Sully 1	0–0	1–0	1–0	3–30	3–0	1–0	1–0	0–0	0–0	1–0
Sully 2	0–0	1–0	0–0	1–0	2–0	0–0	3–7	0–0	0–0	1–0
Teapot 1	3–27	2–3	0–0	0–0	7–0	1–0	5–89	0–0	0–0	0–0
Teapot 2	2–0	1–0	1–0	0–0	7–30	5–0	5–4	1–0	0–0	0–0

Table 4.9: Matches obtained with the two-stage matching procedure computed on the third group of objects (models of the first and second groups vs. tests of the third group). The slots corresponding to the models appearing in tests sequences are emphasized with a darker gray.

	Coffee	Delfina	Kermit	Donald	Scrooge	Pastel	Goofy	Rabbit	Sully	Teapot
Bambi 1	0–0	0–0	1–0	2–15	0–0	0–0	1–0	7–118	0–0	2–0
Bambi 2	0–0	1–0	0–0	1–0	0–0	0–0	1–0	7–44	1–0	3–0
Biscuit 1	0–0	1–0	0–0	0–0	0–0	1–0	2–0	1–0	1–0	2–0
Biscuit 2	2–0	0–0	1–0	0–0	0–0	2–12	1–0	0–0	3–65	0–0
BookGeo 1	0–0	0–0	0–0	1–0	0–0	4–13	1–0	1–0	0–0	0–0
BookGeo 2	2–9	0–0	2–0	1–0	0–0	3–18	1–0	1–0	0–0	2–10
BookGeo 3	0–0	1–0	0–0	2–146	1–0	2–179	1–0	0–0	0–0	4–46
Dewey 1	4–17	1–0	0–0	3–2	0–0	1–0	1–0	0–0	0–0	2–4
Dewey 2	0–0	2–0	0–0	0–0	1–0	0–0	0–0	0–0	0–0	0–0
Dewey 3	0–0	1–0	0–0	0–0	1–0	0–0	3–33	0–0	0–0	0–0
Winnie 1	0–0	0–0	0–0	1–0	2–0	0–0	2–4	2–0	0–0	0–0
Winnie 2	2–4	0–0	0–0	0–0	0–0	0–0	4–3	1–0	0–0	4–6
BookSvm 1	0–0	0–0	0–0	1–0	0–0	0–0	0–0	0–0	0–0	0–0
BookSvm 2	0–0	1–0	0–0	0–0	0–0	0–0	0–0	1–0	0–0	0–0
EasyBox 1	1–0	5–16	1–0	4–6	4–0	2–10	3–0	1–0	2–0	3–0
EasyBox 2	0–0	2–0	0–0	1–0	2–0	1–0	7–4	2–0	1–0	5–3
Eye 1	2–6	0–0	0–0	1–0	0–0	0–0	0–0	0–0	2–0	1–0
Eye 2	0–0	0–0	0–0	1–0	0–0	0–0	1–0	0–0	0–0	1–0
Pino 1	0–0	0–0	0–0	0–0	0–0	2–21	0–0	0–0	2–16	0–0
Pino 2	0–0	0–0	0–0	0–0	0–0	0–0	0–0	0–0	1–0	0–0
Tommy 1	0–0	0–0	0–0	0–0	0–0	3–40	0–0	0–0	0–0	0–0
Tommy 2	0–0	0–0	0–0	0–0	1–0	0–0	1–0	0–0	0–0	0–0

Table 4.10: Matches obtained with the two-stage matching procedure computed on the third group of objects (models of the third vs. test of the first and second groups). There are no slots emphasized with a darker gray, since in these tests no one of these models is appearing.

Conclusions

In this thesis we exploited the compactness and the expressiveness of local image descriptions to address image matching and 3D object recognition. Our aim was to demonstrate that the robustness and the good qualities of local descriptors can improve performances of methods for matching and for recognising three dimensional objects in images.

These issues are strongly correlated, but in the course of this thesis we faced them from two slightly different perspective: while in the matching case our analysis is strongly based on the image and its local keypoints, in the recognition case we base our work on image sequences and trajectories of local keypoints.

The results obtained represent contributions in the following aspects. Firstly, we proposed a novel version of the SVD-matching introduced by Scott and Longuet-Higgins [SLH91] and later modified by Pilu [Pil97]. Scott and Longuet-Higgins were among the first to use spectral methods for image matching. The algorithm is based on the two principles of proximity and exclusion, that is, corresponding points must be close, and each point can have one corresponding point at most. The algorithm presented in [SLH91] was working well on synthetic data, but performance started to fall down when moving to real images. Pilu argued that this behaviour could be taken care of by evaluating local image similarities. He adapts the proximity matrix in order to take into account image intensity as well as geometric properties. Experimental evidence in [Pil97] showed that the algorithm performs well on short baseline stereo pairs but that the performance falls when the baseline increases. Thus in **Chapter 2** we showed that the reason for this behaviour is in the feature descriptor chosen and is not an intrinsic limit of the algorithm. To this purpose we evaluated the appropriateness of SIFT [Low04] for the SVD-matching and we performed an extensive analysis to evaluate different weighting functions which are used to model the probability of the similarity between the feature points.

Secondly, we devised a novel approach to view-based 3D objects recognition which is built upon the well known idea of *visual vocabulary*. The add on of our approach is in the use of temporal constraints for building models and matching test sequences with respect to models. To do so we have worked with video sequences, in which the appearance of the object varies smoothly. **Chapter 4** contains an extensive experimental analysis of the

methods designed in **Chapter 3**. We have observed that to improve object recognition performances it is fundamental to build homogeneous model and test sequences and the robustness of descriptors and the stability of trajectories are two other important factors to be considered. The final goal of our work was to obtain a system capable to recognise an object on the basis of its appearance and on the temporal evolution of its description.

The system described in this thesis achieved very good results in many difficult environments, but so far it is designed for analysing sequences off-line. It is the case to observe that for most real applications (including recognition and localisation systems in robotic and automatic guidance applications) an on-line version of our testing procedure would be advisable. We are currently working on this direction, taking into account that some changes to the use of temporal constraints have to be implemented.

For what concerns the use of local descriptors for object recognition, it is worth reminding that one limits of local approaches is that, while observing minute details, the overall appearance of the object may be lost. In fact, the ability of exploiting context and global aspects is one of the fundamental differences between the human vision competency and the machine vision system. Indeed a machine able to see an object solely as a collection of many details will be similar to *the man who mistook his wife for a hat*[5] as described by Oliver Sacks:

> *His eyes would dart from one thing to another, picking up tiny features, individual features, as they had done with my face. A striking brightness, a colour, a shape would arrest his attention and elicit comment – but in no case did he get the scene-as-a-whole. He failed to see the whole, seeing only details, which he spotted like blips on a radar screen. He never entered into relation with the picture as a whole - never faced, so to speak, its physiognomy. He had no sense whatever of a landscape or a scene.*

And in the same book we find:

> *If we wish to know about a man, we ask "what is his story – his real, inmost story?"– for each of us is a biography, a story. Each of us is a singular narrative, which is constructed, continually, unconsciously, by, through, and in us - through our perceptions, our feelings, our thoughts, our actions; and not least, our discourse, our spoken narrations. Biologically, physiologically, we are not so different from each other; historically, as narratives – we are each of us unique.[...] To be ourselves we must have ourselves – possess, if need be re-possess, our life stories. We must "recollect" ourselves; recollect the inner drama, the narrative, of ourselves. A man needs such a narrative, a continuous inner narrative, to maintain his identity, his self.*

[5] "The Man Who Mistook His Wife For A Hat: And Other Clinical Tales", by Oliver Sacks, 1970

Thus, approaches to object recognition need to take into account the observations raised in these words. Indeed there are research investigating how the inclusion of global information and the use of the context can improve object detection and recognition [OTSM03, Tor03].

Even if in this thesis we have not faced explicitly the problem of building a more global representation of the object, we have introduced a matching procedure which is strongly based on temporal constraints and on spatial constraints for refining the search. In fact we built our descriptors thinking about the informative contents which can be collected by observing an object when its visual appearance changes. In other words we modelled their *story* which is defined on the basis of a tracking process but we also took into account the keypoints relative positions based on local refinement of the matching. Future research will be devoted to exploiting more explicit global information.

La macchina del capo

Dannazione. Questa stupida macchina non parte. E dove hanno messo la prima? Freno e frizione ingrano la prima e giro la chiavetta. E parti! PARTI!!!
Niente.
Mi volto a destra e, dal sedile del passeggero, Steve Smale[6] mi sorride e mi indica il freno a mano. Ancora tirato. Voglio seppellirmi. "Se vuoi guido io", mi dice il rosso, dal sedile posteriore. "No grazie. Ce la posso fare da sola" rispondo io con le mani salde sul volante. Sorrido sicura.
Giro la chiave, partiamo, un balzo in avanti e siamo fermi. Ah! Non è la prima questa... eh eh eh... "Se vuoi guido io", mi dice il rosso, dal sedile posteriore. "No grazie" rispondo io. Partiamo.
In fondo, il percorso è breve: dal dipartimento di Informatica al porto antico sono al massimo 5 o 6 chilometri. L'ho già fatta molte volte questa strada con la mia macchina. È un tragitto che conosco bene. Con la mia macchina, non con quella del capo. E quanto traffico: da dove sono saltati fuori tutti questi automobilisti? Ok, c'è uno Stop in salita. Mi fermo, scalo e vado in prima. Ingrano la prima. La prima ho detto: PRI-MA! Tutto ok: si riparte! Un balzo e il motore si spegne. Sorrido a Smale e dissimulo sicurezza voltandomi verso il rosso sul sedile posteriore.
Mi immetto nella coda di automobili sulla strada in salita. Questa volta mi concentro, anche se c'è la salita la macchina non si spegnerà. Respiro, sono attentissima: freno, frizione, accelero e riparto. Si! Ora ci ho preso la mano, posso andare tranquilla. Metto la freccia a sinistra, lascio passare un autobus e svolto nella piazza alberata. Il motore borbotta, tentenna e si spegne. Di nuovo.
"Se vuoi guido io", mi dice il rosso, dal sedile posteriore. La sua voce è meno sicura rispetto a prima, forse un po' più tremulante. Ma ormai ci siamo quasi e la strada è quasi tutta in discesa o almeno in piano. A questo punto del viaggio ho capito una cosa: mai più con la macchina del capo.
Semaforo rosso: mi fermo dietro alla fila di macchine. Verde: riparto. No, balzello, bal-zel-

[6]Matematico americano, viene insignito della medaglia Fields nel 1966 per aver risolto la congettura di Poincaré per $n \geq 5$. Nel 2006 vince il Premio Wolf per i fondamentali contributi nell'ambito della topologia differenziale, nello studio dei sistemi dinamici e della matematica economica.

lo, bal-zel-lo: no! Non si deve spegnere! Ce la posso fare: non spegnerti, ti prego. Frizione, acceleratore, frizione, borbottio sinistro, batte in testa... ma non la spengo! Guardo il cambio: ero in terza. Oh oh... Però sono in gamba! Partire in terza non è da tutti... Steve Smale, si guarda intorno. Non sembra preoccupato, stringe a sé il borsello di pelle nera e chiacchiera con il rosso.
La giornata è fantastica, il sole, il cielo blu e il mare in lontananza. La trasmissione radio va in onda dal Porto Antico e noi siamo quasi arrivati. In orario perfetto. Ma ora, dove accosto? Dove faccio scendere dalla macchina la medaglia Fields, Steve Smale e il suo rosso accompagnatore? Sulla destra: una selva di motorini. Sulla sinistra: automobili in terza fila. Il passaggio è stretto, abbastanza per una macchina alla volta. Quindi con coraggio premo il pulsante delle quattro frecce e inchiodo: arrivati!
Smale con un balzo è fuori e dietro di lui il rosso scende senza salutarmi. Torneranno con un taxi. Sono sola in macchina mentre loro, con passi decisi, si allontanano da me, prima che riparta. Forse corrono perché sono in ritardo.
Santo cielo: che figura! Che vergogna... Un colpo di clacson da dietro mi risveglia. Tolgo le quattro frecce, frizione, prima acceleratore e parto. Senza scrolloni, neanche un borbottio.

Pare che, salutando il mio capo prima di salire sull'aereo, Steve Smale abbia pronunciato queste parole: "Alessandro, don't let Elisabetta borrow your car any more".

Bibliography

[ADJ00] N. Allezard, M. Dhome, and F. Jurie. Recognition of 3D textured objects by mixing view-based and model based representations. In *Proceedings of the IAPR International Conference on Pattern Recognition*, 2000.

[ADOV06] E. Arnaud, E. Delponte, F. Odone, and A. Verri. Trains of keypoints for 3d object recognition. In *Proceedings of the IAPR International Conference on Pattern Recognition*, 2006.

[AMCF05] E. Arnaud, E. Mémin, and B. Cernuschi-Frías. Conditional filters for image sequence based tracking - application to point tracking. IEEE *Tr. on Im. Proc.*, 1(14), 2005.

[Att54] F. Attneave. Some informational aspects of visual perception. In *Psychological Review*, volume 61, pages 183–193, 1954.

[Bau00] A. Baumberg. Reliable feature matching across widely separated views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 774–781, 2000.

[BCGM98] S. Belongie, C. Carson, H. Greenspan, and J. Malik. Color and texture based image segmentation using the expectation-maximization algorithm and its application to content-based image retrieval. In *Proceedings of the International Conference on Computer Vision*, pages 675–682, 1998.

[BCP96] M.F. Bear, B.W. Connors, and M.A. Paradiso. Neuroscience: Exploring the brain. Technical report, Williams & Wilkins, 1996.

[BET95] H. H. Bulthoff, S. Y. Edelman, and M. J. Tarr. How are three-dimensional objects represented in the brain? *Cerebral Cortex*, 5(3), 1995.

[Bie87] I. Biederman. Recognition-by-components: A theory of image human understanding. In *Psychological Review*, volume 94, pages 115–147, 1987.

[BJ85] P. J. Besl and R. C. Jain. Three-dimensional object recognition. *Computing Surveys*, 17(1):75–145, 1985.

[BL02] M. Brown and D. Lowe. Invariant features from interesting point groups. In *Proceedings of the British Machine Vision Conference*, pages 656–665, 2002.

[BMP02] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(24), 2002.

[BOV02] A. Barla, F. Odone, and A. Verri. Hausdorff kernel for 3d object acquisition and detection. In *Proceedings of the European Conference on Computer Vision*, pages 20–33, 2002.

[BSP93] D. Beymer, A. Shashua, and T. Poggio. Example based image analysis Technical report, A.I. Memo 1431, MIT, 1993.

[BTB05] S. Boughorbel, J. P. Tarel, and N. Boujemaa. The intermediate matching kernel for image local features. In *International Joint Conference on Neural Networks*, pages 889–894, Montreal, Canada, 2005.

[Can86] J. Canny. A computational approach to edge detection. *PAMI*, 8:679–698, 1986.

[CDFB04] G. Csurka, C.R. Dance, L. Fan, and C. Bray. Visual categorization with bag of keypoints. In *Proceedings of the European Conference on Computer Vision*, Prague, 2004.

[CH03] M. Carcassoni and E. R. Hancock. Spectral correspondence for point pattern matching. *Pattern Recognition*, 36:193–204, 2003.

[Cha01] D. Chandler. Semiotics for beginners. http://www.aber.ac.uk/media/Documents/S4B/sem08.html, 2001.

[Chu97] F. R. Chung. Spectral graph theory. *American Mathematical Society*, 92, 1997.

[CK01] C. Cyr and B. Kimia. 3d object recognition using similarity-based aspect graph. In *Proceedings of the International Conference on Computer Vision*, pages 254–261, 2001.

[CM95] J. L. Crowley and J. Martin. Experimental comparison of correlation techniques. In *International Conference on Intelligent Autonomous Systems*, 1995.

[CR00] H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[CWT00] T. F. Cootes, K. N. Walker, and C. J. Taylor. View-based active appearance models. In *International Conference on Automatic Face and Gesture Recognition*, pages 227–232, 2000.

[DAOV06] E. Delponte, E. Arnaud, F. Odone, and A. Verri. Analysis on a local approach to 3d object recognition. In *DAGM Symposium on Pattern Recognition*, 2006.

[DIOV05] E. Delponte, F. Isgrò, F. Odone, and A. Verri. SVD-matching using sift features. In E. Trucco and M. Chantler, editors, *Proceedings of the of the International Conference on Vision, Video and Graphics*, pages 125–132, Edinburgh, UK, 2005.

[DIOV06] E. Delponte, F. Isgrò, F. Odone, and A. Verri. Svd-matching using sift features. *Graphical Models*, 68:415–431, 2006.

[DNOV07] E. Delponte, N. Noceti, F. Odone, and A. Verri. Spatio-temporal constraints for matching view-based descriptions of 3d objects. In *To appear in Proceeding of the International Workshop on Image Analysis for Multimedia Interactive Services*, 2007.

[DZLF94] R. Deriche, Z. Zhang, Q. T. Luong, and O. Faugeras. Robust recovery of the epipolar geometry from an uncalibrated stereo rig. In J. O. Eklundh, editor, *Proceedings of the European Conference on Computer Vision*, volume 800 of *LNCS*, pages 567–576, 1994.

[ECT99] G. Edwards, T. F. Cootes, and C. J. Taylor. Advances in active appearance models. In *Proceedings of the International Conference on Computer Vision*, 1999.

[Ede97] S. Edelman. Computational theories of object recognition. *Trends in Cognitive Sciences*, 1:296–304, 1997.

[EIP] S. Edelman, N. Intrator, and T. Poggio. Complex cells and object recognition. http://kybele.psych.cornell.edu/ edelman/archive.html.

[EIP97] S. Edelman, N. Ingrator, and T. Poggio. Complex cells and object recognition, 1997. Unpublished manustript. Cogprints.

[FA91] W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.

[FFJS06] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. Technical Report 5980, INRIA, 2006.

[Flo93] L.M.J. Florack. *The Syntactical Structure of Scalar Images*. PhD thesis, Universiteit Utrecht, 1993.

[FPZ05] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[FSMST05] J.D.R. Farquhar, Sandor Szedmak, Hongying Meng, and John Shawe-Taylor. Improving "bag-of-keypoints" image categorization: generative models and pdf-kernels. In *bohhhh*, page bohhhhhhhh, 2005.

[FTG03] V. Ferrari, T. Tuytelaars, and L. Van Gool. Wide-baseline muliple-view correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[FTG06] V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous object recognition and segmentation from single or multiple model views. *International Journal of Computer Vision*, 67(2), 2006.

[Gab46] D. Gabor. Theory of communication. *Journal of Institution of Electrical Engineering*, 1946.

[GB05] M. Grabner and H. Bischof. Extracting object representations from local feature trajectories. In *I Cogn. Vision Workshop*, 2005.

[GBE93] J.L. Gallant, J. Braun, and D.C. Van Essen. Selectivity for polar, hyperbolic, and cartesian gratings in macaque visual cortex. *Science*, 1993.

[GD05] K. Grauman and T. Darrell. Efficient image matching with distributions of local invariant features. In *CVPR*, 2005.

[GL83] G. H. Golub and C. F. Van Loan. *Matrix computations*. John Hopkins University Press, 1983.

[GPK98] N. Georgis, M. Petrou, and J. Kittler. On the correspondence problem for wide angular separation of non-coplanar points. *Image and Vision Computing*, 16, 1998.

[Gra04] M. Grabner. Object recognition with local feature trajectories. Master's thesis, Technische Unversität Graz, 2004.

[Gri90] W.E.L. Grimson. *Object recognition by computer: the role of geometric constraints*. The MIT Press, 1990.

[GSBB03] M. Galun, E. Sharon, R. Basri, and A. Brandt. Texture segmentation by multiscale aggregation of filter responses and shape elements. In *Proceedings of the International Conference on Computer Vision*, pages 716–723, 2003.

[HRRS86] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: the approach based on influence functions*. John Wiley & Sons, 1986.

[HS88] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1988.

[Hut89] D.P. Huttenlocher. Three-dimensional recognition of solid object from a two dimensional image. Technical report, M.I.T. AI Lab Memo, 1989.

[ITKS04] F. Isgrò, E. Trucco, P. Kauff, and O. Schreer. Three-dimensional image processing in the future of immersive media. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(3):288–303, 2004.

[Joh04] B. Johansson. *Low level operation and learning in computer vision*. PhD thesis, Linköping Universitet, 2004.

[JP98] M. J. Jones and T. Poggio. Multidimensional morphable models: a framework for representing and matching object classes. *International Journal of Computer Vision*, 2(29):107–131, 1998.

[JU97] S. Julier and J. Uhlmann. A new extension of the kalman filter to non linear systems. In *Int. Symp. Aerospace/Defense Sens., Sim. and Cont.*, 1997.

[Kal60] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering,*, 1960.

[Koe84] J.J. Koenderink. The structure of images. In *Biological Cybernetics*, volume 50, pages 363–370, 1984.

[KvD79] J. J. Koenderink and A. J. van Doorn. The internal representation of solid shape with respect to vision. *Biol. Cyber.*, 24:211–216, 1979.

[KvD87] J.J. Koenderink and AJ van Doorn. Representation of local geometry in the visual system. In *Biological Cybernetics*, 1987.

[Lei04] B. Leibe. *Interleaved Object Categorization and Segmentation*. PhD thesis, ETH Zurich, 2004.

[Lin93] T. Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *International Journal of Computer Vision*, 11(3):283–318, 1993.

[Lin94] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994.

[Lin96] T. Lindeberg. Scale-space: A framework for handling image structures at multiple scales. In *Proc. CERN School of Computing*, 1996.

[Lin98a] T. Lindeberg. Feature detection with automatic scale selection. Technical report, CVAP, Department of numerical analysis and computing science, 1998.

[Lin98b] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):77–116, 1998.

[LL03] I. Laptev and T. Lindeberg. Space-time interest points. In *Proceedings of the International Conference on Computer Vision*, 2003.

[LLS04] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Proceedings of the European Conference on Computer Vision*, 2004.

[LM04] Xiaoye Lu and R. Manduchi. Wide baseline feature matching using the cross-epipolar ordering constraint. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume I, pages 16–23, 2004.

[LMS06] B. Leibe, K. Mikolajczyk, and B. Schiele. Efficient clustering and matching for object class recognition. In *Proceedings of the British Machine Vision Conference*, 2006.

[LN87] P. Locher and C. Nodine. Symmetry catches the eye. In *Eye movements: from physiology to cognition*, 1987.

[Low99] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, pages 1150–1157, Corfú, Greece, 1999.

[Low01] D.G. Lowe. Local feature view clustering for 3d object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 682–688, 2001.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[LTAO03] M. Lourakis, S. Tzurbakis, A. Argyros, and S. Orphanoudakis. Feature transfer and matching in disparate stereo views through the use of plane homographies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2), 2003.

[LZ03] G. Loy and A. Zelinsky. Fast radial symmetry for detecting points of interest. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):959–973, 2003.

[Mar76] D. Marr. Early processing of visual information. *Philosophical Transactions of the Royal Society of London*, 1976.

[Mar82] D. Marr. *Vision. A Computational Investigation into the Human Representation and Processing of Visual Information*. Freeman Publishers, 1982.

[MC02] K. Mikolajczyk and C.Schmid. An affine invariant interest point detector. In *Proceedings of the European Conference on Computer Vision*, volume 1, pages 128–142, 2002.

[MC04] K. Mikolajczyk and C.Schmid. Scale & affine interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[MCUP02] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 384–393, 2002.

[Mik02] K. Mikolajczyk. *Detection of local features invariant to affine transformations*. PhD thesis, Institute national polytechnique de Grenoble, 2002.

[MN95] H. Murase and S. Nayar. Visual learning and recognition of 3d objects from appearance. *International Journal of Computer Vision*, 14(1), 1995.

[Mor77] H.P. Moravec. Towards automatic visual obstacle avoidance. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, page 584, 1977.

[Mor81] H. Moravec. Rover visual obstacle avoidance. In *Proc. of the 7th Intern. Joint Conference on Artificial Intelligence*, pages 785–790, 1981.

[MP79] D. Marr and T. Poggio. A theory of human stereo vision. In *Proceedings of the Royal Society of London*, volume B, pages 301–328, 1979.

[MS03] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume II, pages 257–263, 2003.

[MS04a] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004.

[MS04b] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[MTS+06] K Mikolajczyk, T Tuytelaars, C Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65:43–72, 2006.

[MU81] D. Marr and S. Ullman. Directional selectivity and its use in early visual processing. In *Proceedings of the Royal Society of London, Series B, Biological Sciences*, pages 151–180, 1981.

[OBV05] F. Odone, A. Barla, and A. Verri. Building kernels from binary strings for image matching. In *IEEE Transaction on image processing*, volume 14, pages 169–180, 2005.

[OM02] S. Obdrzálek and J. Matas. Object recognition using local affine frames on distinguished regions. In *Proceedings of the British Machine Vision Conference*, 2002.

[OTSM03] A. Oliva, A. Torralba, M. S.Castelhano, and J. M.Henderson. Top-down control of visual attention in object detection. In *Proceedings of the IEEE Conference on International Conference on Image Processing*, 2003.

[Pil97] M. Pilu. A direct method for stereo correspondence based on singular value decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 261–266, Puerto Rico, 1997.

[PM90] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:629–639, July 1990.

[Pol00] M. Pollefeys, editor. *3-D modeling from images*, 2000. in conjunction with ECCV 2000.

[PP00] C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, 2000.

[PV92] T. Poggio and T. Vetter. Recognition and structure from one 2d model view: observations on prototypes, object classes and symmetries. Technical report, A.I. Memo 1347, MIT, 1992.

[PV98] M. Pontil and A. Verri. Support vector machines for 3d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:637–646, 1998.

[PZ98] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *Proceedings of the International Conference on Computer Vision*, pages 754–760, 1998.

[RBK98] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 23–38, 1998.

[RSSP06] F. Rothganger, S., C. Schmid, and J. Ponce. 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *International Journal of Computer Vision*, 2006.

[SB91] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[SB92] L. S. Shapiro and J. M. Brady. Feature-based correspondence — an eigenvector approach. *Image Vision Computing*, 10, 1992.

[SC96] B. Schiele and J.L. Crowley. Object recognition using multidimensional receptive field histograms and its robustness to view point changes. In *ECCV*, 1996.

[SC00] B. Schiele and J.L. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, 36(1):31–52, 2000.

[Sin95] P. Sinha. *Perceiving and recognizing 3D forms*. PhD thesis, Massachussetts Institute of Technology, 1995.

[SLH91] G. Scott and H. Longuet-Higgins. An algorithm for associating the features of two images. *Proc. Royal Society London*, B244:21–26, 1991.

[SM97] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–534, 1997.

[SMB00] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.

[SP95] S. Sclaroff and A. P. Pentland. Modal matching for correspondence and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):545–561, 1995.

[ST94] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Conference in Conference on Computer Vision and Pattern Recognition*, pages 593–600, June 1994.

[STCW02] J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, pages 419–444, 2002.

[SZ97] C. Schmid and A. Zissermann. Automatic line matching across views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 666–671, 1997.

[SZ03] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, 2003.

[Tan97] K. Tanaka. Mechanisms of visual object recognition: monkey and human studies. *Current opinion in neurobiology*, 7:523–529, 1997.

[TC02] D. Tell and S. Carlsson. Combining appearance and topology for wide baseline matching. In *Proceedings of the European Conference on Computer Vision*, volume I, pages 68–81, 2002.

[TG00] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *Proceedings of the British Machine Vision Conference*, pages 412–425, 200.

[tHR94] B. M. ter Haar Romeny, editor. *Geometry-Driven Diffusion*, chapter Linear scale-space: I. Basic theory and II. Early visual operations, pages 1–77. Kluwer Academic Publishers, 1994.

[TMF] A. Torralba, K. Murphy, and W. Freeman. Sharing visual features for multiclass and multiview object detection. *Trans. on PAMI.* in press.

[TMF04] A. Torralba, K. Murphy, and W. Freeman. Sharing features: Efficient boosting procedures for multiclass object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[Tor03] A. Torralba. Contextual priming for object detection. *International Journal of Computer Vision*, 2003.

[TP86] V. Torre and T. Poggio. On edge detection. *PAMI*, 8(2):147–163, 1986.

[TR71] M. Thurston and A. Rosenfeld. Edge and curve detection for visual scene analysis. *IEEE Transaction Computers*, 20(5):562–569, May 1971.

[Tuc] M. Tuceryan. Perceptual grouping. http://www.cs.iupui.edu/∼tuceryan/research/ComputerVision/perceptual-grouping.html.

[TV98] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision.* Prentice Hall, 1998.

[Ull79] S. Ullman. *The interpretation of visual motion*, chapter 2 and 3. MIT Press, 1979.

[Ume88] S. Umeyama. An eigen decomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10, 1988.

[Vap98] V. Vapnik. *Statistical learning theory*. John Wiley and sons, New York, 1998.

[Vet95] J.K.M. Vetterli. *Wavelets and subband coding*. Prentice Hall, 1995.

[VJ01] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 511–518, 2001.

[VO07] A. Verri and F. Odone. *Kernel methods in bioengineering, communications and image processing*, chapter Image classification and retrieval with kernel methods, pages 325–345. Idea Group Publishing, 2007.

[VP97] T. Vetter and T. Poggio. Linear object classes and image synthesis from a single example image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):733–742, 1997.

[WCG03] C. Wallraven, B. Caputo, and A. Graf. Recognition with local features: the kernel recipe. In *Proceedings of the International Conference on Computer Vision*, page 257ff, 2003.

[Wei98] J. Weickert. *Anisotropic diffusion in image processing*. Teuber Verlag, Stuttgart, 1998.

[Wei99] Y. Weiss. Segmentation using eigenvectors: a unifying view. In *Proceedings of the International Conference on Computer Vision*, pages 975–982, 1999.

[Wer12] M. Wertheimer. Experimentelle studien über das sehen von bewegung. In *Zeitschrift fr Psychologie*, 1912.

[Wit73] L. Wittgenstein. *Philosophical Investigations*. Blackwell, London, 1973.

[Wit83] A.P. Witkin. Scale space filtering. In *Proceedings International Joint Conference on Artificial Intelligence*, pages 1019–1023, 1983.

[WvdM00] E.A. Wan and R. van der Merwe. The Unscented Kalman filter for nonlinear estimation. In IEEE *Symp. on Adapt. Sys. for Sig. Proc., Communication and Control*, 2000.

[You87] R.A. Young. The gaussian derivative model for spatial vision: Retinal mechanisms. In *Spatial Vision*, volume 2, pages 273–293, 1987.

[YP86] A. Yuille and T. Poggio. Scaling theorems for zero crossings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:15–25, 1986.

[ZDFL95] Z. Zhang, R. Deriche, O. Faugeras, and Q. T. Luong. A robust techique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78:87–119, 1995.

[ZW98] R. Zabih and J. Woodfill. A non parametric approach to visual correspondence. *PAMI*, 1998.