

Numéro d'ordre : 2010-ISAL-0099 Année 2010

Institut National des Sciences Appliquées de Lyon
Laboratoire d'InfoRmatique en Image et Systèmes d'information
École Doctorale Informatique et Mathématiques de Lyon

Thèse de l'Université de Lyon

Présentée en vue d'obtenir le grade de Docteur,
spécialité Informatique

par

Anh Phuong TA

Inexact graph matching techniques: Application to object detection and human action recognition

---

Thèse soutenue le 26 décembre 2010 devant le jury composé de :

M.	Chabane Djeraba	Professeur, Université de Lille I	Rapporteur
Mme.	Marinette Revenu	Professeur, ENSICAEN	Rapporteur
M.	Colin de la Higuera	Professeur, Université de Nantes	Examinateur
M.	Rémi Mégret	Maître de Conférences, IPB/Université de Bordeaux	Examinateur
M.	Atilla Baskurt	Professeur, INSA Lyon	Directeur
M.	Christian Wolf	Maître de Conférences, INSA Lyon	Co-encadrant
M.	Guillaume Lavoué	Maître de Conférences, INSA Lyon	Co-encadrant

Laboratoire d'InfoRmatique en Image et Systèmes d'information
UMR 5205 CNRS - INSA de Lyon - Bât. Jules Verne
69621 Villeurbanne cedex - France
Tel: +33 (0)4 72 43 60 97 - Fax: +33 (0)4 72 43 71 17

To my wife and my son.

Acknowledgments

First of all, I would like to express my deepest and sincerest gratitude to my three thesis advisors, Dr. Christian Wolf, Dr. Guillaume Lavoué, and Pr. Atilla Baskurt for their valuable guidance and consistent encouragement throughout this research work. I would like to thank my director of thesis Pr. Atilla Baskurt for giving me an interesting research topic related to several domains such as graph matching, object detection, and human action recognition. I owe a great debt of thanks to both Dr. Christian Wolf and Dr. Guillaume Lavoué for their continuous guidance and invaluable suggestions during my PhD studies. They are always available to answer my questions, and explain to me what I did not understand. During my PhD studies I did face some personal and academic difficulties, but their assistance and encouragement hepled me to sort out these matters. Without them I would not have been able to complete this work. Thank you again, Christian, for having been a guarantor for my apartment.

I am very thankful to Pr. Chabane Djeraba and Pr. Marinette Revenu for the time they spent reviewing my thesis manuscript, and for the valuable feedback and suggestions they provided me with. I am grateful to Pr. Colin de la Higuera and Dr. Rémi Mégret for spending their valuable time as my thesis examiners.

I want to thank the Pinka company for their financial support for the first year of my PhD. It gave me the opportunity to explore an interesting industrial topic: recognizing 3D models from 2D storyboards. I am grateful to thank Pr. Jean-Michel Jolion for giving me the opportunity to participate to the CANADA project, and for his financial support. Without his support, this thesis would have been impossible.

I would like to express my gratitude to all members of the M2DisCo and Imagine teams at LIRIS, in particular Frank, Djamel, V. Vicent, Imane, Atif, Imtiaz, Yuyao, Mingyuan, for various forms of help, advice, and support over the years. My special thanks go to Jérôme for his kind help in implementing the Zernike moments, and to my office-mates/friends Kai and Yi, who helped me to surmount personal difficulties encountered over the last three years.

I want to thank my good friends Huy, Viet, Quynh, Quang Anh, ..., with whom I shared a lot of happy moments playing football on the weekend at the Gratte-ciel stadium. How empty would life be without playing football with you!

I thank Pr. Tien Son Pham, from the University of Dalat in Vietnam, for his encour-

agement during my PhD work.

Finally, I would like to thank my family members, especially my wife Dieu Hang and my son Anh Vu for their constant source of stimulation, support, understanding, and patience over the last years. Without them, this work would not have been possible. I would like to dedicate this thesis dissertation to my wife and my son.

Abstract

Object detection and human action recognition are two active fields of research in computer vision, which have applications ranging from robotics and video surveillance, medical image analysis, human-computer interactions to content-based video annotation and retrieval. At this time, building such robust recognition systems still remain very challenging tasks, because of the variations in action/object classes, different possible viewpoints, as well as illumination changes, moving cameras, complex dynamic backgrounds and occlusions. In this thesis, we deal with object and activity recognition problems. Despite differences in the applications' goals, the associated fundamental problems share numerous properties, for instance the necessity of handling non-rigid transformations. Describing a model object or a video by a set of local features, we formulate the recognition problem as a graph matching problem, where nodes represent local features, and edges represent spatial and/or spatio-temporal relationships between them. Inexact matching of valued graphs is a well known NP-hard problem, therefore we concentrated on finding approximate solutions. To this end, the graph matching problem is formulated as an energy minimization problem. Based on this energy function, we propose two different solutions for the two applications: object detection in images and activity recognition in video sequences. We also propose new features to improve the conventional Bag of words model, which is widely used in computer vision. Experiments on both standard datasets and our own datasets, demonstrate that our methods provide good results regarding the recent state-of-the-art in both domains.

Key-words: Object recognition, object localization, activity recognition, graph matching, pairwise features, multiple actions.

Contents

List of Figures

List of Tables

List of Algorithms

Part I

General introduction

This part consists of two chapters:

In chapter 1, we give a general introduction of this thesis. First, we present the problem of object detection and activity recognition, on which this thesis focuses. Then, the context and the motivation of this work are described. Finally, we present the objective of this thesis, and outline the contributions.

In chapter 2, we review the existing graph matching methods in computer vision. Our idea differs from most other surveys in that we always highlight the difference between purely graph matching methods and graph matching techniques, which can be applied in vision. Besides, we also focus on recent work, which has not been discussed in previous surveys.

Chapter 1

General introduction

Object recognition is one of the fundamental challenges in computer vision, which has been studied for more than four decades [Ull96]. The goal of object recognition is to find all instances of a given object in an image. Figure 1.1 shows an example of recognizing a dog in an image. With a little effort, humans can recognize a large variety of objects, even through severe appearance variations due to changes in pose, illumination, texture, deformation, and under occlusion. In contrast, general robust object recognition is still beyond the capacities of current artificial vision systems. The main reasons do not seem to result from a lack of sufficient research in this domain [Ull96], the problems lie in the inherent difficulties of the object recognition problem. Many approaches have been proposed to object recognition in natural scenes. Indeed, there are many important applications in the following fields, in which object recognition task is involved:

Industrial machine vision: object recognition capabilities can be used to factory automation, inspection, and quality control in an industrial environment.

Robotics and video surveillance: normally, video surveillance often imply recognizing scenarios: moving object extraction, moving object recognition and tracking, e.g. face recognition for video surveillance. Similarly, a robot agent should integrate object identification and detection algorithms to recognize new models from the physical environment. Therefore, reliable object recognition algorithms are necessary for such domains.

Medical image analysis: reliable recognition algorithms are necessary to identify and highlight the features and anomalies on medical images. For example, Radiologists usually use Computer-Aided Detection applications to review medical images.

Content-based image and video indexing and retrieval: with the increase of multime-

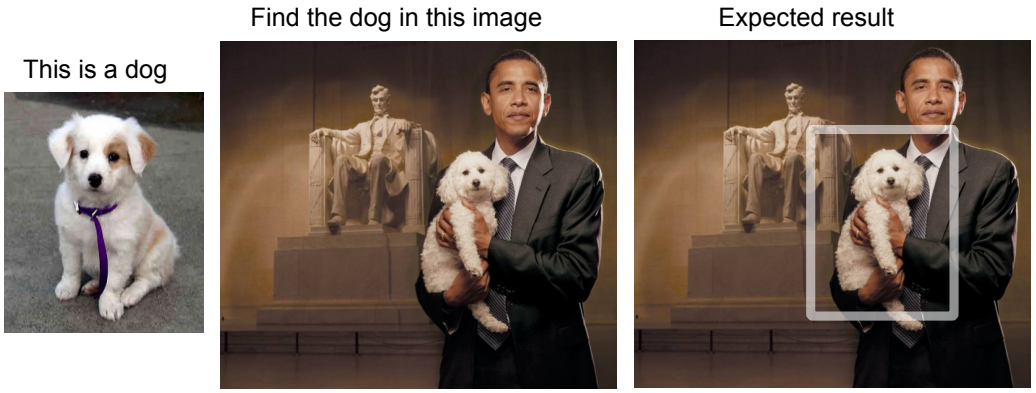


Figure 1.1: An example of object recognition. The model here is the dog (left), which will be searched in the scene image (midle).

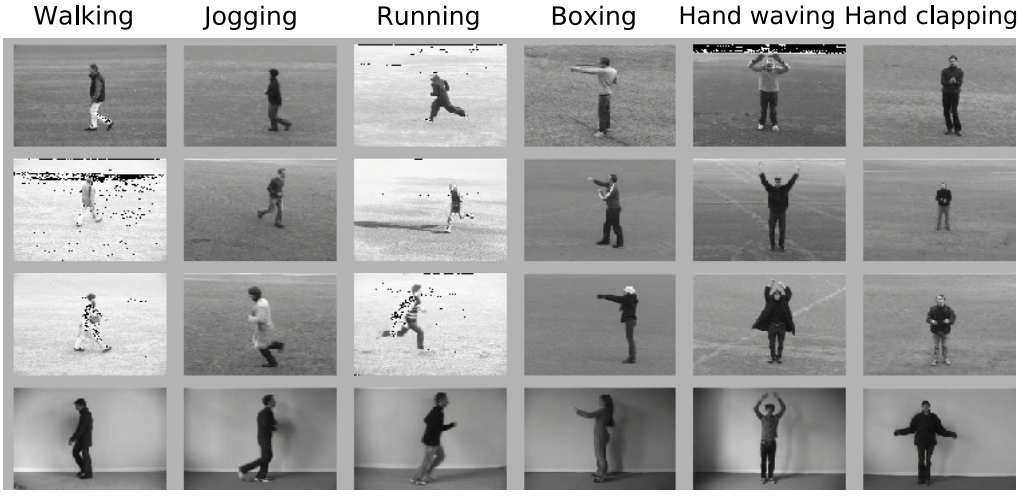


Figure 1.2: An example of human activities. These are the activities from the KTH dataset [SLC04].

dia applications from a variety of sources on the World Wide Web, efficient object recognition algorithms are needed to achieve reliable indexing methods, which help storing multimedia data (image, video, ...) in databases.

Similar to object recognition, recognizing human activities from video sequences is one of the most active fields in computer vision. For a little history, one of the earliest pioneering studies on the nature of human motion was the investigations done by the contemporary photographers Etienne Jules Marey and Eadweard Muybridge [Muy01] in the 1850s. They photographed moving subjects in all kinds of activities - walking, running, playing games, climbing stairs, etc, and revealed several interesting aspects of human and animal locomotion. The earliest computer vision attempt to motion-based approaches is the classic Moving Light Display experiments by Johansson [Joh73], who

gave a great analysis of human motion perception in the field of neuroscience.

The goal of activity recognition is to recognize common activities in life-world such as walking, jogging, jumping, climbing, etc. Figure 1.2 shows an example of such activities. We can state the problem in a simple way: given a video sequence, which contains one or more persons performing an activity, activity recognition means to recognize what activities are being performed, where and when?

Activity recognition has attracted considerable attention in recent years from the computer vision community because of its potential applications, which are described as follows:

Human-machine interaction: most of existing approaches to human-machine communication are modeled by explicitly typed, through spoken instructions, or with dedicated hardware (joysticks, game pods, etc). Implicit human-machine communication has received considerable attention recently. The main idea of such approaches is to interpret human behavior in daily activities. An application of such systems for example is detecting changes in cognitive behaviors of patients with dementia (see `http://immed.labri.fr/` for a concrete example). In such systems, a camera is attached to each patient, which captures their behaviors, and send the information to a computer. Based on installed activity recognition algorithms, the computer will analyse the captured behaviours and make decisions. Another nice example is the Microsoft's Kinect, which enables human-machine interaction in games.

Monitoring and video surveillance: example applications are detecting abnormal behaviors in a very crowded environment, e.g. detecting abandoned luggage and left luggage in an airport. Nowaday, we can encounter video surveillance systems in banks, supermarkets, parking, museums, etc. However, most of these systems are still limited to capure movies and videos from enviroment, making an urgent need for automatic analysis of human behavior. Efficient activity recognition algorithms are necessary to achieve this goal.

Content-based video annotation and retrieval: most traditional approaches to content-based video retrieval have mainly focused on visual features such as color, shape, texture, and motion [RM97]. Despite their success, these methods can represent only low-level information. In [MI00], the authors showed that high-level information such as human activities are necessary to video annotation.

In this thesis, we deal with object and activity recognition problems. Despite differences between in the application goals, both applications share the same type of pattern

matching problem. Describing a model object or a video by a set of local features, we propose a framework for shape matching, which can be applied to both object recognition and action detection. In next section, the motivation and the context of this thesis are presented.

1.1 Research context and Motivation

This thesis was partially funded by an industrial contract from the company Pinka[1], and partially by a grant from the project ANR-Canada of the LIRIS Laboratory. The research in this thesis was therefore carried out within these two projets.

Pinka is an animation production company located in Annecy, France. The goal of this project is to help creating 3D animated films from 2D storyboards. The storyboards are hand drawn by the artists in charge of the scenario of the movie, mostly with traditional non electronic pens. Then, often the modeling specialists manually create the 3D scenes based on the storyboards, using existing 3D models stored in a database, which is repetitive and very time consuming. Hence the company's desires to perform it automatically with a computer vision algorithm. Particularly, the aim of this work is to recognize each 3D model from the corresponding piece of sketch (in the storyboard plan), along with its 3D viewing angle, its scale and its rotation angle in the drawing plane, so that it can be automatically and correctly placed in the 3D scene. An illustration of the problem can be found in figure 1.3. In fact, such a problem belongs to the class of hand-drawn recognition problems, which have not received much attention until now. The practical application of this project is not restricted to the company. Indeed, in an image retrieval system, it is interesting to sketch an object and tell computer to find all images containing instances of this model from a database. Motivated by such an object retrieval application, we investigated a method for recognizing object classes in both natural and storyboard scenes from a hand-drawn model.

Our second application deals with the development of methods to abnormal activity detection in video sequences. Most of existing methods usually involve supervised algorithms, which are used to distinguish normal and abnormal activities. In this project, we focused on multiple activity recognition, i.e. the detection and localization (in time and in space) of the activities of different people performing different activities in the same video and possibly simultaneously. Due to the lack of publicly available challenging datasets for multiple activity recognition, in the case of this project, a new dataset of more than 400 videos was collected by our partners of the ANR Canada project, called

[1] http://www.pinka-prod.com/

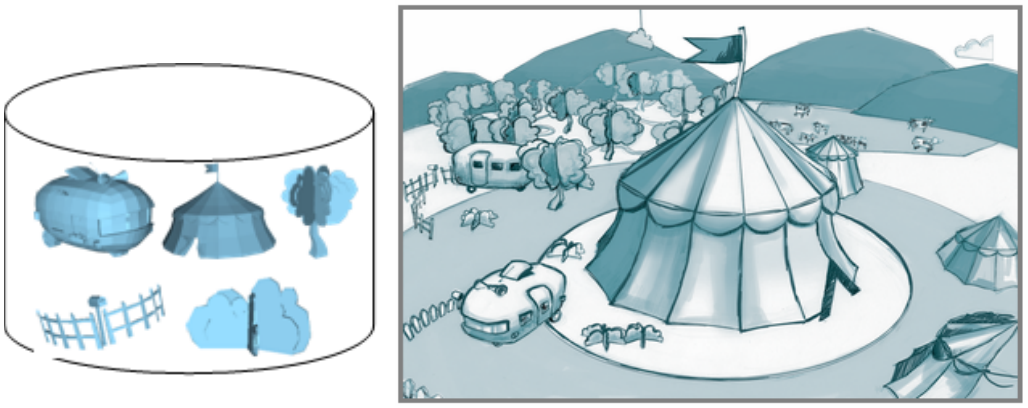


Figure 1.3: Illustration of the database of existing 3D models (left) and a hand-drawn storyboard scene (right). Our objective is to detect the 3D models in the scene storyboard.

Figure 1.4: Illustration of several activities from the canada dataset. In these videos, multiple activities occur simultaneously.

canada dataset. Figure 1.4 illustrates several activities from this dataset. One of the main contributions of this thesis is the design of a method capable of detecting multiple activities performed by several people at the same time in a video sequence. While there are many applications for activity recognition, very few methods exist for multiple activity detection.

1.2 Problem Statement and Objectives

Our goal is to recognize objects and actions of interest in images, storyboard scenes, as well as in video sequences. Although a lot of methods have been proposed for object recognition, few results are available in the literature about stroke images (e.g., storyboards). Similarly, it should be noted that very little work in literature has been done for multiple activity recognition. This is due to the fact that sketch recognition, as well as activity recognition problems face a lot of common challenges, which can be described

as follows:

Poor information. As opposed to natural images, drawings are very poor on texture information or not textured at all, and it is therefore difficult to extract local features of the target object from the scene. This makes it necessary to resort to methods taking into account a very large part of the object. Simularly, texure information in videos - although available - is not related to the requested activity and can therefore not be used.

Occlusion. Occlusion problems are inevitable in sketch recognition. Figure 1.6 gives an example of such occlusions in drawing images. Another example of occlusions occurring in storyboard scenes is shown by the trees in figure 1.3. In a video sequence, the occlusion problem usually appears in the video areas captured from crowded environments such as airports or supermarkets. There is another kind of occlusions in activity recognition, called the self-occlusion problem, which is due to motion overlapping in the same region [ATKI10]. In general, it is difficult to deal with occlusion problems in video processing.

Deformation problem. Figure 1.5 shows two examples of deformations in sketched images: a) a reduced image of a 3D caravan model (top), three different hand-drawn caravans (bottom), and b) non rigid tranformations typical to the articulated motion of living beings. Thus, successful approaches to sketch recognition must cope with the object deformation problem. In activity recognition, there are significant spatial and temporal variations for the same action when performed by different people or even by the same people.

Inter-class and intra-class variations. Inter-class and intra-class variations are the central issues of any pattern recognition problem, they characterize the difficulty of the problem. In sketch recognition, intra-class variations are often larger than inter-class variations, as shown in figure 1.8 for an example. We can see that while the variations among the bushes are quite large, the differences between the two classes (trees and bushes) are quite low. In the case of activity recognition, the intra-class variation between postures of the same class is usually high for many actions, e.g. a walking person may be from left to the right, from right to the left or directly facing the camera. Also, different people perform different actions in different ways, e.g. walking actions can differ in speed and stride length. In addition, the inter-class variation is low for many actions of different classes, e.g. slow running resembles jogging although the application might require the distintion

of these two activities. A successful recognition approach should be able to deal with these difficulties.

Scale variation. In object recognition from natural images, the scale variation problem has partially been solved by the power of the features, e.g. SIFT features from [Low99]. However, in sketch recognition, the traditional approach of assigning a local scale value to a feature is not possible due to the lack of texture (see figure 1.7 for an example). Thus, an efficient method is needed to cope with these problems. Activity recognition also features a large variabilities in the spatio-temporal scale, e.g. the duration of the same actions in different videos may not be equally long.

It is often useful to distinguish between two types of problems:

- Classification – It is defined as a problem of assigning class labels to the scene, i.e. the "recognition" (in a broad sense) of a class of objects or actions, e.g. cars, walking, etc.

- Recognition – The objective is to determine which, if any, of a given set of specific objects (e.g. my red Ferrari) or actions (e.g. gaits) appear in a given scene (image, video).

The two different types of problems are often addressed differently. In general, and in the point of view of classification, object and activity recognition often consist of three main stages:

a) Feature extraction and Representation: this stage first extracts features from objects or actions of interest, which can be either global or local (i.e., sets of primitives). Then, the object or action is represented by the features.

b) Learning models: machine learning algorithms, either generative or discriminative, are used to learn object or action models. The following learning methods are often employed: Support vector machines (SVM), Baysian networks, adaboost, Hidden Markov Models (HMM), conditional random fields (CRF), etc.

c) Recognition: this step decides whether or not object or action instances are present in the scene (image or video) using the models learned from the above step.

However, from the point of view of detection, there exist many methods, which do not need training, i.e. matching methods. Such methods usually consist of the following stages:

a) Feature extraction and Representation: similar to the same stage described above, objects or actions of interest can be represented by either global or local (i.e., sets of primitives) from the extracted features.

b) Matching: this stage performs feature matching. If the objects or actions of interest are represented globally, this stage computes distances between two vectors. Otherwise, it needs to perform feature correspondences. There are several algorithms designed for this perpose, like Hough voting, graph matching, RANSAC, etc.

c) Verification: this step decides whether or not object or action instances are present in the scene (image or video) by using a threshold chosen from experiments.

In this thesis, unless otherwise specified, we slightly relax the distinction between classfitication and recognition, and use both terms for the same kind of algorithm. In particular, the classification problem can be solved by matching a new object/activity against a set of labelled objects/activities using a k-nearest neighbor approach. This is what we chose for our technique described in chapter 7.

Depending on how to represent an object or an action of interest, i.e. using global or local features, a classification/recognition problem can be broadly classified as a global or a local approach. Global methods extract features on the whole object at once, while local methods extract features on each local primitive (i.e. interest points, edges, regions, etc.). Because there is no need to perform feature correspondence matching, global approaches have the advantage of requiring less computation compared to the local ones. They, however, may fail in many practical applications due to partial occlusion or clutter. Local methods, based on the extraction and representation of local features, overcome the global ones. Probably one of the most widely used local methods is Lowe's SIFT algorithm [Low99], which is invariant to location, scale and rotation, and robust to affine transformations. A SIFT (scale-invariant feature transform) descriptor represents a histogram of local gradient information around extrema in a pyramid of Difference of Gaussian (DoG) images (i.e. in scale space). Other local approaches consider other type of features, e.g. the Boundary Fragment Models (BFM) from [OPZ06b, SBC05], and k-adjacent segments (kAS) from [FFJS08b]. Both BFM and kAS use contour information as basic features, but they mainly differ in how to exploit spatial information, i.e. BFM method learns a codebook of contour fragments referenced to an object's centroid, while kAS approach partitions contour segments into lines and combines several adjacent lines to form a robust feature (cf. chapter 3).

Although many impressive results have been reported, most of the existing methods share a common weakness that they have been developed to deal with rigid objects

with limited intra-class variation, which makes it difficult to apply them to articulated models, as described above. Morever, some features like SIFT, BFM, and kAS do not carry enough information in a sketch image.

Based on the difficulties mentioned above, and on the analysis of state-of-the-art approaches, we can point out hereafter, the main points that need to be considered in finding the solution for both sketch and activity recognition problems:

Local features. Global features are not suitable for our context because of their lack in handling occlusions.

Spatial and/or temporal relations. Simple collections ("bag") of local features are not sufficient to recognize objects and activities of interest. We, therefore, take into account the spatial and/or temporal relationships between the local features. Spatial model has to deal with non rigid deformation.

Structured data. Hand-drawn images and activities usually undergo non rigid deformations, thus there are no global transformations (affine, rigid, isometry, etc) between the model and its instances in the sketch scenes or video scenes. This requires flexible matching between structural features, e.g. graphs which represent the data as sets of nodes and edges between the nodes.

Traditional methods often work with non-structured data (histograms or vectors), which - in the context of sets of several local features collected or local primitives - requires abandoning the spatial relationships of the features. The bag of words model is particularly widely used [CD01, VZ02, DC04, DS03, LP05]. The trade-off is a loss in discriminative pover. This reveals the problematics of this thesis: investigating new methods, which do not require any learning, for dealing with the main points described above. To this end, we formulate the recognition problem as a graph matching problem, and find the solutions through the optimization of a cost (energy) function measuring the quality of matching between two graphs as well as the attributes associated with them (node and edge attributes) - see chapter 2. Indeed, graphs are a powerful way to model features extracted from objects where nodes represent local features (local parts) and egdes represent geometric relationships (spatial and temporal relationships) between them. An example of different graph topologies is shown in figure 1.9. In this thesis, we are interested in generic graphs for representing objects or actions of interest.

Once the graphs have been constructed, an important question arising in the context of pattern recognition is how to perform efficiently graph matching, which, depending on the concrete formulation, can be difficult or NP-hard - see chapter 2. The main

issue of graph matching is the problem of computation time and memory space. This is a big problem when applying graph matching to pattern recognition because the computational complexity increases with the size of the graph. In this thesis, our main objective is to develop approximate solutions to graph matching problems for the two applications we are interested in, and for which the computation of exact solutions is hard or even impossible.

1.3 Contribution of this thesis

We focus on object recognition and activity recognition from model templates. As discussed so far, we formulate both problems as graph matching problems. In this thesis, we make several contributions toward that end, which are summarized below:

- We present a decoupled method for solving energy functions. In such a method, the problem is separated into two stages: matching and verification. One of advantages of this approach is that we do not need to solve the pairwise or higher order terms in the energy function, which is known to be NP-hard [KZ04]. This method was tested on a real dataset and a publicly available one, in the context of sketch recognition.

- Graph matching techniques have been studied intensively in the field of pattern recognition, but no method has yet been given for recognizing human activities - a straightforward application of these techniques to video recognition is difficult. While most of state-of-the-art graph matching methods concentrate on finding efficient algorithms for solving the assignment matrix (i.e., the matrix describing the correspondence between two graphs), we focus our attention to complementary problems: how to construct more expressive graphs and how to efficiently estimate edge compatibilities of the assignments. To this end, we adapt a spectral-based method for high order graph matching to activity recognition. In particular, we propose a modified version of third-order graph matching for video matching.

- In activity recognition, the temporal order of the local features is very important for characterizing action. However, such an information has been largely ignored in state of the art. We propose a new feature, which encodes both the appearance and spatio-temporal relationships of the local features. Our proposed feature has been tested on two stardard datasets for action recognition.

1.4 Thesis outline

We organize this thesis into three parts: part *I* consists of two chapters. In the first chapter, we present the goal of this thesis, and describe our approach to the problematics. Then, we describe existing methods for graph matching, on which this thesis is based.

Part *II* and *III* present our contributions to object detection and activity recognition, respectively. Each of these parts starts with an overview, and follows with our contributions to the related domains. The remainder of this manuscript is organized as follows.

Part II: Object recognition

Chapter 3 presents a review of the state of the art in object recognition. We aim at recognizing hand-drawn sketches, which consitute very poor texture information. Only contour descriptors are suitable for such applications, we therefore review only methods, which are based on contour features.

Chapter 4 presents our first contribution to object recognition. We proposed a decoupled method which consists of two steps: in a first step, we use only local features (associated to each node) to peform matching between the model and the scene graphs. Then we validate the result by verifying their geometric relationships.

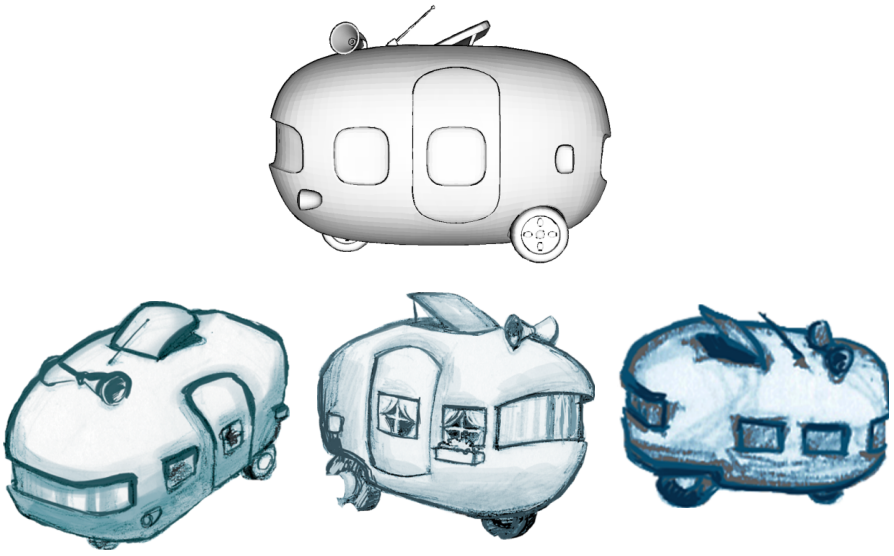
Part III: Activity recognition

Chapter 5 presents some background knowledge on action recognition. We also describe some publicly available datasets, which are used to test activity recognition systems.

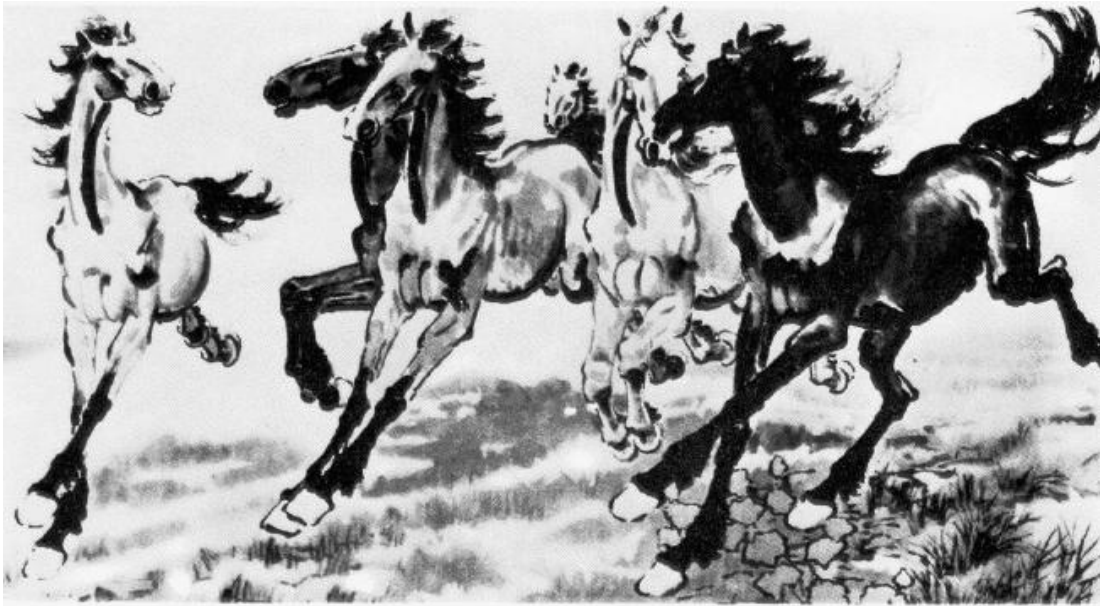
Chapter 6 introduces our improvements to the conventional Bag of Words (BoW) representation. Most of existing works in activity recognition are mainly based on the BoF paradigm, which discards spatio-temporal relations between local features. We bring some improvements to BoF models by proposing a new feature, which encodes both local appearances and geometric relationships between local points. Moreover, this work gives us motivation to apply graph matching tecniques for action recognition.

Chapter 7 presents an adaptation of high order graph matching based on spetral techniques for activity recognition. We consider the action recognition as a graph matching one, in which template videos are used as the model graphs, and videos containing activities to be recognized are the scene graphs.

Chapter 8 summarizes the contributions of this dissertation, gives some conclusions of this work and provides several future perspectives for both object and activity recognition.



a)



b)

Figure 1.5: Examples of different types of deformations in drawings: a) The deformation of caravans, which are given by the company Pinka; b) A dand drawn picture drawn by a Chinese painter, Xu Beihong, in 1942. This illustrates non-rigid tranformations typical for living beings.

a) b)

Figure 1.6: a) An example of occlusions that occur in the hand-drawn image. This picture was drawn by a Belgian surrealist artist, René Magritte, in 1957; b) Illustration of occlusions occuring in the storyboards from the company Pinka.

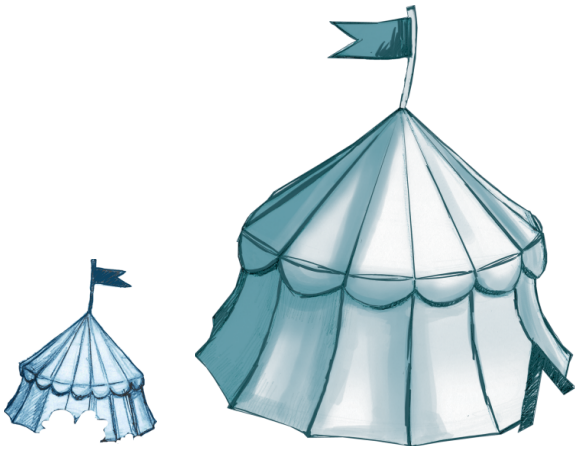


Figure 1.7: Illustration of scale variations from two drawn tents.

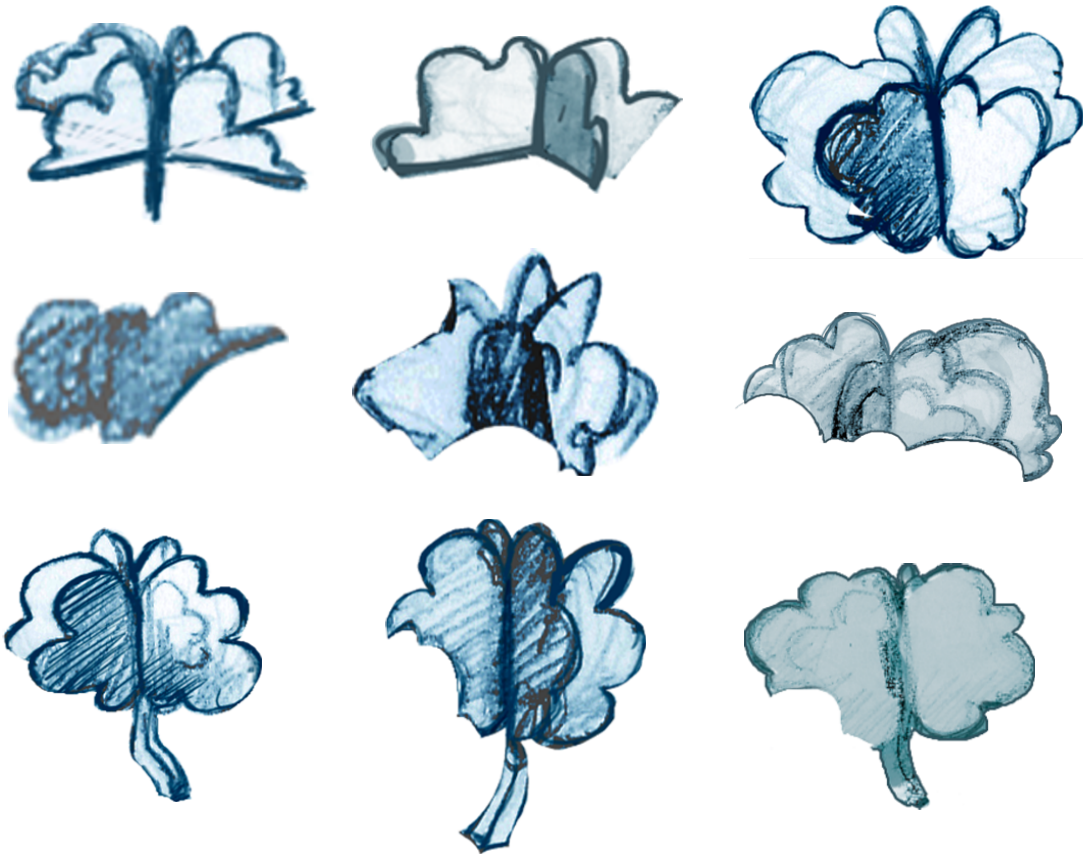


Figure 1.8: Illustration of inter-class and intra-class variations in the hand-drawn images: the first two rows are the bushes and the last row shows some trees. These hand-drawn images are from the company Pinka.

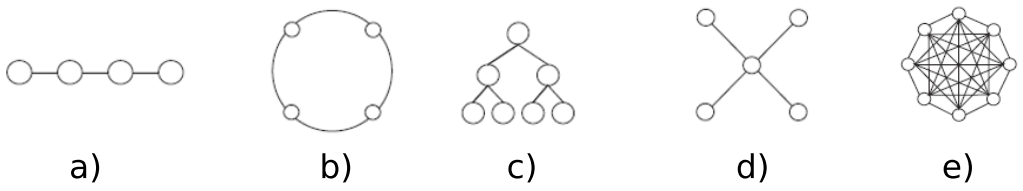


Figure 1.9: Example of graph topologies: a) Chain; b) Wheel or Ring; c) Tree; d) Star; and e) Fully connected graph. This figure is reprinted from [OGM08].

Chapter 2

Graph matching in pattern recognition

Contents

In this chapter, we first review the existing graph matching methods used in computer vision. Then, we introduce our general framework, from which we propose three different solutions for graph matching. It is not possible to review all the available literature on graph matching, which has a long history on computer science. We provide here the minimum background needed for understanding this thesis and placing it within appropriate context. The interested readers could refer to additional references [CFSV04, CFSV07] for some futher reading. Our taxonomy is in some respects based on the one introduced by Conte et al. in [CFSV04]. For each part of the taxonomy, we review the widely used methods and describe more recent work in the field. Particularly, we attempt to highlight the effects and impact of graph-based techniques and their use in pattern recognition applications. Besides, we also focus on recent work, which has not been discussed in previous surveys.

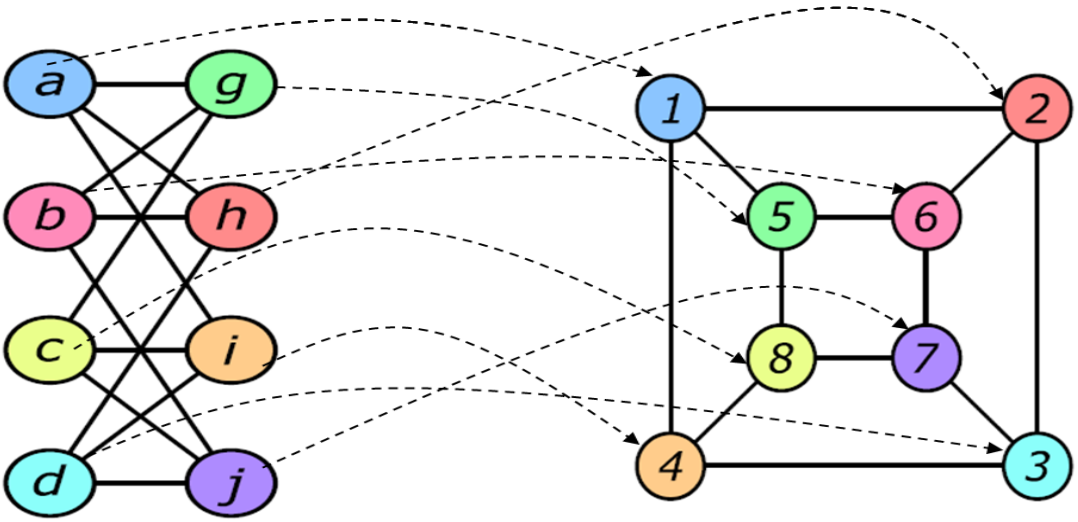
2.1 Introduction

Graphs are flexible and powerful representation techniques, which have been successfully applied in computer vision and pattern recognition [CFSV04, DBKP09, LH05b, TKR08a, ZS08]. In such domains, graphs are used to represent objects, and the recognition problem often turns into the task of graph matching, i.e searching a transformation of one graph into another. In this chapter we briefly review efficient algorithms for graph matching, including exact and inexact methods. Generally speaking, we can state the object recognition problem through using graph matching techniques as follows: let $G^m = (\mathcal{V}^m, \mathcal{E}^m, F^m)$ and $G^s = (\mathcal{V}^s, \mathcal{E}^s, F^s)$ be two graphs (the model and the scene graph, respectively), where $\mathcal{E}$ represents a set of edges, $\mathcal{V}$ a set of vertices, and F the set of their associated unary measurements (i.e appearance features extracted from local regions in the image, corresponding to the nodes $\mathcal{V}$)[1]. In the following we denote the number of nodes in both graphs as $N^m = |V^m|$ and $N^s = |V^s|$, respectively. We also denote by v_i^m and v_j^s for the $i-th$ node in the model graph and the $j-th$ node in the scene, respectively. Object recognition using local features is then casted as a graph matching problem which consists of finding correspondences between the nodes of the two graphs, in which vertices (nodes) often represent local features (objects parts), and edges represent the proximity relationships between nodes (cf. figure 2.2).

The advantage of graph matching compared to alternative techniques, e.g. RANSAC [FB81], is that non-rigid tranformations can be handled easily. Here RANSAC is an abbreviation for "RANdom SAmpling Consensus", which is an iterative method to estimate the parameters of a rigid transformation, e.g. a homography, from observed data, which contains outliers. Generally, the graph matching methods can be divided into two broad categories:

Exact methods – The first contains exact matching methods called graph isomorphism and sub-isomorphism (cf. figures 2.1 and 2.3) that require a strict correspondence between the two objects being matched or at least between subparts of them. Note that apart from the term "exact", this first category can also be known as **structural** methods. The term *structural* (i.e., exact graph matching) means that the matches are strictly based on graph structure (the set F is often null in this case). In practice, most structural graph matching methods are mainly based on the exploitation of the adjacency matrix. This matrix is an $n \times n$ matrix $A = (a_{ij})$, in which the entry $a_{ij} = 1$ if there is an edge from vertex i to vertex j, and equals 0 if otherwise (e.g.,

[1]Note that for readability or according to the method at hand, we sometimes omit the features and denote $G^m = (\mathcal{V}^m, \mathcal{E}^m)$ and $G^s = (\mathcal{V}^s, \mathcal{E}^s)$ for the model and the scene graph, respectively.



$$\begin{pmatrix} 0&0&0&0&1&1&1&0\\ 0&0&0&0&1&1&0&1\\ 0&0&0&0&1&0&1&1\\ 0&0&0&0&0&1&1&1\\ 1&1&1&0&0&0&0&0\\ 1&1&0&1&0&0&0&0\\ 1&0&1&1&0&0&0&0\\ 0&1&1&1&0&0&0&0 \end{pmatrix} \quad \begin{pmatrix} 0&1&0&1&1&0&0&0\\ 1&0&1&0&0&1&0&0\\ 0&1&0&1&0&0&1&0\\ 1&0&1&0&0&0&0&1\\ 1&0&0&0&0&0&1&1\\ 0&1&0&0&1&0&1&0\\ 0&0&1&0&0&1&0&1\\ 0&0&0&1&1&0&1&0 \end{pmatrix}$$

(i) (ii)

Figure 2.1: Exact and structural graph matching: node positions and features are not used. There exists an isomorphism f between these two graphs: $f(a) = 1, f(b) = 6, f(c) = 8, f(d) = 3, f(g) = 5, f(h) = 2, f(i) = 4, f(j) = 7$. The corresponding adjacency matrices are also listed at the bottom.

see figure 2.1).

Inexact methods – The second category defines inexact matching methods, where a strict correspondence between the two graphs being compared does not need to be found. In a real application, exact matching methods are often inapplicable due to the complexity of shape, distortions or errors in the underlying data. On the other hand, inexact matching algorithms can deal with errors and noise, thus inexact graph matching is also referred to as error-tolerant graph matching. These methods are often **attributed**: due to the existence of noise and distortion, in real-world applications of pattern matching, the graph structure itself is not sufficient to recognize patterns, thus node attributes are needed. The attributes can be symbolic (name, function, etc) or numerical (position, size, descriptors). Figure 2.2 illustrates an example of inexact graph matching, which takes into account nodes' attributes (i.e., geometric information).

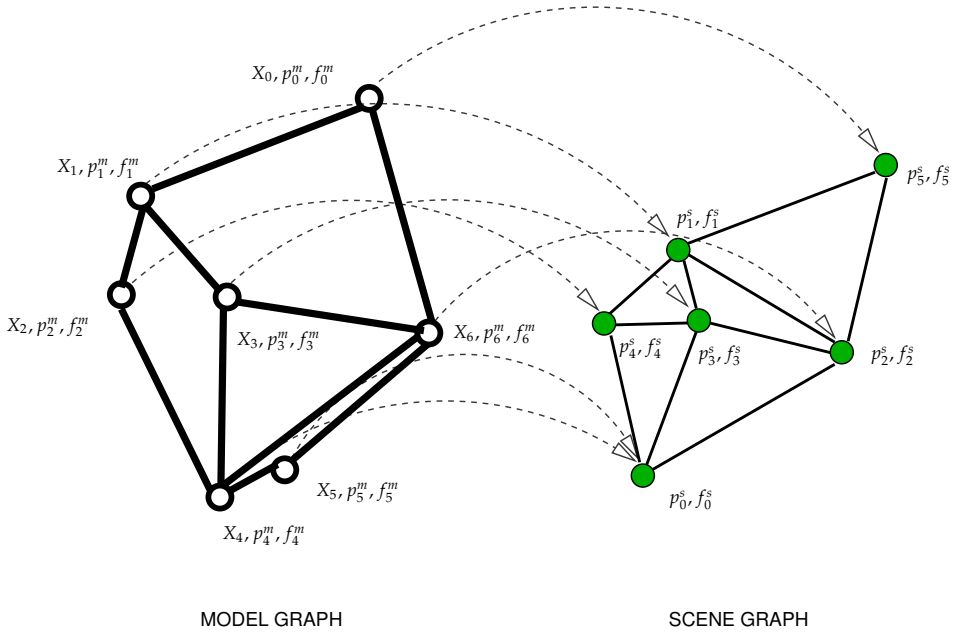


Figure 2.2: Inexact and attributed graph matching is driven by the geometry associated with the graphs. The values p_i and f_i for each node correspond, respectively, to the position and the feature vector of the patch. x_i is a variable indicating the assignment to the scene node.

Before explaining the two types of matching problems, we briefly illustrate common techniques for the creation of graphs from images.

2.2 Graph construction

Graph construction is the first step of the application of the graph matching techniques to pattern recognition. For a given image, the creation of the corresponding graph is composed of two steps:

Construct the nodes – The nodes of the graph often represent one of the following types of information:

- Points of interest (2D or 3D): the interest points represent a rich set of local points that are robust to geometric transformations of the image.
- Regions of interest.
- Edgels, i.e. contour fragments.

Construct the edges – Once the nodes are contructed, the edges of the graph are often determined as follows:

- Full-connection: a fully-connected graph is a graph in which every pair of

vertices is connected by an unique edge. This kind of graph is sometime used in inexact matching methods restricted to finding rigid transformations.

- Proximity: proximity graphs are contructed by connecting the neighbour nodes based on thresholding techniques (i.e. the distances between the nodes).
- Adjacency: the edges of the graph are etablished from the adjacency relations of image regions (e.g. in a segmented-image).
- Delaunay: the edges are constructed by connecting each node to its natural neighbors such that for each edge we can find a circle containing the two edge's nodes but not containing any other nodes. The lack of stability restricts the usefullness of this choice.

We give hereafter an overview of standard techniques for exact as well as inexact graph matching methods.

2.3 Exact graph matching

Generally, exact graph matching is an edge-preserving mapping from the vertices of one graph (G^m) to the vertices of another graph (G^s) in the sense that vertices adjacent in G^m must also be adjacent in G^s. Usually, appearance features F are not used. There are several forms of exact graph matching as follows:

- Graph isomorphism: there must be a bijective correspondence (i.e. one-to-one correspondence) between the vertices of the two graphs that preserves edges of both graphs - implying that the numbers of vertices and edges of the two graphs must be the same (cf. figure 2.1).
- Subgraph isomorphism: that is a slightly form of graph isomorphism, which requires an isomorphism to hold between one of the graphs and a subgraph of the other. In practice, this is useful for searching objects in larger scenes.
- Monomorphism: this is an instance of subgraph isomorphism, in which the mapped subgraph allows additional edges.
- Homomorphism: this is an instance of Monomorphism, in which more than one vertex of G^m may be mapped to the same vertex of G^s.
- Maximum common subgraph isomorphism (cf. figure 2.3): the two graphs are similar as they have a subpart including their adjacent edges in common. In other words, the largest part of two graphs that is identical in terms of structure, is

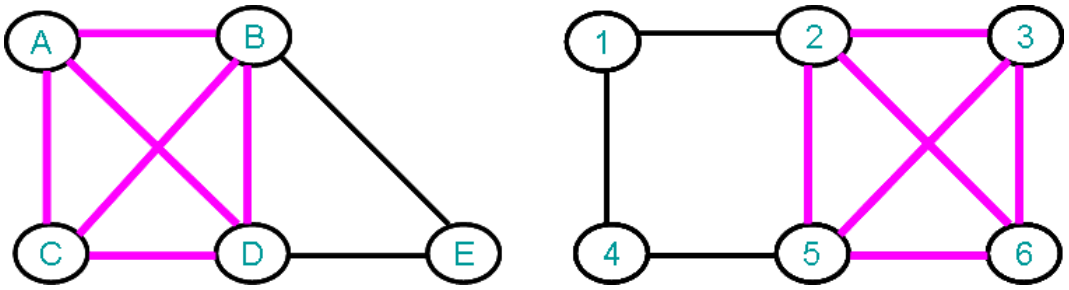


Figure 2.3: Illustration of a maximum common subgraph isomorphism between two graphs.

refered to the maximum common subgraph. In literature the problem of finding the maximum common subgraph isomorphism is usually related to the maximum clique problem.

In the following we briefly review some typical methods for exact graph matching. As mentioned before, these methods only exploit the graphical structure to establish correspondences, and do not model geometry and/or features associated to each node.

2.3.1 Tree search-based algorithms

Most of the algorithms for exact graph matching are tree-based search methods with backtracking. We present Ullmann's algorithm [Ull76], which is one of the most popular algorithm used in exact graph matching, despite of its age. It addresses all of the forms of exact graph matching mentioned above, but is less suited for maximum common subgraph isomorphism. This method is designed to find all of the isomorphisms between a given graph $G^m = (\mathcal{V}^m, \mathcal{E}^m)$ and subgraphs of a another graph $G^s = (\mathcal{V}^s, \mathcal{E}^s)$. It uses a mapping matrix M with N^m rows and N^s columns, with elements $m_{ij} \in \{0, 1\}$, where $m_{ij} = 1$ means the ith vertex in G^m corresponds to the jth vertex in G^s, such that each row contains a single 1 and each column has no or a single 1.

$$\begin{aligned} &\textstyle\sum_{j=1}^{N^s} m_{ij} = 1, \forall i \in \{1, 2, ..., N^m\} \\ &\textstyle\sum_{i=1}^{N^m} m_{ij} \in \{0, 1\}, \forall i \in \{1, 2, ..., N^m\} \end{aligned} \tag{2.1}$$

Let $A^m = [a^m_{ij}]$ and $A^s = [a^s_{ij}]$ be adjacency matrices of G^m and G^s, respectively. If a permutation matrix M can be found, which permutes rows and columns of A^s, resulting in a matrix C:

$$C = M(MA^s)^T \tag{2.2}$$

And the following expression is true:

$$(a_{ij}^m = 1) \Rightarrow (c_{ij} = 1), \forall i, j \in \{1, 2, ..., N_1\}, \tag{2.3}$$

then M specifies an isomorphism between G^m and a subgraph of G^s, i.e. if $A^m = C$ for a certain M, then the two graphs are isomorphism. The Brute-Force algorithm exhaustively evaluates every possible matrices M. As an enhancement for this algorithm, not all possible matrices M holding the constraint 2.1 are verified, but only those that map vertices V^m to vertices V^s of the same or a higher degree[2]. This is performed by constructing another matrix $M^0 = [m_{ij}^0]$ of the same size as M in accordance with:

$$m_{ij}^0 = \begin{cases} 1 & \text{if } deg(v_i^m) \leq deg(v_j^s) \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

All possible matrices M are generated by setting all but one of the 1's of each row of M^0 to 0, subject to the constraint 2.1. For each version of M, C is computed using eq. 2.2. If the condition 2.3 holds, a subgraph isomorphism has successfully been detected.

The search space for those matrices M that make C holding the constraint 2.3 is realized in a tree. Indeed, this tree has a height of N^m (i.e., $N^m = |V^m|$). Each nonterminal node at depth $d \leq N^m$ corresponds to a distinct matrix $M^{(d)}$ that is created by changing to 0 all but one of the $1's$ in a row of $M^{(d-1)}$, which was not already considered in constructing $M^{\{1,2,...,d-1\}}$. Consequently, the leaf nodes of this tree are at depth $d = N^m$, and they correspond to different matrices of M^{N^m}, which are used as M to compute C.

The algorithm presented so far is also known as the enumeration algorithm (i.e., enhanced bruce-force one), which has a high complexity. Ullmann [Ull76] proposed to reduce the computations required in the above algorithm for finding subgraph isomorphisms by introducing a constraint on the elements $m_{ij}^{(d)}$ of $M^{(d)}$:

$$\forall x \in \{1, 2, ..., N^m\}, a_{ix}^m = 1 \Rightarrow (\exists y \in \{1, 2, ..., N^s\}, (m_{xy}^d \times a_{yj}^s = 1)) \tag{2.5}$$

where $a_{ix}^m \in A^m$, i indicates the $i-th$ point in V^m, and $a_{yj}^s \in A^s$, j indicates the $i-th$ point in V^s. The $1's$ in M^d ($m_{ij}^d = 1$) define correspondences between vertices of v^m and v^s. According to the definition of subgraph isomorphism, if v_i^m corresponds to v_j^s in the isomorphism, then there must exist a vertex v_y^s that is adjacent to v_j^s, such that v_y^s corresponds to v_x^m in the isomorphism, where v_x^m is an adjacent vertex of v_i^m. Condition 2.5 is used to check this point. The expression 2.5 is a necessary and sufficient condition,

[2]The degree of a node is the number of its neighbors.

which is used as a test for subgraph isomorphism instead of using condition 2.1. In the tree search, 2.5 is tested for every element $m_{ij}^d = 1$. If the condition is not satisfied, then all M based on M^d cannot lead to a subgraph isomorphism, and therefore m_{ij}^d is set to 0. This way eliminates some of the $1's$ from the matrices of M^d, thus reducing the number of successor nodes to evaluate in the tree search.

Tree-based search is also used to find the maximum common subgraph isomorphismis between two graphs. In general the maximum common subgraph problem is related to the maximum clique (i.e. a fully connected subgraph) one. A typical example is the work of Bron and Kerbosh [BK73], who use tree search approach to find the maximum clique in an *association graph*, which represents node-to-node correspondences between the two given graphs. More recent algorithms, which are still based on tree search, are the VF and VF2 algorithm presented by Cordella et al. [CFSV00, CFSV01, PCFSV04]. They define a heuristic that is based on a depth-first strategy with a set of rules, which sigficantly prunes the search space. They have shown that this heuristic is fast to compute, and leads to improvement over Ullmann's algorithm. Their later VF2 algorithm [PCFSV04] reduces the memory requirement from $O(N^2)$ to $O(N)$ with respect to the number of nodes in the graphs. Another interesting tree-based algorithm is the one published by Larrosa and Valiente [LV02]. The authors reformulate the graph isomorphism as a contraint satisfaction problem (CSP), which is widely studied in discrete optimization and operational problem. Therefore they employ heuristics derived from state-of-the-art research in CSP.

2.3.2 Other techniques

Besides tree search-based algorithms, many other exact matching algorithms have been proposed in litterature, e.g. [McK81, BM97, MB99]. Probably the *Nauty* algorithm introduced by [McK81] is the most interesting method from the non tree-based search approaches. The basis idea is to transform the graphs to be matched to a canonical form, and the test for isomorphism of two graphs is performed by simply checking the equality of their canonical forms. The *Nauty* algorithm is regarded as the fastest isomorphism algorithm available today [FSV01]. It deals only with isomorphism problems, but the construction of the canonical form takes exponential time in the worst case.

Several algorithms are specifically designed to match an input graph against a library of preprocessed graphs. We can cite here a very impressive algorithm of this family proposed by Messmer and Bunke [BM97, MB99], addressing both isomorphism and subgraph isomorphism problems. The algorithm needs a preprocessing phrase, for

which a decision tree is built from the graph libary. Based on this decision tree, the consuming time to match an input graph against the whole libary is of $O(N^2)$ with respect to the graph size, which is independent of the number of graphs in the library.

2.4 Inexact graph matching

For having an isomorphism (subisomorphism) between two given graphs, exact graph matching requires that the topology together with the corresponding vertices are similar. These stringent constraints make exact graph matching applicable to only a few applications. In many applications, these constraints need to be relaxed to deal with intrinsic variability of the object class, presence of noise, and occlusion problem. Moreover, the major drawback of exact graph matching methods is their high computational complexity, which limits the applicability of these approaches in complex applications. Besides, geometric information of the features (e.g. node positions), which is very important in computer vision applications, is not taken into account for exact graph matching. For these reasons, a significant number of inexact graph-matching algorithms have been proposed, dealing with a general graph structure. Usually, these inexact matching algorithms do not impose the edge-preservation constraint used in exact methods. Instead, they penalize edges that do not satisfy edge-preservation constraint with a respective cost. In other words, the aim of inexact methods is to determine a mapping from one graph to another such that the overall cost of the matching is minimized. In general, in the field of machine vision, the cost function often takes into account the following information:

- Topological differences between model graph and scene graph.
- Feature distances between the unary measurements (features) F^m and F^s assigned to vertices in G^m and their corresponding vertices in G^s.
- Geometrical distances between the position of the nodes in G^m and the position of their corresponding nodes in G^s.

Summation form

Hence, the graph matching problem is often formulated as the minimization of an energy function (cost or objective function):

$$\widehat{x} = \arg\min_{x} E(x) = \lambda_f \sum_{i \in \mathcal{V}} \psi_1(i; x_i) + \lambda_d \sum_{ij \in \mathcal{E}} \psi_2(i, j; x_i, x_j) \tag{2.6}$$

Where each node i of the model graph is assigned a discrete variable x_i, $i = 1..N^m$ (the size of the model graph) which can take values from a discrete set $L = \{1..N^s\}$ (the size of the scene graph); Please note that only the arguments x are variables over which we need to optimize. ψ_1 is therefore an unary potential function, which is also called the linear assignment affinity measure, and ψ_2 a pairwise potential function. More precisely, these functions are defined as follows:

- ψ_1 is a distance function between two feature vectors associated to the model node i (i.e., f_i^m described in figure 2.2) and its corresponding scene node (i.e., $f_{x_i}^s$ described in figure 2.2).

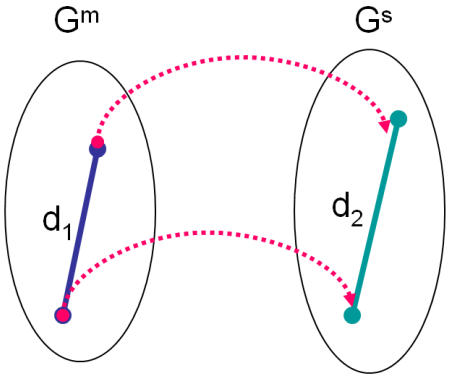


Figure 2.4: Illustration of possible pairwise measurements, which are used to verify the compatibility between a pair of neighboring nodes in the model and its corresponding scene nodes: e.g. d_1 and d_2 illustrate the coherence between edges, i.e. these two distances need to be checked for the existence of an isometry.

- ψ_2 expresses the compatibility between a pair of neighboring nodes in the model and its corresponding scene nodes, e.g. edge compatibilities such as angle, length. In our experiments, the pairwise function is taken nonnegative to ensure the tractability of $E(x)$ as the function of x for some energy optimization algorithms (see section 2.4.4).

Matrix form

As we will present below, this energy function can also be expressed in a matrix form. First of all, let us recall the linear assignment problem applied in pattern recognition. Given two graphs $G^m = (\mathcal{V}^m, \mathcal{E}^m)$ and $G^s = (\mathcal{V}^s, \mathcal{E}^s)$, which represent two sets of points or primitives, respectively, a matching between G^m and G^s is equivalent to looking for an $N^m \times N^s$ assignment matrix X such that X_{ij} is set to 1 when v_i^m is matched to v_j^s, and to 0 otherwise. Thus, the search space is the set C of assignment matrices:

$$C = \{X_{ij} \in \{0,1\}^{N^m \times N^s}\} \tag{2.7}$$

A correspondence $x = \{0,1\}^{N_1 N_2}$ is the row-wise vectorized replica of the assignment matrix $X \in \{0,1\}^{N^m \times N^s}$. The similarity measure of the node-to-node matching is represented by an affinity matrix A such that:

$$A_{ij} = \psi_1(i;j), \quad \forall i, \quad \forall j \tag{2.8}$$

where $\psi_1(.;.)$ is the same unary meaasement as described in eq. 2.6. The best assignment obtained by solving:

$$\widehat{x} = \arg\max_x(x^T a), \quad \widehat{x} \in \{0,1\}^{N_1 N_2} \tag{2.9}$$

where $a \in \mathbb{R}^{N_1 N_2}$ is a row-wise vectorized replica of A. Note that in concrete applications, several constraints can be put on eq. 2.9, e.g. to allow a one-to-one corresponding, one can impose conditions on the X so that each row and each column contain only one 1. The optimal solution for eq. 2.9 can be obtained by Brute force search for each vertex, or by the Hungarian algorithm [Mun57].

If we add pairwise constraints, the optimal assignment is achieved by solving the following binary quadratic problem:

$$\widehat{x} = \arg\max_x(x^T A x), \quad \widehat{x} \in \{0,1\}^{N_1 N_2} \tag{2.10}$$

Here the pairwise function ψ_2 is reshaped into a $N_1 N_2 \times N_1 N_2$ symmetric matrix A, called pairwise affinity matrix, with elements in $\mathbb{R}$.

Inexact graph matching problem is solved by optimizing an energy function, which is described by either Eq 2.6 or 2.10. It should be noted that each of these forms of the energy function has its own advantages and disadvantages. For instance, the matrix-form representation is naturally suited for algorithms from linear algebrac, e.g. singular value decomposition (SVD), statistical properties, e.g. PCA algorithm for computing eigen-space, etc. In constrast, the discrete-form representation is naturally suited for algorithms from discrete optimization (see section 2.4.1).

In litterature, efficient algorithms for such problem can be either optimal or approximate methods. *Optimal* inexact matching algorithms always find an exact solution if it exists. These algorithms have exponential time complexity in the worst case, which makes them unattractive for many applications. In contrast, *approximate* or *suboptimal* matching algorithms find local minima of the matching cost. Generally, there are no guarantees to reach the global minimum, but often the approximation is not very far from the global one [CFSV04]. On the other hand, these algorithms usually have poly-

nomial time complexity with respect to the number of nodes. Various formalisms have been proposed to perform inexact matching.

Alternatively, matching cost can also be based on some sort of edit operations (e.g., node insertion, node deletion, node substitution, ...), which are called *graph edit cost* [SF83]. This is an extension of a well known method, called *The Levenshtein or string-edit distance* [Lev66]. The idea is to define the cheapest operations needed to transform one graph into the other. The A^* technique is usually employed to compute graph edit cost [Bunoo]. Note that these alternative techniques do not use the energy function described above.

A large number of approaches to inexact graph matching have recently been proposed, which are based on continuous optimization, quadratic programming, and spectral decomposition of graph matrices. In the following, some important categories are briefly described.

2.4.1 Discrete optimization - MRF

The graph matching problem formulated in eq. 2.6 can also be interpreted as an energy potential function of a corresponding Markov Random Field (MRF), i.e. as the logarithm of the joint probability distribution of hidden variables X (the assignments of model nodes to scene nodes) and observed variables Y (positions of nodes, features, etc):

$$E(X = x, Y = y) = \log P(X = x, Y = y) \tag{2.11}$$

where the X are the same values as in eq. (2.6), and the observed variables Y have been omitted in (2.6). This means that methods developped for the optimization of MRFs are directly applicable for the minimization of eq. (2.6), although they may not always be the best choice. In the next two sections we review, respectively, general methods for MRF energy minimization, as well as MRF methods specifically developped for graph matching.

2.4.1.1 General opimization methods for MRFs

All general methods presented below, can also be used to optimize the energy function (2.6).

Graph-cuts – Graph-cuts only give the exact solution for submodular energy functions, but approximate versions exist for general classes. The graph cut was first introduced by Greig et al. [GS] as a combinatorial optimization method in the context of Markov random fields to recover binary images. Later, Boykov et al. [BVZ01] developed

graph cuts based technique to find a local minimum of eq. 2.6 with various smoothness constraints. In 2004, Kolmogorov and Zabih [KZ04] suggested a method to construct the graph for more general energy functional. Komodakis et al. [KTP07] proposed a fast optimization approach, called Fast-PD, based on the duality theory of Linear Programming. Rother et al. [RKLS07] introduced an approximate algorithm for general functions.

Message passing methods – Message passing algorithms such as Belief Propagation (BP) have been shown to produce excellent results for many energy functions. Probably the the first message passing algorithm for inference on Bayesian networks is belief propagation developed by Pearl [Pea88]. Based on this belief propagation, several improvements have been developed, including [WF01a, WF01b, Hes02, YFW05]. More recently, Duchi et al. [DTEK06] exploit the tractable substructures in MRFs within the context of max-product belief propagation. However, as pointed out in [KPT10], the convergence properties of these algorithms were not well analysed, i.e., one does not really know when and why these methods fail.

Other methods of this family are based on the tree-reweighted max-product (TRW-MP) algorithm, which was first introduced by Wainwright et al. in [WJW05]. This algorithm is directly related to the LP relaxation of the integer program. Kolmogorov [Kol06] improved the TRW-MP algorithms by introducing TRW-S algorithm, in which messages are updated in a sequential order. Recently, Ravikumar et al. [RAW08] solve the LP relaxation based on proximal minimization schemes using Bregman divergences. Very recently, Nikos et al. [KPT10] have proposed to decompose the LP problem into a set of smaller solvable subproblems, and then combine the solutions from these subproblems. Their method works for both the original problem (primal decomposition) or its Lagrangian dual (dual decomposition).

2.4.1.2 MRF graph matching

In this section, we will present several successful MRF methods for graph matching. In most of the methods presented above, the problem of graph matching is solved approximately. Still based on MRF optimization, but contrary to the state-of-the-art methods, where the authors try to use a complete data model and an approximate inference algorithm, Cateano et al. [CCB04, TCSB06] approximate the graphical structure (Junction tree) with a k-tree so that an exact inference algorithm can be applied. First, the model graph is approximately represented as k-tree structure (e.g., $k = 3$), motivated by the assumption that there is an isometry tranformation between the model graph and the

scene graph (e.g., see figure 2.5). The main idea of constructing the approximated structure model comes from the fact that under rigid tranformations, when we eliminate a sufficient number of edges from the full connected graph (to obtain k-tree structure in this work), the globally rigidness properly is still preserved. Figure 2.6 illustrates this point. The authors construct the potential function by comparing the relative distance

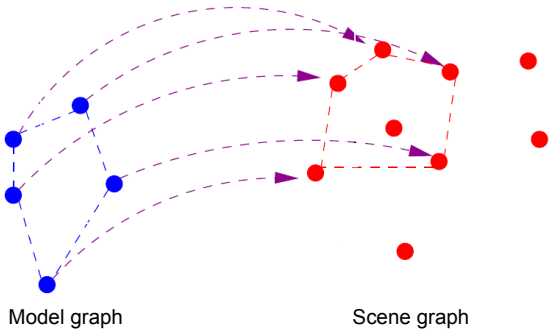


Figure 2.5: Illustration of a graph matching problem under isometry transformation, e.g from [TCSB06]. We can see in this figure, a global rigid transformation between the model and the scene graph is preserved, i.e. distance between a pair of model nodes and its corresponding scene nodes is equally.

between a pair of nodes in the model graph (i.e., k-tree structure) with its corresponding scene nodes (i.e., the assigned labels, cf. figure 2.5). This method has polynomial time, $(O(N^m.N^{s4}))^3$, which is still not applicable for large point sets. McAuley et al. [MCB08] further improved this work by trying to construct another graphical model, of which the complexity is reduced to $(N^m.N^{s3})$.

Recently, Torresani et al in [TKR08b] have presented a new graph matching optimization technique for the correspondence problem. They casted the correspondence problem as an energy minimization and introduced a novel algorithm to minimize this function based on the dual decomposition approach (DD) for graph matching. They use maxflow (graph cuts) techniques to solve the subproblem of the energy function in the case of submodular.

2.4.2 Tree search based methods

Tree search with backtracking can also be used for inexact matching. In this case the cost of the current partial matching along with the estimated matching cost for the remaining nodes direct the search process. This estimated total cost is used either to prune search paths in a branch and bound traversal, or to determine the order of branches to be

[3]The running time of this method is quite slow compared to spectral methods.

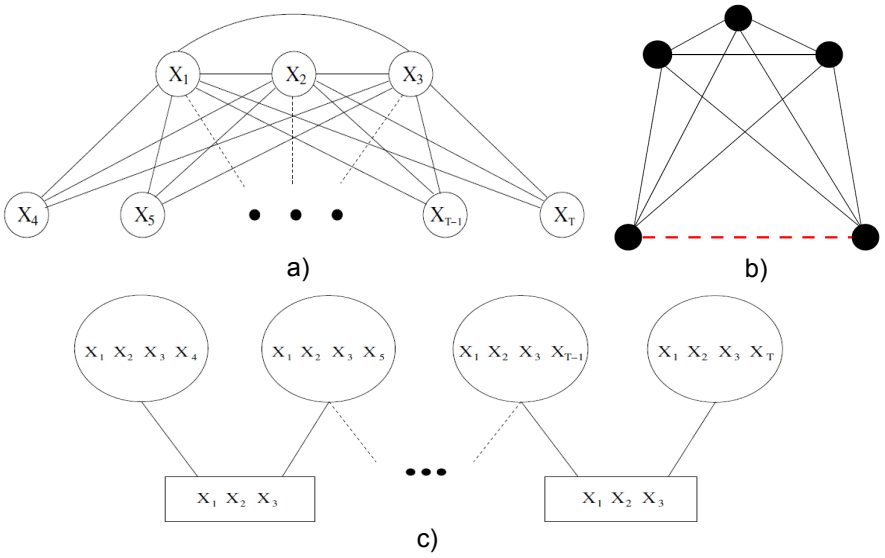


Figure 2.6: Illustration of the approximated structure model from a *full* model: a) A k-tree model for k = 3; b) The globally rigidness is preserved from the k-tree structure, i.e. every other point is contected to the basic clique (etablished from 3 non-collinear points), making a global rigid structure; c) The Junction Tree obtained from the model (a). These figures are reprinted from [TCSB06].

traversed. Examples of inexact matching based on tree search can be found in [TF79, TF83, WYC90, EF84].

2.4.3 Continuous optimization

Most of the methods mentioned so far rely on a formulation of a discrete optimization problem. Such methods usually require generating all the permutations of possible vertex matchings over the mapping matrix, or rely on special chosen of evergy potential functions. In order to overcome the limitations, different inexact matching methods based on continuous optimization have been proposed. The objective is to transform a discrete representation into a continuous one, and solve the continuous optimization problem by applying continuous optimization algorithms. These techniques guarantee finding local optimal results, which must finally be converted back from the continuous domain to the initial discrete space. We review in the following paragraphs some important inexact matching methods based on continuous optimization.

2.4.3.1 Probabilistic relaxation labeling

Probabilistic relaxation labeling is an iterative process, which tries to assign labels to set of objects using contextual constraints. Relaxation labeling methods for graph matching

have been investigated by [KH89, CKP95, HW02a, Li94, WH97, HH99]. These methods are typically iterative, and globally convergent to local minima [CKP95]. In these approaches each vertex of the first graph is to be assigned to one label out of a discrete set of possible labels that are described by a vector, which holds the estimated probabilities of correspondence to each vertex of the other graph. In the initial labeling step, these probabilities are computed based on vertex attributes, vertex connectivity, , or other information available. During the matching process, these probabilities are refined in an iterative procedure until either the labeling converges, or a maximum number of iterations is reached. The pioneering method of relaxation labeling was developped by Rosenfeld et al. [RHZ76], who applied relaxation techniques for the scene-labeling problem (i.e. image segmentation problem). In fact, this work was motivated from the relational graph matching between two sets of features introduced by Fischler and Elschlager [FE73]. Based on this pioneering research, several contributions to relaxation labeling have been proposed. Kittler and Hancock [KH89] present a probabilistic framework for relaxation labeling, in which a theoretical foundation for the update rules was proposed. The main drawback of this approach is that vertex and edge attributes are only used during initialization. Christmas et al. [CKP95] overcome this by introducing a probabilistic framework, which considers both vertex and edge attributes in the iterative process. The relaxation labeling has been also successfully applied to compute edit distances between trees [TH03]. Despite their success, relaxation methods suffer from increasing point set sizes, as well as from the choice of the starting solution (i.e., initialization) [CKP95].

2.4.3.2 The weighted graph matching problem (WGM)

A weighted graph is a graph, where each edge has been assigned a real nonnegative number, called the weight of the edge. The adjacency matrix of a weighted graph is defined by the weights of their edges, in which the entries in the main diagonal are assigned to zero. Obviously, if all edges are weighted by a binary value, a weighted graph becomes an ordinary graph. In pattern recognition applications, edges are usually labeled by using information on vertex coordinates, e.g. vertex distances. In a weighted graph matching problem, a mapping matrix M that has real valued elements, is usually used to express the mapping of nodes between two graphs. The matching then must optimize a suitably defined goal function, which depends on the weighted edges preserved by the match. By allowing continuous values for M, the problem is usually transformed into a continuous one, which is known as a quadratic optimization prob-

lem, i.e. finding solution for eq. 2.10. One main limitation of weighted graph matching is that vertices cannot contain attributes and edges cannot have other attributes than their weight. Almohamad and Duffuaa [AD93] were among the first, who solved the quadratic problem by using the simplex algorithm. Rangarajan and Mjolsness [RM96] propose a method based on Lagrangian relaxation network for Graph Matching. Also in the same year 1996, Gold and Rangarajan [GR96] proposed a graduated assignment graph-matching algorithm, which allows only one-to-one matching. In this algorithm, a technique called *graduated nonconvexity*[4] is employed to avoid poor local optima. This method is considered as the basic method of solving the graph matching problem by minimizing an heuristic energy function.

2.4.3.3 Other techniques

Several other methods for inexact graph matching based on continuous optimization have been proposed. Examples include the fuzzy graph matching [MK99, MKC01], Kernel Methods for graph matching such as Hilbert kernel in [WW02, WWH02].

2.4.4 Spectral methods

Spectral graph theory is the theory in which graphs are studied by means of eigenvalues of their adjacency matrix. The main idea of spectral graph matching methods is based on the following observation. The eigenvalues and eigenvectors of the adjacency matrix representing a graph are invariant to vertex permutations. Thus, if two graphs are isomorphic, their adjacency matrices have the same eigenvalues and eigenvectors (i.e. the same eigendecomposition), but the inverse is not necessarily true [CK04]. Spectral methods for Graph matching have received considerable attention [Ume88, SB92, CH03, CSS07a, LH05a, SB92, WH04] due to the fact that the computation of eigenvalues/eigenvectors is a well studied problem, which has polynomial time complexity. As shown in [CH03, WH04], spectral methods are not robust for matching patterns of very different sizes. Besides, and as pointed out in [NB07], the main problem of spectral methods is that they are sensitive to structural errors, such as missing or spurious vertices.

The first work on spectral matching is due to Umeyama in 1988 [Ume88], which addresses the ismorphism and subgraph ismorphism problem, but with the assumption that the graphs to be matched must have the same number of vertices and the matching matrix must be a permutation matrix. Shapiro and Brady [SB92] proposed a method for

[4]Graduated nonconvexity is a deterministic annealing to find an approximate global solution for nonconvex optimization.

point set matching by comparing the eigenvectors of the point proximity matrix, where the proximity function is build using the Gaussian function. A more recent paper by Xu and King [XK01], proposed a solution to the weighted isomorphism problem that combines the use of Principal Component Analysis with gradient descent to find the optimum of the cost function. Spectral graph matching can be done in an hierarchical manner: first, finding a correspondence between cluster and then between vertices within the clusters [CH01, KC02]. Differently from Umeyama's method, this method does not suffer from the limitation that the graph must have the same number of nodes. We detail here some spectral based methods, which have widely been applied in pattern recognition. As already mentioned, the quadratic optimization problem in eq. (2.10) is NP-complete. However, we can obtain an approximate solution by way of spectral relaxation. Probably, one of the most recent successful spectral methods, which have been applied to solve eq. (2.10), is the one proposed by Leordeanu and Herbert [LH05a]. The authors used the principal eigenvector along with its largest eigenvalue of the pairwise affinity matrix A as the confidence to decide if the matching candidates belong to the optimal set:

$$\widehat{w} = \arg\max_{w} \frac{w^T A w}{w^T w}, \quad w \in \mathbb{R}^{N^m N^s} \tag{2.12}$$

Note that the original quadratic problem described in eq. (2.10) is now relaxed into a continuous domain (i.e. $x \in \{0,1\}^{N^m N^s}$, while $w \in \mathbb{R}^{N^m N^s}$), which is solved by computing the principal eigenvector w of A. According to the Rayleigh theorem, the solution $\widehat{w}$ of eq. (2.12) is given as ${N^s}^{1/2}$ times the eigenvector associated with the largest eigenvalue of the affinity matrix A [GVL96]. An important constraint imposed on A is that it is non-negative, which is easy to guarantee given the usually employed affinity functions. Thus, by the Perron-Frobenius theorem, $\widehat{w}$ has only non-negative coefficients, which helps to interpret the results, i.e. to find an assignment matrix from the final result. Leordeanu and Herbert [LH05a] discretize $\widehat{w}$ using a greedy algorithm (i.e., thresholding the leading eigenvectors). Note that the mapping constraints (i.e. one-to-one or one-to-many) are not considered in the spectral relaxation step (eq. 2.12), but during the discretization step. In related work, Cour et al. [CSS07a] improved this work by introducing affine constraints (eq. 2.10) into the spectral decomposition.

In most existing methods mentioned above, only unary and binary relations between vertices are taken into account. This is due to the fact that the complexity of the graph matching approaches are significantly high, making very difficult to take into account all types of dependencies between the vertices and edges. In many real application, however, unary or pairwise interactions are not sufficient — we will come back to this point

later —, and that is why many authors have recently trended to tackle with higher order matching [ZS08, DBKP09, CK10]. Duchenne et al. [DBKP09] generalize the spectral matching method from [LH05b] by replacing the affinity matrix with an affinity tensor and using a tensor-based algorithm for high order graph matching. Before going into detail of this work, we would like to say a little about what tensors and tensor-based algorithms are. In physics and mathematics, tensors are generalizations of scalars, vectors and matrices to higher orders. In general, tensors can be represented as a multidimensional array of numerical values, based on which algebraic operations generalizing matrix operations can be performed. A scalar is a tensor of rank zero (order zero), a vector is a tensor of rank one (first-order tensor), and a matrix is a second-order tensor, etc, i.e. k-th order tensor is understood as a k-tuple of subscripts. In pattern matching, a tensor might capture an n-way interaction, e.g., $H(i, j, i', j')$ is a tensor element, representing the interaction between a pair of points (i, j) and its corresponding (i', j'). Tensor-based algorithms are the algorithms that use *tensor products* as the multiplication operation, which can be applied in different contexts to vectors, matrices, etc. Note that tensor product is also referred to as *outer product*, which differs from the matrix product in that it multiplies *elements to elements*.

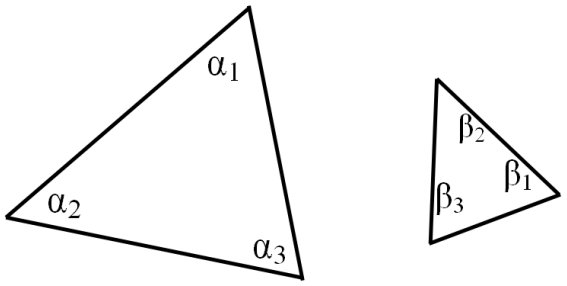


Figure 2.7: The second triangle is obtained from the first one by a scale change. In this case, pairwise affinity matching will fail, while triplet matching (based on the angles for example) will be scale invariant.

The power iteration methodis a very simple way to find the maximum eigenvalue/eigenvector of a symmetric matrix, which is useful for solving equation 2.12. Indeed, to compute the main eigenvector and its largest eigenvalue of the affinity matrix A for the pairwise assignment, we can describe a power iteration algorithm as follows:

Les us now return to the advantages of high-order matching, introduced in [DBKP09], as opposed to unary and pairwise matching from [LH05b]:

- The performance of spectral matching from [LH05b] hinges on the affinity matrix, which may be tainted with poor pairwise affinity values. In constrast, affinity

Require: affinity matrix A
Ensure: Principal eigenvector of A
1: initialize w randomly;
2: **repeat**
3: $w \leftarrow Aw$
4: $w \leftarrow w/||w||_2$
5: **until** convergence;

Algorithm 2.1: Tensor power iteration for computing the principal eigenvector w of the pairwise affinity A. This algorithm was introduced in [DBKP09].

tensor can represent higher order geometric affinities. This is illustrated in figure 2.7, where triplet matching is able to handle scale changes of triangles.

- High order matching algorithm from [DBKP09] converges to a local optimum the relaxed problem, from which we can easily extract the final result (i.e., a binary assignment), while Leordeanu and Herbert [LH05a] must discretize the eigenvectors to obtain the final result.

High order assignment is an extension of a pairwise one by considering m pairs of vertices to define the m-order affinity measure (m-order potential function):

$$\psi_m = \psi_m(i_1, i_2, ..., i_m; i'_1, i'_2, ..., i'_m) \tag{2.13}$$

Note that throughout this thesis, we denote i' as the matched vertex (node or point) of i, and for readability we also denote $i = (i, i')$ as a pair of potentially matched vertices. In [DBKP09, CK10] as well as in most tensor-based methods, the m-order affinity ψ_m is represented as a m-order tensor T_m, i.e., $\psi_m = T_m$. Thus, the optimal assignment is obtained by maximizing the following assignment affinity:

$$\widehat{x} = \arg\max_x \sum_{i_1,...,i_m} T_m(i_1, ..., i_m) x_{i_1} x_{i_2} ... x_{i_m} \quad , \quad \widehat{x} \in \{0,1\}^{N^m N^s} \tag{2.14}$$

For instance, in the triplet matching we obtain:

$$\widehat{x} = \arg\max_x \sum_{i,j,k} T_3(i, j, k) x_i x_j x_k \quad , \quad \widehat{x} \in \{0,1\}^{N^m N^s} \tag{2.15}$$

Note that similar to the matrix affinity A, the tensor T_m is nonnegative. Furthermore it is symmetric, i.e., invariant by permutation of indices in $\{i_1, ..., i_m\}$.

Finding the optimal solution for eq. 2.14 and 2.15 is known to be NP-complete, as for the binary case described before. Approximation results can be obtained by computing the rank-one approximation [ZS08, CK10] or the leading eigenvector and the largest

eigenvalue of T_m [DBKP09]. Analogue to algorithm 2.1, Duchenne et al. [DBKP09] propose a power iteration algorithm for computing the eigenvector. An example of such a power iteration algorithm for third-order tensor is given in algoritm 2.2.

Require: Tensor symmetric T_m
Ensure: Principal eigenvector of T_m
1: initialize w;
2: **repeat**
3: $\quad w \leftarrow \prod_{i=1}^{m} T_m \otimes w$
4: $\quad \forall i \; w(:,i) \leftarrow w(:,i)/||w(:,i)||_2$
5: **until** convergence;

Algorithm 2.2: Finding the principal eigenvector w of the tensor affinity $T - m$ with unit norm constraints. This algorithm is reprinted from [DBKP09].

Line 4 of the algorithm 2.2 normalizes w to obtain columns with unit ℓ^2, which is different from the Frobenius norm used in [LH05b]. Note that, as pointed out by Duchenne et al. [DBKP09], this normalization is necessary to guarantee the existence of a non-negative principal eigenvector and its largest eigenvalues. In their experiments, Duchenne et al. [DBKP09] shown that using ℓ^1 keeps the same complexity as the classical ℓ^2, but more easily to interpret the final results.

Zass and Shashua [ZS08] model the tensor T_m as the joint probability of the assignments, and they decompose it by the Kronecker product of these probabilities under the assumption that different assignments are statistically independent. Recently, Chertok and Keller[CK10] have improved this work by marginalizing T_m as a matrix and computing its rank-one approximation by eigendecomposition. These high order graph matching methods differ from the one presented by Duchenne et al. [DBKP09] in that Duchenne et al. 's algorithm always keeps the full degree of the hypergraph, i.e. does not marginalize it in a lower order.

2.4.5 Other techniques

Besides the three main classes of solution for graph matching mentioned above, we can find various important works in literature: Maciel and Costeira [MC03] construct a concave objective function and relax the search domain into its convex-hull. By constructing a metric tree representation of the graphs, Demirci et al. [DSD*04] translate the many-to-many graph matching problem into a many-to-many geometric point matching one, which is solved by the Earth Mover's Distance algorithm. Schellewald and Schnorr [SS05] uses semi-definite programming (SDP) relaxation for graph matching. As pointed out in [CSS07a], the SDP relaxation approach in [SS05] squares the problem size and does not scale to large problems.

There also exist several other techniques for inexact graph matching which will not be discussed here, such as decomposition methods [MB98, FLM00], neural networks [SA96, SY98], genetic algorithms [WcFtH97, PBB*99], methods based on bipartite matching [WZC94, ESI98, BYV00] and methods based on local properties [DTS96, OAW99, HW02b]. In the next section, we will present our contributions to graph matching prolems.

2.5 Graph matching algorithms studied in this thesis

As mentioned in chapter 1 (section introduction), in this thesis we address the inexact graph matching problem. The choice of graph matching and inexact graph matching has been motivated by the following reasons: i) both supported applications (i.e. object recognition and activity recognition) require non-rigid tranformations; ii) both applications need to deal with occlusion problem, therefore local features are used instead of global ones.

Given a model (object or action of interest), our aim is to find all its instances in a scene image (or video). To this end, the recognition problem is formulated as a graph matching problem. As mentioned before, we solve the graph matching problem by minimizing the energy function formulated in eq. 2.6 .

Our approach is somehow similar to the MRF approaches (see section 2.4.1), and most closely related to the work of Torresani et al. [TKR08b], who formulate feature correspondence as a graph matching problem by defining an objective function taking into account the similarity as well as the geometric constraints between features. Our method differs from the work of Torresani et al [TKR08b] as well as the earlier works on graph matching in 2 key points. First, most of these methods try to preserve a transformation (affine [BBM05, FJS07], rigid [TCSB06], geometric transformation [LHS07], ...) between the model and the scene. Instead, we check only the neighboring interactions for each model feature, which enables us to use a large number of features and to deal with both rigid and non-rigid transformation. Second, we exploit the angle interactions of all neighboring assignments for each model feature, as opposed to the pairwise angles between all features. We summarize our contributions for graph matching in the following paragraphs.

2.5.1 Decoupled method

Despite many existing methods and improvements for approximately solving energy minimization (e.g., graph cuts), these methods are only applicable for graphs with small

size. Moreover, the existing methods for such energy minimization have to face the problem of time complexity, making it intractable for real applications. Recall that our goal is to find an assignment x minimizing the sum of all costs of eq. 2.6, which is known to be NP-hard. We propose a fast approximative solution, namely a decoupled approach, for solving eq. 2.6. The main idea is to divide the problem into subproblems, and combine the obtained results from these subproblems. The proposed approach consists of two steps: the first step solves only the first order terms in eq. 2.6, i.e. matching unary features. The second one evaluates the matching results through a verification of the compatibility of the pairwise terms. The details of this method can be found in chapter 4.

2.5.2 High order graph matching

As already mentioned in above sections, the inexact graph matching problem can be cast in general as a quadratic assignment one, where a linear term in the objective function represents the unary potential functions (vertex compatibility) and a quadratic term encodes edge compatibility functions. The literature research mainly focuses on solving approximately the quadratic assignment problem. In constrast, we turn our attention to another viewpoint, which is to answer the question of how to estimate compatibility functions such that the matching process is speeded up and the result is not far from that based on global optimization.

Graph matching techniques have been studied intensively in the field of pattern recognition [CFSV04, DBKP09, LH05b, TKR08a, ZS08], but no method has yet been given for recognizing human activities — a straightforward application of these techniques to video recognition is difficult. In this thesis, we present a first attempt in applying graph matching techniques for activity recognition. In particular, we consider a third-order graph matching. The details of this method can be found in chapter 7.

Part II

Object recognition

This part introduces a new shape matching method for object recognition. This part consists of two chapters:

In chapter 3, we present an overview of existing methods for object recognition. As mentioned in chapter 1, in this part of our thesis, we are interested in recogninizing objects from hand-drawn models, only shape descriptors can be suitable for this context of sketch recognition. Thus, in this chapter we discuss only the shape-based representation and recognition methods, which might be used for sketch recognition.

In chapter 4, we present a new method for recognizing hand-drawn models in both natural and storyboard scenes. In our approach, each model, as well as each scene image, is represented by a set of patches. Then, graphs are constructed from these sets of patches, and a decoupled matching method is proposed to solve the graph matching problem. Experiments on a standard dataset and an industrial dataset, demonstrate that our method gives good results.

Chapter 3

Background on object recognition

Contents

As stated in chapter 1, we study the object recognition problem from hand-drawn models, from which only shape descriptors can be informative and useful for recognition compared with other descriptors such as SIFT descriptor. For that reason, in this chapter, we discuss the existing shape-based methods in object recognition. Given the diversity of available researches in the field of object recognition and detection, it is impossible to make an exhaustive bibliographical study. Therefore, we only provide the approaches that are essential for the understanding of the following chapters. We refer the readers to the surveys in [VH99, ZL04] for further details.

3.1 Introduction

Object recognition is a fundamental problem in computer vision. Recognizing real three-dimensional objects in a scene is a well known problem. We can find, in state of the art, a great variety of 2D-3D object recognition methods, some successful methods are those based on local features, e.g. from Schmid and Mohr [SM97], Jurie and Schmid [JS04], Lowe's SIFT [Low04], and Mikolajczyk05 and Schmid [MS05]. However, it is almost impossible to apply these methods for sketch retrieval or 2D-3D drawings recognition because drawings are very poor on texture information, and thus it is difficult to extract local features of the target object from the scene. Moreover, storyboard (see figure 3.1 for several examples of storyboard scenes) scene understanding still remains an open problem, few results are available in the literature about stroke images.

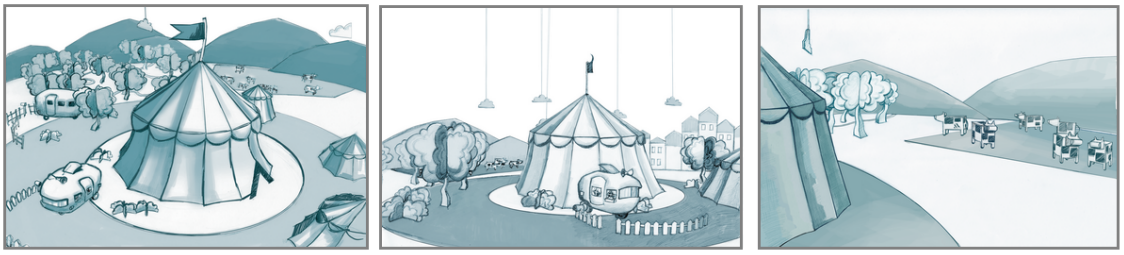


Figure 3.1: Some examples of storyboards.

Because sketches constitute a special type of image with little, if any, semantic contribution from colour and texture, only shape descriptors can be used in our context. We present hereafter a review of existing techniques for shape-based object recognition and detection.

When talking about the object detection problem, there is an ambiguity that needs to be clarified. In general, object detection implicitly includes object localization, i.e. the object's position and size also need to be found. However, object detection may sometimes also refer to answer the binary question of whether or not an object instance is present in the scene. In this thesis, we consider the object detection problem as the first situation and we refer it using the terms object recognition, i.e. the terms *object-recognition* and *object-detection* are used with the same meaning in this thesis.

The issue of shape-based object recognition has been one of the central interests in the area of computer vision for a long time. In the literature, we can find a great variety of shape representation approaches. Some excellent surveys are available: Veltkamp and Hagedoorn [VH99], Zhang and Lu [ZL04]. Zhang and Lu [ZL04] present a classification

of the existing shape representation and description techniques. They classify them in a hierarchical framework that consists of two main groups: contour-based methods and region-based methods. Then, each group is subdivided into several sub-groups such as the global approaches and the structural approaches.

There exist other shape representation methods that don't belong to contours and nor to regions, such as point-set matching. In this work, we choose to present a slightly different hierarchical structure of existing shape representation and description techniques. The whole hierarchy of this classification is shown in figure 3.2. The contour-

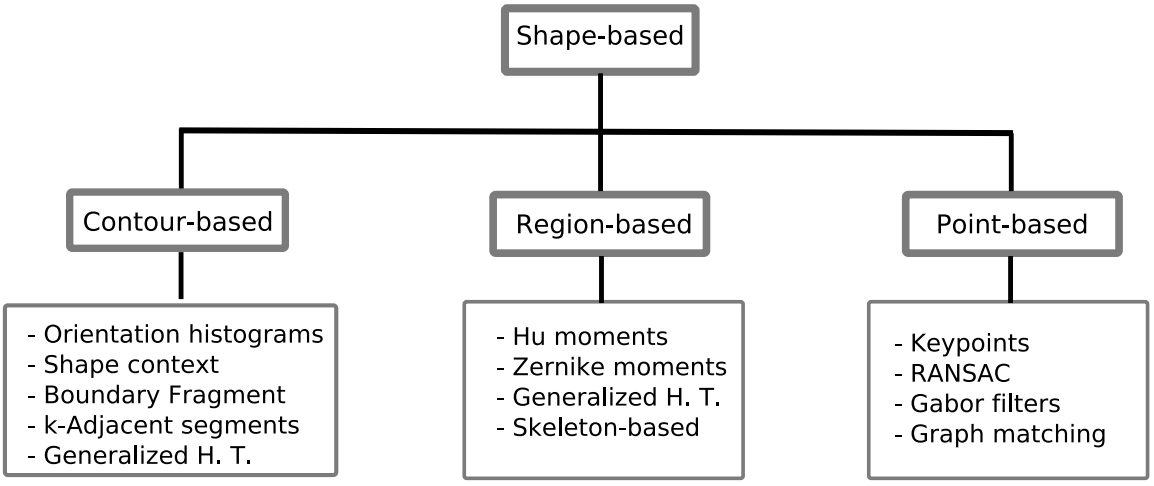


Figure 3.2: Classification of existing techniques for shape representation and description into contour-based, region-based, and point-based approaches.

based methods and the region-based methods differ on whether the shape features are extracted from the contours, or from the shape regions, while point-bases methods rely on interest points.

Of course, the classification shown here is only relative, because the hierarchical structure is only built using the characteristics of the features (contours or regions).In fact, shape representation methods can be classified based on the use of matching methods, i.e. global approaches vs. feature-based correspondence methods (local approaches). This second hierarchical structure is shown in figure 3.3.

Note that, in shape-based methods, there are several widely-used approaches such as Fourier descriptors [ZR72, Rei93] and CSS (Curvature Scale Space) descriptors [Mok95, MS98], which are not present in both above hierarchical classification. The reason is that, although these methods have low computation cost, they cannot deal with disjoint shapes, e.g. storyboards, sketches, where objects are not represented by a single closed contour. Also, for the same reason, we do not focus on active contours, the interested

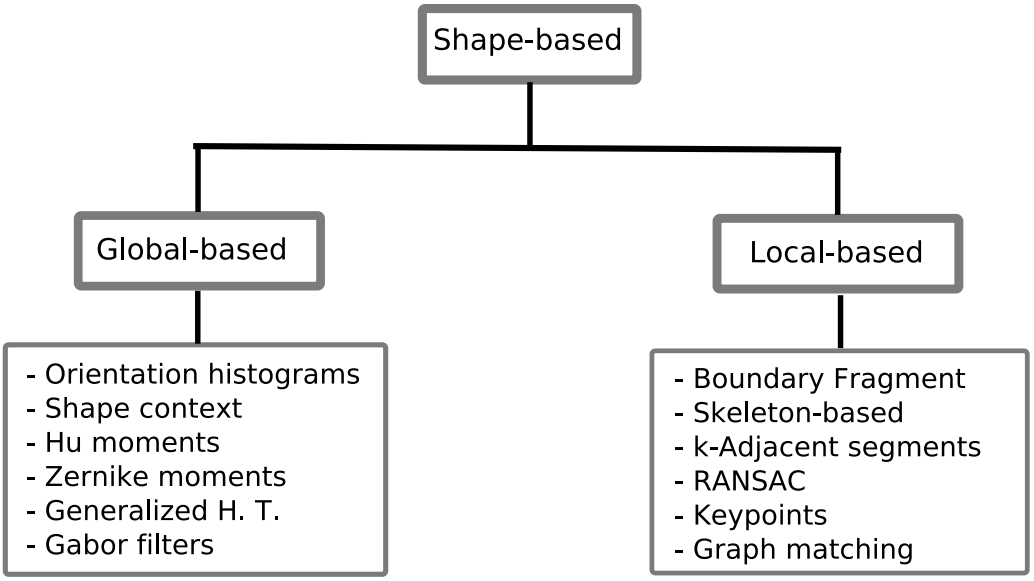


Figure 3.3: Classification of existing techniques for shape representation and description into global-based and local-based approaches.

readers are referred to the recent work on region-based active contours from [LFJB*10].

The keypoints-based methods usually rely on an interest point detector. Schmid and Mohr [SM97] propose to use local greyvalues extracted from the interest points detected by the Harris detector. These local greyvalues are invariant to scale. Lowe [Low99] introduces a scale-invariant feature transform (SIFT), which is defined as a histogram of local gradient information around extrema in a pyramid of Difference of Gaussian (DoG) images (i.e. in scale space). The SIFT descriptor is invariant to location, scale and rotation, and robust to affine transformations. Wu and Bhanu [WB95] used Gabor filter descriptors for 3D object recognition. Althrough the above methods have been successfully used in object recognition, they work on textured objects only, and therefore are not suitable for our context of sketch recognition. The RANSAC (RANdom SAmple Consensus) method is an iterative method to estimate parameters of a mathematical model (i.e. a tranformation) from the observed data. This method works well in the case of the existence of a global rigid tranformation between the model object and its instances in the scene. In our context of storyboard recognition, this method is not suited because there is no rigid transformation between the model and the scene. The graph matching techniques, which are selected for our approach, are described in chapter 2.

In the next sections, we discuss only the methods, which might be suitable for sketch retrieval. First, we present the appoaches based on regions. Then, we introduce the appoaches based on contours. And lastly, we talk about the object detection from hand-

drawn models, for which our approach gives promising results.

3.2 Region-based approaches

Region-based approaches use the entire shape regions to represent objects, i.e. exploit the information encoded inside the object regions. In general, these techniques rely on image segmentation, and do not use low level features such as textures and colors. Region-based approaches are generally less sensitive to noise compared to the contour-bases ones, but they usually need reliable segmentation algorithms to obtain image regions. We review hereafter several region-based approaches, which can be applied to sketch recognition.

3.2.1 Hu moments (geometric moments)

Moment-based invariants are the most common region-based image invariants, which have been used in many computer vision applications [MP85, RPAK88]. Classical geometric moment invariants were first introduced by Hu [Hu62]. They were derived from the theory of algebraic invariant, and consist of groups of nonlinear centralised moment expressions. The result is a set of absolute orthogonal moment invariants, which can be used for scale, position, and rotation invariant pattern identification. They were used in a simple pattern recognition experiment to successfully identify various typed characters. For a gray image of size $N \times M$, which is represented as a function $f(x, y)$, the moment of order $(p + q)$ is defined as:

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^p y^q f(x, y) \tag{3.1}$$

where p,q = 0,1,2,3, ...
The central moment μ_{pq}, which represents a normalized version of the previous one, is given as:

$$\mu_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (x - \bar{x})^p (y - \bar{y})^q f(x, y) \tag{3.2}$$

Where $\bar{x} = \frac{m_{10}}{m_{00}}$ and $\bar{x} = \frac{m_{01}}{m_{00}}$. In order to obtain scale invariant moments, the central moments is once again normalized as:

$$\eta_{pq} = \frac{\mu_{pq}}{{\mu_{00}}^{\gamma}}, \quad \gamma = 1 + (p + q)/2 \tag{3.3}$$

Using these moments, the seven Hu's moments are given below:

$$\begin{cases} M_1 = \eta_{20} + \eta_{02} \\ M_2 = (\eta_{20} - \eta 02)^2 + 4{\eta_{11}}^2 \\ M_3 = (\eta_{30} - 3\eta 12)^2 + (3\eta_{21} - \eta_{03})^2 \\ M_4 = (\eta_{30} + \eta 12)^2 + (\eta_{21} + \eta_{03})^2 \\ M_5 = (\eta_{30} - 3\eta 12)(\eta_{30} + \eta 12)[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ \quad +(3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta 12)^2 - (\eta_{21} + \eta_{03})^2] \\ M_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \mu_{12})(\mu_{21} + \mu_{03}) \\ M_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ \quad +(\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta 12)^2 - (\eta_{21} + \eta_{03})^2] \end{cases}$$

Hu demonstrated the discriminative power of these seven moments in the case of recognition of printed capital characters. Since then, there have been various improvements and generalizations of Hu's moments. The interested readers can find more details in [Fluoo]. Even recently, Rizon et Al. in [RYS*06] obtained good recognition results using Hu's moments.

3.2.2 Zernike Moments

Althrough Hu's moments are invariant to scale, translation and rotation, the main problem with them is that there are only a few invariants derived from low order moments, which is not sufficient to accurately describe shapes. Higher order invariants are difficult to derive. Besides, it is very difficult to reconstruct the image from Hu's invariants. To overcome this, Teague [Tea80] introduced Zernike moments based on the basis set of orthogonal Zernike polynomials.

Zernike polynomials were first introduced by F. Zernike in 1934 [Zer34]. The moment formulation of the Zernike polynomials [KH90] appears to be one of the most popular feature, outperforming the alternatives [TC88] in terms of noise resilience, information redundancy and reconstruction capability. Complex Zernike moments are constructed using a set of complex polynomials which form a complete orthogonal basis set defined on the unit disc in polar coordinates [Tea80]. Zernike moments are given below:

$$A_{nm} = \frac{n+1}{\pi} \sum_{x=0} \sum_{y=0} {V^*}_{nm}(x,y) P(x,y), \quad x^2 + y^2 \leq 1 \tag{3.4}$$

where $m = 0, 1, ..., \infty$ is the order of the moment and n represents the repetition, which

is subject to the below conditions:

$$m - |n| = even, |n| \leq m \tag{3.5}$$

and $*$ denotes the complex conjugate (i.e. $V^*_{n,m} = V_{-n,m}$). The Zernike polynomials are defined in polar coordinates as follows:

$$V_{nm}(x, y) = V_{nm}(\rho \cos\theta, \rho \sin\theta) = R_{nm}(\rho) \exp(jm\theta) \tag{3.6}$$

Where $R_{nm}(\rho)$ is radial polynomial given as:

$$R_{nm}(\rho) = \sum_{s=0}^{(n-|m|)/2} (-1)^s \frac{(n-s)!}{s!((n+|m|)/2-s)!((n-|m|)/2-s)!} \rho^{n-2s} \tag{3.7}$$

Here ρ and θ are, respectively, the radius and the angle of the pixel (x, y) with respect to the center of gravity of the shape.

Zernike Moments are widely used for object recognition. Teh and Chin [TC88] have presented a detailed study of orthogonal moments: Legendre moments, Zernike moments, pseudo-Zernike moments, and non-orthogonal moments: geometric moments, complex moments, rotation moments. Their results showed that Zernike Moments have the best reconstruction rate. In additional, Choksuriwong et Al. in [CLE05] made a comparison evaluation of three object invariant descriptors: Hu moments, Zernike moments and Fourier-Mellin descriptors. Their experimental results showed that Zernike moments and Fourier-Mellin are much more efficient than the Hu moments.

One of the advantages of Zernike moments is that it is easy to reconstruct the original image from them. Indeed, if we know all Zernike moments A_{nm} of $f(x, y)$ up to a certain order, we can reconstruct the image by using the following equation:

$$f'(x, y) = \sum_n \sum_m A_{nm} V(x, y) \tag{3.8}$$

Despite their good results, Zernike moment descriptors still have several shortcomings as pointed out in Zhang and Lu [ZL04]. First, the kernel of Zernike moments is complex to compute, and the shape has to be normalized into a unit disk before computing the moment features. Second, they are not scale and translation invariant. In practice, to archive translation invariance, we need to move the original image's center to the centroid before the Zernike moment's calculation.

Another drawback of Zernike moments is that they can easily fail when objects appear partially hidden (partial object occlusion) in the image or when a complex back-

ground is present. To improve these drawbacks, a combination of Zernike descriptors with a local approach based on the detection of image points of interest is proposed by Choksuriwong et Al. in [CLRM05]. In this approach, the Zernike moments are calculated in a neighborhood of each detected keypoint. As compared to the global approach, the authors showed that the use of local points of interest allows to obtain better results in the case of occlusion (the authors perform their experiments on the COIL database[1]). Recently, Revaud et al. [RLB09], members of our team at the LIRIS laboratory, have presented a modification of Zernike moment comparison for optimal similarity and rotation angle retrieval. Their experiment results outperform the state-of-the-art methods.

3.2.3 Generalized Hough Transform (GHT)

The Hough Transform (HT) [Hou62] is a technique that locates shapes in images, i.e. which matches a model against an image containing model instances. In particular, it has been used to extract lines, circles and ellipses, etc. However, in realistic applications, shapes are much more complex than lines, circles or ellipses, and thus HT cannot be used for such realistic shapes. It may be possible to decompose a complex shape into topologically simple geometric primitives, which can be described by a mathematical equation like lines, circles, etc, but this can lead to highly complex data structures. Ideally, a good solution is to develop techniques that can find arbitrary shapes using the idea of the HT. As a response, Ballard [Bal81] introduced a technique, called generalized hough transform (GHT), which can be used to recognize arbitrary shapes with unknown position, size and orientation. GHT consists of two main stages:

Initial parameterization – An R-table is generated for the shape model (see table 3.1). The R-table is a lookup table used to determine the relationship between contour coordinates and orientations, and Hough parameters (i.e. length, angles, see figure 3.4) of the object to be detected.

$\phi_1 = 0$	(r_1^1, β_1^1)	...	(r_1^n, β_1^n)
...	...	...	...
$\phi_k = \pi$	(r_k^1, β_k^1)	...	(r_k^n, β_k^n)

Table 3.1: The R-table

Figure 3.4 illustrates how to construct R-table. First, a reference point is chosen somewhere inside the object (e.g., the gravitational center). For each point (x, y) on the boundary of the shape, two parameters ϕ and r, which represent the gradient

[1] http://www1.cs.columbia.edu/cave/research/softlib/coil-100.html

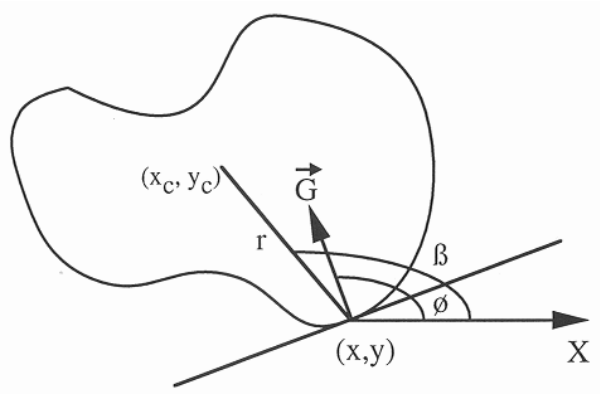


Figure 3.4: Geometric information used to form R-table.

angle and the distance from this point to the reference point, respectively (cf. figure 3.4), are computed. An additional angle β is used to describe the direction between the contour-point and the reference point. Then, R-table is indexed by ϕ for every pair (r, β) - see table 3.1.

The location of the reference point can be recomputed using the R-table:

$$\begin{cases} x_c = x + r\cos(\beta) \\ y_c = y + r\sin(\beta) \end{cases} \tag{3.9}$$

Detection – In this stage, each contour-point in the scene image will cast votes for object candidates (i.e., locations of object's reference point) using the R-table constructed above. More precisely, for every point the gradient angle ϕ is first computed and all pairs (r, β) indexed by ϕ are retrieved from R-table. Then, a reference point is generated for each pair (r, β) by using eq. 3.9. Once the votes are computed, the local maxima from the resulting accumulator array, which satisfy a specific threshold, represent the locations of the objects found in the image.

In conclusion, the generalized Hough transform is a powerful method for object recognition. There are some advantages in using GHT, it is robust to partial or slightly deformed shapes, tolerant to noise, and it can find multiple occurrences of a shape during the same processing pass. Nevertheless, it requires a lot of storage space and expensive computation.

It should be noted that in object recognition, GTH is not limited to the edge-points, it can be applied to patches, segments or to any kind of local features (e.g. Boundary Fragment models, and codebook-based methods, as we will present later). There are now a lot of methods and techniques proposed in literature for object detection improving the

generalized hough transform. Most of them concentrate on improving the voting stage. Interested readers can find more approaches in [ACS07, Cho06, USB03].

3.2.4 Skeleton-based techniques

The skeleton of an object is conceptually defined as the locus of the centers of maximal 2d disks inside the object. The skeleton of a shape aims at capturing the shape's part structure. The skeleton can be created by an algorithm called the "Medial Axis" [Blu73], which calculates the locus of the center points of all the maximal circles contained within the shape boundary. Shape similarity based on skeleton matching usually performs better than contour or other shape descriptors in the presence of partial occlusion and articulation of parts [SK05]. However, the medial axes tend to be very sensitive to boundary noise and variations. In addition, for complex shapes, the problem of computation of skeletons becomes a difficult task. Probably the most important challenge for skeleton similarity is the fact that the topological structure of skeletons of similar objects may be completely different (cf. figure 3.5). Recently, Bai et al. [BYL08] have proposed a novel graph matching algorithm to match skeleton graphs through comparing the geodesic paths between skeleton endpoints. Their experimental results showed that the proposed method is able to produce correct results in the presence of articulations, stretching, and contour deformations. Shock Graphs are a special skeleton-based technique. Siddiqi and Kimia [SSDZ98] were the first to define the shock graph concept, which is an abstraction of the skeleton onto a directed acyclic graph (DAG). The skeleton points are first labelled according to the local variation of the radius function at each point along the medial axis. Type 1 shocks form a segment of skeleton points, in which the radius function varies monotonically, as is the case for a protrusion. A type 2 arises at a neck, and is immediately followed by two type-1 branches flowing away from it in opposite directions. Type 3 shocks belong to an interval of skeleton points, in which the radius function is constant. Finally, a type 4 arises when the radius function achieves a strict local maximum. which means the boundary collapses to a single point. Figure 3.6 is an example of these shock-types. Since the shock graph forms a hierarchical representation of the shape and naturally captures its part structure, the shape matching problem can be then reduced to a tree matching problem. The shock graph benefits from the skeleton-based shape descriptor's features, including particularly the robustness to articulation and occlusion. However, like shape skeletons, the shock graph encounters difficulties in dealing with boundary noise. Most recent research work on shock graph has concentrated on graph matching techniques. For instance, Sebastian et al. [SKK04]

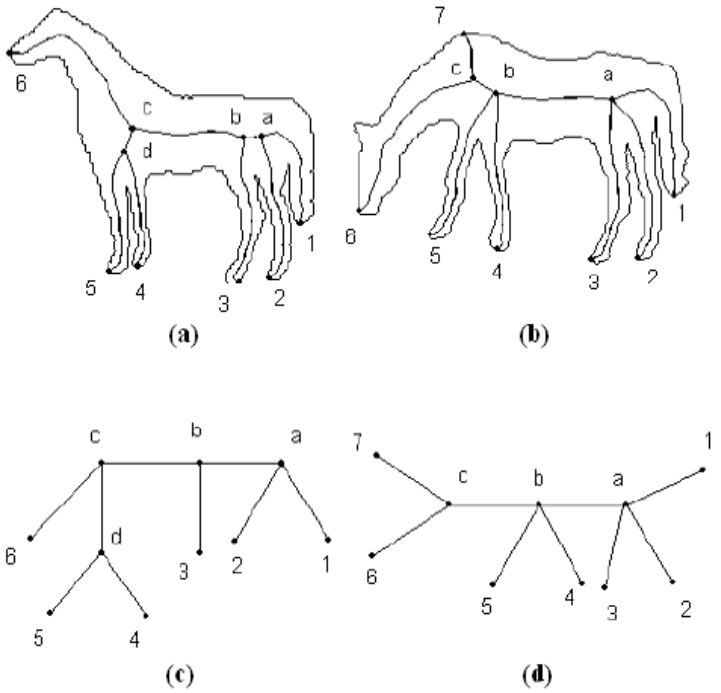


Figure 3.5: Visually similar shapes in (a) and (b) have very different skeleton graphs in (c) and (d).(reprinted from [BYL08])

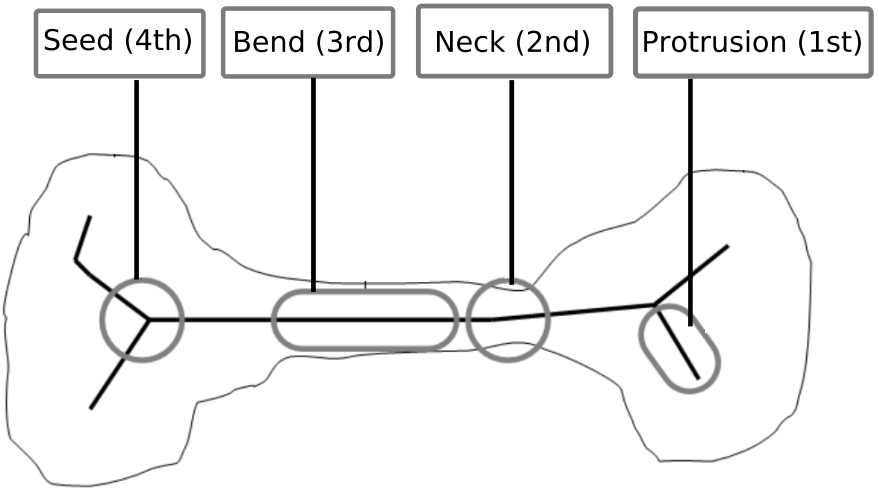


Figure 3.6: Example of four shock-types.

proposed an edit-distance between shock graphs to effectively match shapes.

3.3 Contour-based approaches

Contour-based approaches are different from the ones based on region in that only the contour information of the shape is taken into account. Contours are more efficient than appearance features, e.g. texture, for poorly textured objects. We will present several important contour-based methods in the following.

3.3.1 Shape context

Shape contexts are reminiscent of orientation histogram descriptors, which are computed from the gradient image, e.g. from Freeman et Al. in [FR95], Lowe's SIFT [Low04]. The SIFT descriptor does not work on poorly textured objects such as sketchs, drawings, storyboards. This is confirmed by our experimental trials, in which we have attempted to use SIFT descriptor for sketch recognition with very poor results.

Shape context [BMP01, BMP02] is similar to the SIFT descriptor (i.e. also create histograms of local points), but it is based on edges (contour). Shape matching using shape contexts is an improvement to traditional Hausdorff distance based methods. Figure 3.7 shows the main steps of shape context representation for a given shape.

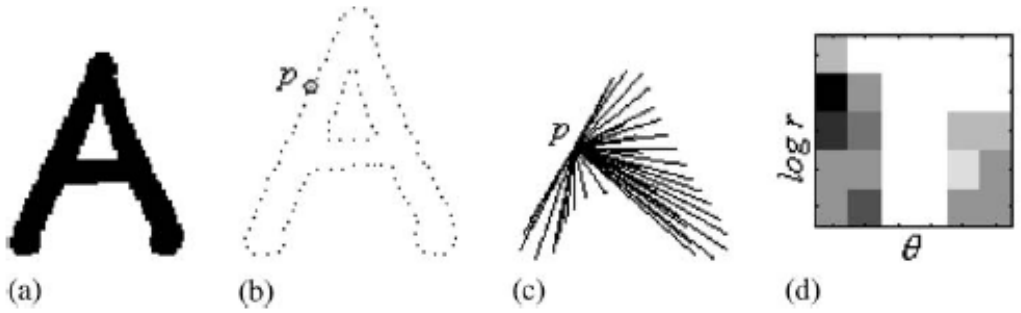


Figure 3.7: Shape context. (a) a character shape; (b) edge image of (a); (c) a point p on shape (a) and all the vectors started from p; (d) the log-polar histogram of the vectors in (c), the histogram is the context of point p. (reprinted from [BMP02])

The set of vectors going from a point (see figure 3.7.c) to all other points in the shape describe the appearance of the shape relative to this point. However, considering all these vectors as a shape descriptor for each point is not appropriate due to noise. Therefore the authors define shape context as the coarse distribution (histogram) of the rest of the shape with respect to a given point on the shape. More precisely, as shown

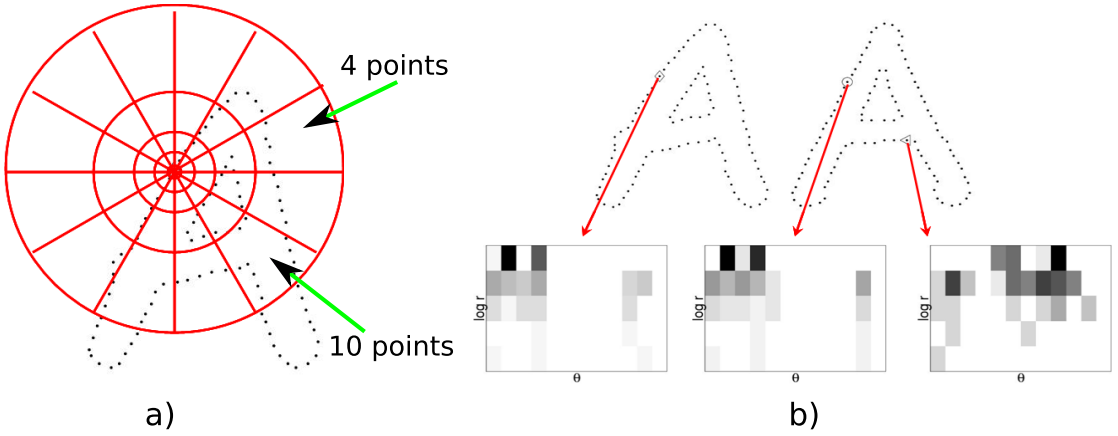


Figure 3.8: (a) Illustration of shape context computed for the point p from figure 3.7.a on a log-polar space; (b) Shape contexts of three different points.

in figure 3.8, the shape context of a point is a log-polar histogram of the coordinates of the other points measured using the current point as origin (i.e. the origin of the polar space). Belongie et al. [BMP02] suggested to use 5 bins for $\log r$ (log distance) and 12 bins for θ (polar angle), and therefore each shape context histogram is of 60 bins. In practice, this histogram is computed by counting the number of points for each bin (see figure 3.8). In term of shape context, the distance between two points from two different shapes is measured by the difference between the two corresponding histograms using χ^2 statistic. Finding the correspondence between two shapes is now equivalent to finding for each point on the first shape, exactly one point on the other shape that has the most similar shape context, i.e. the minimum χ^2 distance between the two shape contexts.

Once the correspondence between two point sets is done, Belongie et al. [BMP02] use a technique, called thin-plate-splines [Boo89] (SC+TPS), to estimate the transformation between them. TPS is a technique that allows to find coordinate (homogeneous coordinates) transformations, which maps points in one shape to another shape. Figure **??** illustrates this idea through warped gridlines. In fact, TPS models the deformations by interpolating displacements between two shapes under the assumption that all the data points are distributed on a thin, elastic plate, or spline. The results are achieved by minimizing the so-called "bending energy function" of the spline, i.e., the energy required to achieve the bend in all directions over the entire plate.

Due to its simplicity and discriminability, shape context has become quite popular recently. Grigorescu and Petkov [GP03] proprosed a new distance method called "Distance Multiset" for shape context matching. Thayananthan et al. [TSTC03] suggested including a figural continuity constraint for shape context matching via an ef-

ficient dynamic programming scheme. The interested readers can find more details in [TY04, ZM03, MDS05, LS03].

3.3.2 Boundary Fragment Model

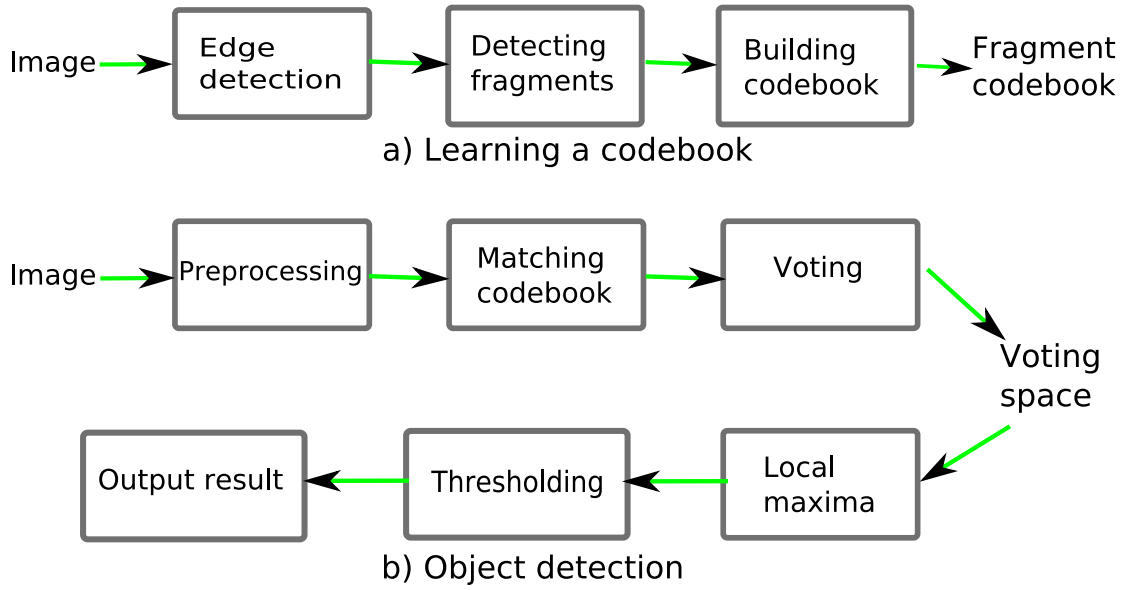


Figure 3.9: General diagram illustrating object detection using codebook.

As shown in figure 3.10, fragments are represented by disjointed boundary contours of an object. The fragment matching can handle both complete and partial shapes, i.e. occlusion. The key idea of the approaches based on contour fragments is to learn a codebook of geometrical fragments from images, and use it for detection. Figure 3.9 demonstrates the basic principle of approaches based on a codebook. Generally, these approaches consist of two main stages:

Learning a codebook – In this stage, fragments are extracted from training images. Then, clustering techniques like K-mean, k-medoids[2], Normalized Cut are employed to train a codebook. This codebook can include spatial relationships among fragments.

Detection stage – This stage is also known as a voting stage, in which fragments extracted from test images are first matched to the codebook. Next, the matched codewords vote for every possible object candidates. The non-maxima suppression and thresholding techniques are employed to find all object instances.

[2]k-medoids algorithm is very similar to k-means, but differs on the choice of datapoints as centers (i.e. medoids).

Examples include the methods of Leibe et al. [LLS04], Fergus et al. [Fer03], Vidal-Naquet and Ullman [VNU03]. These methods differ on the details of the codebook, but more fundamentally they differ in how strictly the geometry configuration of object parts is exploited. For example, Agarwal et al. [AAR04], and Vidal-Naquet and Ullman [VNU03] use loose relationships between parts (i.e. pairwise relationships), while Fergus et al. [Fer03] utilize a strongly parametrized geometric model consisting of a joint Gaussian over all the parts. There are other methods using a simple *bag of words* model [CDF*04, BHHW05, ZMLS07], where geometric relationships between object parts are discarded. In general, the approaches using geometric relations usualy use a voting mechanism such as the Generalized Hough transform for the detection stage. For instance, by casting votes for object centroid as in the Generalized Hough transform, Leibe et al. [LLS04] have obtained good detection performance on various object classes of the Caltech database. We will review in the following two typical methods, which are based on Boundary Fragment Model (BFM).

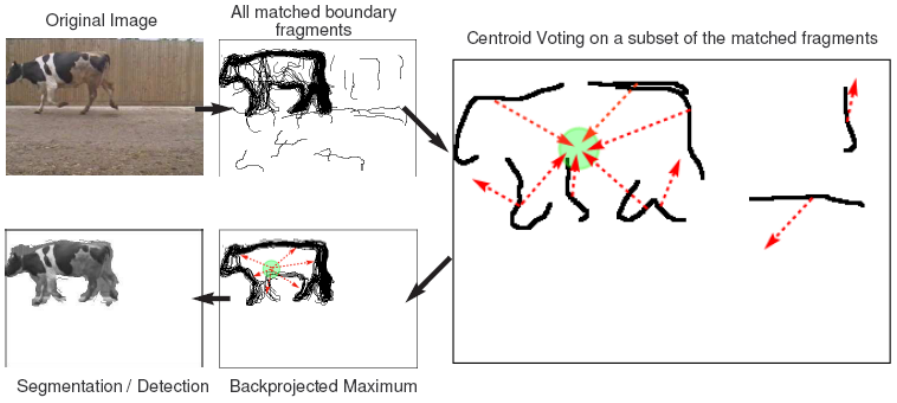


Figure 3.10: An overview of applying the BF model detector. (reprinted from [OPZ06b])

Independently, Opelt et al. [OPZ06b] and Shotton et al. [SBC05] have also explored the idea of vote for the object centroid, which has previously been used in state of the art, (e.g. from [LLS04]), but they use only the *boundaries* of the object (silhouette) and discard interior edges. They aim at detecting and localizing object classes (e.g. cows, horses), see figure 3.10 for an overview of the idea used in [OPZ06b]. In this method, the codebook consists of not only the boundary fragment, but also the objects' centroid. The boundary represents the "shape" of the object class without requiring learning the appearance (e.g. texture). This method consists of three stages:

Learning a codebook and an object detector – First, candidate boundary fragments are extracted from a training image set. Then, these candidates are optimized over the validation set. In particular, the Chamfer distance calculates a score, which is used

to find candidate boundary fragments by growing the fragment size and calculating the cost (on the validation set) at every step. Then, the best fragment, for which the cost is minimized, is chosen. After the validation process, a set of fragments which is taken as the codebook is found. The clustering algorihtm used is k-medoids. In order to reduce redundancy in the codebook, the fragments can be merged using agglomerative clustering techniques.

Once the codebook of fragments is constructed, an object detector is also learned. First, to form weak detectors (weak learners), two or three fragments are combined (into a single one) by learning combinations which fit well on all the positive validation images. Note that some fragments, which are not suitable, are filtered out. Then, a strong learner is constructed from those weak detectors using a boosting algorithm.

Object detection – This stage uses the output of the strong detector to decide whether or not an instance of object is present in the image based on thresholding technique. More precisely, this stage is composed of three steps: first, edges are extracted from test images; Secondly, the fragments of the weak detectors, which form the strong detector, are matched to this edge image. Finally, each weak detector votes with a weight in a Hough voting space. The accumulated votes in Mean-Shift-Modes that are above a specific threshold are taken as object instances.

Both works from Opelt et al. [OPZ06b] and Shotton et al. [SBC05] use boosting to select fragments from a list of candidates, but they differ on how to etablish these candidates (rectangles sampled from training segmentation masks in [SBC05], whereas Opelt et al. [OPZ06b]] construct them by varying the length of fragments so as to maximize Chamfer matching score and have a good accuracy of object centroid prediction in validation images). Another difference is the localization techniques used for finding object centroid: grid in [SBC05] and mean shift in [OPZ06b].

According to the experimental reports in [SBC05] and [OPZ06b], the BFM methods give good results. However, an important drawback of these approaches is that the relative spatial distribution of contour fragments is captured through the mediation of an object centroid, making it sensitive to partial occlusion. The lack of the relative spatial relationship among contour fragments restricts the power of the method. In addition, as pointed out in [FFJS08a], although this method can have good performance in the learned class, these kinds of fragments are harder to re-use within other applications, compared to generic features which depend only on local properties of images. Besides, the fragments used in this method are not scale-invariant and those of [SBC05] need

segmented training images to be learned, which has limited applications.

3.3.3 k-Adjacent Segment (kAS)

As mentioned above, one of the important drawbacks of the Boundary Fragment approach is its lack of genericity, making it hard to reuse within other recognition applications. To overcome this limitation, Ferrari et al. [FFJS08a] have proposed a generic method that considers the relation of local groups of contour segments. k-Adjacent Segments (kAS) have been proposed as an extension of contour segment networks [FTG06], which constitute a graph-based method for matching hand-drawings against natural images. We summarize below the techniques to build the contour segment network of the image on which kAS features are dectected.

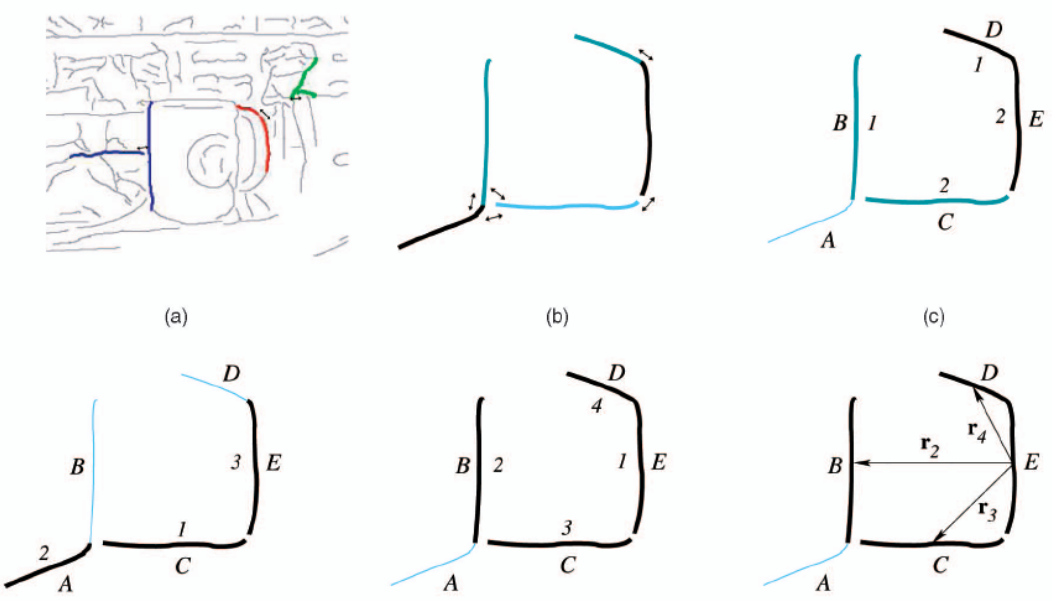


Figure 3.11: (a) An example image with three 3AS and the underlying connections (arrows).(b) Three edgel chains with five segments and their interconnections (arrows) in the network. (c) Two detected 2AS (B,C) and (D,E). (d) 3AS (C,A,E). (d) 4AS (E,B,C,D). (f) r_i vectors involved in the description of the 4AS in (d). (reprinted from [FFJS08a])

Building the contour segment network – Given a set of images, edgels are detected by the Berkeley natural boundary detector [MFM04]. Next, they are chained and the edgels' tangent orientations are computed. The resulting edgel chains are linked at their discontinuities, see figure 3.11 for an illustration. These connections take into account the fact that a contour can continue beyond the gap between two edgel chains and make it possible to capture junctions (in T, in L. .). The edgel chains are partitioned into roughly straight segments, which are then organized in the form of a network. In this

network, two segments are connected if they are adjacent. Each fragment is typically linked to several other ones, thus the final network is relatively complex. Two segments being connected if, intuitively, they can be adjacent within the contour of an object.

Detecting kAS – kAS is defined as a group of k Adjacent Segments [FFJSo8a] that are connected within the segment network. By growing k, kAS can form complex local shape structures: individual segments for k = 1; L shapes and 2-segment T shapes for k = 2; C, Y , F, Z shapes, 3-segment T shapes, and triangles for k = 3. To detect kAS, the authors suggested to use a depth-first search starting from every segment, taking into account the elimination of equivalent paths. In general, kAS representation has some advantages. The first one is the capability of accessing directly a part of the contour of an object; the second one is that they can be described in order to achieve robustness to translation and scaling-invariance.

Describing kAS – In order to compare different kAS, each kAS is described by a numerical vector. A kAS is considered as a list $P = (s_1, s_2, ..., s_k)$ of segments. For each kAS, one segment is picked as reference, and the layout of the others described relatively to that reference segment. Let segment s_1 be the reference segment (cf. figure 3.11), $r_i = (r_i^x, r_i^y)$ the vector going from the midpoint of s_1 to the midpoint of s_i, θ_i and $l_i = ||s_i||$ respectively the orientation and length of s_i, then the descriptor of P is given as follows:

$$\left(\frac{r_2^x}{N_d}, \frac{r_2^y}{N_d}, ..., \frac{r_k^x}{N_d}, \frac{r_k^y}{N_d}, ..., \theta_1, \theta_k, \quad \frac{l_1}{N_d}, , ..., \frac{l_k}{N_d}, \right) \tag{3.10}$$

where N_d is the distance between the two farthest midpoints, which is used as a normalization factor, making the descriptor scale invariant. The dimensionality of k-AS features is $n = 4 * k - 2$. After describing kAS, the codebooks of kAS are learnt from different image sets. The interest of such codebooks is evident in order to characterize a kAS thereafter we have to only find in which cluster it belongs to. This is much faster than to compare it with all the kAS. The dissimilarity measurement between two kAS P_a and P_b is defined by the following formula:

$$D(a,b) = w_r \sum_{i=2}^{k} ||r_i^a - r_i^b|| + w_\theta \sum_{i=1}^{k} D_\theta(\theta_i^a, \theta_i^b) + \sum_{i=1}^{k} |\log(\frac{l_i^a}{l_i^b})| \tag{3.11}$$

where $D_\theta \in [0, \pi/2]$ measures the difference between segment orientations, the weights w_r and w_θ were set to 4 and 2 in [FFJSo8a], respectively.

Object class detection by using kAS – We present now how to detect the objects using kAS. The main idea is to take into account the frequency and the spatial distribution

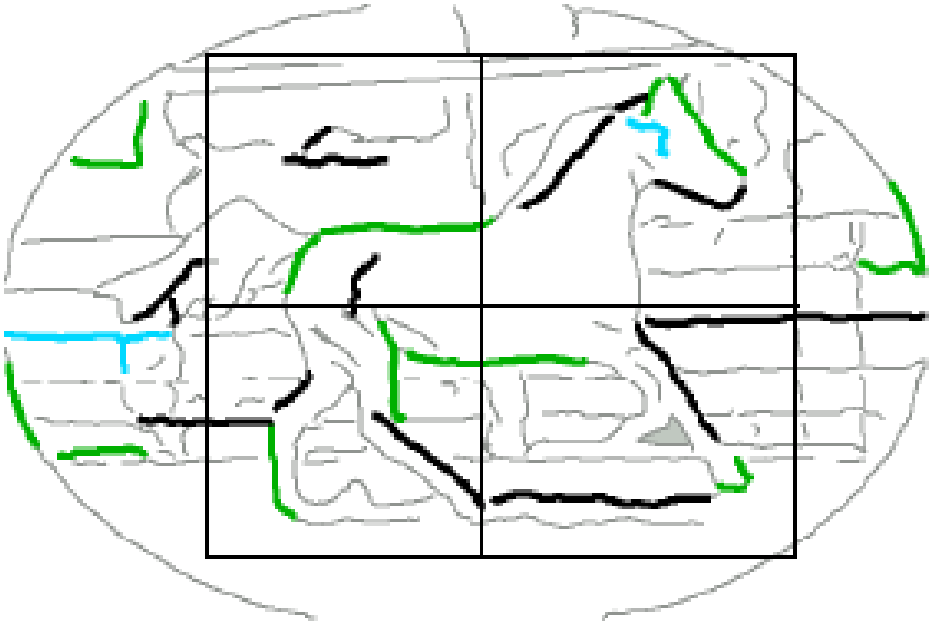


Figure 3.12: A positive training example, with bounding box, tiling, and a few kAS (k = 2). (reprinted from [FFJS08a])

of the kAS in local windows. This approach needs a training stage in which positive images are annotated by a bounding box (window) around the instances of the class. Each window is subdivided into a set of tiles (cf. figure 3.12) and then kAS histogram is computed by counting how many kAS of each type there are inside the tiles. In this method, the SVM is used for classifier training. Having trained a linear SVM window classifier, the detection stage is processed by using a sliding-window mechanism at multiple scales for computing the histogram of kAS types within a large number of image tiles. The resulting local windows, which give highest histogram scores (i.e. local maxima), represent object candidates.

Recently, Yu et al. [XYCD07] have proposed a probabilistic voting method for kAS codebook descriptors. This method can be considered as an extension of the approaches proposed by Shotton et al. [SBC05] and Opelt et al. [OPZ06a]. In this approach, the authors have used a two-layer structure of the shape codebook for object detection. The main idea is similar to other codebook-based methods, which are illustrated in figure 3.9, but they separately learn the kAS descriptors (i.e., shape codewords as they called) and geometric information (i.e., shape grammar as they called). First, TAS (Triple-Adjacent-Segments) descriptors are used as shape codewords and geometrical constraints are used as the shape grammar. The TAS codewords are learned from TASs in a training

image set. The clustering algorithm used is the Normalized Cut algorithm [XYCD07]. After obtaining the TAS codewords, the TAS grammar is learned from training images with the object delineated by bounding boxes. In this step, each TAS is associated with its position relative to the object centroids. During detection, each TAS from the test image casts votes for all possible object centroids based on the geometrical information learned and stored in the codebook. Then, object category detection is accomplished by searching for local maxima in the probabilistic voting space after applying Parzen window probability density estimation.

According to the experiment results in [XYCD07], their method gives better results compared to the fragments-based methods described above. However, the relationship between the codewords are not taken into account in this method.

3.4 Object detection from hand-drawn models

Object recognition from hand-drawn models gains widespread interest recently. Examples include the methods of Ferrari et al. [FTG06, FJS07], Zhu et al. [ZWWS08] and Ravishankar et al. [RJM08]. Ferrari et al. [FTG06] build a contour segment network and find paths most resembling to the model chains. In their further work [FJS07], Ferrari et al. learn automatically shape models from training images, and combine Hough-style voting with a non-rigid point matching algorithm. Although this approach shows a significant robustness to clutter, it is not rotation invariant. Ravishankar et al. propose [RJM08] a multi-stage approach for detecting deformable objects. The model is first decomposed into segments at high curvature points. In the first stage, these segments are passed through a series of deformations such as rotation, scale and bend and are searched independently in the gradient image. In the next stage, they try to group k-adjacent segments, and in the final stage, they localize the objets up to boundary contours by searching for contours in the gradient image that connect the points of the matched k-segment groups. Recently, Zhu et al. [ZWWS08] have presented a method which breaks the model into informative semantic parts and extracts salient contours from the scene by using a bottom-up contour grouping algorithm. The key idea is to find simultaneously the maximal contour subsets of the scene and the model so that the objects composing these contour subsets match. These contour subsets are determined during matching by minimizing a cost function that measures the shape dissimilarity between contour subsets. This method gives a good way for dealing with accidental alignments and supporting object detection and segmentation simultaneously. However, it manually determines the distinctive parts and the control points of the model.

3.5 Conclusion

We have reviewed the existing shape-based recognition and detection techniques. Generally, there are three groups of approaches: contour-based, region-based and point-based. In fact, each method has its advantages and disadvantages. Moment shape descriptors are usually robust, easy to compute and match, but it is difficult to effectively exploit high order moments. Another drawback of moment approaches is that they are sensitive to spatial occlusion. One possible solution is to apply the moment approaches in a local manner.

In this work, the objects to detect are hand-drawn sketches that can only be represented by contour descriptors, we should investigate a generic method based on contours. Therefore, skeleton approaches are not recommended because of their strong application dependence (e.g. they are mostly suitable for articulated objects) and expensive computation. However, they can be combined with other shape descriptors to create flexible descriptors.

Shape contexts are a way of representing the shapes of objects in the plane. They model the relative positions of points on an object. Shape contexts can be applied to recognize an object model in an image scene (which typically contains multiple object instances) by using sliding windows, but this costs a lot of computation time because shape contexts work on the pixel level. Thus, they are not suitable for our context.

The Generalized Hough Transform (GHT) is a well known method to detect arbitrary objects. Fundamentally, however, this method relies on a voting mechanism, which needs a sufficient amount of features to accurately make decisions, while our hand-drawn models are usually constituted by only several contour fragments. Another drawback is that this method is very time-consuming.

Besides, as analyzed above, the point-based methods are also not suitable for our context.

As mentioned above, each method has its advantages and disadvantages. To better compare the performance of the different approaches described in this chapter, we summarize the advantages and disadvantages of each approach in the following table (cf. table 3.2).

Methods	Advantages	Disadvantages

Moments-based approaches	• Robust to deformable objects • Easy to compute and match.	• Difficult to effectively exploit high order moments. • Sensitive to spatial occlusion.
GHT	• Robust to partial or slightly deformed shapes, tolerant to noise • Able to find multiple occurrences of a shape during the same processing pass.	• Requires a lot of storage space and expensive computation.
Boundary Fragments Models (BFM)	• Outperforms appearance features when the texture is very variable.	• The relative spatial distribution of contour fragments is captured through the mediation of an object centroid, making it sensitive to partial occlusion • Lack of the relative spatial relationships among contour fragments • Limited rotation invariance.

Shape contexts	• Informative: each point is described in the context of overall shape • Not require much reprocessing.	• Limited rotation invariance • Expensive computation • Only limited to boundary based shape representations.
Approaches using kAS descriptors	• Generic and suitable for many recognition applications based on shape • Robustness to translation and scaling-invariance	• Not rotation invariant • Partitioning edgels into roughly straight contour segments destroys valuable contour information • Requires bounding boxes, which delineate objects for training.
Skeleton-based techniques	• Robust to articulation and occlusion.	• Encounters difficulties in dealing with boundary noise • Problem of tree matching for complex articulated shapes.

Table 3.2: Summarizing the advantages and disadvantages of shape based object recognition methods

Among the different approaches presented above, Zernike moments are the most promising for sketch recognition due to the following reasons: a) they are robust to

noise and deformed shapes (e.g. drawing), b) they are invariant to rotation and easy to compute. Hence, we select these features for our recognition method. Nevertheless, as pointed out in the above table, Zernike moments are quite sensitive to occlusion, which is inevitable in storyboard scenes. To cope with this problem, we introduce a local-Zernike representation based on a decomposition of the model object and the scene into patches.

Chapter 4

An approximate decoupled graph matching method for object recognition

Contents

In this chapter, we present an approach for recognizing objects in cluttered scenes, which is robust to occlusion, scale and rotation. Our system takes only a single hand-drawn sketch as input model. The existing approaches using local descriptors like interest points fail in such images mainly due to the lack of texture and color. We introduce a new local patch-based representation using Zernike moments for shape matching. The object recognition task is formulated as a global energy minimization problem by defining a flexible objective function as described in eq. (2.6), which takes into account not only the features themselves, but also their local relationships. Our objective function

depends on three constraints: (a) Zernike descriptors, (b) consistency of the neighborhood relationships between patches and (c) consistency of the rotation angle among the neighboring patches. We propose a fast approximative solution, namely, approximation by decoupled matching for energy minimization. We demonstrate the effectiveness of our approach through experiments on two databases: an industrial database (i.e. project Pinka, cf. chapter 1) and a standard challenging database.

A part of this chapter was published in the IEEE 7th International workshop on Content-Based Multimedia Indexing (CBMI), Crete 2009 [TWLB09].

4.1 Introduction

As humans we are capable of recognizing a variety of object classes based on 2D sketches only. It is with this intuition that we chose to explore a sketch recognition system. Our work focuses on detecting and localizing objects in both storyboard and natural scenes, given only hand-drawings as input of the object shape. The model is represented by a set of patches extracted on different spatial locations. Our goal is to recognize this model in the scene, on which we calculate a set of overlapping patches in a scale-space.

One of the main difficulties is the frequent occlusion problem, which in image indexing and in object detection is commonly tackled using local features [CLRM05][KKK02]. In these local approaches, local descriptors are calculated on keypoints [SM97][Low99][WJKB00] or on edges [FFJS08a]. However, interest points being very unstable on hand-drawn sketches[1], these methods are not applicable in our case. In such an object-from-sketch recognition scenario, global Zernike moments are particularly robust; they have been successfully used for 2D/3D object recognition in [ADV07][HR07][RLB09]. Especially, Revaud et al. [RLB09] have successfully used Zernike moments for recognizing 3D models in sketch scenes (storyboards). Inspired from these works, we chose patch-based Zernike moments as basic descriptors. However, we employ them in a local manner, which allows us to overcome the occlusion problems.

In our approach, each shape (model and scene) is described by a set of locally extracted patches. Thus, object recognition can be solved with a general correspondence problem between two sets of local features. In general, feature correspondence is a difficult problem, which is known to be NP hard in object recognition [TKR08b] (cf. chapter 3). We cast the shape matching and object recognition problem as a global energy minimization task, which takes into account the following constraints:

a) The Zernike moment distance between the model and the scene patches.

b) The consistency of the neighborhood relationships between model and scene patches. If i and j are two neighboring model patches and i', j' are their corresponding scene patches, respectively, then i', j' should be neighbours, i.e the distances between neighboring points should be preserved under isometry (relaxed isometry).

c) The consistency of the rotation angle among the neighboring patch assignments. Assuming local rigidity, this constraint allows us to verify the consistency of the transformation across the object, i.e. if the object has been subject to a rotation between

[1] This is confirmed by our experiments, in which we have attempted to use SIFT keypoints and descriptors for sketch recognition with very poor results.

teh model and the scene, then the rotation is supposed to be roughly uniform across the object. *To our knowledge, this is a novel contribution.*

Based on this energy function, a fast approximative solution is proposed for the corresponding energy minimization, namely a decoupled approach which consists of two steps: the first step matches features and the second one evaluates the matching results through a verification of constraints (b) and (c).

Our method is related to the inexact graph matching methods and the existing methods based on local contour features, which are described in chapter 2 and 3, respectively. We briefly summarize hereafter some of them, those are most related to ours.

Our method is similar to the ones that take into account geometric information between local features, e.g Leordeanu et al. [LHS07] who exploit pairwise geometric interactions (i.e distance, angles) between all pairs of object parts; Similarly, Elidan et al. [EHK06] encode pairwise spatial relations between landmark points from the training shapes and solve the problem of outlining the objects by applying standard MRF (Markov random field) inference algorithms over these landmarks. Our approach gives an alternative among these approaches: we also use pairwise interactions between features but we check only pairwise neighbour interactions for each model patch as opposed to the fully-interconnection between all features [EHK06, LHS07]. Our method is non-parametric, non-iterative, and does not involve any learning or semi-learning such as in [EHK06, LHS07].

There are other methods, which are based on local features and also taken into account spatial relationships between them like Boundary Fragment Model from [OPZ06a, SBC05], kAS method from [FFJS08a], which are described in chapter 3. In general, these methods need a training stage before detection, therefore the results depend on the number of training examples. Moreover, the codebook training methods are sensitive to the distance metric and depend on the clustering algorithm used. In contrast, no prior information about the object to detect (object centroid, ...) and no training is required for our method.

In recent years, recognition from hand-drawn models has received attention from the computer vision community. The existing methods that take hand-drawn models as input, for which our method offers an alternative, have been presented in chapter 3. As opposed to these methods, which manully decompose the model into parts, we introduce a new local patch-based representation to extract model features without manual intervention, where patches are small rectangular subimages distributed across an image. This representation boosts the probability of finding the objects even in the

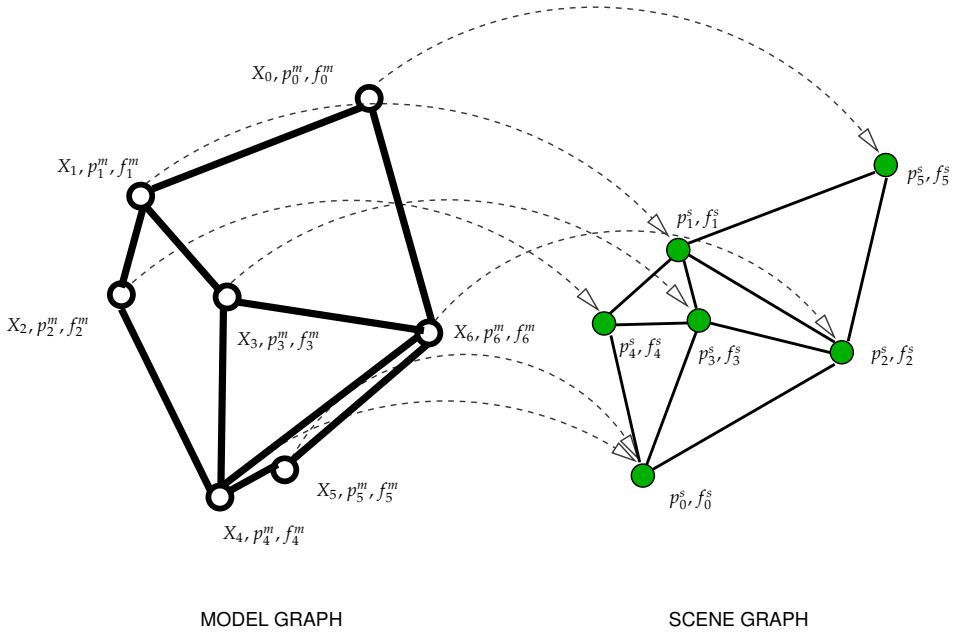


Figure 4.1: The problem translates into a graph matching task. The values p_i and f_i for each node correspond, respectively, to the position and the feature vector of the patch. x_i is a variable indicating the assignment to the scene node.

case of partial occlusion. Moreover, our descriptor provides a natural rotation-invariant quantity based on Zernike moments. In constrast, Ravishankar et al. [RJM08] deal with rotation by rotating the model segments in the range of $[-\delta, \delta]$ degrees; Zhu et al. [ZWWS08] allow rotation but rely on the rotation invariance property of Shape context descriptors.

4.2 The energy minimization formulation

We start from a list of M model patches as well as a list of S scene patches. In this section we consider patches of a single scale, a multi-scale representation will be introduced in section 4.3. As stated in chapter 2, the problem is formulated as a graph matching task, therefore the patches of a given image, or model, are organized as a graph $\mathcal{G} = \{G, E\}$, where G is the set of nodes of the graph (the patches) and E is the set of edges between nodes. We will abrevate this be denoting $i \sim j$ if nodes i and j are neighbors in the graph, i.e. $(i, j) \in E$ (see figure 4.1). We describe the creation of the graph in section 4.4.1.

Each node i of the two graphs (model and scene) is also assigned a position p_i as well as a feature vector f_i describing the image content of the corresponding patch. When necessary, we will distinguish between model and scene values by the superscripts ^{m} and ^{s}: $p_i^m, f_i^m, p_i^s, f_i^s$.

As mentioned above, the correspondence problem translates to an assignment task between the two graphs, a model graph having M nodes as well as a scene graph having S nodes. To this end, each node i of the model graph will be assigned a discrete variable x_i, $i = 1..M$ which can take values from a discrete set $\Lambda = \{1 \dots S\}$. The whole set of variables x_i is also abbrevated as x. A value of $x_i = j$ is interpreted as model node (patch) i being assigned to scene node (patch) j (see figure 4.1).

We suppose the existence of a function between two feature vectors f_i and f_j of two nodes i and j, which returns both the distance (in the sens of feature similarity) and the rotation angle between the two patterns (supposing that the two patches are similar). In the following we will denote the feature distance between two patches as $d_{zd}(\cdot,\cdot)$ and the retrieved rotation angle as $d_{za}(\cdot,\cdot)$. A method providing this function has been given, in [RLB09] (see chapter 3) for Zernike descriptors.

The correspondence between the two graphs, i.e. the assignment of model nodes to scene nodes, will be determined by minimizing a global energy function based on several criteria:

a) The distances $d_{zd}(\cdot,\cdot)$ between the corresponding feature vectors

b) The spatial coherence of the correspondence, i.e. the distances between neighboring patches should be similar to the distances between corresponding patches in the scene (Relaxed Isometry) [2].

c) The coherence of the rotation angles in neighboring patch assignments.

These constraints are illustrated in figure 4.2. They are integrated into the inexact valued graph matching framework described in chapter 2. In particular, the matching algorithm minimizes equation (2.6), which we repeat for convenience as equation 4.1 below:

$$\widehat{x} = \arg\min_{x} E(x) = \lambda_f \sum_{i \in \mathcal{V}} \psi_1(i; x_i) + \lambda_d \sum_{ij \in \mathcal{E}} \psi_2(i, j; x_i, x_j) \tag{4.1}$$

The constraints are integrated into the unary and pairwise functions ψ_1 and ψ_2. More precisely, the unary terms correspond to the distances between the features f^m and f^s of matching patches:

$$\psi_i(i; x_i) = d_{zd}(f_i^m, f_{x_i}^s), \tag{4.2}$$

and the pairwise terms consist of the two distance and angle measures, i.e. the coherence between two model nodes and their corresponding scene nodes, and the coherence

[2] The distances should eventually be normalized with a global constant in the case of a scaling transformation between the model and the scene

between two neighbourhood assignments. We therefore combine these two measements into ψ_2 as follows:

$$\begin{aligned} \psi_2(i,j;x_i,x_j) &= \lambda_d \sum_{i\sim j} \psi_2^d(d_e(p_i^m,p_j^m),d_e(p_{x_i}^s,p_{x_j}^s)) \\ &+ \lambda_a \sum_{i\sim j} \psi_2^a(d_{za}(f_i^m,f_{x_i}^s),d_{za}(f_j^m,f_{x_j}^s)) \end{aligned} \tag{4.3}$$

ψ_2^d is a function that measures the consistency between the associations variables x_i and x_j, i.e the difference in spatial distance between two model patches and their corresponding scene patches (cf. figure 4.2). This distance should be close to zero if the object is subject to an isometry. In the case of exact matching under isometric transformations, this measure can be computed as follows:

$$\psi_2^d(d_e(p_i^m,p_j^m),d_e(p_{x_i}^s,p_{x_j}^s)) = \begin{cases} 1 & \text{if } d_e(p_i^m,p_j^m) = d_e(p_{x_i}^s,p_{x_j}^s) \\ 0 & \text{if } d_e(p_i^m,p_j^m) \neq d_e(p_{x_i}^s,p_{x_j}^s) \end{cases} \tag{4.4}$$

where d_e is the euclidean distance. Since this measure works only for exact matching, we use a gaussian kernel to adopt the distance to realistic conditions:

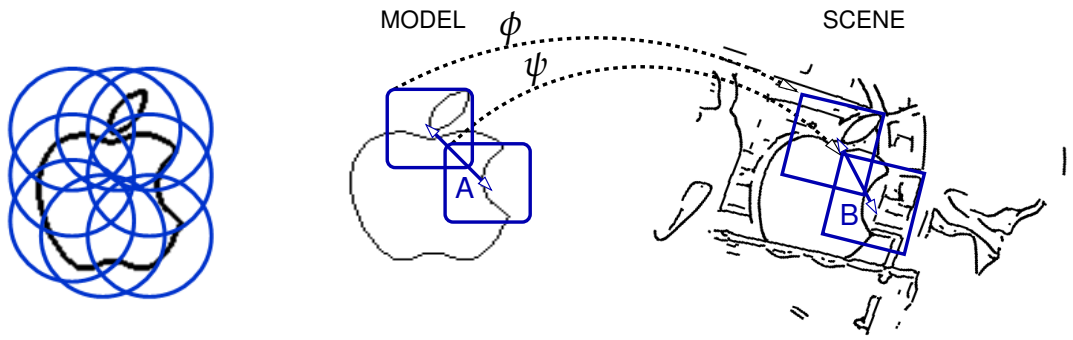
$$\psi_2^d(d_e(p_i^m,p_j^m),d_e(p_{x_i}^s,p_{x_j}^s)) = \frac{(\frac{d_e(p_i^m,p_j^m)}{PatchSize^m} - \frac{d_e(p_{x_i}^s,p_{x_j}^s)}{PatchSize^s})^2}{2\sigma_{de}^2} \tag{4.5}$$

where $PatchSize^m$ is the size of model patch and $PatchSize^s$ is the size of scene patch, which are used for normalization in the case of a scaling transformation between the model and the scene. σ_{de}^2 is the variance of the distances.

The second function ψ_2^a measures the consistency of rotation angles between neighboring nodes, i.e the consistency in the in-plane rotation angles between neighboring patches (cf. figure 4.2), given as follows:

$$\psi_2^a(d_{za}(f_i^m,f_{x_i}^s),d_{za}(f_j^m,f_{x_j}^s)) = \frac{(d_{za}(f_i^m,f_{x_i}^s) \circ d_{za}(f_j^m,f_{x_j}^s))^2}{2\sigma_{da}^2} \tag{4.6}$$

where d_{za} is a function which gives the retrieved rotation angle of an assignment. The operator $\circ$ computes the difference between two angles taking into account their circular domain (i.e. $\psi_2^a(0,\epsilon) = \psi_2^a(2\pi,\epsilon)$). Similarly to σ_{de}^2 described above, σ_{da}^2 is the variance of the angle distances. Note that the score α_i is independent of the number of neighbors since the score in eq. (4.8) has been normalized by the number of neighbors for each model patch.



(i) (ii)

Figure 4.2: (i) Illustration of the overlapping patches extracted on a model: sizes range from 65% to 75% of the model size. (ii) The different constraints listed in section 4.2: a) the Zernike distance assigns model patches to scene patches (dotted arrows); b) the euclidean distance A between neighboring patches is checked to be consistent with the euclidean distance B of neighboring scene patches; c) the rotation angle ϕ of one assignment is checked to be consistent with the rotation angle ψ of a neighboring assignment.

4.3 Approximation by decoupled matching

In this section we present a fast approximative solution for minmizing the energy function (4.1)

Starting from a list of M model patches as well as a list of S scene patches, our goal is to find the assignment $\widehat{x}$, which minimizes eq. (2.6). For the full correspondence problem, generally there are M^S possible combinations of assignments. For each of these assignments, $\approx M \cdot \bar{D}$ consistency criteria need to be checked, where $\bar{D}$ is the average number of neighbors of a patch (i.e. the average vertex degree). By considering the list of patches as a point set (as described in section 4.2, each patch is associated to a position), solutions for this problem can be found in the literature: e.g by adopting the neighborhood structure (k-tree) in order to find the maximum *a posteriori* solution with the junction tree algorithm [TCSB06], which is of complexity $O(M \cdot S^4)$ and applicable to few primitives only; or by formulating the problem as a sampling procedure and checking for the consistency of each sample (RANSAC) [ZK06, BSW05]. Finding an assignment to minimize the energy in (2.6) can be also solved as an Integer Quadratic Programming problem [GS08], which is known to be NP-hard in general. However, dividing the problem into subproblems can be easier to solve. We follow a two step process:

1. First the assignment problem is solved using the first order terms ψ_1 only, which corresponds to the constraint (a). This is trivial, since the solution can be calcu-

lated separately for each node by taking the assignement having minimum feature distance. Further speed-up can be achieved with a kd-tree.

2. In a second step, the detection itself is performed by evaluating the consistency of the other terms ψ_2 locally only for each node i and its neighbors, given the assignments calculated in the first step.

In the following we describe the algorithm in details. As described above, in the first step we assign model patches to scene patches according to the minimal Zernike distance:

$$\begin{aligned} \forall i : x_i &= \arg \min_{j \in [1..S]} \quad \psi_1(i;j) \\ &= \arg \min_{j \in [1..S]} \quad d_{zd}(f_i^m, f_j^s) \end{aligned} \tag{4.7}$$

we recall that x_i denotes the scene patch assigned to model patch i. Since the spatial relationships and the rotation angle coherence are not used, this sub problem is of low complexity compared to the full problem.

In a second step, we calculate the local evidence, i.e the coherence of the local transformation:

$$\alpha_i = \frac{1}{|\{j : i \sim j\}|} \sum_{j : i \sim j} \psi_2(i, j; x_i, x_j) \tag{4.8}$$

The final match score $\alpha(s)$ is calculated for each scale s of a scale space representation as the evidence of the patch which best satisfies these geometrical constraints (note that the smaller this value is, the more constraints are satisfied):

$$\begin{aligned} \alpha(s) &= \min_i \alpha_i \\ \bar{\alpha}(s) &= \arg \min_i \alpha_i \end{aligned} \tag{4.9}$$

We obtain the best consistency value $\alpha(s)$ for scale s. Finally, the patch $\bar{\beta}$ giving the smallest score β over scale-space is selected as the best patch for the input model across all scales:

$$\begin{aligned} \beta &= \min_s \alpha(s) \\ \bar{\beta} &= \arg \min_s \alpha(s) \end{aligned} \tag{4.10}$$

This score β is used for deciding whether or not the object is present in the scene, i.e the existence of an assignment satisfying a threshold. Note that once we have found a significant score, we also obtain the node (model patch) $\bar{\beta}$ that gives minimum score β, i.e the partial energy. Since the position of the model patch in the model is known, this model node alone allows us to localize the object instance in the scene.

In order to detect multiple objects (object instances), after a successful detection step

we use a standard non-maxima suppression technique in the scale-space to remove the detected patches from the list of scene patches and restart a new detection process.

4.3.1 Occlusion and outliers

In our framework, in the first step, each model patch i ($i = 1..M$), is assigned to a scene node (scene patch) j ($j = 1..S$). The outliers do not influence this assignment as we always take the scene patch j which gives the minimum Zernike distance even if the model patch i is not present in the scene. Some constraints discussed so far can become meaningless if one of the involved variables is an outlier. For instance, the distance function $D_a(.,.)$ becomes meaningless if one of the variables is set to "outlier", since there is no *real* associated scene patch to which the rotation angle between model patch and scene patch can be determined. In a graph matching problem, outlier detection can (and must) be tackled by adding an additional binary node that is set to true whenever the value of an involved variable (model node) is "outlier". Our method is a natural and straightforward solution to this problem, since only the best terms influence the detection decision, the other terms are ignored.

4.3.2 Non rigid transformations

Although we check spatial and angular consistencies based on the principle of preserving rigid transformations, a very large class of non-rigid transformations is handled by the gaussian kernels in equation (4.5) and equation (4.6), as well as the fact that the consistencies are verified locally between neigboring patches only.

4.4 Experiments and Results

As described in chapter 3, for sketch recognition we use Zernike moments as basis features in all experiments. Usually, the classical euclidean distance is used to compare two Zernike descriptors [KKSO00, GSTL02, ZL02]. Since the rotation angle (between two shapes) is exploited in our energy function, we use the comparator proposed by Revaud et al. [RLB09], which returns both the distance (in the sense of similarity) and the rotation angle between the two patterns. Our approach differs from the work of Revaud et al. [RLB09] in that we employ Zernike moments in a local maner, i.e. calculate Zernike moments from patches.

4.4.1 Creation of the graph structure

As shown in figure 4.2, the model is indexed by calculating Zernike moments on overlapping patches: the model patches are extracted on different spatial locations, and are of sizes varying from 65% to 75% of the model size. The model graph is based on these extracted patches. The neighborhood relationship between nodes is established by thresholding the Euclidean distance with a threshold T_e, i.e $i \sim j \Leftrightarrow d_e(p_i, p_j) < T_e$. Note that the model graph is fixed, and we construct several scene graphs for dealing with multiple scales. First, overlapping patches [3] are extracted at multi-scales from the scene. Then, Zernike moments are calculated from these scene patches. We construct a scene graph separately for each scale. With the given model graph and scene graph, we can now solve the energy function (cf. eq 4.1) by applying our algorithm presented in section 4.3.

To test our method, we conducted two experiments in the context of object recognition. The first one was performed on a standard dataset (i.e., natural scenes), while the second one was applied to a real industrial application, i.e. project Pinka (cf. chapter 1).

4.4.2 Recognizing hand-drawn models in natural scenes

In a first experiment, we tested our method on the ETHZ database [FTG06], which contains five diverse object categories with 255 images in total: apple logos (40), bottles (48), giraffes (87), mugs (48) and swans (32). Each image contains one or more object instances. The dataset features significant scale changes, illumination changes, deformations and intra-class variations, which make it highly challenging for object detection. We use the same experimental setup as [FJS07, ZWWS08], using only a single hand-drawn model for each class and the whole dataset as test set. Figure 4.3 illustrates an example of models as well as the scenes in this dataset. First, edgels are detected by the Berkeley natural boundary detector [4]. Then, all images are thresholded to obtain binary images. In all experiments, we calculate Zernike moments up to order 12, which is usually used in the literature. We start by constructing the model graph and the scene graph as mentioned in section 4.4.1. The Zernike filters are precomputed and used for extracting all scene patches at the same scale for accelerating the computation. In practice, we first preselect n (set to 25 in our experiments) best patches for each model patch by using only the Euclidean Zernike distance (classical comparator). Then, we apply

[3] In all our experiments the sliding step is about 15% of the window size in each direction, while the scale step is $2^{\frac{1}{4}}$

[4] V. Ferrari provides all segmented images on
http://www.vision.ee.ethz.ch/~ vferrari/datasets.html

Figure 4.3: ETHZ dataset for testing object class detection and shape matching algorithms: the hand-drawn models are shown in top, and natural scenes in bottom.

Revaud et al.'s Zernike comparator to find the best scene patch from this preselected n best patches.

Note that by doing a two-step process, λ_f (see eq. 4.1) is not required anymore. After some simple experiments, we found that $\lambda_d = 51$ and $\lambda_a = 6.25$ give the best results for all objects and we fixed these values for all our experiments. The variance of gaussian kernels has been fixed to a constant (0.4) since the feature variability is handled by the coefficients λ_d and λ_a.

Figure 4.4 illustrates how our method can localize the objects. Once we have detected the best scene patch, we also know its corresponding model patch. The pose of the corresponding model patch, along with the retrieved rotation angle, allow us to localize the object in the scene (white rectangle on figure 4.4.iii) to axis-alignment.

As an evaluation criterion, Ferrari et al. [FTG06][FJS07] considered detection rate (DR) vs. false positive per image (FPPI). Because the DR/FPPI measure depends on the ratio of the number of positive and negative test images, we chose Precision/recall curves (P/R)[5] for the quantitative analysis. Note that recall is defined as the amount of correctly detected objects with respect to the total amount of objects in the ground truth, whereas precision is the amount of correctly detected objects with respect to the total amount of detected objects. Since the two measures are dependent, i.e. varying the threshold T_d in order to increase recall will generally decrease precision. Another reason

[5]These curves are also called ROC curves (receiver operating characteristic)

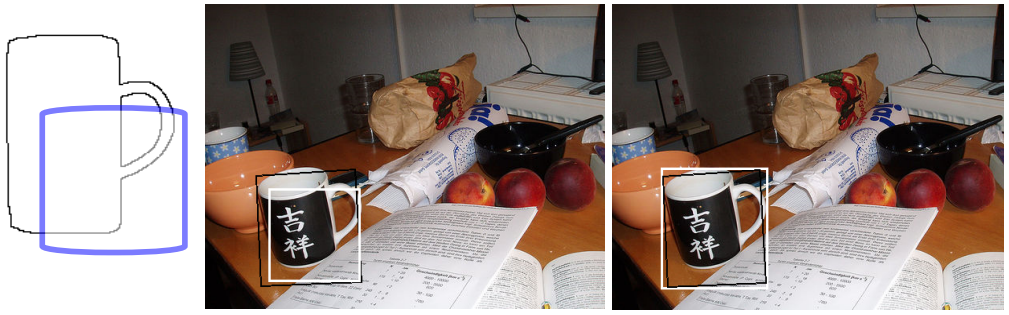


Figure 4.4: (i) the input model, in which the model patch corresponding to the best detected patch is marked. (ii) the best scene patch detected (delineated with a white bounding box) with an angle of 18°. The oriented rectangle (in black) is the bounding box calculated from the best patch, the position of its corresponding model patch and the retrieved rotation angle. (iii) the same bounding box (in black) together with the returned axis-aligned bouding box (while).

for choosing P/R criterion is that the latest results presented by Zhu et al. [ZWWS08], against which we compare our method, have been evaluated by P/R curves and they outperformed the method of Ferrari et al. [FJS07].

Figure 4.5 illustrates our results on this dataset. For evaluation analysis, we used the DetEval tool developed by Wolf and Jolion [WJ06] [6]. Table 4.1 shows detection rates at equal-error-rates for our method against the method from Zhu et al. [ZWWS08] (i.e operating points where Precision equals Recall). Compared to this latest state-of-the-art result [ZWWS08], our performance is better on three classes among five. We obtain lower values for the Mug and the Bottle models. This is mainly caused by the fact that these sketched models are quite simple (i.e they comprise only straight segments), and thus local shape features like patch descriptors are not suitable enough.

Beyond this evaluation, our method offers an important advantage over the method of Zhu et al. [ZWWS08] and the state-of-the-art methods [FTG06][FJS07][RJM08]: our method recognizes and detects the object along with its rotation angle in the plane of the scene. This can further help in 3D scene understanding, for instance 3D scene reconstruction from 2D views (considering the model as a 3D viewpoint). Note that the method of Zhu et al. [ZWWS08] manually determines the distinctive parts and the control points of the model.

Figure 4.6 shows some visual results for this database. The detected objects are delineated with white bounding boxes. Images a-1,a-2, a-3 demonstrate that our approach

[6] DetEval is available on http://liris.cnrs.fr/christian.wolf/software/deteval/index.html.

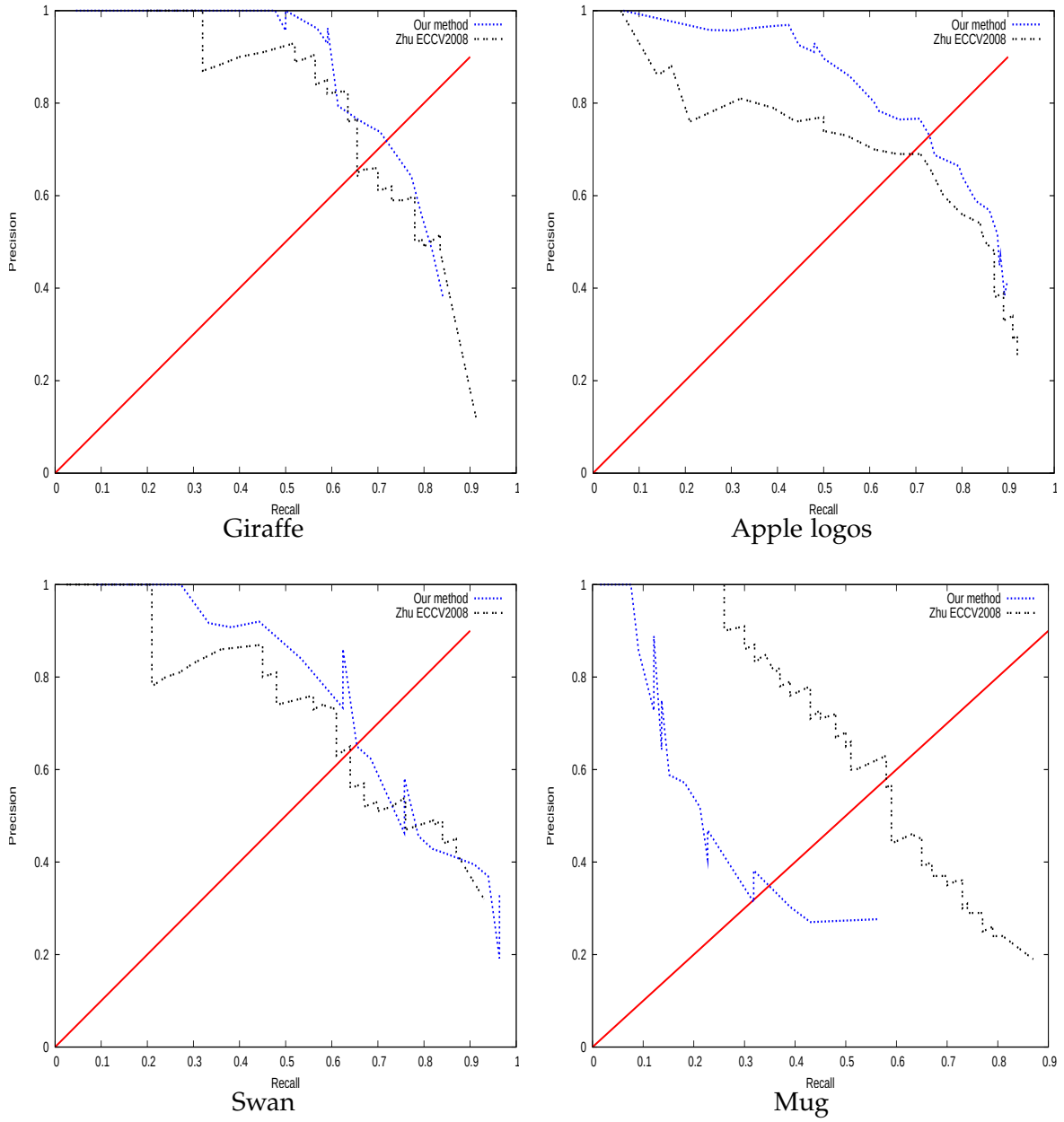


Figure 4.5: Precision vs. recall curves on four classes of ETHZ database. Our precisions on "Apple logos", "Giraffes" and "Swans" are better than the latest state-of-the-art results [ZWWS08].

	Zhu et al. [ZWWS08]	Our approach
Giraffes	68.00 %	**73.00 %**
Mugs	**58.00 %**	36.50 %
Apples	65.50 %	**72.00 %**
Bottles	**78.00 %**	46.50 %
Swan	64.00 %	**66.00 %**

Table 4.1: Comparison of detection rates at equal error rates (ERR) between the method of Zhu et al. [ZWWS08] and the proposed approach.

is capable of detecting multiple instances of objects in scenes. Images b-1, b-2, b-3 show detection results, on severely deformed objects. The results for giraffes (c-1, c-2) illustrate the robustness of our approach to occlusion. Images c-3, d-1, d-2, d-3 show some detection results in very cluttered scenes. Finally, the detection results for swans (e-1, e-2, e-3) show that our method handles intra-class variability very well. Note that the bounding-boxes for Giraffes are sometimes smaller than the ground truth ones. The reason is due to the fact that the giraffe's legs have not been well-drawn in the model from Ferrari et al. [FTG06], and as noticed in [FJS07], the legs of the animal are hard to detect.

Figure 4.7 shows some false positives. The first column presents the input models and some typical false-alarms are shown in the second and the third column.

Note that the models used here have been well sketched by Ferrari et al. [FTG06]. One of the main issues of such object-from-sketch application is to answer the question about how the performance is stable when changing the input model, e.g different people making different sketches. By using Zernike moments for extracting contour information, our method is robust to sketch variation [ADV07][HR07][RLB09]. This is also confirmed by our experiments, in which we have tried to retrieve the horses from the Weizmann-Shotton horse database [SBC05], composed of 327 positive images, and 327 negative images. Figure 4.8 shows a quantitative analysis of P/R curves, in which we have used two different hand-drawing horses as the input models and the results are quite similar. Figure 4.9 shows some visual results when using "horse drawing 1" (see figure 4.8) for this dataset. We can see that our method can capture large deformations such as the articulation of the horse's neck.

Accuracy of detections – We use the same criterion which has been used in [FJS07]: a detection is counted as correct if its bounding-box overlaps more than 20% with the ground-truth bounding-box, and vice-versa. Any other detection is counted as a false-positive.

The dependence of object recall and precision on this evaluation criteria is important

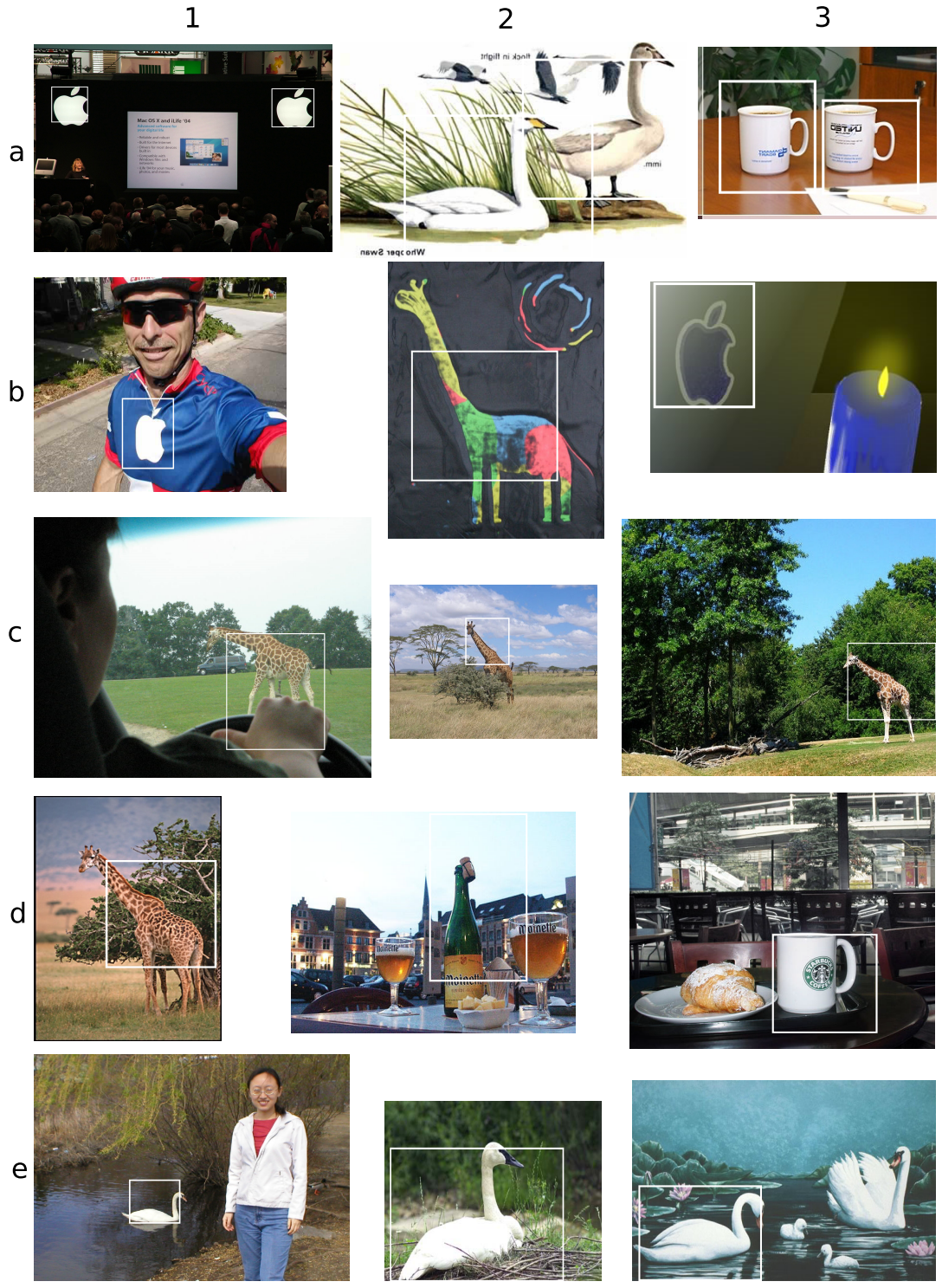


Figure 4.6: Results on some example images of the ETHZ shape classes.

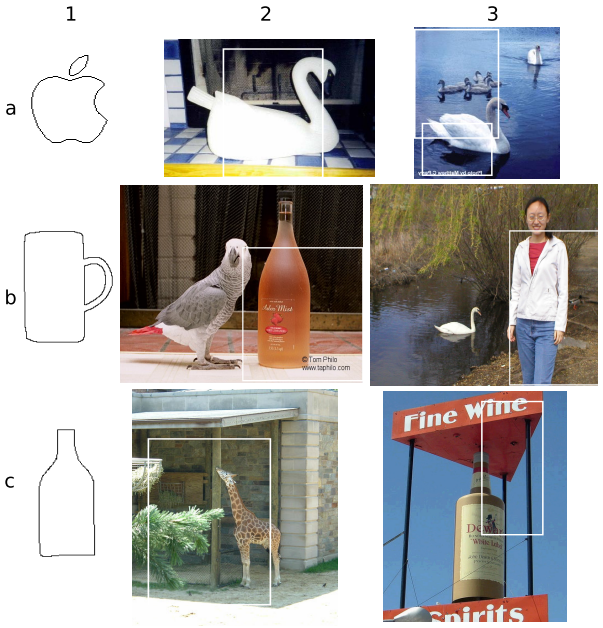


Figure 4.7: Some typical false positives.

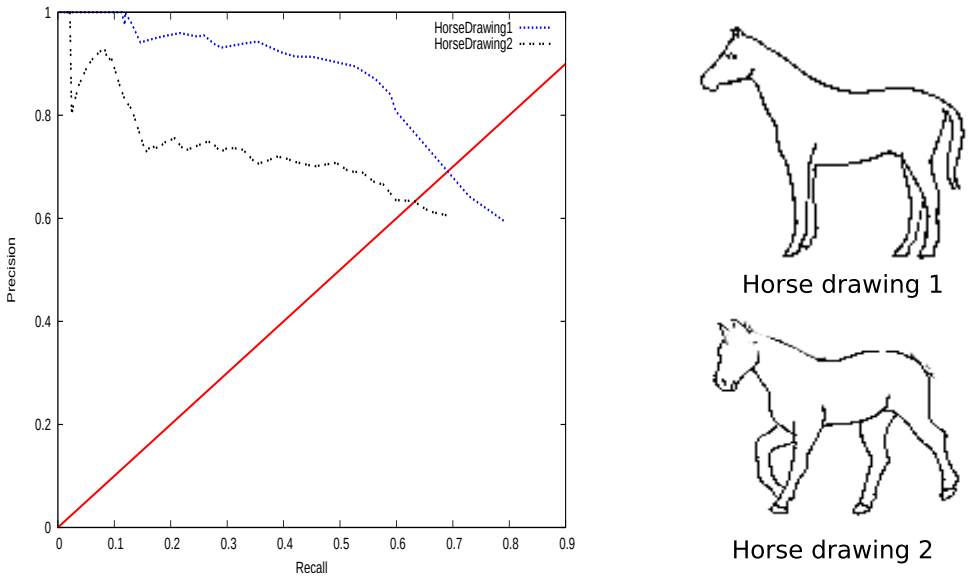


Figure 4.8: An example to demonstrates how the perfomance is stable when changing the input model. The left image shows the P/R curves for two corresponding sketch models in the right image. These tests are performed on the Weizmann-Shotton horses [SBC05].

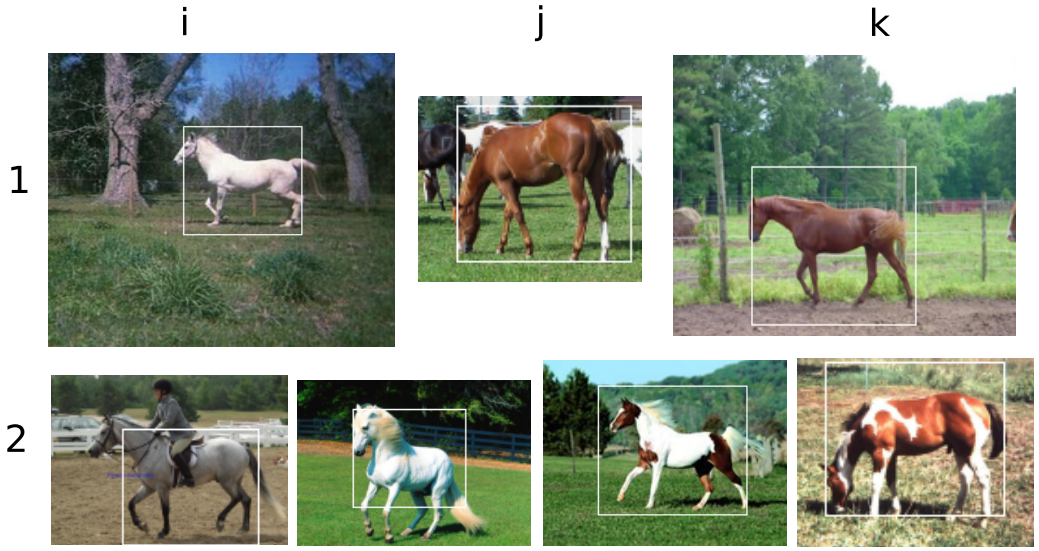


Figure 4.9: Some visual results of the Weizmann-Shotton horse database [SBC05]. The input model used here is the horse drawing 1 in figure 4.8.

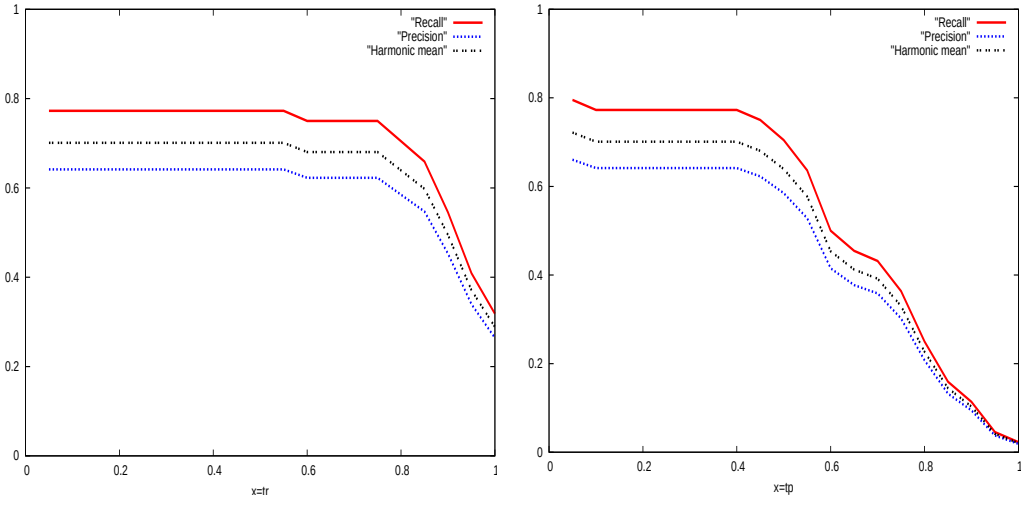


Figure 4.10: Detection results on the Apple logos of the ETHZ shape classes with varying evaluation criteria. Left: varying constraint area recall (t_r) while area precision (t_p) is constant and equal to 0.2. Right: varying constraint t_p while t_r is constant and equal to 0.2.

and should be subject to evaluation itself [WJ06]. More precisely, an object should be detected if a sufficiently large part of the ground truth is detected (i.e area recall is sufficiently high) and the detection bounding box is high enough (i.e area precision is high enough) [WJ06]. As proposed in [WJ06], figure 4.10 shows the dependence of detection performance on the evaluation criteria as graphs. We can see in the left column that object recall and precision decrease slowly when the area recall (t_r) approaches 1, which indicates that most of the object rectangles are detected with their entire area.

4.4.3 Recognizing 3D models from 2D storyboard scenes

In a second experiment, we tested our method on a real industrial dataset provided by the Pinka company (cf. chapter 1). Note that our objectives in this project are two-fold:

- Detecting and recognizing the 3D models as well as their location and size
- Recognizing the 3D pose: detecting the viewpoint for each model

The latter viewpoint will be obtained by selecting the correct 2D view stored in the database as well as retrieving the in-plane rotation angle of this view.

Althrough recognizing three-dimensional objects in a 2D scene is a well known problem, very few work exists, e.g [HRW07], on direct 3D object recognition. We chose to tackle the problem by recognizing a 3D object through a set of 2D images each corresponding to a single viewpoint. Figure 4.11 presents our general scheme for recognizing 3D objects in storyboards. Our method is based on the following principle: each 3D model is represented by a set of 2D views (model images); then, edges are detected in the model images with a canny detector − the storyboard images are already stroke images which do not necessite edge detection. All images are thresholded before subsequent preprocessing, e.g. construction of the graphs (see section 4.4.1).

As mentioned above, we aim at detecting the viewpoint in the plane of storyboard, i.e. the input of our algorithm (presented in section 4.3) is now a set of model views, and each view contains a list of model patches. For this end, first we apply our algorithm to find the best patch for each view. Then, the view that gives highest score is selected as the best view for the 3D model.

Experiment-setup – We applied our method to a real industrial application comprising five different 3D models: tents (2 different models), trailers, bushes, and trees provided by the Pinka company. As shown in figure 4.11, the proposed method consists of two processing stages:

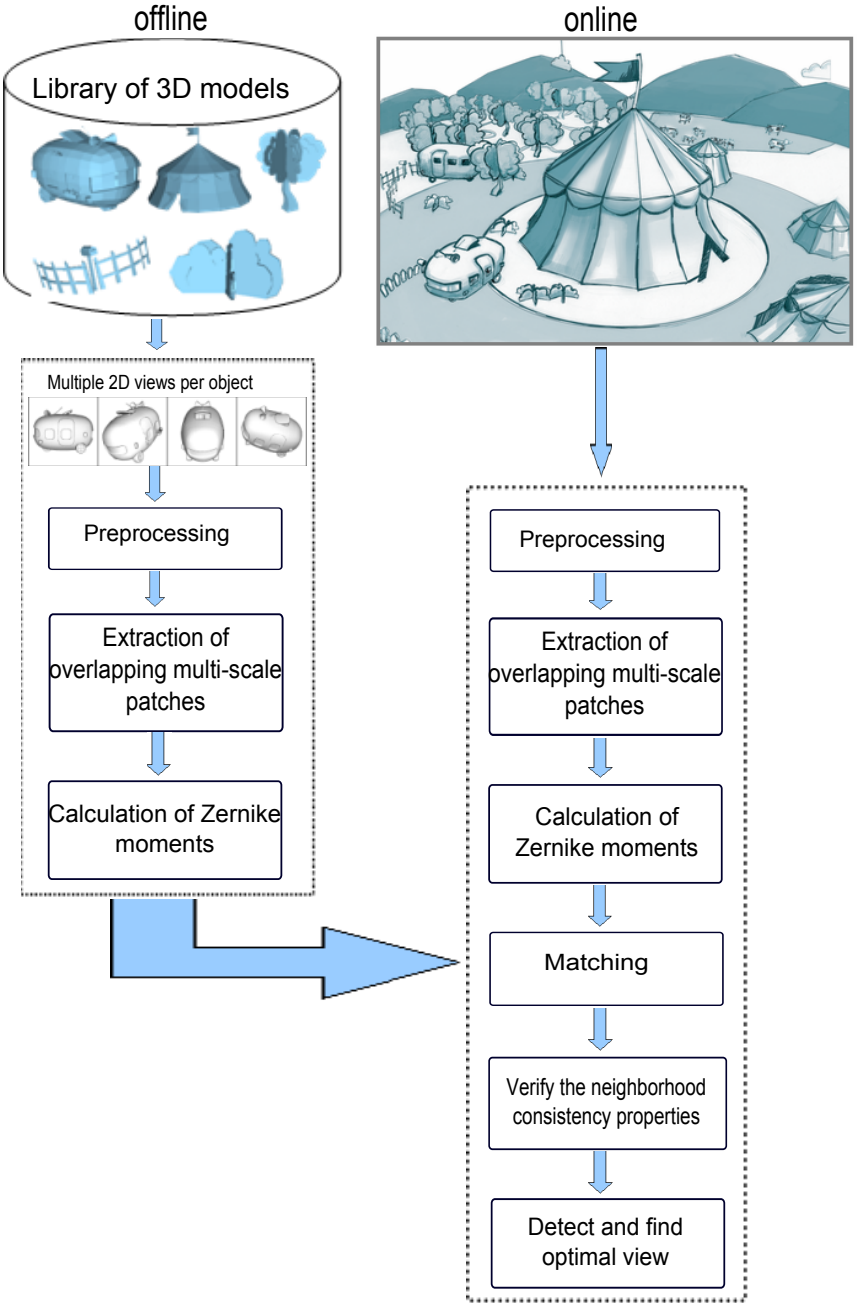


Figure 4.11: The proposed 3D patch-based object detection in sketch images.

- For the offline process, about 120 views from each 3D model (i.e. $5 \times 120 = 600$ views on total) are extracted and then the views are indexed by calculating Zernike moments on 16 overlapping patches: the model patches are extracted on different spatial locations and are of sizes varying from 65% to 75% of the model size.

- In the online process, similar overlapping patches are extracted from the storyboard and Zernike moments are calculated at multiple-scales. We used 4 storyboards of size 3350×2260 for testing. We chose the same variance parameter which has been used in [TCSB06] for Gaussian kernels measuring differences in euclidean distances: σ_{de}=σ_{za}=0.4.

Figure 4.12 illustrates some results of our algorithm. Each object detected is marked with a bounding box and its corresponding 3D model view is displayed. Note the excellent results on the very difficult storyboard images. Most of the objects are occluded, some of them significantly, which does not prevent our method from correctly detecting and recognizing them. Furthermore, even very similar 3D models, as for instance the two different tents, are distinguished correctly. Several typical false positive detections are reported in figure 4.12d. These false positive results occur in the case of intra-class variation, i.e. a bush model is very similar to a part of a tree model. Note that we chose to report the recall obtained for 100% precision, i.e. no false alarms.

Table 4.2 shows the comparison between Revaud et al's global approach [RLB09] and our approach. The comparison results show that the proposed method attains higher recall than that of the global one. Both methods obtain low recall for the tree models. This is mainly caused by the fact that sometimes their sketch deviates too much from their 3D model. In addition, the sketched trees belong to highly cluttered scenes (forest − see the tree models in figure 4.12). We do not consider the errors of the detected viewpoint in calculating recall and precision. A comparison of these errors for both methods is given in table 4.3.

Table 4.3 presents the mean error in viewpoint detection compared with the global method. We here show only the mean errors for 100% precision, which are calculated according to the results in table 4.2. Note that each view is characterized by two angles $((\alpha \in [0, 2\pi], \beta \in [0, \pi])$. To compute the error of the viewpoint detection, we translate the angle pair into a corresponding 3D point on the unit sphere. The error between the detected viewpoint and the sketched viewpoint is then calculated as the euclidean distance between the corresponding 3D points on the sphere. The comparison results in table 4.3 demonstrate that our method performs better than the global one in viewpoint detection.

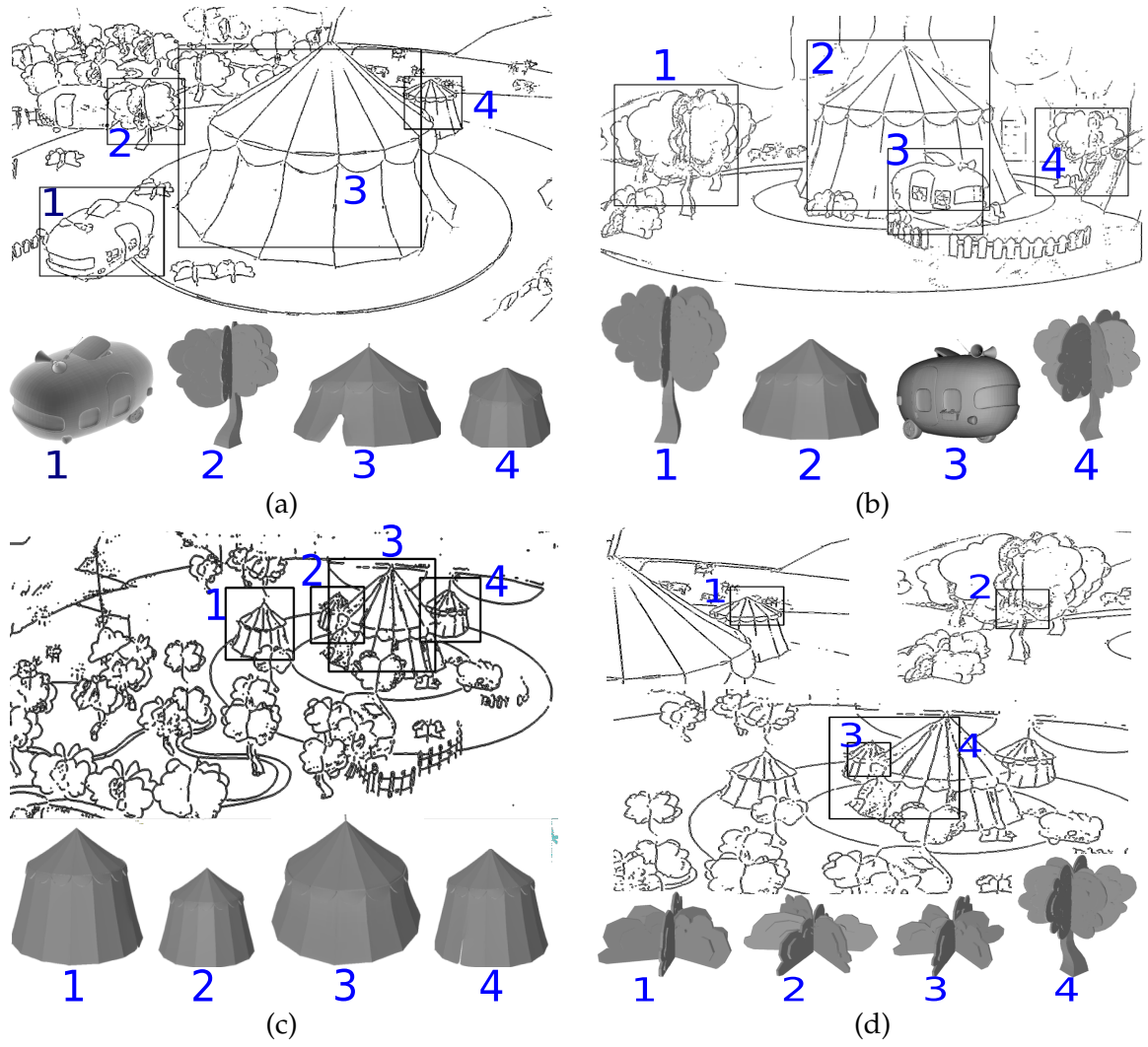


Figure 4.12: Examples of detection results on several storyboards. Note the successful detection in spite of many occlusions. Images (a)-(c) show detection results for 100% precision, i.e. no false alarms. Figure (d) illustrates the difficulty of the tree and bush models on an image created as a mixture of images (a)-(c). Searching for four tree and bush models, the best response for each detection are wrong models or parts of wrong models (tents etc.).

In figure 4.13 some visual results are presented for both approaches: in the first row are some sketched objects extracted from the storyboards, the second and third row show the detected views of the global approach and the proposed approach, respectively. We can see that the global approach is very sensitive to occlusions: for objects 1 and 4, which are slightly occluded, the global approach returned the correct object model but views which are not very similar to the views of the sketched objects, while our method can find the best views (the view which is closest to the sketched object in the database). For objects 2 and 3, which are more severely occluded, the global method failed, whereas

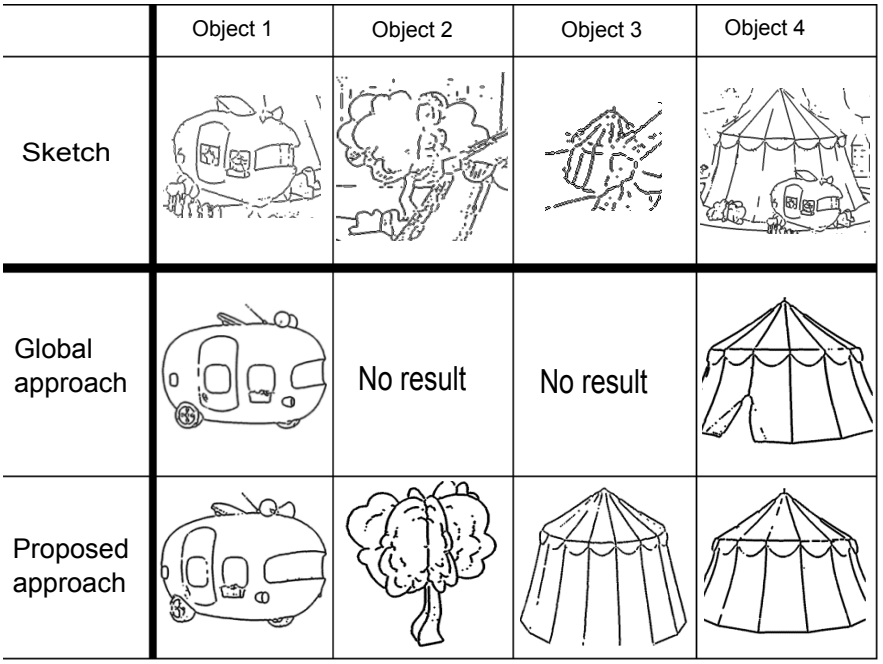


Figure 4.13: Examples of detection results, where the detected views are given for both compared methods.

	Global approach [RLB09]		Proposed approach	
	total	%	total	%
tents	4/8	50.00	7/8	**88.00**
trailers	2/3	**67.00**	2/3	**67.00**
bushes	3/10	30.00	6/10	**60.00**
trees	1/31	3.00	4/31	**13.00**

Table 4.2: Comparison of recall for 100% precision for the global approach and the proposed approach.

our method can still recognize the 3D models although the detected views are sometimes slightly different from the original ones.

4.5 Conclusion and Discussion

In this chapter, we have presented an efficient approach for object recognition in cluttered images that is robust to occlusion, scale and rotation. Our contributions are twofold. First, we present a new shape matching method, which incorporates two main relationships between patches: spatial interactions and rotation angle consistency. This makes the proposed method flexible and easy to use with other descriptors. Experiments on both industrial and challenging databases confirm that our framework can

	Global approach [RLB09]	Proposed approach
tents	0.81	**0.31**
trailers	0.29	**0.06**
bushes	**1.16**	1.41
trees	2.70	**1.36**

Table 4.3: Comparison of the mean error in viewpoint detection between the global approach and the proposed approach.

deal with any transformation, e.g rigid, non-rigid, affine, etc. Secondly, we propose a new local patch-based representation using Zernike moments, which is suitable for object recognition from hand-drawn models, and which is more robust to occlusion.

Our work can be extended in several directions. First, we plan to combine our Zernike moments descriptors with contour descriptors like kAS in [FFJS08a]. Secondly, in our current experiments, the contribution of the topological coherence of the correspondence has not been taken into account. We believe that considerable gains in object detection performance can be achieved by taking into account this information.

We currently work on a method which approximates the model graph as a linearly ordered chain, which allows to quickly calculate the global minimum of eq. 2.6 without decoupling the terms.

Part III

Human activity recognition

This third part describes our contributions in the field of activity recognition. This part is composed of 3 following chapters:

In chapter 5, we present an overview of existing methods for human activity recognition. There are several available surveys on action recognition, which introduce different taxonomies of action recognition approaches. They are based on either the methods used or features used for action classification. In this chapter, we first introduce features, which are widely exploited in action recognition, and then classify recognition approaches into several categories.

In chapter 6, we introduce new features for action classification. Our goal is to improve the conventional bag of words models. To this end, we propose new features, which take into account spatio-temporal information. Experiments on two standard datasets show that our approach is comparable to state of the methods.

Chapter 7 presents a new activity recognition method by using graph matching techniques. We do not focus directly on classification techniques, which have been widely used in action recognition. Instead, we formulate the problem as a graph matching one, in which a model graph represents a template video, and a scene graph represents a test video. As opposed to most other previous works on graph matching, which focused on directly solving the correspondence problem, we approach the problem in another angle: how to efficiently estimate the compatibility matrix, from which the solution can be computed efficiently. We evaluate our method on both two standard datasets and our own dataset. The experimental results show that the proposed method gives good results.

Chapter 5

State of the art in human activity recognition

Contents

In this chapter, we discuss the existing approaches for action recognition. Due to the large number of existing methods, and variations in evaluation practices, we review only main ideas of each approach, without discussing detailed results. There are different taxonomies proposed for action recognition. Our taxonomy is in some respects based on the ones used by Moeslund et al. [MHK06], as well as by Poppe et al. [Pop10]: feature representation and action representation. In this thesis, we focus on using local features and perform action recognition from local features. Thus, we do not consider the work on gesture recognition, interactions between persons, human poses, and tracking of human motion. We refer the readers to surveys [MHK06, AC99, Gav99, MG01, AP04, FAI*05, Pop10].

5.1 Introduction

Human action recognition has been an active research area in recent years due to its wide number of applications which include video-surveillance but also annotation and retrieval, human computer interaction etc. A considerable amount of literature exists on different applications of action recognition. Based on the features used for recognition, existing action recognition methods can be broadly divided into two categories: local approaches [DRCB05, NWFF08, SDPNL08, SLC04] and holistic approaches [LAS08, WBR07, WGLR08] and some methods which do not neatly fall into these categories, e.g. Sun et al. [SCH09] combine local and holistic features. However, there exist different taxonomies in literature: Bobick [Bob97] classifies action recognition methods into movement recognition, activity recognition and action recognition; Aggarwal and Cai [AC99], and Wang et al. [WHT03] use a taxonomy of body structure, tracking and recognition; Gavrila et al. [Gav99] propose another taxonomy based on 2D approaches, 3D approaches and recognition; Moeslund et al. [MG01] discuss the main steps that an action recognition sytem needs to consider: initialization, tracking, pose estimation and recognition; Recently, Turaga et al. [TCSU08] discuss two levels of action recognition: "action", and "activities"; Very recently, Poppe [Pop10] introduces a taxonomy of image representation and action classification. It should be noted that several researchers (e.g. from [TCSU08]) consider "actions" as the simple motion patterns performed by one person, and "activities" as more complex motion patterns performed by several humans. In this thesis, we consider the term action with the same meaning as activity.

Each taxonomy has its advantages and disadvantages, and has been developed for different specific purposes. Among the above taxonomies, we found that the ones introduced by Moeslund et al. [MG01], and later by Poppe [Pop10], are most *informative* to review current state of the art approaches. In this thesis, we present two contributions to action recognition, which are related to both the Bag of words models, spatio-temporal relations based methods, and video matching methods. We therefore decide to discuss the literature in two main steps: first, we focus on the features used in most action recognition systems, as well as in our method; Second, we review the recognition methods, which can be classified into: statistical, probabilistic graphical models, spatio-temporal relations based methods, and video matching.

The remaining part of this chapter is organized as follows. First of all, we summarize two common datasets, which are widely used in action recognition. Next, we first discuss features, which are mainly employed for action recognition. Then, we review the recognition methods.

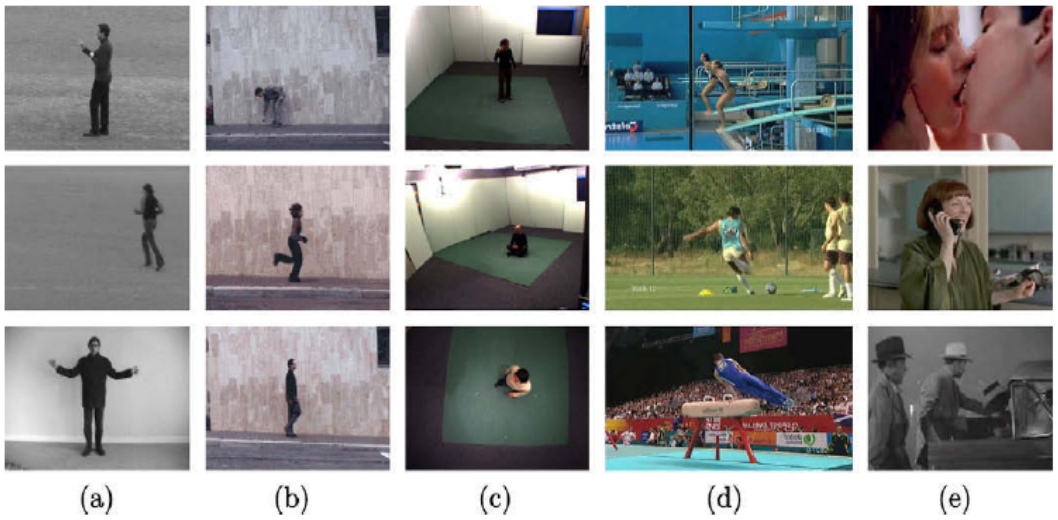


Figure 5.1: Illustration of several publicly available datasets for action recognition: (a) KTH dataset, (b) Weizmann dataset, (c) Inria XMAS dataset, (d) UCF sports action dataset and (e) Hollywood2 human action dataset. This figure is reprinted from [Pop10].

5.2 Common datasets

In this section, we present the publicly available datasets, which are widely used for activity recognition evaluation. Figure 5.1 illustrates frames of these datasets, including the KTH dataset, Weizmann dataset, Inria XMAS dataset, UCF sports action dataset, and Hollywood2 dataset. In this thesis, we chose the most two widely used KTH and Weizmann datasets to evaluate our methods.

5.2.1 KTH dataset

The KTH dataset (cf. figure 5.1a) was provided by Schuldt et al. [SLC04] in 2004 and is the largest public human activity video dataset. It contains a total of 2391 sequences, comprising 6 types of actions (boxing, hand clapping, hand waving, jogging, running and walking) performed by 25 subjects in 4 different scenarios including indoor, outdoor, changes in clothing and variations in scale. Each video clip contains one subject performing a single action. The image resolution is of 160×120, and temporal resolution is of 25 frames per second. There are considerable variations in duration, and in viewpoint. Several actions (e.g. walking, jogging and running) are performed either from left to right, or in the opposite direction. The background is homogeneous and static, but hard shadows are present.

5.2.2 Weizmann dataset

The Weizmann dataset (cf. figure 5.1b) was first used from Blank et al. [BGS*05a] in 2005, which consists of 90 video clips of 10 actions (walking, running, jumping, gallop sideways, bending, one-hand-waving, two-handswaving, jumping in place, jumping jack, and skipping) performed by 9 different subjects. Each video clip contains one subject performing a single action. The image resolution is of 180×144 pixels and the temporal resolution is of 25 frames per second. The duration of each sequence is about 3 seconds on average and the backgrounds are static.

5.3 Video representation

In this section, we present features and extraction methods for activity recognition. Generally, video representation methods can be classified as holistic (global) or local (patch-based). Holistic representations encode the image frames as a whole. Although they can capture the shape of the moving objects and require much less computation than the local representations, holistic representations are quite sensitive to occlusions or noise. Another drawback of holistic representations is that they usually need more preprocessing such as background substraction or tracking. On the other hand, local representations can deal with change in background, and occlusion. In general, local representations are described as a set of local patches, where a patch is a small neighborhood of pixels around a spatio-temporal interest point.

We discuss holistic features in section 5.3.1 and local features in section 5.3.2.

5.3.1 Holistic features

MEI and MHI – Bobick and Davis [BD96, BD01] exploit silhouettes of a person, and introduce Motion Energy Image (MEI) and Motion-History Image (MHI). MEI is a binary image which represents where motion has occurred in an image sequence. In practice, MEI is archieved by the cumulative absolute difference in pixel intensity, i.e. returned by background subtraction - see figure 5.2a. MHI is a grey-level image which represents how a person has moved during the action - see figure 5.2b. MHI is computed by the difference between the gray-scale images of t successive frames. Moments (see chapter 3) are usually computed from MEI and MHI, and used as global features for action recognition. A part from Hu moments, Zernike moments or other moments, or shape context descriptors [LN07] can also be computed from MEI and MHI silhouettes. Recently, Sharif and Djeraba [SD09] use MHI to estimate *spatiotemporal region of interest*

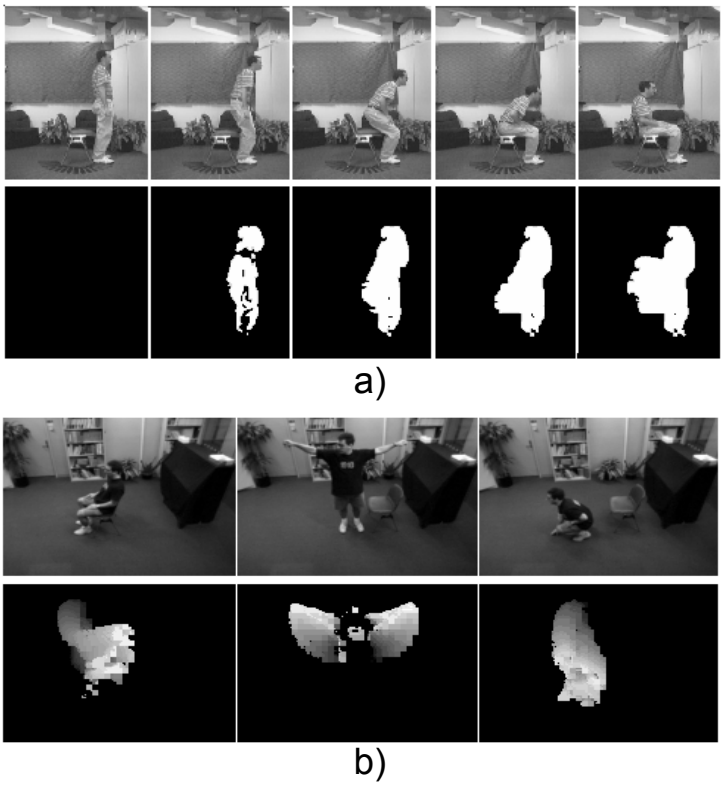


Figure 5.2: Examples of Motion Energy Images (a): bottom row shows a cumulative binary MEI corresponding to the frames above, and Motion-History Images (b) for three actions (sit-down, arms-raise, crouch-down). This figure is reprinted from [BD01].

(ST-RoI). Then, based on ST-RoI features, they propose a new method to detect exceptional motion frames in real video sequences for both static and dynamic backgrounds.

MEI and MHI representations are insensitive to color, texture, and illumination changes. However, they usually fail in the case of occlusions, and depend on the background segmentation performance.

Optic flow – Optic flow, which is defined as the set of apparent velocities of the brightness, is widely used in many vision systems. It is a vector velocity field defined on sequences of images (see figure 5.3 for an illustration). There are several algorithms to calculate the optical flow, e.g. from Lucas and Kanade [LK81] (based on local smoothness constraints), the regularization based algorithms [HS80, BA93], blurred optic flow [EBMM03] (cf. figure 5.3). We refer the readers to [BFB94] for a comparison of various optical flow techniques. Optical flow provides important information about the regions undergoing motion and the velocity of motion, but it is quite sensitive to noise and illumination changes.

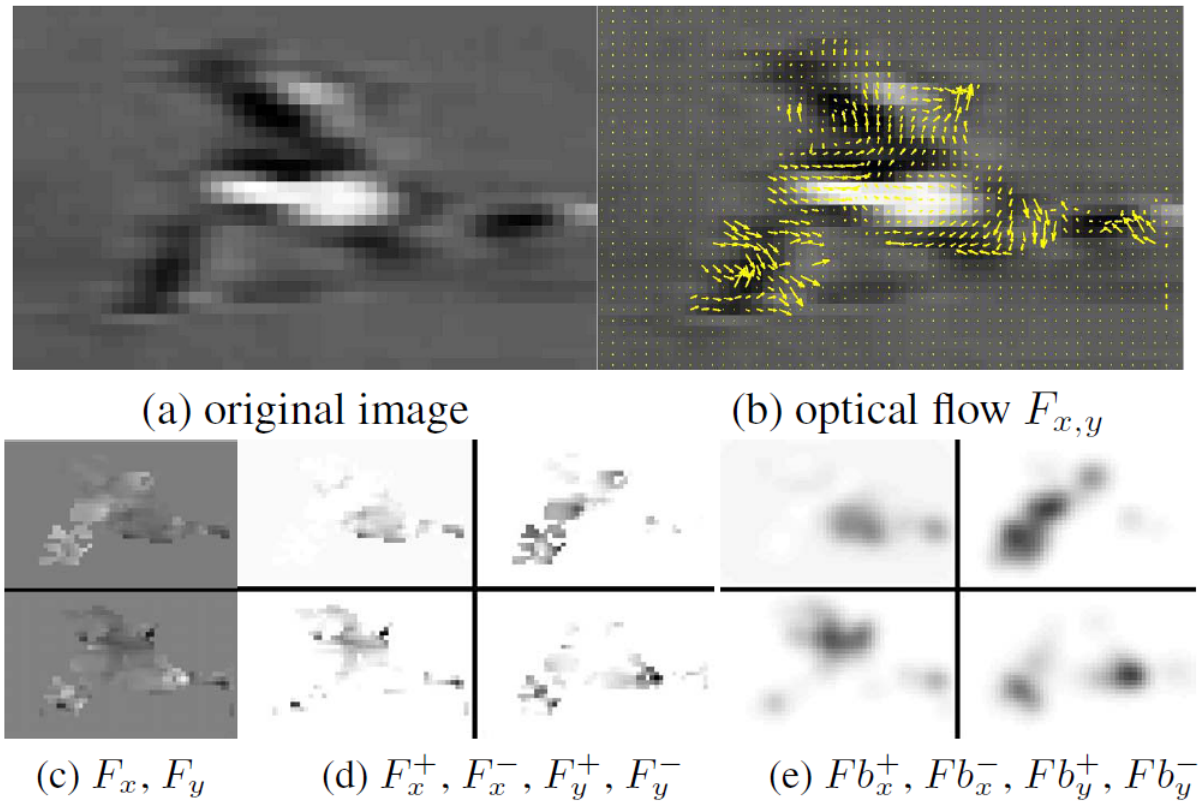


Figure 5.3: Optical flow split into directional components, then blurred (reprinted from [EBMM03]).

5.3.2 Local features

Generally, local features are features extracted from 3D regions around some spatio-temporal interest points or other local primitives. Different methods differ on the nature of features which are extracted from interest points. Usually, the extracted descriptors are then flattened or concatenated into a single vector.

It should be noted that the most successful methods for action recognition are local-based methods, often a combination of local features, bag of words models, and a classification algorithm. We detail the bag of words model, and briefly review classification techniques used in literature in the following.

5.3.2.1 Bag of words models

The bag of words models (BoW)[1] was first used in natural language processing and information retrieval. In this model, a document [SM83] is represented as an unordered

[1]Not that, in computer vision, bag of words is sometimes referred as Bag-Of-Visual-Words (BOVW) [SZ03].

collection of frequencies of words. BoW models have been very successful for several computer vision tasks, such as image classification or topic discovery, where an image is considered as a document and keypoints extracted from this image are the words. Inspired by this, a number of works have shown very good results for human action recognition [DRCB05, NWFF08, SDPNL08, SLC04] using BoW models, i.e. considering a video sequence as a bag of keypoints. In general, a BoW model consists of the following steps (cf. figure 5.4):

a) Feature extraction. Local features are extracted from training images/video sequences. There are different ways to feature extraction, e.g. using a regular grid or applying an interest point detector (see the next section).

b) Learning a "visual vocabulary". In this step, a clustering technique like K-mean is applied to the set of local features extracted from the previous step for learning a visual vocabulary (i.e. a codebook). Then, for each cluster (after learning), a center word, which is representative for this cluster, is selected as codeword.

c) Video representation. First, a set of local features, which represents a video, is mapped to the codebook, and the indices of the nearest codewords are selected. Then, a frequency histogram is constructed from these indices. This histogram is used as input for a classifier such as Support Vector Machines.

In computer vision, the BoW model is very popular due to the following reasons:

- Generally, hundreds of local features are extracted for each video sequence or image, while a lot of them are very similar in nature.

- The sizes of different images/videos are not the same, yielding different dimensional feature spaces that cannot used to perform classification, which requires feature vectors of fixed dimension as input.

Bow models are very flexible and effective for many applications. However, a question arises as how to choose a vocabulary size k? If k is too small, then codewords are not representative of all local features. Otherwise, if k is too large, then quantization artifacts, or overfitting can appear. In practice, the codebook size is usually empirically chosen through experiments.

In the next sections, we present how to extract local features from video sequences: first, several interest point detectors are described. Then, major local descriptors used in activity recognition are presented.

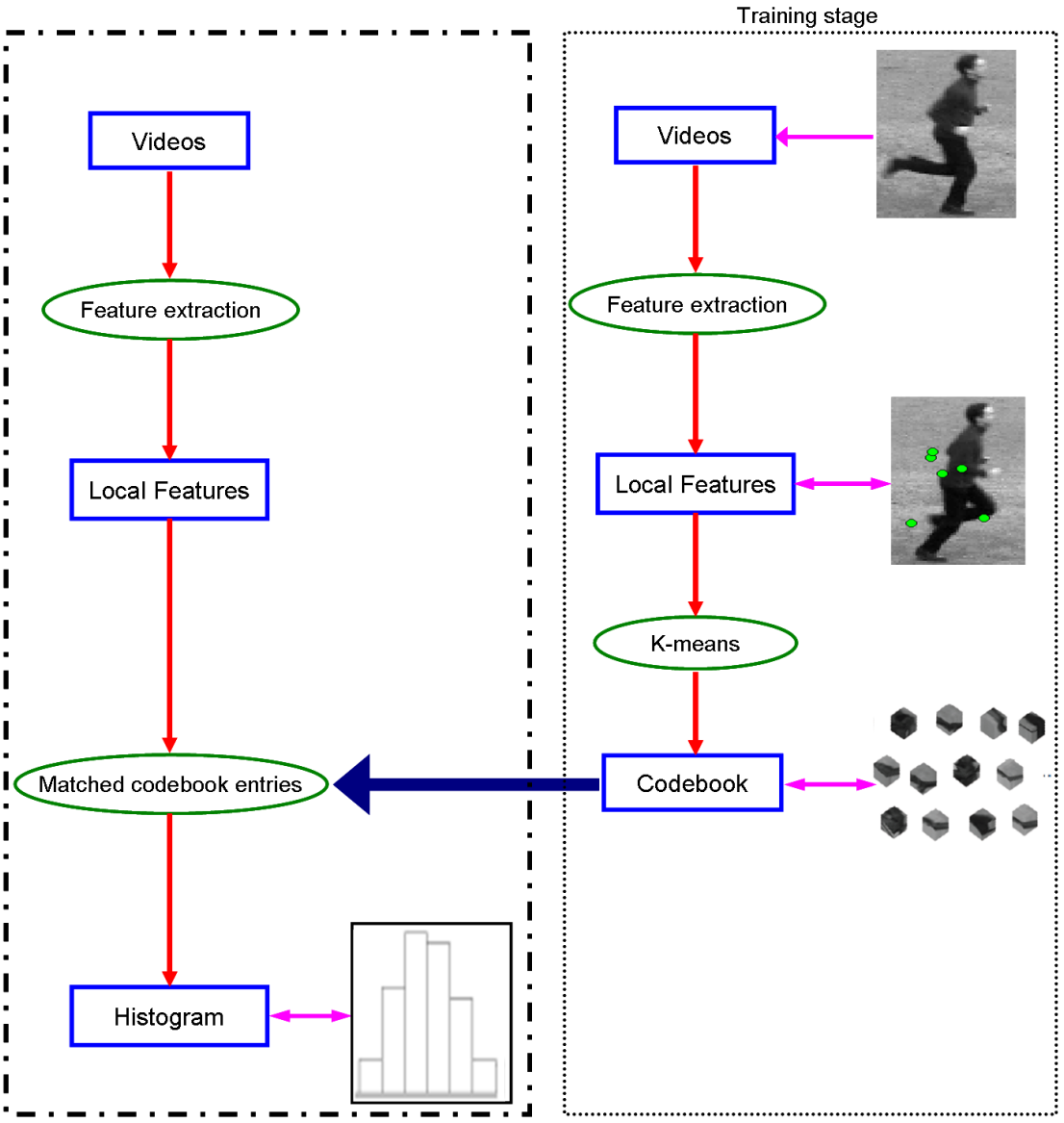


Figure 5.4: Illustration of two steps of the Bag of words model used for action recognition (see the text for more details).

5.3.2.2 Interest point detectors

Spatio-temporal interest points (STIP) are locations in a video sequence, where intensity values present significant variations in both space and time directions. There are several methods to detect STIPs in video. Typically, a response function is computed at every point and salient points, which correspond to local maxima of the response function, are selected as STIPs. Among the different approaches to detect STIPs, the ones introduced by Laptev and Lindeberg [LL03], and by Dollar et al. [DRCB05] are most widely used

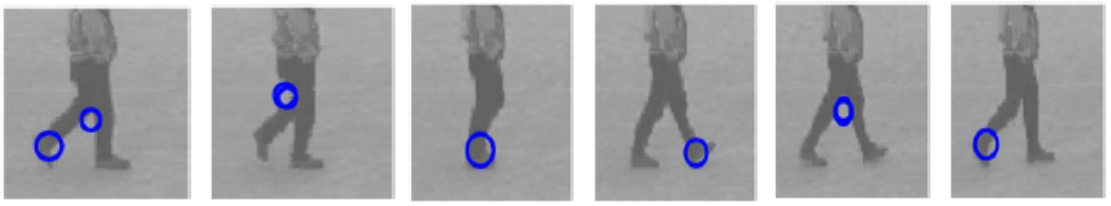


Figure 5.5: Illustration of several spatio-temporal interest points of a walking action, detected by Harris3D [LL03]. We can see that the sptio-temporal interest points are mostly located at corners in spatio-temporal directions.

in action recognition. We present below the details of these two methods, and briefly introduce other methods.

Harris3D – Laptev and Lindeberg [LL03] extended the 2D Harris detector from [HS88] to a 3D space-time detector, called Harris3D. A spatio-temporal second-moment matrix is computed for each video point:

$$\mu(x, y, t, \sigma^2, \tau^2) = g(x, y, t, \sigma^2, \tau^2) * \begin{pmatrix} L_x^2 & L_x L_y & L_x L_t \\ L_x L_y & L_y^2 & L_y L_t \\ L_x L_t & L_y L_t & L_t^2 \end{pmatrix} \tag{5.1}$$

where $L_x = \partial_x L$ and $L_t = \partial_t L$ denote the partial derivatives (space-time gradients) of L in the spatial and temporal dimensions; and σ, τ are spatial and temporal scale values, respectively; g is a Gaussian smoothing function:

$$g(x, y, t, \sigma^2, \tau^2) = \frac{1}{\sqrt{(2\pi)^2 \sigma^4 \tau^2}} e^{-(x^2+y^2)/2\sigma^2 - t^2/\tau^2} \tag{5.2}$$

The locations (x, y, t) of positive local maxima of the following corner function H are selected as spatio-temporal interest points:

$$H = det(\mu) - k * trace^3(\mu) \tag{5.3}$$

where k is a parameter, which was set to 0.0005 as suggested in [LL03].

One drawback of Harris3D detector is that it provides quite sparse interest points (see figure 5.5). This detector favors corners in 3 space-time directions. Inspired by Gabor filters used in image processing, Dollar et al. propose a dense spatio-temporal interest point detector.

Dollar detector – Dollar et al. [DRCB05] used two linear separable gaussian filters in space and a gabor filter in time to detect spatio-temporal interest points. The form of

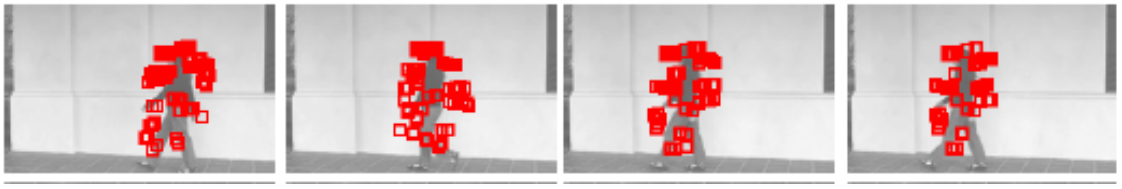


Figure 5.6: Illustration of several spatio-temporal interest points of a walking action, detected by Dollar detector [DRCB05]. We can see that this detector provides very dense interest points. This figure is taken from [NWFF08].

the response function is given as:

$$R = (I * g * h_{ev})^2 + (I * g * h_{od})^2 \tag{5.4}$$

where g is the 2D spatial Gaussian smoothing kernel, h_{ev}, h_{od} are a pair of 1D Gabor filters which are applied temporally.

$$h_{ev}(t, \tau, w) = -cos(2\pi\tau w)e^{-t^2/\tau^2} \tag{5.5}$$

$$h_{od}(t, \tau, w) = -sin(2\pi\tau w)e^{-t^2/\tau^2} \tag{5.6}$$

where $w = 4/t$. The local maxima of the response function are selected as spatio-temporal interest points. Figure 5.6 shows an example of interest points of a walking person, detected by the Dollar detector.

A part from the Laptev detector (Harris3D) and the Dollar detector, there also exist several other methods to detect spatio-temporal interest points.

Other detectors – Oikonomopoulos et al. [OPP06] extended the work on 2D salient point detection by Kadir and Brady [Kad03] to 3D. Positions with local maximum energy, estimated by changes in local information content over different scales, are selected as salient points. Willems et al. [WTG08] introduces the Hessian detector, where the saliency of the determinant of a 3D Hessian matrix are selected as interest points. Bregonzio et al. [BGX09] estimate the focus of attention by subtracting subsequent frames, then apply Gabor filters to detect salient points.

5.3.2.3 Descriptors

It should be noted that the approaches to extract global descriptors, can be applied locally around the interest points to create local features. In this section, we preview

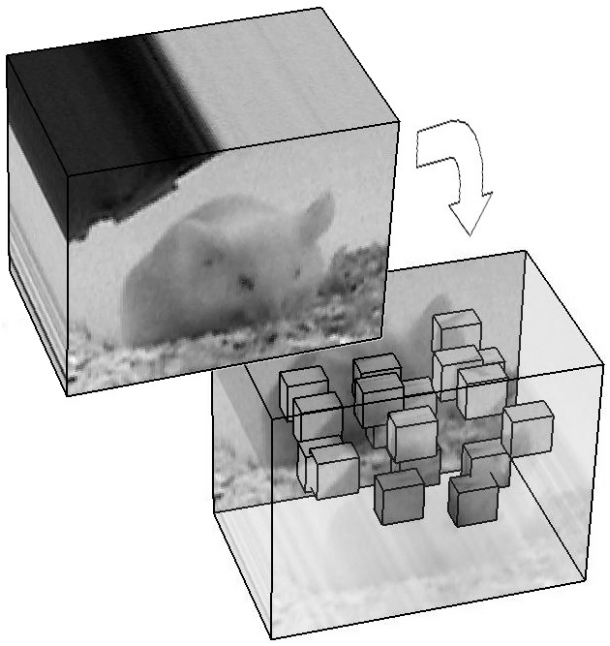


Figure 5.7: Illustration of cuboids from Dollar detector [DRCB05].

several popular local descriptors used in action recognition.

Cuboid descriptor – Along with their detector, Dollar et al. [DRCB05] also proposed the cuboid descriptor. A cuboid is a 3D patch extracted from an interest point (see figure 5.7 for an illustration), and its descriptor is a vector of gradients. First, brightness gradients of pixels in the cuboid region are computed and concatenated into a single vector. The obtained vector is then projected to a lower dimensional space using the Principal Component Analysis (PCA) technique.

HOG/HOF descriptors – The HOG/HOF descriptors were proposed by Laptev et al. [LMSR08] (see figure 5.8). First, each cuboid window (volume) around interest points is subdivided into a 3D grid of cells, then 4-bin histograms of gradient orientations (HOG) and 5-bin histograms of optic flow (HOF) are computed. These histograms are normalized, and concatenated into HOG, HOF, HOG/HOF descriptor vectors.

3D SIFT descriptors – Scovanner et al. [SAS07] extended the idea of 2D SIFT descriptor to 3D (cf. figure 5.9). Similar to 2D SIFT, the 3D SIFT feature descriptor is based on Gaussian space. Each 3D cuboid is subdived into 3D sub-volumes, and the spatio-temporal gradient, the gradient magnitude and orientations in 3D are computed for each sub-volume. The results for each sub-volume are accumulated into its sub-histogram. These histograms are concatenated into a vector histogram (see figure 5.9).

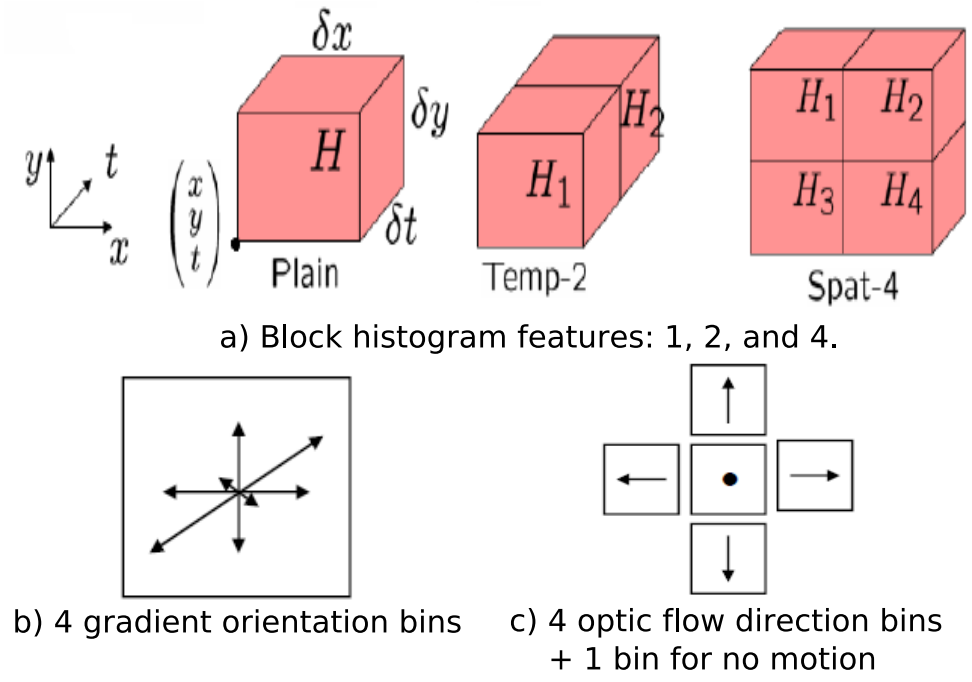


Figure 5.8: Illustration of histograms of spatial gradient and optic flow computed from interest points. First, a cuboid window is subdivided into a 3D grid of celles, 4-bin histograms of gradient orientations (HOG) and 5-bin histograms of optic flow (HOF) are computed for each cell. (reprinted from [LMSR08]).

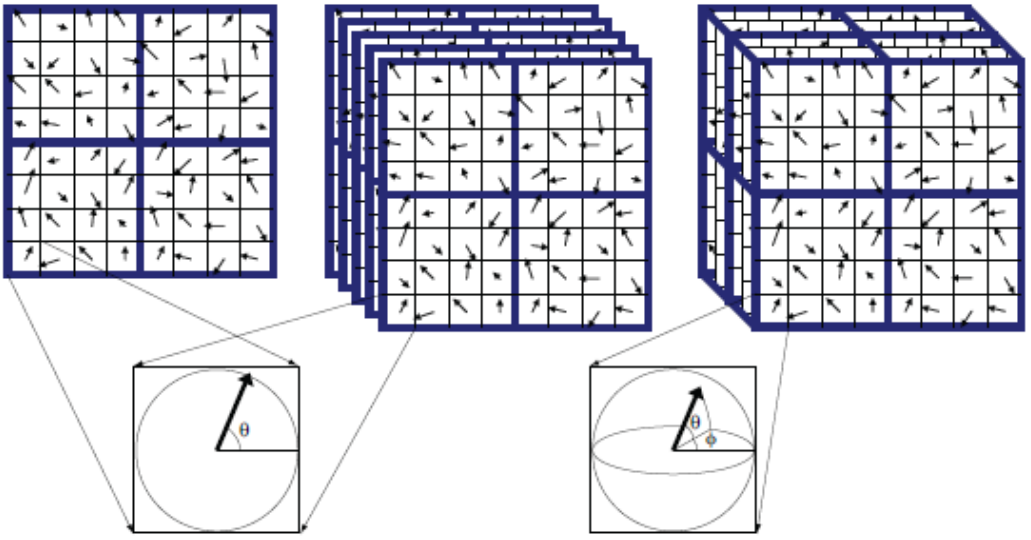


Figure 5.9: The left image illustrates the 2D SIFT descriptor (e.g. [Low04]). The center image shows multiple 2D SIFT (frames level). The last shows how to compute 3D SIFT descriptors from [SAS07].

5.3.3 Hybrid features

We can combine several different features to form hybrid features. Examples from the work of Sun et al. [SCH09], who show that a combination of holistic descriptors (Zernike moments) and local SIFT descriptors give good results.

We would like to note that the evaluation of the performance of different interest point detectors and descriptors is beyond this thesis. We refer the readers to [WUK*09] for an extensive evaluation.

5.4 Statistical classification methods

Statistical approaches have been widely used in object categorization, due to the effectiveness of machine learning techniques. Statistical methods to action recognition usually consist of three main steps (see figure 5.10):

Feature extractions – Usually, local features are extracted from video sequences (see the previous section).

Bag of words representation – Each video sequence is represented by a Bag of words model, i.e. a fixed dimension vector. See section 5.3.2 for more details about BoW model.

Classification – In this step, machine learning techniques are applied to classify the actions. There are two different types of classifiers: discriminative vs generative. Note that, in training stage, the bag of words representations (i.e. histograms), along with labels of the sequences, are provided. However, generative approaches can be used as unsupervised learning.

It should be noted that, in general setting, local features are used in statistical methods, but if global features are used instead of local ones, we can omit the first two steps, i.e. apply classification techniques for vectors representing global features.

5.4.1 Discriminative approaches

Discriminative classifiers are a popular approach to solve classification problems. The main idea is to find optimal frontiers between classes during the training stage. Probably the most popular classifiers of this family are support vector machines (SVM). SVM learn a hyperplane in the feature space, which best separates the two classes (multiple classes are learned in the same principle) - See figure 5.10.e for an illustration.

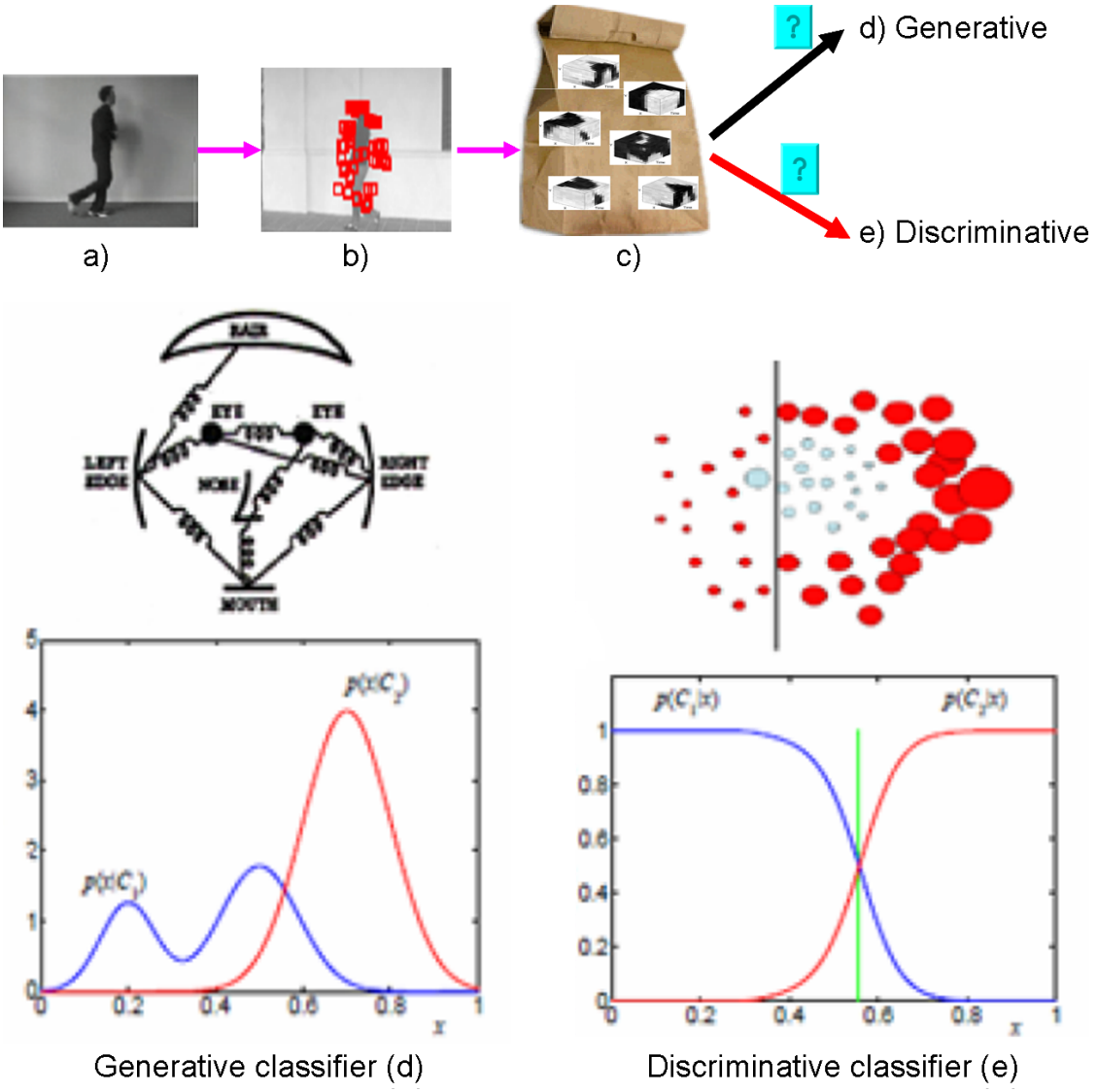


Figure 5.10: Illustration of statistical approaches: a) input video sequences; b) Extraction of local features; c) Bag of words representation; d) Generative classifiers; e) Discriminative classifiers.

Schuldt et al. [SLC04] use the SVM classification combined with motion descriptors (spatio-temporal jets) for action classification. Laptev et al. [LMSR08] apply SVM with DOG/HOG features to recognize actions in realistic videos. Scovanner et al. [SAS07] combine SVM with 3D SIFT descriptors for action classification. SVM has also been used with velocity motion features in [JSWP07] [LCSL07].

Another popular discriminative method is boosting. The principle of boosting is that a strong classifier is formed from a set of weak classifiers, where each weak classifier takes only a single dimension of the features. A most widely used boosting technique is Adaboot, which is used in [KSH05, OCKI06].

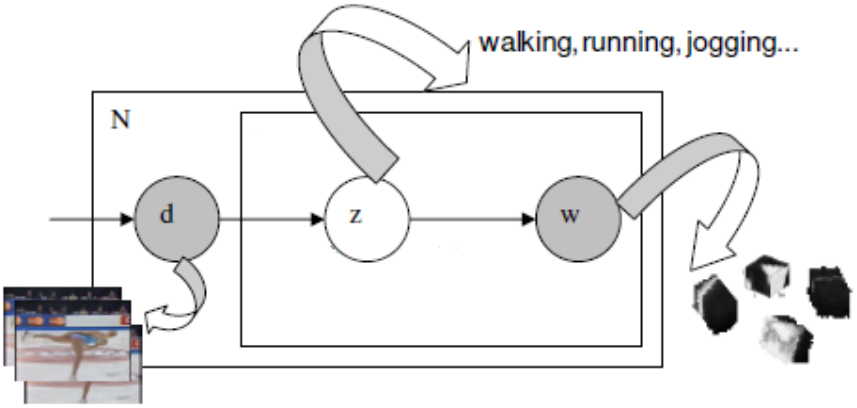


Figure 5.11: pLSA graphical model. Nodes are random variables. Shaded ones are observed and unshaded ones are hiddens. The plates indicate repetitions. This figure is reprinted from [NWFF08]. Here N is the number of video sequences, d represents video sequences, z are action topics (walking, running, etc), and w are spatio-temporal words. We need to find the probability of topic z_k occurring in video d_j, and the probability of a word w_i occurring in the topic z_k.

5.4.2 Generative approaches

While discriminative classifiers find optimal frontiers between classes without explicitly modeling them, generative methods model the generation of the observed data, i.e. the features, from the hidden class information and often from additional latent variables, such as topics. The idea is to learn a model of the joint probability of the inputs and the labels. This learned model is used to predict the activity classes by using Bayes rules to calculate the posterior probability.

One of the most successful used generative methods belonging to the statistical family is the probabilistic Latent Semantic Analysis (pLSA) introduced by Hofmann [Hof99]. pLSA aims at analyzing the co-occurrences of terms in a corpus of documents in order to find hidden/latent topics. Niebles et al. [NWFF08] apply pLSA for unsupervised action classfication. Figure 5.11 illustrates the main components of their method, where d represents video sequences, z are latent topics (i.e. the action categories), and w are spatial-temporal words. Given M videos containing words from a codebook of size V, and the co-occurrence matrix C of size $V \times M$, where $c(w_i, d_j) \in C$ is the number of occurences of a word w_i in video d_j, this method is composed of two stages:

Learning stage – We compute the probability of topic z_k occurring in video d_j, and the probability of a word w_i occurring in the topic z_k. These probabilities are easy to compute because we know the co-occurrence matrix between words and videos C.

Testing stage – Assuming that d_{test} is a new test video. First, for every topic z_k, we

compute the probability of z_k occurring the video d_{test}. Then, the topic that gives highest probability is selected as the action label for the video d_{test}.

There are several improvements on pLSA. Wong et al. [WKC07] integrate the location of object centroid into pLSA model. Wang et al. [WSM07] use a Semi-latent Dirichlet allocation (sLDA), which is similar to pLSA. However, in SLDA, each frame is considered as a codeword, instead of the entire video sequence as used in [NWFF08].

5.5 Other probabilistic graphical models for action classification

Approaches based on probabilistic graphical models are beyond this thesis. We therefore present only major methods used in literature of this family, and do not explicitly discuss it. They differ from the purely statistical ones in that they relate hidden and observed data in more complex (structural) way. In some cases, the probabilistic graphical model is directly related to graph matching, the suject of this thesis. We refer the readers to chapter 2 for more details. Note also, that topic models like pLSA and LDA fall into this category. In the following, we just mention some methods which are not related to these types of approach.

5.5.1 Discriminative approaches

Conditional Random Fields (CRF) are probabilistic discriminative models, that can model multiple overlapping features. We refer the reader to recent work [NXGH08] [SCSW] that have successful applied CRF for action recognition.

5.5.2 Generative approaches

Hidden Markov Models (HMM) are popular generative models, which are used in action recognition. HMM models both hidden states (i.e. transition probabilities between the current state to other state), and observation probabilities. HMM works under the assumptions that there is a linear order of the variables and that observations in time are independent. Several examples of HMM in action recognition, include: Yamato et al. [YOI92] used HMM to recognize the tennis shots such as backhand volley, smash, etc; Weinland et al. [WBR07] use HMM to condition the observation on the viewpoint.

5.6 Other classification methods

Of course, there are other methods that do not neatly fall into the above categories, e.g. Nearest neighbor classification (KNN). KNN is probably the simplest method for action classification. It simply selects k training sequences closest to the test video, and the class label with the highest number of "votes" is chosen for the test sequence. KNN has several advantages: it can be either used with global features (both on frame level or sequence level), and local features. In addition, due to its non-parametric nature, KNN is easy to use. However, also due to its simplicity, with high dimensional data, KNN gives not good results.

Some sucessful methods used KNN: Blank et al. [BGS*05b] use 1-NN for global features; 1-NN has also been used for histograms of codewords in [BCS08]; Dollar et al. [DRCB05] apply 3-NN for cuboid descriptors; Bobick and Davis use [BD01] KNN for Hu moments extracted from silhouettes.

5.7 Spatio-temporal relations based methods

Despite their success, the methods based on bag of words models discard the spatio-temporal relations between local features. As a response, efforts to explore the spatio-temporal information have been undertaken. These improvements concentrate on either constructing new features, taking into account such information, or exploiting such information through co-occurrences between codewords. Our contributions in this thesis fall into the first situation.

Scovanner et al. [SAS07] extended the popular 2D SIFT descriptor for three dimensions (3D SIFT). In order to take into account the spatio-temporal information, they construct a co-occurrence matrix of codewords, and iteratively merge codewords having similar co-occurrences until the difference between them exceeds a given threshold. In a related work, Liu and Shah [LS08] explore the correlation of video-word clusters (i.e. codeword) using a modified correlogram. They optimize the codebook size by using maximization of Mutual Information (MI). MI is defined between codewords and action videos. In this method, two codebook entries are merged if they have similar distributions given MI measures. In addition, the authors integrate spatio-temporal information into the bag of words model by exploiting spatial correlogram and spatio-temporal pyramid matching. Wong et al. [WKC07] introduce both semantic (i.e. appearance) and structural information (i.e. location) of local features into the pLSA model (see section 5.4.2). Gilbert et al. [GIB08] extract all corners in 3 planes (x,y,t), then spa-

tially concatenate these corner descriptors. As there are a large number of features obtained, data mining techniques are applied to select informative features. Niebles and Fei-Fei [NFF07] propose a constellation of bags of words, which combines both spatial and spatial-temporal features, i.e. a video sequence is a hierarchical constellation of frames. They use both dynamic and static features. Zhang et al. [ZHCC08] introduce the concept of motion context to capture both spatial and temporal distribution of motion words. Oikonomopoulos et al. [OPP09] encode the spatial co-occurrences of pairs of codewords and propose a probabilistic spatiotemporal voting framework for localizing and recognizing human activities in unsegmented image sequences. Very recently, Ryoo and Aggarwal [RA09] present a spatio-temporal relationship matching method for action recognition. They first define a set of spatial and temporal predicates such as vertical near, horizontal near, vertical/horizontal far, before, etc. Next, a codebook is used for video representation. Finally, to perfom action recognition, they introduce a method to count the number of pairs of codewords satisfying one of the above logical predicates that two given videos share in common. This results in a "score", which is used as a detection threshold.

Among the methods presented above, conceptually, one of our contributions developped in chapter 6 is similar to the work of Gilbert et al. [GIB08], who propose compround features, which take into account spatio-temporal information.

5.8 Video matching

There exists another direction in action recognition, namely video matching or video correlation. These approaches differ from the traditional classification-based ones in that they do not explicitly model the actions to be recognized, nor do they learn these actions. One of our contributions in this thesis falls into this category. We review hereafter related works in literature.

Zelnik-Manor and Irani [ZMI06] compute the distance between two sequences by the distance between two corresponding distributions of histograms of appearance-normalized gradient patches in a temporal pyramid. In this approach, only patches undergoing moving areas are taken. As a result, this method does not work on non-moving backgrounds. Shechtman and Irani [SI07b] define a motion consistency measure to match space-time volumes directly. However, the distance between pair of videos is computed by exhaustively comparing patches extracted from every space-time point. Ke et al. [KSH07] combine a part-based shape and flow matching framework from [SI07b] for event detection in crowded videos. In their later work, Shechtman and Irani [SI07a]

introduce a self-similarity descriptor. A distance between two video sequences are measured by the distance between two sets of corresponding local self-similarity descriptors. They also propose several heuristics to ignore homogeneous regions. However, this method is still time consuming, due to the exhaustive search for computing features. Recently, Seo et al. [SM09] introduce space-time local steering kernels, which are the generalization of the self-similarity descriptors described in [SI07a]. They organize these local kernels into a matrix, so-called *steering* matrix. Then, PCA is applied on this matrix to select salient features. A matrix cosine similarity is also introduced to compute the distance between two given videos. This method requires sliding windows in both space and time, making it computationally expensive. In the spirit of video matching, Ryoo and Aggarwal [RA09] have presented a histogram-based match kernel for video matching (see section 5.7, as we also classify this method as the Spatio-temporal relations based methods).

Among the methods mentioned above, one of our approaches developped in chapter 7 is most closely related to the work of Ryoo and Aggarwal [RA09], who perform video matching from two sets of spatio-temporal interest points. Our method differs from their work in two main points. First, the authors in [RA09] define a set of logical predicates for taking into account the pairwise relationships among points. Instead, we *naturally* take into account higher-order relationships between points through a graph-matching technique. It should be noted that such logical predicates are difficult to extend to higher relationships. Second, their method needs to train a codebook from the training sets, while our method does not require any learning.

5.9 Conclusion

In this chapter, we have reviewed the existing methods for action recognition. We have discussed separately the features and methods used in action recognition. Generally, there are two main types of features used for video representation: holistic vs local. We have classified the existing methods into two main groups: Statistical vs Probabilistic graphical models. In this section, we summarize the advantages, as well as limitations of each method:

Holistic vs Local – Holistic features have several advantages such as they capture global information of activities, and are easily extracted with low cost. However, they usually require preprocessing steps, e.g. background subtraction. In addition, holistic features cannot deal with oclusion. Local features have been proposed to deal with this problem. Local features are usually combined with Bag of words models to video repre-

sentation. Althrough they can deal with changes in background, scale and illumination, and occlusion, local features based approaches are computational expensive. Moveover, the bag of words models much valuable information such as the spatio-temporal information. Many efforts have been proposed to overcome this restriction. These approaches concentrate on exploiting the spatio-temporal relationships between local features or among the codewords.

Statistical vs Probabilistic graphical models – Statistical methods can benefit from learning techniques, which have been well studied in machine vision domain. Statistical-based methods usually ignore the spatial relations between features. In constrast, probabilistic graphical models methods can model the temporal and spatial relationships between features. Hoverver, the choice of an inference algorithm is not trivial.

Generative vs Discriminative – Generative approaches can naturally model the prior class distributions, and can handle missing (unsupervised learning) or partially labelled data. However, they seem less suitable for distinguishing between two actions that are very similar, e.g. running vs jogging. Discriminative methods can improve this limitation by directly modelling the mapping from inputs to outputs. However, when data be learned become complex, and training samples are not large enough, Discriminative approaches may give unreliable results.

Currently, the state-of-the-art approaches in action recognition implicitly assume that videos are already segmented into a single sequence, i.e. each sequence contains exactly one action instance. This limits the applicability to realistic environments, where one video often contains multiple different activities. Besides, the localization task is often ignored in an action recognition system (except the work from Ryoo and Aggarwal [RA09], and from Oikonomopoulos et al. [OPP09]). These problems are addressed by our work described in chapter 7.

Chapter 6

Exploring the spatio-temporal relationships for activity recognition

Contents

Existing action recognition approaches mainly rely on the discriminative power of individual local descriptors extracted from spatio-temporal interest points (STIP), while the geometric relationships among the local features[1] are ignored. In many applications, e.g. actions of interest in complex scenes (i.e. cluttered backgrounds or partially

[1]We consider the features extracted from interest points as local features

occluded crowds), simple collections of local features are not sufficient to describe activities. In this chapter, we overcome this restriction by introducing new features, called pairwise features (PWF), which encode both the appearance and the spatio-temporal relations of the local features for action recognition. For a given video sequence, STIPs are first extracted, then PWFs are constructed by grouping pairs of STIPs which are both close in space and close in time. We also propose a combination of two codebooks for video representation. Experiments on two standard human action datasets: the KTH dataset and the Weizmann dataset show that the proposed approach gives good results.

The work described in this chapter was published in the 20th international conference on Pattern Recognition 2010 [APTJ10].

6.1 Introduction

Human activity recognition is an active field of research in computer vision, which has applications ranging from video surveillance and monitoring, human-computer interactions to content-based video annotation and retrieval. Due to the non-rigidness of the human body, activity recognition is a very challenging task. One way to tackle the problem is to recognize behaviors of people in entire video sequences, or in spatio-temporal sublocks handled with a temporally sliding windows. Thus, the activity recognition problem can be treated as a classification problem, where each activity is considered as a class label. As mentioned in chapter 5, existing action recognition methods can be broadly divided into two categories: local approaches [DRCB05, NWFF08, SDPNL08, SLC04] and holistic approaches [MU, LAS08, WBR07, WGLR08]. Our work falls into the first category. Local approaches to activity recognition, which are usually based on bag-of-words models (BoW) (cf. chapter 5), are very popular because they can deal with changes in background, scale, illumination and occlusion. However the BoW models discard the spatio-temporal layout of the local features, which may be almost as important as the features themselves. Our goal is to improve BoW models by constructing new features from local features, which are extracted from a video sequence. To this end, we take into account the following information:

Temporal aspects – The temporal direction, which has been ignored in the BoW models, is very important in an activity recognition system. Each action usually takes a certain amount of time, therefore capturing temporal relations (e.g. temporal order) between STIPs describes the motion information, which characterize activities.

Spatial aspects – Spatial relationships between local features, which are represented in different forms (e.g. star topologies, fans models), have been widely used in object recognition. However, they have usually been ignored in activity recognition. This is due to the fact that a straightforward application of the techniques used in object recognition to action recognition is very difficult. In this chapter, we propose to exploit spatial relations between local features.

Combining spatial and temporal information – Intuitively, if two local features are adjacent in both space and time dimensions, they should belong to the same action. From this intuition, we construct our new feature from pairs of *adjacent* local features. Previous works (e.g. from [OPP09]), which exploit spatio-temporal relationships, usually consider spatial distance in the same way as temporal distance.

However, in video processing, one unit of time is much important than one unit of space. Thus, we adjust the difference between these two domains by weighting.

Our algorithm is related to the works that exploit information on the spatial and temporal distribution of interest points to improve the BoW models. Wong et al. [WKC07] introduce both semantic (i.e. appearance) and structural information (i.e. location) of local features into the pLSA model (see chapter 5). Scovanner et al. [SAS07] extended the popular 2D SIFT descriptor for three dimensions (3D SIFT). In order to take into account the spatio-temporal information, they construct a co-occurrence matrix of codewords[2], and iteratively merge codewords having similar co-occurrences until the difference between them exceeds a given threshold. In a related work, Liu and Shah [LS08] explore the correlation of video-word clusters (i.e. codewords) using a modified correlogram. Gilbert et al. [GIB08] spatially concatenate corner descriptors detected on different regions and apply data mining techniques to construct compound features. Zhang et al. [ZHCC08] introduce the concept of motion context to capture both spatial and temporal distribution of motion words. Oikonomopoulos et al. [OPP09] encode the spatial co-occurrences of pairs of codewords and propose a probabilistic spatiotemporal voting framework for localizing and recognizing human activities in unsegmented image sequences. Ryoo and Aggarwal [RA09] present a spatio-temporal relationship matching method by defining the spatial and temporal predicates such as near, far, etc.

The methods mentioned above are mainly based on the discriminative power of individual local features for codebook construction and thus the performance depends on the local features used. In this chapter, we present new features, called pairwise features (PWF), which capture both appearance and geometric relationships among local spatio-temporal interest points (STIPs). The main point is that the performance of our method is usually due to the geometric information, and thus does not really depend on the local appearance descriptors (STIP descriptors) used. Our method differs from most the state-of-the-art methods in two main points:

- Our feature is not limited to the appearance information but also includes geometric information.
- Our method incorporates both the spatial and temporal relationships among local space-time points in a significant manner, i.e such relationships are constrained to time and space.

Actually, spatio-temporal information can be either considered at the codebook construction stage, or directly encoded into the features, i.e. to form new features. For a

[2] A codeword is considered as a representative of a group of similar words (patches, keypoints, etc).

Table 6.1: Comparison of the performance dependence on local appearance descriptors of different methods, taking into account spatio-temporal information. The term *levels* indicates at which level spatio-temporal information is encoded, and the term *dependencies* indicates the *amount* of dependencies on local appearance descriptors.

Approaches	Levels	dependencies
Wong et al. [WKC07]	codeword level	largely
Scovanner et al. [SAS07]	codeword level	totally
Liu and Shah [LS08]	codeword level	totally
Gilbert et al. [GIB08]	feature level	partially
Zhang et al. [ZHCC08]	frame and codeword levels	largely
Oikonomopoulos et al. [OPP09]	co-occur and codeword level	totally
Ryoo and Aggarwal [RA09]	codeword level	totally
Our approach	feature level	little

better comparison, we listed, in table 6.1, a comparison of the difference between other methods taking into account spatio-temporal information and our method. The different levels used in this table are explained in the following:

- The term *codeword level* means that appearance features are firstly used to construct a codebook, then several codewords are combined (i.e. to take into account spatio-temporal information) to represent a video sequence.

- The *frame level* based methods try to combine several adjacent frames before extracting local features. Some techniques like frame differencing is usually used.

- Finally, *feature level* based methods encode both appearance features and spatio-temporal information to construct compound features. Note that the work from Gilbert et al. [GIB08] differs from our method in that they encode the local appearance descriptors such as the gradient orientation, the channel and the scale of interest points (returned by an interest point detector), into their comproud features, thus their method still depends on the appearance descriptors used. In contrast, we encode separately the spatio-temporal relations of local features into our new features (see section 6.3).

As shown in table 6.1, while most other methods rely heavily on the local appearance descriptors (e.g. SIFT, cuboid, etc) extracted from interest points, our method only slightly depends on the choice of local appearance descriptors. We will confirm this fact through experiments presented in section 6.7. In the next section, we present the overview of our method.

6.2 Overview of the proposed method

We follow the standard framework which is widely used in litterature, i.e. Bag-of-Words model using local features (cf. chapter 5), to test our pairwise features. Figure 6.1 illustrates the main components of our approach. The spatio-temporal interest points (STIPs) are first extracted from the video sequences, then PWFs (pairwise features) are constructed from them. We apply the bag of words (BoW) model using PWFs for video representation, which requires creating a visual vocabulary. To this end, we generate two codebooks (vocabularies) according to the appearance descriptors and the geometric similarity of the PWFs. A video sequence is then represented by two histograms of visual words. These two histograms are combined into a single vector, and Support Vector Machines (SVM) are used for action classification.

6.3 Pairwise descriptor

We propose new features, namely pairwise features (PWF), which encode both STIP descriptors and the spatio-temporal relations among the STIPs. Essentially, two STIPs are connected to form a PWF if they are adjacent in space and in time. Intuitively, two STIPs that are close both in space and in time often belong to the same human activity.

A spatio-temporal detector usually detects interest points locating salient changes in a video sequence, and descriptors are extracted around these interest points. Thus, in general a local feature contains two types of information: appearance information, and space-time coordinate information. We denote a spatio-temporal local feature as $f = (f_{des}, f_{loc})$ where f_{des} is an arbitrary appearance descriptor and f_{loc} is its space-time coordinates. Let $f_1 = (f_{1des}, f_{1loc})$ and $f_2 = (f_{2des}, f_{2loc})$ be two local descriptors (i.e., local STIP descriptor), a PWF $p = (F_{des}, \mathbf{F}_{vec})$ is established if the conditions below are satisfied:

a) $d_s(f_{1loc}, f_{2loc}) \leq t_s$, AND

b) $d_t(f_{1loc}, f_{2loc}) \leq t_t$

where F_{des} is a concatenation of f_{1des} and f_{2des}; $\mathbf{F}_{vec}$ is a geometric vector from the first location (f_{1loc}) to the second one (f_{2loc}) in temporal order. If f_1 and f_2 are in the same frame, the STIPs in the PWF are ordered from left to right; t_s is a spatial threshold, t_t is a temporal threshold; $d_s(.,.)$ and $d_t(.,.)$ are spatial distance and temporal distance functions, respectively. We can imagine each PWF as a segment in the 3D space (see figure 6.2 for an illustration).

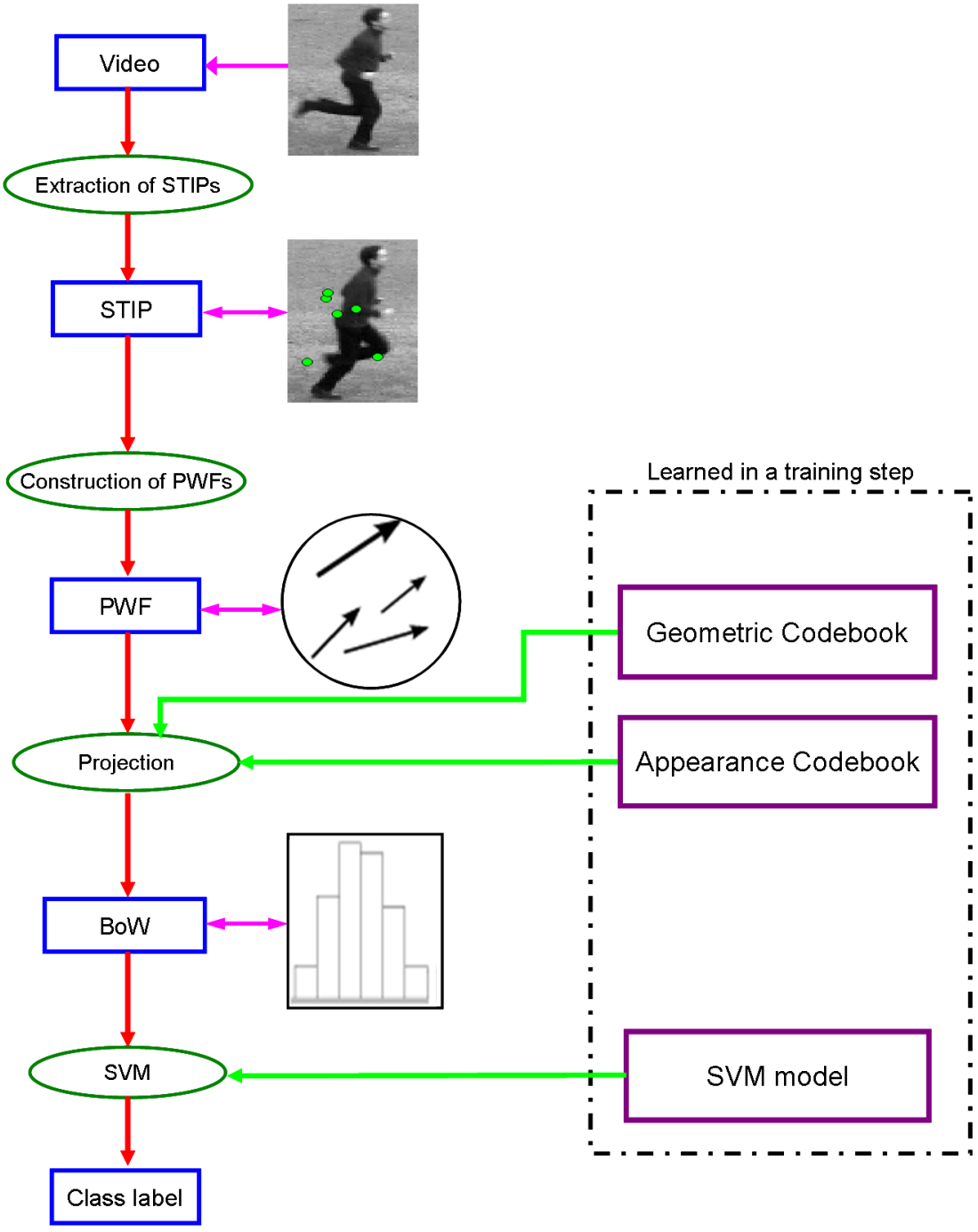


Figure 6.1: The diagram of the proposed action recognition framework. Here, STIP is the abbreviation of "Spatio-Temporal Interest Point", BoW for "Bag of Words model", PWF for "Pairwise features".

6.4 Extracting local features

Recall that our pairwise features are established from pair of STIPs. Therefore, we need to extract STIPs from video sequences. Our method does not require any specific STIP

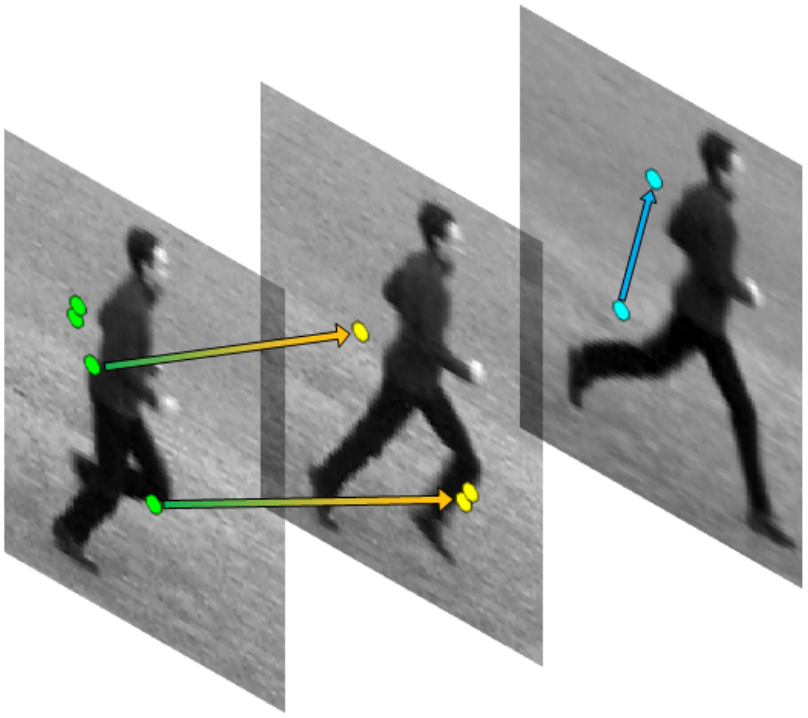


Figure 6.2: Illustration of several pairwise features as segments in space-time. Note that apart from spatio-temporal relationships, we are also motivated to take into account spatial relationships (i.e., PWFs in the same frame).

detector, and we chose the Dollar detector [DRCB05] to detect STIPs (cf. chapter 5). This detector is based on temporal Gabor filters: first 2D Gaussian kernels are applied in space domain, and 1D Gabor filters (in time dimension) are then applied to the obtained result. Interest points are detected as local maxima of the response function.

We also chose the cuboid descriptor from [DRCB05] as the appearance descriptor for our PWF. Cuboid descriptor is a vector of gradients: first spatio-temporal gradients of pixels in a cuboid region are computed and concatenated into a single vector. The obtained vector is then projected to a lower dimensional space using the Principal Component Analysis (PCA) technique.

6.5 Comparing pairwise features

Once the PWFs are constructed, the video sequence is considered as a collection of PWFs. In order to compare different PWFs, we need a similarity measure. Note that a PWF p contains not only the appearance descriptor F_{des} but also the geometric information $\mathbf{F}_{vec}$, which is translation invariant. Since the physical meanings are different, it

is not desirable to construct a single codebook based on PWFs. We propose to generate two codebooks: the first codebook is generated by clustering only the appearance descriptors of the PWFs, and the second one is generated through the clustering of the geometric descriptors. In order to use a clustering algorithm (e.g. k-means) for codebook construction, we need to define two distance functions corresponding the two feature parts of PWFs. These distance functions are given as follows:

Distance metric between two geometric descriptor parts of two PWFs — We present hereafter a distance metric for the geometric descriptors of the PWFs. Let p^i and p^j be two PWFs, a measure $d_g(p^i, p^j)$ of the geometric similarity between two PWFs is given as follows:

$$d_g(p^i, p^j) = \frac{(\mathbf{F}^i_{vec} - \mathbf{F}^j_{vec})^T \Sigma^{-1} (\mathbf{F}^i_{vec} - \mathbf{F}^j_{vec})}{||\mathbf{F}^i_{vec}|| + ||\mathbf{F}^j_{vec}||} \tag{6.1}$$

where Σ is a matrix capturing variations and correlations between the data dimensions similar to a covariance matrix; In our case we propose to set it to a diagonal matrix:

$$\Sigma = \begin{vmatrix} \lambda_s & 0 & 0 \\ 0 & \lambda_s & 0 \\ 0 & 0 & \lambda_t \end{vmatrix}$$

where λ_s and λ_t are thus weights adjusting the differences between the spatial and the temporal domains.

Note that our distance $d_g(.,.)$ without denominator corresponds to the Mahalanobis distance, where Σ is a covariance matrix. The normalization factor $1/(||\mathbf{F}^i_{vec}|| + ||\mathbf{F}^j_{vec}||)$ is used to make the distance scale invariant.

Similarity between two appearance parts of two PWFs — Since the appearance part of a PWF is a histogram vector (concatenation of two local appearance descriptors), we can use the Euclidean distance to measure the distance between two appearance parts of two PWFs. This appearance distance between two PWFs p^i and p^j is given as follows:

$$d_d(p^i, p^j) = D_e(F^i_{des}, F^j_{des}) \tag{6.2}$$

where D_e is the Euclidean distance.

6.6 Video representation by using pairwise features

6.6.1 Codebook construction

We first extract local features from training videos, and then PWFs are constructed from them in the following way: the descriptor part of the PWF is represented as a concatenation of the two cuboid descriptors (cf. section 6.4), and the geometry part is a geometric vector constructed from two space-time coordinate components (cf. section 6.3). Once PWFs are constructed, K-means algorithm is used to construct the two codebooks:

Geometric codebook – Apply k-means algorithm to **geometric parts** of PWFs using the measure described in eq. 6.1.

Appearance codebook – Apply k-means algorithm to **appearance parts** of PWFs using the measure described in eq. 6.2.

6.6.2 Video representation

By mapping the PWFs extracted from a video to the vocabularies (i.e. the two codebooks constructed above), a video sequence is represented by the two histograms of visual words, which are denoted as H^d and H^g. We introduce a combination of these two histograms to form a feature vector as input to a classifier such as SVM:

$$H = \{\alpha * H^d, (1-\alpha) * H^g\} \tag{6.3}$$

where α is a weighting coefficient. The coefficient α here does not interfere with the classifier intself, but only compresses (or increases) the dynamic rang of the corresponding dimensions in the feature vector. With a perfectly adoptive learning machine, changing α would not lead to any changes in the results. Since the decision functions of realistic learning machine (SVM, MLP, etc.) are necessarily regularized, changing α leads to the desired result of controlling the influence of parts of the feature vector.

The advantage of this combination method is that the two histograms do not necessarily have the same size. Another advantage of this combination is that we can use each of PWF parts alone, i.e. setting $\alpha = 1$ for using only appearance part of PWF, and $\alpha = 0$ for using only geometric part.

6.7 Experiments

6.7.1 Datasets

Our experiments are carried out on the standard KTH and Weizmann human action datasets (cf. chapter 5). The Weizmann dataset contains 10 different actions (bend, jack, jump, jump in place, run, side, skip, walk, wave1, wave2) performed by 9 different subjects. There are totally 93 sequences in this dataset. The KTH contains 599 sequences of six types of human actions (walking, jogging, running, boxing, hand waving and hand clapping) performed by 25 subjects in four different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes and indoors.

6.7.2 Activity recognition

Testing protocol — We perform leave-one-out cross-validation and report the average accuracies. More precisely, for each run, we leave the videos of one subject as test data, and use the rest of the videos for training. K-means clustering is applied to define the visual vocabulary, with 512 visual words for KTH and 250 for Weizmann for both descriptor and geometric codebooks, respectively (cf. section 6.6). For constructing PWFs, t_s was set to 20 and t_t was 5. To reduce complexity, we select the 1500 most significant PWFs for each video sequence, where significance is measured as the product of the response functions of the two interest points corresponding to the PWF. Recognition was performed using a non-linear SVM with a χ^2 kernel. To adjust the differences between the spatial and the temporal domains, λ_s and λ_t were set to 7 and 1, respectively for all experiments.

Classification accuracy — Figure 6.3 shows the performance obtained on the KTH dataset. Threshold α indicates the percent contribution from the appearance parts of PWFs, which has been taken for classification, e.g. figure 6.3a ($\alpha = 0$) shows visual results achieved by using only geometric information of PWFs, while figure 6.3b) illustrates obtained results by using only descriptors parts. As shown in this figure, running is frequently confused with jogging due to the similar characteristic of these two actions.

It can be observed (fig. 6.3a) that even with only geometric descriptors used for recognition, the result is very good. This result is very promising and suggests that it is possible to avoid the non-trivial problem of choosing an appearance descriptor for action recognition, i.e. exploiting only geometric distribution among STIPs.

Through experiments we found that, the recognition results obtained using only appearance parts of PWFs (i.e. figure 6.3b) are in general, better than the ones using

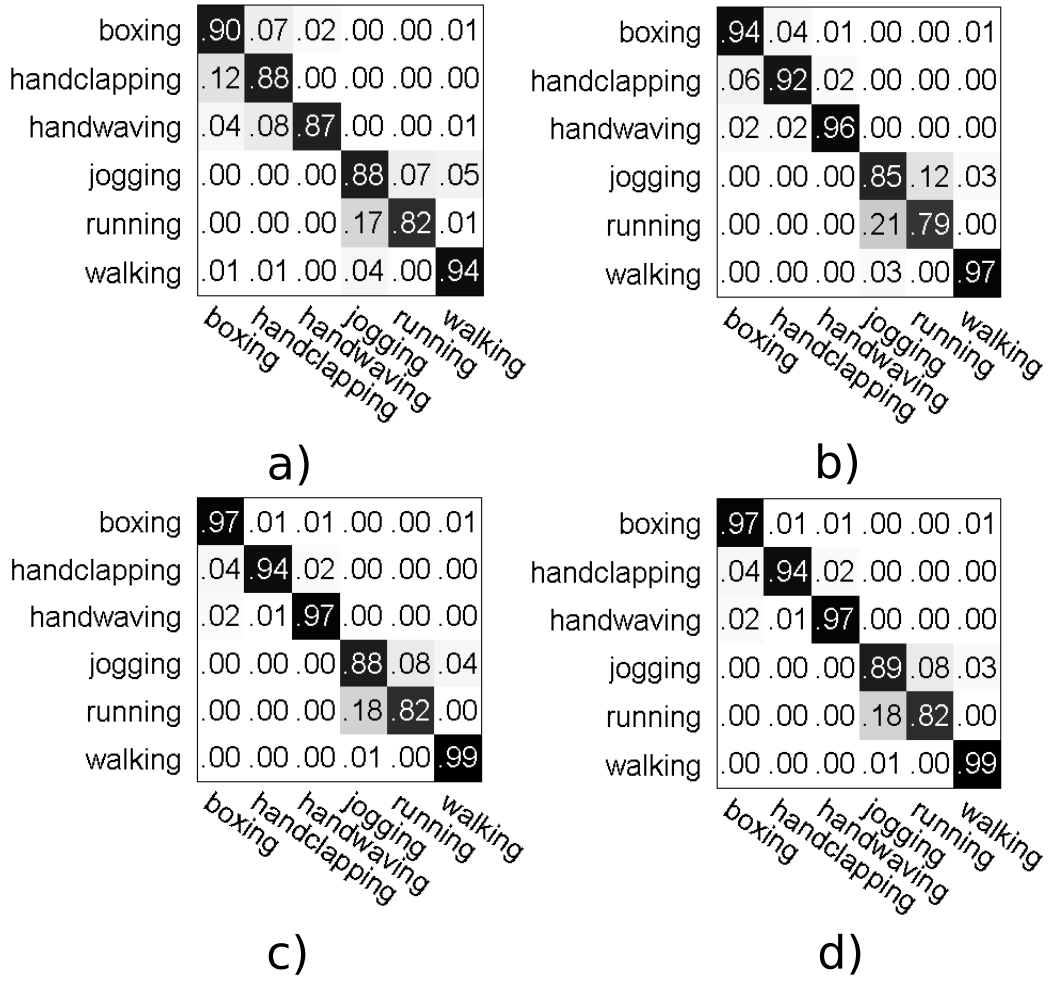


Figure 6.3: Confusion matrices on the test KTH dataset: a) $\alpha = 0$ and accuracy (ac) = 88.2%; b) $\alpha = 1.0$ and ac = 90.5%; c) $\alpha = 0.5$ and ac = 92.8%; d) $\alpha = 0.6$ and ac = 93.0%;

only geometric parts (i.e. figure 6.3a). However, for some difficult actions, e.g. the running and jogging actions, geometric features are more robust (i.e., 82% versus 79% for running, and 88% versus 85% for jogging). It should be noted that even if the results from appearance parts of PWFs (i.e. figure 6.3b) are better on average, these features also contain spatial-temporal information, i.e. intrinsic to the construction of PWFs (see section 6.3).

The best result obtained for the KTH dataset is reported in figure 6.3d, which indicates that the appearance and the geometric descriptor of the PWF are complementary to each other, e.g. adding geometric information of PWFs improved the performance of jogging action by up to 4% (from 85% (6.3b) to 89% (6.3d)).

The motivation for showing figure (fig. 6.3c) here is that we would like to demon-

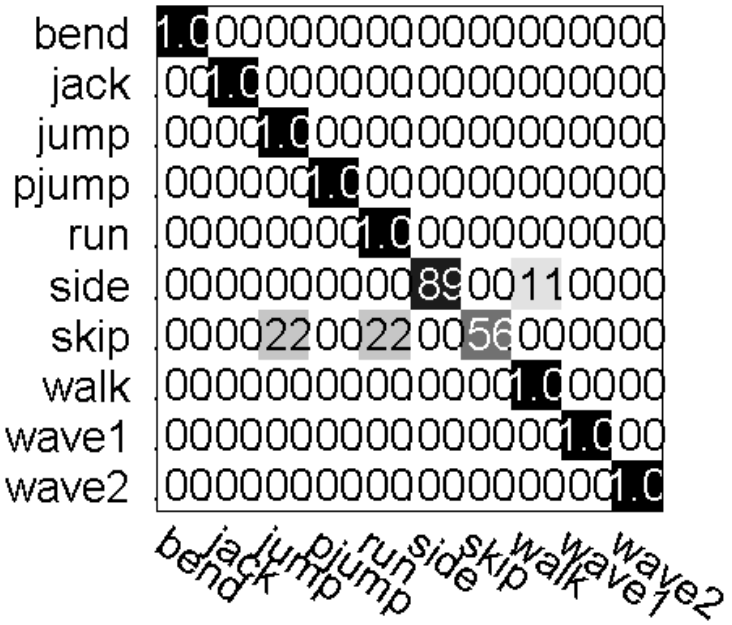


Figure 6.4: Confusion matrices on the test Weizmann dataset by setting $\alpha = 0.6$. The average accuracy is of 94.5%.

strate the *simplicity* of our combination codebook method (cf. eq. 6.3) for choosing an optimal weight, i.e. it is easy to find a *near* optimal weighted parameter α. Indeed, by setting equally the contributions of the two feature parts of PWFs, the obtained result is of 92.8%, which is not far from the optimal case with 93.0%.

Figure 6.4 illustrates the results obtained on the Weizmann dataset. We report here only the performance for 60% contributions from geometric information and 40% contributions from appearance descriptors of PWFs. For this dataset, the recognition accuracy is 100% for several actions. The worst recognition occurs for the "skip" action, which was often confused with the "run" and the "jump" actions. It should be noted that the KTH dataset is much more challenging than the Weizmann dataset. The currently best reported recognition rate on the Weizmann dataset is of 100% (see table 7.1).

Finally, to evaluate the stability of our method, we also performed the tests with $\lambda_s = \lambda_t = 1$, the accuracies moderately decrease by roughly 1% for both datasets.

6.7.3 Comparison to the state-of-the-art

Table 1 presents a comparison of our results with state-of-the-art results reported recently. This table is divided into three groups: the first group consists of the methods which can be directly compared with ours, i.e use the same features (cuboid descriptor) and the same experimental set-up for action classification; the second one includes the

Table 6.2: Comparison of our method with different methods, tested on KTH and Weizmann datasets.

Method	KTH	Weizmann	
Our method	**93.0**	**94.5**	
Dollar et al. [DRCB05]	81.2	-	- Same features
Niebles et al. [NWFF08]	83.3	90.0	as our work
Savarese et al. [SDPNL08]	86.8	-	
Oikonomopoulos et al. [OPP09]	80.5	-	
Scovanner et al. [SAS07]	-	82.6	
Gilbert et al. [GIB08]	89.9	-	BoW + Spatio-temporal
Zhang et al. [ZHCC08]	91.3	-	relationships
Wong et al. [WKC07]	91.6	-	
Ryoo and Aggarwal [RA09]	93.8	-	
Liu and Shah [LS08]	94.2	-	
Wong and Cipolla [WC07]	86.6	-	
Ballan et al. [BBDB*09]	92.1	92.4	
Sun et al. [SCH09]	94.0	97.8	
Laptev et al. [LMSR08]	91.8	-	
Fathi and Mori [FM08]	90.5	100.0	
Gorelick et al. [GBS*07]	-	99.6	Other methods
Schindler and Gool [SvG08]	92.7	100.0	
Liu et al. [LAS08]	-	90.4	
Klaser et al. [KMS08]	91.4	84.3	
Willems et al. [WTG08]	84.3	-	

methods that exploit spatio-temporal relations among STIPs or among visual words. Among the methods in the first group, our method obtained the best accuracy for both datasets. Our method outperforms most existing methods which take into account the spatio-temporal relationships (second group). The advantages and gains of our method are thus two-fold:

- Our method is able to boost the performance obtained from the cuboid descriptor (first segment in table 7.1)
- Our method boosts the perfromance of the widely used BoW models (second segment in table 7.1).

Note that the results which are most close to ours, i.e the work from Ryoo and Aggarwal [RA09], requires encoding more semantic information about the interactions (17 in total) between STIPs.

On the KTH dataset, our recognition rate of 93.0% is very close to the current best rate of 94.2% . It should be noted that the selection of different numbers of codewords sometimes lead to different recognition rates. Note that we cannot directly compare to results reported in the last group of table 1, because they exploit both holistic and local representation [SCH09], included more data given by segmentation masks [GBS*07,

Table 6.3: Testing the performance of the feature parts (appearance, geometry) of the PWF in a new experimental set-up: learning on one dataset and testing on another one.

Feature part	Learning	Testing	ac.
Appearance descriptor	KTH	WM	40.0
	WM	KTH	48.2
Geometric descriptor	KTH	WM	70.0
	WM	KTH	83.3

FM08], or use another experimental set-up.

6.7.4 Robustness of geometric features across datasets

In the previous section, we demonstrated that each feature part of the PWFs gives good results, even used alone. In this section, we will demonstrate that in some very difficult cases, the geometric part is still robust enough to produce accurate results, while the appearance descriptor part seems to be unstable.

To this end, we perform another experiment to verify the discriminative power of our feature across different datasets. In fact, the geometric part of our PWF is a generic feature, which is robust against database changes. To verify this fact, we performed an experimental evaluation through training on one dataset and testing on another one. More precisely, the experimental tests are carried out using three common actions which exist in both datasets: handwaving, running, walking. Because there are only 29 videos of these actions in the Weizmann dataset, we randomly select 30 videos containing these three actions from the KTH dataset to perform our tests. We first extract PWFs from training dataset, and construct codebooks, with size of 100 words, for each feature part (geometric part and appearance part). Then, the other dataset is used as test set. Note that, in this experiment, we also use SVM with χ^2 kernel to train the model.

In table 6.3, we show the results obtained using each feature part of the PWFs alone. From this table, it can be seen that the appearance descriptors of the PWFs failed, while the geometric descriptors work very well. This is due to the fact that our geometric part depends only on spatio-temporal relationships of the interest points. This is an important advantage over all other methods, which take into account spatio-temporal information based on the local appearance descriptors (cf. table 6.1). These results are very interesting for future research in human action recognition, i.e. to avoid the non-trival problem of choosing an appropriate appearance descriptor.

Note that the video sequences in the Weizmann dataset are of much shorter duration compared to the ones in the KTH dataset. This explains why there are large differences

in the performance in our second test (e.g. from %70 to 83%).

6.8 Conclusions

In this chapter, we have presented a new feature for action recognition. Our method differs significantly from previous local-based approaches in that our feature is a semi-local representation, which encodes both the appearance and geometric information among local features, while most other methods exploit spatio-temporal information in the video representation stage (i.e. codeword level - see table 6.1), and thus still depend heavily on the local appearance descriptors extracted around interest points.

Through experiments, we have proved that exploiting the location information of local features gives valuable improvements to the conventional BoW models.

Beyond this contribution, we have demonstrated that our feature (i.e. geometric feature part) can keep its discriminative power well across different datasets, i.e. when learning in one dataset and testing on another one.

The work in this chapter prepares us to take one more step in capturing more complex geometric structures among the local features (e.g. triplet descriptors), which are described in the next chapter.

Chapter 7

Human activity recognition through graph matching techniques

Contents

In the previous chapter, we have demonstrated that employing spatio-temporal relationships between local features significantly improves recognition performance. In this chapter, we try to exploit relationships of higher order structure between local points. In this chapter we tackle the problem of detecting individual and multiple human actions in video sequences. A major improvement with regard to our previous work is that the proposed method can localize multiple activities in continous video sequences. While the most successful methods are based on local features, which proved that they can deal

with changes in background, scale and illumination, most existing methods, as well as our previous method, have a major shortcoming: these methods mainly focus on direct classification techniques to classify the human activities, as opposed to detection and localization. We propose a new approach, which is based on a graph matching algorithm for activity recognition. In contrast to most previous methods which classify entire video sequences, we design a video matching method from two sets of spatio-temporal interest points (STIPs) for human activity recognition. First, STIPs are extracted, and hyper graphs are constructed from them, i.e. graphs with edges involving more than 2 nodes (3 in our case). The activity recognition problem is then transformed into a problem of finding instances of model graphs in the scene graph. By matching local features instead of classifying entire sequences, our method is able to detect multiple different activities which occur simultaneously in a video sequence. Experiments on two standard datasets demonstrate that our method is comparable to the existing techniques on classification, and that it can, additionally, detect and localize activities.

The work described in this chapter was published in the 7th International Conference on Advanced Video and Signal-Based Surveillance 2010 [APCGA10], and has been selected as the best paper for the track “recognition”.

7.1 Introduction

As mentioned in the previous chapters (cf. chapter 5, and chapter 6), bag of words models discard the spatio-temporal layout of the local features which may be almost as important as the features themselves. To overcome this, efforts have been made to exploit information from the spatial and temporal distribution of interest points [LS08, ZHCC08]. These extensions, however, still suffer from some of the inherent problems involved in classification: they do not allow to localize activities, and they require selecting the optimal number of codewords for codebook formation as well as fine-tuning of parameters.

In this chapter, we overcome some of the above shortcomings of the earlier works by introducing a new approach for both detection and localization of multiple activities in video sequences. In particular, our work is motivated from two aspects:

Multiple activity recognition – The problem of multiple activity recognition has many applications such as video retrieval, video indexing and video mining, but this problem is still incompletely understood. We are therefore investigating this problem.

Continuous recognition – Most existing methods assume that video sequences are well segmented in temporal direction, i.e. each video sequence contains only one activity. This requires a video segmentation preprocessing. However, the problem of temporal segmentation of activities still remains open or is only partially solved yet. In real-time applications such as video surveillance, this problem should be part of recognition systems, i.e. does not require a separate video segmentation task.

To this end, we consider the following choices:

Local features – We use local features because they are robust to noise, changes in background, and occlusion. Moreover, by using local features, we can avoid preprocessing steps such as tracking.

Spatio-temporal information – In our previous work (cf. chapter 6), we have proved that using spatio-temporal relationships significantly improves the BoW models. In this chapter, we exploit higher order relationshisps between local features, i.e. third order. A major improvement with regard to our previous work, as well as most other works, is that we no longer require training a codebook, which usually suffers from selecting the optimal number of codewords.

To deal with these points, we formulate the activity recognition problem as a graph matching one, i.e. the problem translates into a matching task: a model graph represents a template video, and a scene graph represents a test video. Hence, our approach is related to the work on matching videos (cf. chapter 5), as well as the work on graph matching (cf. chapter 2). In the following, we briefly summarize these related works subsequently. For each field, we highlight the differences between our approach and those of related works.

Video matching — First of all, it should be noted that our work shares some similarities with previous research on video correlation, e.g [KSH07, SI07b, RA09]. Shechtman and Irani [SI07b] define a motion consistency measure to match space-time volumes directly. However, the distance between pair of videos is computed by exhaustively comparing patches extracted from every space-time point. Ke et al. [KSH07] combine a part-based shape and flow matching framework from [SI07b] for event detection in crowded videos. Recently, Ryoo and Aggarwal [RA09] have presented a histogram-based match kernel for video matching. Among the methods mentioned above, our approach is most closely related to the work of Ryoo and Aggarwal [RA09], who perform video matching from two sets of STIPs. Our method differs from their work in two main points. First, the authors in [RA09] define a set of logical predicates for taking into account the pairwise relationships among points. Instead, we *naturally* take into account higher-order relationships between points through a graph-matching technique. It should be noted that such logical predicates are difficult to extend to higher relationships. Second, their method needs to train a codebook from the training sets, while our method does not require any learning.

Graph matching — In this chapter, we advocate for the advantages of graph matching methods and their capability of exploiting relationships between local primitives. The main contribution of our method is the introduction of a graph-based matching method for detecting and localizing multiple actions in video sequences. Graph matching techniques have been studied intensively in the field of pattern recognition [CFSV04, DBKP09, LH05b, TKR08a, ZS08] - see chapter 2, but no method has yet been given for recognizing human activities — a straightforward application of these techniques to video recognition is difficult. As mentioned in chapter 2, it is widely known that computing the exact solution to inexact graph matching problem is NP-complete [TKR08a], as is sub graph isomorphism [ZDS09]. Approximate solutions have been proposed for various applications. For instance, let N_1, N_2 be the number of vertices (nodes) in graphs G_1 and G_2, respectively. Optimally assigning each node from G_1 to one of the nodes in G_2 is of complexity $O(N_1 \cdot N_2)$ if only the unary measurements (e.g., SIFT descriptors)

associated with each node are used, i.e. for each node in G_1 we assign the node G_2 having minimum feature distance. However, this is highly suboptimal. If neighborhood relationships are taken into account, i.e., coherence of distances and/or angles associated to the edges in the graphs, the complex interactions between assignment variables make the complexity exponential: there are $N_1^{N_2}$ possible assignments over the whole set of nodes in G_1, where each assignment takes $O(N_1^2)$ to check. Although fast approximative algorithms do exist, e.g. with graph cuts [TKR08a], the problem remains very difficult.

As pointed out in [ZS08], the vertex correspondance problem can be transformed into an edge correspondence problem. In this approach, each edge in graph G_1 is assigned an edge in graph G_2 according to a minimal distance which involves, both, the feature distances of the 4 involved nodes as well as edge compatibility, e.g. a comparison of the lengths of the model edge and the assigned edge. In [LH05b], the resulting optimization problem is solved approximately with a spectral method, which relaxes the discrete assignemnt variables into continuous ones and then solves it numerically by removing some constraints during the optimization procedure itself. This principle can be naturally extended to higher order interactions. Zass and Shashua [ZS08] present a hyper-graph[1] matching method, which is of complexity $O(|N_1|.|N_2|.z^{(2d-1)})$, where z is the closest hyper-edges per vertex and d is the order of hyper-edges considered (d=2 for pairs). This method is still very time consuming, because in a real application z could not be far from $min(|N_1|, |N_2|)$. Very recently, Duchenne et al. [DBKP09] generalized the spectral matching method from [LH05b] by using a tensor-based algorithm for high order graph matching, which is of complexity $O(n^3 + n^d \log(n))$ where $n = max(|N_1|, |N_2|)$. This is still too high to be applied for practical systems, in particular for the applications in videos such as video surveillance. A modified version of this triangle based algorithm (i.e., d=3) is the basis of our work on activity recognition in video sequences. More precisely, our work on graph matching differs from the work of Duchenne et al. [DBKP09], as well as other graph matching methods in several aspects:

In terms of solving the problem – Recall that the graph matching problem is described by either eq. 2.6 or eq. 2.10 (as introduced in chapter 2). In the case of the edge correspondence problem (e.g. from [LH05b][ZS08][DBKP09], as well as our work in this chapter), the graph matching problem is usually formulated in its matrix

[1] A hyper-graph is a generalization of a graph, where an edge can connect any number of vertices, and hyper-edge is an arbitrary number of nodes [ZS08].

form described in eq. 2.10. For convenience, let us repeat this objective function:

$$\widehat{x} = \arg\max_x (x^T A x), \quad \widehat{x} \in \{0,1\}^{N_1 N_2} \tag{7.1}$$

where x is a row-wise representation vector of an assignment matrix X (cf. chapter 2).

While most of state-of-the-art graph matching methods concentrate on finding efficient algorithms for solving the correspondence problem (i.e., solving directly eq. 7.1 - which is known as NP-hard), we focus our attention to the complementary problems: how to construct more expressive graphs and how to efficiently configure the compatibility matrix A, i.e., the matrix describing edge compatibilities of the assignments. Indeed, a reliable compatibility matrix significantly speeds up the solving algorithm.

In terms of checking the edge compatibilities – To deal with complexity, one of the possible solutions is to reduce the edges in the graphs. Zass and Shashua [ZS08] connect each vertex to a number of its closest vertices for creating hyper-edges. In practice, to achieve reliable results, this number of closest vertices should be not so far from the graph size. Duchenne et al. [DBKP09] encode empirically an approximate structure of the second graph but exploit the fully connected interactions in the firs graph. In contrast, we create hyper-edges in a significant way, i.e. we keep only hyper-edges which might characterize action behavior (see section 7.5).

In terms of evaluating the final result – Once the matching process is done, a question arises: is it a good result? While most other methods rely on the matching score to find the answer, i.e. the value of the objective funtion 7.1, we interpret the final assignment to make decisions. This will be detailed later.

Before presenting our method, let us summarize the advantages, as well as the contributions of the proposed approach in the following:

Advantages – Exploiting the full potential of a graph based representation, our method offers several important advantages over other activity detection methods for videos:

- The proposed method can not only classify but also detect and localize activities, which occur simultaneously in the same video sequence[2].

[2]Several methods can be adapted for detecting multiple actions by using sliding windows in both space and time dimensions, e.g. from [NWFF08, SDPNL08]. However, measures and statistics designed for entire video sequences do not always successfully scale down to smaller windows.

- It does not require any parameter tuning, training, foreground/background segmentation, or any motion estimation or tracking.
- By verifying the spatio-temporal constraints in a significant way, our method needs only a small number of features points (i.e., the *important* ones) extracted from two given videos to perform matching. In constrast, the conventional BoW methods need to collect dense points from the videos to perform classification.

Contributions – Besides these advantages, our method features several contributions compared to the original graph matching method introduced by Duchenne et al. [DBKP09] for object detection:

- By significantly reducing the number of hyper-edges in the graph, our method is of much lower complexity compared to the original one [DBKP09] (c.f section 7.5).
- We incorporate both triangle geometries and their orientations to find potential corresponding triangles (c.f section 7.6), which speeds up the convergence by eliminating incompatible triangles.
- We benefit from the features calculated at each STIP to initialize the algorithm (c.f section 7.7), which reduces the number of false alarms and speeds up convergence.
- Most of methods on graph matching consider the score returned by the objective function as a detection criteria. However, this is not optimal, as one cannot distinguish the case in which several vertices in the scene graph are matched to the same vertex in the model graph. We propose to interpret the projection of the set of vertices of the first graph onto the second one to compute a second score, called the detected score, which, along with the matching score, is used for detection (c.f section 7.8).

The rest of this chapter is organized as follows. After briefly summing up the hypergraph matching algorithm in section 7.2, we introduce our adaptation and extension of this algorithm to video matching in section 7.3. In section 7.9, we discuss the computational complexity of the proposed method. The experimental results are presented in section 7.10. Finally we conclude and give some perspectives of this work.

7.2 Hyper-graph matching formulation

In this section, we briefly summarize the hyper-graph matching method introduced in [CSS07b] and [LH05b] and refined in [DBKP09], which has been presented in chapter 2. Let $G^m = (V^m, E^m, F^m)$ and $G^s = (V^s, E^s, F^s)$ be two hyper-graphs (the model and the scene graph, respectively) where hyper-edges correspond to a d-tuple of vertices. In our case, where d=3, E represents a set of triangles (our d-tuples), V a set of vertices, and F the set of their associated unary measurements (i.e an appearance descriptor). In the following we denote the number of nodes in both graphs as $N^m = |V^m|$ and $N^s = |V^s|$, respectively. A matching between G^m and G^s is equivalent to looking for an $N^m \times N^s$ assignment matrix X such that X_{ij} is set to 1 when the model node v_i^m is matched to the scene node v_j^s, and to 0 otherwise. Thus, the search space is the set X of assignment matrices:

$$C = \{X_{ij} \in \{0,1\} : \sum_i X_{ij} = 1\} \tag{7.2}$$

Note that we constrain each model node v_i^m to be matched to exactly one scene node v_j^s, but a scene node v_j^s may be matched to several model nodes. For pairwise matching, the graph matching problem is formulated as the maximization of the following objective function [CSS07b][LH05b]:

$$score(X) = \sum_{i,j;i',j'} \psi_2(i,j;i',j') X_{i,i'} X_{j,j'} \tag{7.3}$$

where, as described in chapter 2, ψ_2 is a pairwise potential function that measures the compatibility between a pair of model nodes (i,j) and its corresponding scene nodes (i',j'). This objective function can be rewritten as the one described in eq. 7.1, where ψ_2 is now reshaped into matrix A:

$$A(i(N^s-1)+i', j(N^s-1)+j') = \psi_2(i,j;i',j')$$

and x is the $N^m \times N^s$ row-wise vectorized replica of the X.

For triplet matching, as described in [DBKP09], the matching problem is similarly formulated as:

$$score(X) = \sum_{i,j,k,i',j',k'} \psi_3(i,j,k;,i',j',k') X_{i,i'} X_{j,j'} X_{k,k'} \tag{7.4}$$

where $\psi_3(\cdot;\cdot)$ is an energy potential estimating the compatibility of pairs of triplets $(i,j,k) \in V^m$ and $(i',j',k') \in V^s$. High values of ψ_3 correspond to similar triplet pairs.

Here, the product $X_{i,i'}X_{j,j'}X_{k,k'}$ will be equal to 1 if (i,j,k) are all matched to (i',j',k') (the original formulation of this problem in [CSS07b] and [LH05b] involved pairs).

As we stated in chapter 2, the inexact graph matching problem can be described by either eq. 2.6 or eq. 2.10 (i.e. eq. 7.1). Indeed, eq. 7.4, which describes the scoring function for triplet matching, can also be rewritten as eq. 7.1, in which the potential function ψ_3 is reshaped into a $N^m N^s \times N^m N^s$ matrix A. In this chapter, for easy readability, and for the sake of convenience with respect to existing works on hypergraph matching (e.g. [CSS07b][DBKP09]), we prefer to use the form of eq. 7.4.

In [CSS07b, LH05b], the scoring function is maximized through a continuous optimization procedure which requires relaxing the values of X such that each element takes continuous values in the interval $[0,1]$ subject to the constraint that the norm of the column vectors of X is 1. Exploiting this constraint, and further requiring the elements of ψ_3 to be non-negative, the maximum value of (7.4) can be calculated as the largest eigenvalue of A.

The basis of our method is the refined method proposed by Duchenne et al. [DBKP09], which improves upon [CSS07b] and [LH05b] in two ways:

- Whereas the interactions in [CSS07b, LH05b] are pairs, [DBKP09] extends the order to triplets (triangles), which may boost the discriminative power of the method.
- The new organization into triplets in [DBKP09] allows to change the L_2 norm of the relaxation to the L_1 norm, which makes the de-relaxation of the continuous values into discrete assignment decisions more robust. In addition, this new form allows to compute the largest eigenvalue using the power-iteration algorithm (cf. algorithm 2.2 in chapter 2.)

We will not detail this algorithm any further since, as already mentioned, solving directly the correspondence problem is beyond the objective of this work.

7.3 Overview of the proposed method

The main objective of our method is to measure the similarity of two videos through a graph-based matching technique. The proposed method consists of the following steps:

Extraction of spatio-temporal interest points – Since our graphs are contructed from spatio-temporal interest points (STIPs), we first need to extract STIPs. As presented in chapter 5, there exist several interest point detectors. In this work, we are not interested in comparing the performance of different interest point detectors, we chose the IP-detector from Dollar et al. [DRCB05].

Graph construction – After having extracted local points from the video sequences, proximity graphs are constructed from them. We benefit from both location and temporal information of STIPs to create the graphs. The activity recognition problem is now formulated as (sub) graph matching between a model graph and a (potentially larger) scene graph (see section 7.5).

An objective function for video matching – We define a new objective function, which takes into account both the similarity between triangles and their orientations. The graph matching problem is now equivalent to maximizing this objective function (see section 7.6).

Matching initiation – We initialize the iterative matching process using appearance STIP descriptors associated with each node of the graphs (see section 7.7). The local appearance descriptors are extracted around the STIPs.

Action detection – In this final step, we propose a new way to verify the matching result in order to decide whether the scene graph contains an instance of the model graph, i.e., the two videos contain the same human activity (see section 7.8).

We evaluate our method through experiments based on two standard datasets for action classification and on our own dataset for multiple activity recognition.

7.4 Extraction of spatio-temporal interest points

We take advantage of the Dollar detector [DRCB05] to detect STIPs (cf. chapter 5). This detector applies two separate linear filters to spatial and temporal dimensions. Interest points are detected at local maxima of the response function. Compared to other detectors (chapter 5), the Dollar detector produces dense features, yielding the complexity of the graph matching problem. By exploiting the spatio-temporal relationships through graph representation, it is not necessary to use all dense detected points, which usually contain noise. In practice, we keep the important points, according to the response function. In order to achieve reliable results, we suggest to keep at least 60 ST-interest points for one second video, for the model videos (note that each model video contains only one activity).

We also extract the cuboid descriptor from [DRCB05] for each STIP. This local cuboid descriptor, which calculates a histogram of gradients at the position of each STIP (see chapter 5), will be employed to initialize the matching algorithm (see section 7.7).

7.5 Constructing more expressive graphs

The complexity of the method from [DBKP09] highly depends on the number of the d-tuples (triangles) in the two graphs constructed from the videos. The original algorithm constructed fully connected graphs, i.e. graphs with close to N^{m3} and N^{s3} hyper-edges, respectively. We present hereafter a graph construction method producing graphs with far fewer but more expressive hyper-edges, which significantly reduces complexity and also increases robustness to non-rigid transformations. Given two sets of STIPs, we construct two corresponding graphs for the model video and the scene video, i.e. we construct the two sets E^m and E^s of hyper-edges (triangles). Without loss of generality, we present the construction of the model graph G^m, the scene graph G^s is constructed in a similar way. We take into account the following information for graph construction from ST-interest points:

Temporal aspects – One of the most important properties of a video is the nature (and importance) of the temporal order, which is very often dominated by causal relationships. We exploit this in two ways:

- We put a constraint on the preservation of the correct temporal order in a triangle (see section 7.6)
- We restrict the number of hyper-edges, i.e., we keep only one hyper-edge per triplet. This first filter allows us to sample the number of triplets from n^3 to $|C_n^3|$, where n is the graph size.

Spatio-temporal proximity – Fully connected graphs contain triangles linking all possible triplets of points, even between very distant STIPs (in time and/or space). While, generally speaking, a higher number of triangles tends to increase the discriminative power of the matching method, it also tends to decrease robustness. This is especially true for triangles between very distant points, as the space-time geometrical transformation between two instances of the same human activity is not necessarily rigid. Moreover, intuitively, if three local STIPs are adjacent in both space and time, they should belong to the same humain activity. From this intuition, and the above reasoning, we propose to filter the set of triangles by a simple thresholding rule keeping only triangles of which three STIPs are close in space and time using two different thresholds, one for the spatial dimensions and one for the temporal dimension (see figure 7.1 for an illustration). The points in a triangle are then ordered according to the temporal order. If there are more than one point in the same frame, they are ordered according to their spatial coordi-

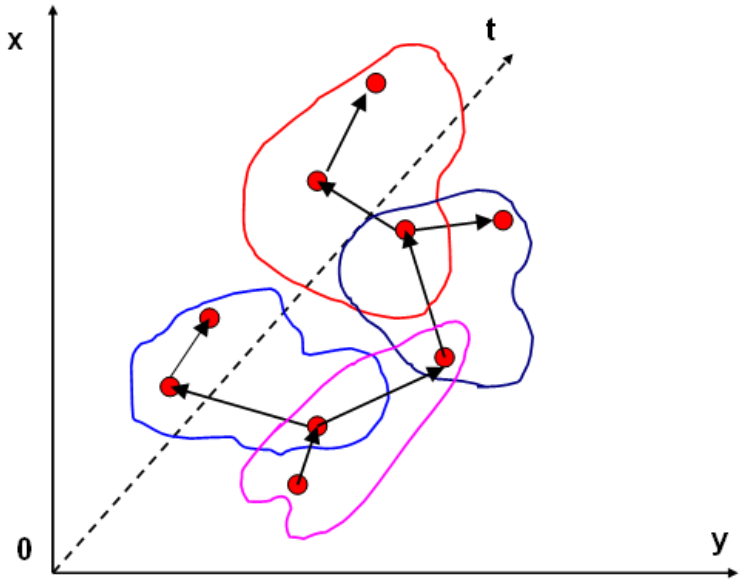


Figure 7.1: Illustration of a partial view of our graph: circles are spatio-temporal interest points; three close points are grouped to form a triangle; arrows indicate the temporal order of the points in a triangle.

nates. We would like to note that the reason for separately considering spatial adjacency and temporal adjacency of spatio-temporal interest points is due to two things: i) a 3D coordinate is significantly different from a spatio-temporal coordinate (i.e 2D + t); ii) using a separate temporal threshold is more flexible, which helps to work on unsegmented sequences. In practice, we fix the thresholds such that $|E^m| \leq 8 \times |V^m|$.

The scene graph G^s is constructed in a similar way, but contains more triplets (i.e., using bigger thresholds) to deal with noise and scale changes. This gives the sub graph matching algorithm a larger set of scene triangles to choose from for each model triangle. In practice, we choose the thresholds such that $|E^s| \leq 50 \times |V^s|$.

7.6 An objective function for video matching

In order to boost the discriminative power of the matching algorithm, we propose the following compatibility matrix ψ_3 for the objective function in equation (7.4) designed for activity recognition in videos:

$$\psi_3(i,j,k;i',j',k') = \psi_3^t(i,j,k;i',j',k') \times \phi((i,j,k),(i',j',k')) \tag{7.5}$$

where $\phi(.,.)$ and ψ_3^t govern temporal and space-time geometrical aspects of the transformation:

Temporal aspects — As mentioned in section 7.5, we think that it is crucial to exploit the important nature of the temporal dimension in video sequences. We therefore introduce the functional $\phi(.,.)$ which verifies if two triangles are equally oriented taking into account the (temporal) order of their points:

$$\phi((i,j,k),(i',j',k')) = \begin{cases} 1 & \text{if } D_{ijk} = D_{i'j'k'} \\ 0 & \text{otherwise} \end{cases} \tag{7.6}$$

where D_{ijk} is the sign of the determinant of the triplet computed from their spatio-temporal coordinates:

$$D_{ijk} = \text{sign}\left(\begin{vmatrix} i_x & i_y & i_t \\ j_x & j_y & j_t \\ k_x & k_y & k_t \end{vmatrix} \right) \tag{7.7}$$

Note that the row order in the determinant must be the same as the point order in the triplet. We can also interpret this measure as follows: if two triangles have the same direction when they are projected onto the temporal axis, it is not excluded that characterize the same activity.

ST-geometrical distance — The factor ψ_3^t in (7.5) describes the geometric similarity between two triangles and is given as follows:

$$\psi_3^t(i,j,k;i',j',k') = \exp\left\{ -\frac{||\boldsymbol{\alpha}(i,j,k) - \boldsymbol{\alpha}(i',j',k')||}{\sigma_g} \right\} \tag{7.8}$$

where $||\cdot||$ is the L_2 norm and $\boldsymbol{\alpha}(\cdot,\cdot,\cdot)$ is a vector of the cosines of the first and second angles in the triangle (recall that the points of a triangle are ordered), as the third one is linearly dependent. The parameter σ_g governs the intra-class variations of the angles, we set it to half the mean over the distances between all triangles.

To further decrease complexity, and to increase discriminative power, triangles in the model graph are only allowed to match triangles in the scene graph if their geometrical shapes (given above) are close enough. More precisely, only the first ranked distances as given by (7.8) are kept non-zero in ψ_3^t, where the ranking can be computed efficiently using a k-d tree. In practise we keep the first k (350 in our experiments) scene triangles for each model triangle. Note that a small value of k may give bad results.

7.7 Matching initialization

Iterative algorithms often suffer from bad initialization, therefore a reliadble initialization is necessary. In this section, we take the advantage of local descriptors to initialize the matching, i.e. to initialize the set of *potential* assignment matrices X (see section 7.2).

Recall that the proposed matching compatibility function $\psi^t - 3$ does not use any features associated with the STIPs — we argue that the matching algorithm itself is more robustly controlled through spatio-temporal geometry. This is consistent with the experimental results we obtained in chapter 6. However, we propose to benefit from the power of local descriptors associed to STIPs to better initialize the matching algorithm. Denoting by f_i^m and f_j^s the features calculated on point i of the model graph and point j of the scene graph, respectively, we initialize the relaxed assignment matrix X as follows:

$$X_{ij} = \exp\left\{-\frac{||f_i^m - f_j^s||}{\sigma_f}\right\} \tag{7.9}$$

where the parameter σ_f captures the variation of distances in feature space. For each model node, only the values for Nb of its nearest neighbors in feature space (efficiently identified through a k-d tree) are kept non-zero in the matrix X. In practice, we set Nb to $0.35|N^s|$. Additionally, the required norm constraint described in section 7.2 is then enforced through normalization.

In our experiments, we use the L_1 norm, which enables us to easily project the model nodes onto the final solution and evaluate the obtained results (see section 7.8). It should be noted that in spetral methods, the normalization for the compatibility matrix is needed to: i) ensure that the algorithm converges; ii) guarantee the existence of non-negative eigenvalues; and iii) verify some constraints, e.g. one-to-one or one-to-many matching.

7.8 Recognition score

In order to decide whether an instance of the graph (a video in our case) has been detected or not, most energy based graph matching methods consider the final matching score, which is denoted by $score(X^*)$ and given as the maximum of equation (7.4), calculated as a sum of all matching scores between triangles after matching. This, however, is not an optimal choice. For instance due to noise, several triangles in G^s could be matched to the same triangle in G^m, and we cannot remove such irrelevant matches by filtering based on only the matching score. Moreover, these matching scores depend on

the geometric features of triangles used to verify compatibility between triangles, and therefore make the choice of a threshold difficult.

We propose a different score, called $score_d(X^*)$, which removes the uncertainties from the matrix X. More precisely, we de-relax the values of the optimal matrix X^* (obtained as the maximum of (7.4)) by taking the maximum for each node in the model graph and setting the others to zero:

$$Z_{ij} = \begin{cases} X_{ij} & \text{if } X_{ij} = \max_k X_{ik} \\ 0 & \text{otherwise} \end{cases} \tag{7.10}$$

Then, $score_d$ is defined by the mean projection:

$$score_d(X) = \frac{1}{N^m} \sum_{ij} Z(i,j) \tag{7.11}$$

$score_d \in [0..1]$ and it is equal to 1 for the ideal case where every node in G^m is well matched. While the matching $score(X)$ indicates how similar two graphs are, $score_d(X)$ measures the percentage of correctly matched points. We combine them through a global score as follows:

$$score_{Global}(X) = \begin{cases} score_d(X) & \text{if } score(X) \geq \tau_0 \\ 0 & \text{otherwise} \end{cases} \tag{7.12}$$

where τ_0 is a preliminary threshold obtained through experiments.

7.9 Computational complexity and running time

Our system has been implemented entirely in Matlab and for the moment processes videos as entire sequences. However, the algorithm itself can be implemented to deal with video streams by processing small overlapping consecutive blocks.

Our matlab implementation does not run in realtime, as 1 second of video for the moment requires 46.7 seconds of processing on a single core processor with 2GHz and 2GB of RAM, including feature extraction, scene graph construction, and matching against 98 model graphs. However, a reimplementation in C++ should provide real-time or near-realtime performance on a recent machine. We would also like to point out that the algorithm is inherently parallel since matching with different model graphs can be done in parallel. Furthermore, the matrix computations of the power-iteration method should run very efficiently on a GPU.

7.10 Experimental results

We evaluated the performance of our proposed algorithm with regards to two different tasks:

- The classification of entire video sequences according to the activity of a single activity performed by a single person in the video. Since standard databases are available for this kind of task, we are able to give quantitative performance figures (classification accuracy).

- Detection and localization of multiple activities performed by multiple people at different locations in the same video. Up to our knowledge, no standard database is available for this more difficult problem. We illustrate the performance of our method qualitatively on our dataset.

7.10.1 Classification of entire sequences

Datasets — Our experiments are carried out on the standard KTH and Weizmann human action datasets (cf. chapter 5). The KTH dataset was provided by Schuldt et al. [SLC04] in 2004 and is the largest public human activity video dataset. It contains a total of 2391 sequences, comprising 6 types of actions (boxing, hand clapping, hand waving, jogging, running and walking) performed by 25 subjects in 4 different scenarios including indoor, outdoor, changes in clothing and variations in scale. Each video clip contains one subject performing a single action. The Weizmann dataset was first used in by Blank et al. [BGS*05a] in 2005, which consists of 90 video clips of 10 actions (walking, running, jumping, gallop sideways, bending, one-hand-waving, two-handswaving, jumping in place, jumping jack, and skipping) performed by 9 different subjects. Again, each video clip contains one subject performing a single action.

Testing protocol — We extracted ST-interest points and their cuboid descriptors from video sequences (see section 7.4). Using leave-one-out cross-validation, we apply our method for the classification of activities and report the average accuracies, even though the focus of our method is to detect/localize the activities. To this end, we employ a group of videos from a single subject in the dataset as the testing videos (i.e., scene graphs), and the remaining videos as the model videos (i.e., model graphs). This was repeated so that each group of videos in the dataset is used once as the testing videos. For each loop, we match each test video against all model videos, and take the label (i.e., the activity) of the model video which gives maximum score from equation (7.12).

Table 7.1: Comparison of our method with different methods, tested on KTH and Weizmann datasets.

Method	KTH	Weizmann	
Our method	**91.2**	**100.0**	
Dollar et al. [DRCB05]	81.2	-	Same features
Niebles et al. [NWFF08]	83.3	90.0	as our work
Savarese et al. [SDPNL08]	86.8	-	
Oikonomopoulos et al. [OPP09]	80.5	-	
Scovanner et al. [SAS07]	-	82.6	
Gilbert et al. [GIB08]	89.9	-	BoW + Spatio-temporal
Zhang et al. [ZHCC08]	91.3	-	relationships
Wong et al. [WKC07]	91.6	-	
Ryoo and Aggarwal [RA09]	93.8	-	
Liu and Shah [LS08]	94.2	-	
Schindler and Gool [SvG08]	92.7	100.0	
Fathi and Mori [FM08]	90.5	100.0	
Gorelick et al. [GBS*07]	-	99.6	Other methods
Sun et al. [SCH09]	94.0	97.8	
Kim et al. [KC09]	95.3	-	
Ballan et al. [BBDB*09]	92.1	92.4	
Laptev et al. [LMSR08]	91.8	-	
Klaser et al. [KMS08]	91.4	84.3	
Willems et al. [WTG08]	84.3	-	

Classification accuracy — Table 7.1 presents a comparison of our results with state-of-the-art results. Although we adopted graph matching techniques for action classification, we cannot directly compare our method with other graph matching methods applied in object recognition because of the nature difference between these two domains. Therefore, we compared the performance of the proposed method with other methods tested on the two standard datasets KTH and Weizmann.

In order to better compare the performance between different approaches, we divide this table into three groups: the first group includes the methods, which have not taken into account spatio-temporal relations; the second one consists of approaches that employ spatio-temporal relations in different ways; and the last groups presents some latest results tested on these two datasets. Our method outperforms the existing methods based on a bag of words model (first group), i.e. using only local features. The results shown in the second group demonstrate that our method is comparable to those exploiting spatio-temporal information. Note that the results which are slightly higher than ours, i.e. the work from Ryoo and Aggarwal [RA09], Liu and Shah [LS08], require training a codebook and tuning parameters. Moreover, the work from Ryoo and Aggarwal [RA09] needs to encode much more logical relationships between local features. We would also like to note that the comparision between our method with these two methods is only relative, because they have not tested on the Weizmann dataset. It is

the fact that some methods may work well for one dataset and not so well for another. Finally, the comparison between our method and those in the last group is not direct, since they either combine both global and local features [SCH09], taken more data from segmentation mask [GBS*07] [FM08], or use a different experimental set-up.

In conclusion, the results obtained for action classification indicate that our method is comparable with previous methods. In addition, our method can detect activities in continuous videos. In the next section, we will demonstrate the ability of our method to recognize multiple activities in the same video sequence.

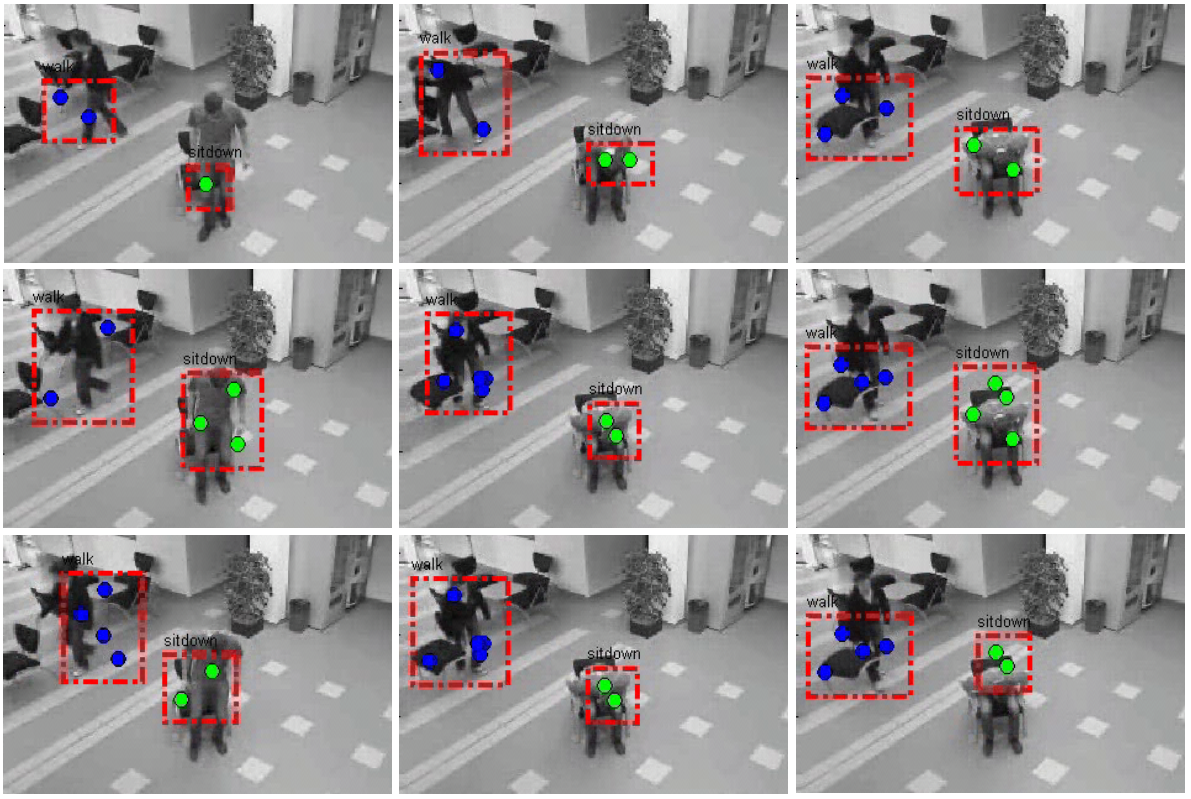
7.10.2 Detection and localization of multiple and individual actions

The main contribution of our method is the possibility of detecting and localizing multiple activities from unsegmented video sequences. Unfortunately, to the best of our knowledge, there is no standard dataset for evaluating such applications. This is due to the fact that this problem is still incompletely understood.[3] To evaluate our system, we have performed a third experiment on our own dataset, which contains 120 videos in 4 classes (run, walk, vertical jump, sitdown) (cf. chapter 1). Our dataset is different from those of KTH and Weizmann in two points: i) different activities are performed in different directions with respect to the camera; and ii) we included videos as short as between 3 and 10 frames.

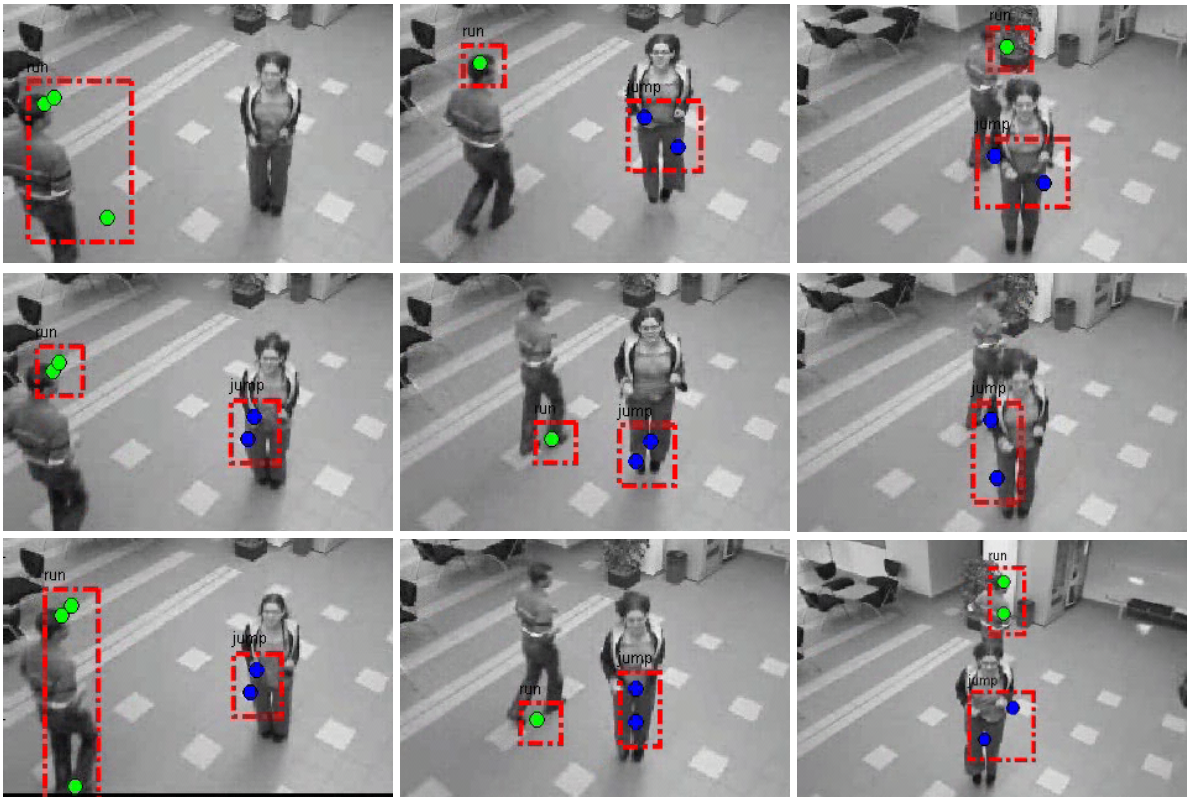
For each pair of videos, we first perform matching as described in section 7.10.1 and we localize the action by projecting the points in the model graph onto the scene graph based on the obtained solution. Finally, the detected action is localized around the detected points.

Figure 7.2 shows some visual results for our dataset. The detected actions are delineated with red bounding boxes at the frame level. Circles are the detected interest points, from which we localize activities. Figure 7.2.a shows visual results for simultaneous walking and sitdown actions. From this figure, we can see that our method can detect any kind of activities, i.e. not limited to the periodic activities (e.g. walking). Note that in our dataset, several actions are of very short duration (e.g. sitdown), the methods that are based on bag of word models often fail for such actions. In figure 7.2.b, we demonstrate other visual results for multiple activity recognition, i.e. simultaneous recognition of running and jumping actions. It can be seen that, even when the walking person is partial occluded by the jumping person, our method can still distinguish the two actions. It should be noted that, in order to obtain our results, most existing meth-

[3]Several teams have proposed algorithms and tested them on their own datasets which are either not publicly available, or only used to qualitatively evaluate, e.g. [SI07b, RA09].



a) recognition results for videos containing simultaneous walking and sitdown



b) recognition results for videos containing simultaneous running and jumping

Figure 7.2: Recognition results on several consecutive frames of two videos of our dataset (top to bottom and left to right). This figure should be best viewed in color.

ods need a tracking system that allows to determine where the actions are happening, and the classification techniques are then applied to the tracked objects.

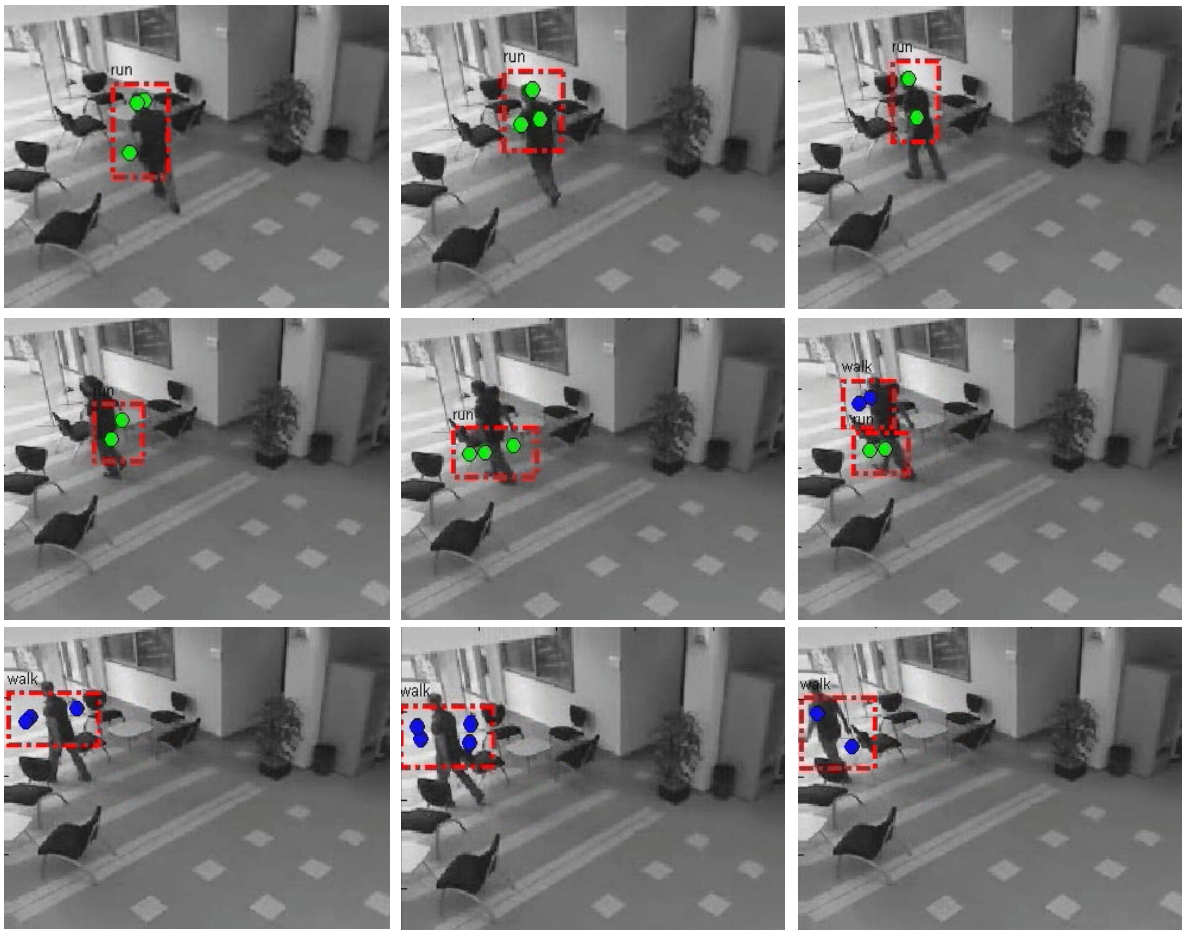


Figure 7.3: Recognition results for continuous videos: a person performing two consecutive actions: first running, then walking. Note that the temporal order is shown from left to right, then top to bottom. This figure should be best viewed in color.

Besides the above evaluations, we have also tested our method on continuous videos. Figure 7.3 demonstrates some visual results for recognizing two consecutive actions: running, then walking. This test has to deal with at least two difficulties: i) inter-class problem, i.e. slow running becomes walking; ii) actions performing by the same person, i.e. appearance features exacted from these actions may be similar. From this figure, we can see that our method gives promising results for the problem of activity recognition in continuous videos. Again, in order to achieve our results, most classification-based methods require a tracking or a temporal segmentation task.

We also tested our objective function (section 7.6) without taking into account the temporal aspects (function $\phi(.,.)$), which significantly decreased the perfomance.

7.11 Conclusion

In this chapter, we propose a first attempt to address the problem of action recognition and localization by using graph matching techniques. Our method features several advantages for activity recognition such as: it can detect/localize multiple activities in the same video sequence without any preprocessing; our method avoids the non-trivial problem of selecting the optimal number of codewords for codebook construction; unlike many other methods it does not need training, background/foreground segmentation, tracking, and does not require any prior knowledge on the action. Through experiments, we have shown that it is feasible to apply graph matching techniques to action recognition.

We also tested our method for the problem of multiple activity recognition on our own dataset. The visual results demonstrate that many complex activities are recognized and detected simultaneously. It should be noted that, in our dataset, there are several actions, which are of very short duration, e.g. sit-down action. The conventional classification-based methods may fail to detect such actions. To our opinion, a potential application of our approach is related to video surveillance, where scene videos are *online* represented by blocks of short duration, which will be matched against a set of labeled video models (often of shorter duration).

Besides the advantages mentioned above, our method still has limitations, e.g. detecting multiple instances of the same activity in a video. In this case, we need to detect the first instance, eliminate it, and restart a new process for other instances. These limitations will be considered in our ongoing work. Another drawback of our method is the need to choose the two thresholds (one for space and another for time) for graph construction. It should be noted that the final results may suffer from the thresholds used. In practice, the temporal threshold should be less than one action cycle.

As a future work, we aim at extending this method to capture higher-order geometric structures among the local features, e.g. relationships between triangles.

Chapter 8

General conclusion and discussion

In this final chapter, we first conclude our work, and review our contributions. Then, we discuss the limitations of our methods, and some perspectives for future work.

8.1 Summary of our contributions

In this thesis, we have presented our research study on object recognition and activity recognition. Our objective was to adopt graph matching techniques for the recognition problem, but fundamentally, our aim was to take into account spatial and spatio-temporal information to improve recognition performance. In activity recognition, we focused on both individual activity classification and multiple action recognition (i.e. including localization). In the context of object recognition, we were interested in recognizing objects in both natural and storyboad scenes from hand-drawn models. In the following, we would like to recall the reason we chose graph matching techniques for both object recognition and activity recognition problems.

Althrough these two problems are quite different in nature, they share several common points that need to be considered to find efficient solutions:

Local features – Global features usually cannot deal with occlusion, thus local features must be used.

Spatial and/or temporal relations – Local features themselves are not discriminative enough to recognize complex objects/actions. Hence, spatial and/or temporal relationships between the local features are needed.

Structured data – In general, there are complex transformations (not simply rigid, or affine)between the model and the scene for both drawing images and human ac-

tivities. Thus, we need a powerful way to deal with such complex non rigid transformation.

We decided to adopt graph matching techniques, which can handle all these problems in a robust manner. First, local features are extracted from images or videos, then the graphs (model and scene) are constructed from them, where nodes represent local features, and edges represent the relationships between them. The recognition problem is then translated into a graph matching problem. Several solutions have been proposed to solve the graph matching problem. We have evaluated our methods on several standard datasets. The experimental results have demonstrated the effectiveness of the proposed methods. Let us briefly summarize the major contributions of this thesis:

Object recognition from hand-drawn models – We introduced a new local patch-based representation using Zernike moments. By employing Zernike moments in a local manner, the object recognition task was formulated as a graph matching problem, for which the solution is found by minimizing an energy function. We have proposed to integrate the following constraints into our objective function: (a) Zernike descriptors, (b) consistency of the neighborhood relationships between patches and (c) consistency of the rotation angle among the neighboring patches. We have proposed a decoupled approximate method, which is composed of two steps (Matching and Verification), for this energy minimization problem. Our method can handle non-rigid transformations, since the consistencies are checked locally between neighboring nodes only. We demonstrated the effectiveness of our approach through experiments on two databases: an industrial database (i.e. project Pinka, cf. chapter 1) and a standard challenging database. The experimental results indicate that our method is robust and achieves high accurate performance.

Exploring the spatio-temporal relationships for improving the bag-of-words models – The conventional bag of words models (BoW) have been successfully used in activity recognition. Despite their success, BoW models discard spatio-temporal relations between local features. We have proposed new features, called pairwise features (PWF), which encode both the appearance and the spatio-temporal relations of the local features for action recognition. Our PWFs are constructed by grouping pairs of local features which are both close in space and close in time. Because our PWFs contain both geometric and appearance information, we have constructed two separate codebooks for each feature parts of PWFs, and a combination of these two codebooks was also proposed for training/classification. Experimental results demonstrated that our PWF features significantly improve the

bag-of-words performance.

Beyond this contribution, we have also demonstrated that our pairwise features (i.e. geometric feature part) can keep their discriminative power even across different datasets, i.e. when learning in one dataset and testing on another one.

Human activity recognition through graph matching techniques – Our previous work in chapter 6, indicated that taking into account location information and temporal direction among local features give valuable improvements for action classification. In our second work on graph matching, we exploited higher order relationships between local features. In particular, we have adopted a hypergraph matching technique to perform triplet matching. First, local features are extracted, and hypergraphs are constructed from them, i.e. graphs with edges involving 3 nodes. The action recognition problem was translated into a matching task between two hypergraphs. We have adopted a spectral-based method to solve the correspondence problem. Our contributions were mainly focused on the following stages:

- **Graph construction –** We have proposed a significant way to construct more expressive graphs from local features extracted from video sequences.
- **Compatibility matrix estimation –** We have incorporated both triangle geometry and their orientations into our new objective function to accurately compute the compatibilies between model triangles and scene triangles.
- **Matching initialization –** We took advantage of local descriptors associated to each node, to initialize the matching process.
- **Score calculation –** We have proposed to interpret the final result to compute a more informative score.

Experiments on two standard datasets demonstrated that our method is comparable with state of the art methods on classification. Besides, our method offers several advantages over most other methods:

- By matching local features instead of classifying entire sequences, our method is able to detect multiple different activities which occur simultaneously in the same video sequence.
- The visual results tested on our own dataset confirmed that our method can detect activities in continous videos.

General conclusion

From the obtained results, we can conclude that spatial and spatio-temporal relations bring a significant improvement over local features. In addition, we have shown that formulating the recognition problem as a graph matching problem allows to take into account any relationships between local features (e.g. spatial and/or spatio-temporal relationships), despite the difference across domains (e.g. in this thesis, we used graph matching techniques to solve both object recognition and activity recognition problems).

Our experimental results from chapter 6 and 7 indicated that spatio-temporal information and temporal direction among local features, which have been largely ignored in the conventional bag of words models, have a significant influence on activity recognition.

Our methods have several advantages over existing work as we described above. However, there are also limitations to our approaches for which we will mention and discuss several potential solutions in the next section.

8.2 Limitations and Future work

In this section, we first propose some potential solutions that can be considered as short term work to resolve the limitations of our current works. Then, we would like to further discuss about the future work which could be done to improve the two problems studied in this thesis.

8.2.1 Limitations and some potential solutions

There are several possible extensions for this work, including:

Improvement of the patch-based Zernike moment features

Through experiments, we have proved that Zernike moments are suitable for sketch recognition, in particular for storyboard scenes. Nevertheless, as shown by our experimental results in chapter 4, our method gives worst results for simple model objects such as bottles and mugs. This can be explained as follows: first, such model objects are composed of only basic line-segments, and patches extracted from these objects are themselve line-segments. Because Zernike moment descriptors calculated for such segments are not enough discriminative, the first step in our decoupled matching algorithm, which searches the best match for each model patch, may return arbitrary best matches from a "pool" of similar scene patches, i.e. pieces of segments can be present anywhere in the scene. Therefore, the verification step, which checks for the consistencies, will fail.

We think that a combination of our patch-based Zernike moments with contours descriptors like a group of k-adjacent segments (kAS) from [FFJS08a] (cf. chapter 3) can

overcome this limitation.

Improvement of the decoupled approximate algorithm

By decoupling terms of the energy function (cf. function 2.6), our method can deal with any graph size, and non-rigid transformation. Nevertheless, the second step of our decoupled algorithm needs some best matches returned from the first step, to ensure a reliable result. In practice, this requirement is sometimes not reached due to noise, cluttered scenes, or descriptors used are not discriminative enough.

It is known that the energy function described in eq. 2.6 can be solved exactly if the graph is a chain and the edges are between neighbors in the chain. The solution can be calculated with complexity $O(N^m.N^{s^2})$ with dynamic programming. In the context of hidden markov models, this algorithm is known as the Viterbi algorithm.

Inspired from this, we could represent the graphs as chains of their nodes, which allow us to find exact solutions without decoupling the terms of function 2.6. Figure 8.1 illustrates this idea. Based on the chain representation, dynamic programming algo-

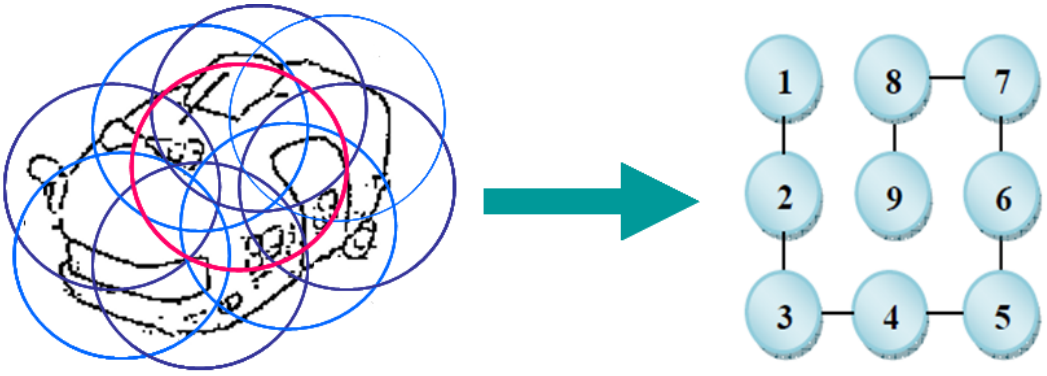


Figure 8.1: Illustration of linear approximate matching. Here our patch-based representation on the left, can be represented as a linear order (right), which allows to quickly calculate the global minimum.

rithm can be used for finding a global solution for eq. 2.6, which is a special case of the belief propagation (BP) algorithm from [Pea88].

Improvement of the graph matching algorithms for activity recognition Although we have used the high order matching (i.e. third order) to exploit the spatio-temporal information, the relationships between triangles, which could speed up detection, have not been taken into account. We intend to capture such relationships, e.g. pairs of triangles.

Improvement of the detection of multiple instances of the same object/activity

Probably one of the major limitations of our methods proposed for both object and activity recognition applications is the ability of our algorithm to detect multiple in-

stances of exactly the same object or activity in the scene. Actually, we need to detect the first instance, eliminate it, and restart a new process for other instances. This is quite computationally expensive, and not optimal.

From the success of existing works [MGMR02][TKR08c][RLAB10], it seems that association graphs can solve this problem. The main idea is to use another graph, called association graph, where each entry holds a match hypothesis between a model node and a scene node. This graph is constructed during matching, and the hypothesis are iteratively updated. Once the matching is done, the number of connected components retained in the association graph, represents potential candidats of the model.

How could we integrate this association graph into our actual algorithms? In our second application of hypergraph matching for activity recognition, it would be easy to do so, because the spectral-based algorithm, which we employed, is itself an iterative process. For our decoupled approximate algorithm, we think that the association graph would be updated in the first step of our algorithm. However, for each model patch, we need to keep several of its corresponding scene patches to construct the association graph. Then, in the second step, the consistencies need to be verified for all connected components of the obtained association graph.

8.2.2 Future research

In this section, we reflect on what future work should be undertaken.

Learning graph matching or traditional graph matching? – From what we have learned from this work, we believe that the future research for graph matching in computer vision will lie in learning graph matching techniques instead of standard graph matching algorithms used so far. This would be due to the fact that traditional graph matching techniques have several disadvantages. For instance, to recognize a model class, it requires various template models, which will be matched against a scene graph. However, the obtained results are sometimes not reliable, because althrough they express the same class model, two different pairs of graphs may give very different scores, i.e. due to the problem of intra-class variation. Learning graph matching techniques can solve this limitation.

For the problem of drawing recognition, Ferrari et al. in [FJS10] have already introduced an idea for learning a handrawn model (see figure 8.2) for each class (Mug, Giraffe, Apple, ..). In fact, their method is like a voting-based mechanism, which relies on the occurrences and connectedness of kAS features (cf. chapter 3) on training images. We would like to note that our tests with kAS features have failed on storyboard scenes.

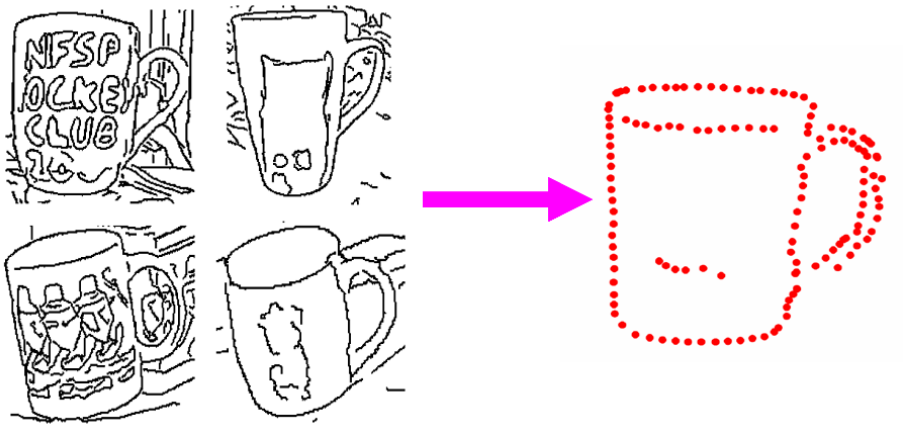


Figure 8.2: Illustration of learning a model class from several sample examples. This figure is adapted from [FJS10]

We think that it would be interesting to "learn" these models using graph matching, i.e. learning graph matching. We may benefit from the work on learning graph matching from Caetano et al. [CRM*09].

How to reduce the complexity of graph matching algorithms? - Application to activity recognition.

Although we have proposed several significant ways to reduce the complexity of the matching problem, it is still of high computational cost. Moreover, our graph matching algorithm is not scalable, since the complexity of our algorithm increases with the graph sizes. We construct our graphs from local points extracted as local maxima of response function. We think that the local maxima values of the response function, which have been largely ignored in the recognition stage, may be useful to speed up detection and reduce the complexity. Generally, according to the response function (of any detectors), a greater value is more *significant* (*important*) than a small one, i.e. not all points contribute equally to recognition. Based on this, it would be interesting to design a multi-level graph matching algorithm, known as a hierarchical matching, where the first level contains only a few *important* points (nodes) of the model. The matching process would start from the first level, and during matching, the hypothesis (i.e. consistencies constraint) would be checked to decide to continue or stop searching. This solution may not require searching for all possible assignments, thus would reduce the computation time significantly, however it would heavily rely on the response function and the hypothesis defined. We may benefit from a similar idea introduced by Revaud et al. [RLAB10], which has been successful applied for specific object recognition.

Bibliography

[AAR04] Agarwal S., Awan A., Roth D.: Learning to detect objects in images via a sparse, part-based representation. *IEEE Trans. Pattern Anal. Mach. Intell. 26*, 11 (2004), 1475–1490. 61

[AC99] Aggarwal J. K., Cai Q.: Human motion analysis: a review. *Comput. Vis. Image Underst. 73*, 3 (1999), 428–440. 100, 101

[ACS07] Anelli M., Cinque L., Sangineto E.: Deformation tolerant generalized hough transform for sketch-based image retrieval in complex scenes. *Image Vision Comput. 25*, 11 (2007), 1802–1813. 56

[AD93] Almohamad H. A., Duffuaa S. O.: A linear programming approach for the weighted graph matching problem. *IEEE Trans. Pattern Anal. Mach. Intell. 15*, 5 (1993), 522–525. 36

[ADV07] Ansary T. F., Daoudi M., Vandeborre J.: A bayesian 3-d search engine using adaptive views clustering. *IEEE Transactions on Multimedia 9*, 1 (2007), 78–88. 73, 85

[AP04] Aggarwal J. K., Park S.: Human motion: Modeling and recognition of actions and interactions. In *3DPVT '04: Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 640–647. 100

[APCGA10] Anh Phuong T., Christian W., Guillaume L., Atilla B.: Recognizing and localizing individual activities through graph matching. In *International Conference on Advanced Video and Signal-Based Surveillance (AVSS)* (Sept. 2010), IEEE, (Ed.). 138

[APTJ10] Anh Phuong Ta Christian Wolf G. L. A. B., Jolion J.-M.: Pairwise features for human action recognition . In *International Conference on Pattern Recognition (ICPR)* (Aug. 2010), IEEE, (Ed.). 122

[ATKI10] Ahad M. A. R., Tan J. K., Kim H., Ishikawa S.: Analysis of motion self-occlusion problem due to motion overwriting for human activity recognition. *Journal of Multimedia 5*, 1 (2010), 36–46. 10

[BA93] Black M. J., Anandan P.: A framework for the robust estimation of optical flow. In *ICCV* (1993), pp. 231–236. 104

[Bal81] Ballard D. H.: Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition 13*, 2 (1981), 111–122. 54

[BBDB*09] Ballan L., Bertini M., Del Bimbo A., Seidenari L., Serra G.: Recognizing human actions by fusing spatio-temporal appearance and motion descriptors. In *ICIP'09: Proceedings of the 16th IEEE international conference on Image processing* (Piscataway, NJ, USA, 2009), IEEE Press, pp. 3533–3536. 134, 153

[BBM05] Berg A. C., Berg T. L., Malik J.: Shape matching and object recognition using low distortion correspondences. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1* (Washington, DC, USA, 2005), pp. 26–33. 41

[BCS08] Batra D., Chen T., Sukthankar R.: Space-time shapelets for action recognition. In *WMVC '08: Proceedings of the 2008 IEEE Workshop on Motion and video Computing* (Washington, DC, USA, 2008), IEEE Computer Society, pp. 1–6. 116

[BD96] Bobick A., Davis J.: An appearance-based representation of action. In *ICPR '96: Proceedings of the 1996 International Conference on Pattern Recognition (ICPR '96) Volume I* (Washington, DC, USA, 1996), IEEE Computer Society, p. 307. 103

[BD01] Bobick A. F., Davis J. W.: The recognition of human movement using temporal templates. *IEEE Trans. Pattern Anal. Mach. Intell. 23*, 3 (2001), 257–267. xvi, 103, 104, 116

[BFB94] Barron J. L., Fleet D. J., Beauchemin S. S.: Performance of optical flow techniques. *Int. J. Comput. Vision 12*, 1 (1994), 43–77. 104

[BGS*05a] Blank M., Gorelick L., Shechtman E., Irani M., Basri R.: Actions as space-time shapes. vol. 2, pp. 1395–1402 Vol. 2. 103, 152

[BGS*05b] Blank M., Gorelick L., Shechtman E., Irani M., Basri R.: Actions as space-time shapes. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 1395–1402. 116

[BGX09] Bregonzio M., Gong S., Xiang T.: Recognising action as clouds of space-time interest points. pp. 1948–1955. 109

[BHHW05] Bar-Hillel A., Hertz T., Weinshall D.: Object class recognition by boosting a part-based model. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 702–709. 61

[BK73] Bron C., Kerbosch J.: Algorithm 457: finding all cliques of an undirected graph. *Commun. ACM 16*, 9 (1973), 575–577. 27

[Blu73] Blum H.: Biological shape and visual science. *Journal of Theoretical Biology 38*, 2 (February 1973), 205–287. 56

[BM97] Bunke H., Messmer B.: Recent advances in graph matching. *Journal of Pattern Recognition and Art. Intelligence 11*, 1 (1997), 169–203. 27

[BMP01] Belongie S., Malik J., Puzicha J.: Matching shapes. In *ICCV* (2001), pp. 454–463. 58

[BMP02] Belongie S., Malik J., Puzicha J.: Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 24*, 4 (2002), 509–522. xiv, 58, 59

[Bob97] Bobick A.: Movement, activity, and action: The role of knowledge in the perception of motion. 1257–1265. 101

[Boo89] Bookstein F. L.: Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence 11*, 6 (1989), 567–585. 59

[BSW05] Brown M., Szeliski R., W. S. A. J.: Multi-image matching using multi-scale oriented patches. In *CVPR (1)* (2005), pp. 510–517. 78

[Bun00] Bunke H.: Graph matching: Theoretical foundations, algorithms, and applications. In *Proc. Vision Interface 2000* (2000), pp. 82–88. 31

[BVZ01] Boykov Y., Veksler O., Zabih R.: Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell. 23*, 11 (2001), 1222–1239. 31

[BYL08] Bai X., Yang X., Latecki L. J.: Detection and recognition of contour parts based on shape similarity. *Pattern Recogn. 41*, 7 (2008), 2189–2199. xiv, 56, 57

[BYV00] Baeza-Yates R., Valiente G.: An image similarity measure based on graph matching. In *SPIRE '00: Proceedings of the Seventh International Symposium on String Processing Information Retrieval (SPIRE'00)* (Washington, DC, USA, 2000), IEEE Computer Society, p. 28. 41

[CCB04] Caetano T. S., Caelli T., Barone D. A. C.: Graphical models for graph matching. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on 2* (2004), 466–473. 32

[CD01] Cula O. G., Dana K. J.: Compact representation of bidirectional texture functions. In *CVPR (1)* (2001), pp. 1041–1047. 13

[CDF*04] Csurka G., Dance C. R., Fan L., Willamowski J., Bray C.: Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV* (2004), pp. 1–22. 61

[CFSV00] Cordella L., Foggia P., Sansone C., Vento M.: Fast graph matching for detecting cad image components. pp. Vol II: 1034–1037. 27

[CFSV01] Cordella L. P., Foggia P., Sansone C., Vento M.: An improved algorithm for matching large graphs. In *In: 3rd IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition, Cuen* (2001), pp. 149–159. 27

[CFSV04] Conte D., Foggia P., Sansone C., Vento M.: Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI) 18*, 3 (2004), 265–298. 20, 21, 30, 42, 140

[CFSV07] Conte D., Foggia P., Sansone C., Vento M.: How and why pattern recognition and computer vision applications use graphs. In *Applied Graph Theory in Computer Vision and Pattern Recognition*. 2007, pp. 85–135. 20

[CH01] Carcassoni M., Hancock E. R.: Weighted graph-matching using modal clusters. In *CAIP '01: Proceedings of the 9th International Conference on Computer Analysis of Images and Patterns* (London, UK, 2001), Springer-Verlag, pp. 142–151. 37

[CH03] Carcassoni M., Hancock E.: Spectral correspondence for point pattern matching. *Pattern Recognition 36*, 1 (January 2003), 193–204. 36

[Cho06] Cho T.-H.: Object matching using generalized hough transform and chamfer matching. In *PRICAI* (2006), pp. 1253–1257. 56

[CK04] Caelli T., Kosinov S.: An eigenspace projection clustering method for inexact graph matching. *IEEE Trans. Pattern Anal. Mach. Intell. 26*, 4 (2004), 515–519. 36

[CK10] Chertok M., Keller Y.: Efficient high order matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence 99*, PrePrints (2010). 38, 39, 40

[CKP95] Christmas W. J., Kittler J., Petrou M.: Structural matching in computer vision using probabilistic relaxation. *IEEE Trans. Pattern Anal. Mach. Intell. 17*, 8 (1995), 749–764. 35

[CLE05] Choksuriwong A., Laurent H., Emile B.: Comparison of invariant descriptors for object recognition. In *ICIP (1)* (2005), pp. 377–380. 53

[CLRM05] Choksuriwong A., Laurent H., Rosenberger C., Maaoui C.: Object recognition using local characterisation and zernike moments. In *ACIVS* (2005), pp. 108–115. 54, 73

[CRM*09] Caetano T., Rio S., McAuley J. J., Cheng L., Le Q. V., Smola A. J.: Learning graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 31*, 6 (2009), 1048–1058. 165

[CSS07a] Cour T., Srinivasan P., Shi J.: Balanced graph matching. In *Advanced in Neural Information Processing Systems* (2007), MIT Press, Cambridge, MA, 2007., pp. 313–320. 36, 37, 40

[CSS07b] Cour T., Srinivasan P., Shi J.: Balanced graph matching. In *NIPS* (2007). 144, 145

[DBKP09] Duchenne O., Bach F. R., Kweon I.-S., Ponce J.: A tensor-based algorithm for high-order graph matching. In *CVPR* (2009), pp. 1980–1987. xxi, 21, 38, 39, 40, 42, 140, 141, 142, 143, 144, 145, 147

[DC04] Dance Chris W. J. F. L. B. C., Csurka G.: Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision* (2004). 13

[DRCB05] Dollar P., Rabaud V., Cottrell G., Belongie S.: Behavior recognition via sparse spatio-temporal features. VS-PETS, pp. 65–72. xvi, 101, 106, 107, 108, 109, 110, 116, 123, 128, 134, 145, 146, 153

[DS03] Dorkó G., Schmid C.: Selection of scale-invariant parts for object class recognition. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision* (Washington, DC, USA, 2003), IEEE Computer Society, p. 634. 13

[DSD*04] Demirci M. F., Shokoufandeh A., Dickinson S., Keselman Y., Bretzner L.: Many-to-many feature matching using spherical coding of directed graphs. In *In Proceedings, 8th European Conference on Computer Vision* (2004), pp. 332–335. 40

[DTEK06] Duchi J., Tarlow D., Elidan G., Koller D.: Using combinatorial optimization within max-product belief propagation. In *NIPS* (2006), pp. 369–376. 32

[DTS96] DePiero F. W., Trivedi M. M., Serbin S.: Graph matching using a direct classification of node attendance. *Pattern Recognition 29*, 6 (1996), 1031–1048. 41

[EBMM03] Efros A. A., Berg A. C., Mori G., Malik J.: Recognizing action at a distance. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision* (Washington, DC, USA, 2003), IEEE Computer Society. xvi, 104, 105

[EF84] Eshera M., Fu K.: A similarity measure between attributed relational graphs for image analysis. pp. 75–77. 34

[EHK06] Elidan G., Heitz G., Koller D.: Learning object shape: From drawings to images. In *CVPR06* (2006), pp. 2064–2071. 74

[ESI98] El Sonbaty Y., Ismail M.: A new algorithm for subgraph optimal isomorphism. 205–218. 41

[FAI*05] Forsyth D. A., Arikan O., Ikemoto L., O'Brien J., Ramanan D.: Computational studies of human motion: part 1, tracking and motion synthesis. *Found. Trends. Comput. Graph. Vis. 1*, 2-3 (2005), 77–254. 100

[FB81] Fischler M. A., Bolles R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM 24*, 6 (1981), 381–395. 21

[FE73] Fischler M. A., Elschlager R. A.: The representation and matching of pictorial structures. *IEEE Trans. Comput. 22*, 1 (1973), 67–92. 35

[Fer03] :. *Object class recognition by unsupervised scale-invariant learning* (2003), vol. 2. 61

[FFJS08a] Ferrari V., Fevrier L., Jurie F., Schmid C.: Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 30*, 1 (2008), 36–51. xiv, xv, 62, 63, 64, 65, 73, 74, 94, 162

[FFJS08b] Ferrari V., Fevrier L., Jurie F., Schmid C.: Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 1 (2008), 36–51. 12

[FJS07] Ferrari V., Jurie F., Schmid C.: Accurate object detection with deformable shape models learnt from images. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on* (2007), pp. 1–8. 41, 66, 81, 82, 83, 85

[FJS10] Ferrari V., Jurie F., Schmid C.: From images to shape models for object detection. *International Journal of Computer Vision 87*, 3 (May 2010), 284–303. xvii, 164, 165

[FLM00] Fuchs F., Le-Men H.: Efficient subgraph isomorphism with 'a priori' knowledge (application to 3d reconstruction of buildings for cartography). In *Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition* (London, UK, 2000), Springer-Verlag, pp. 427–436. 41

[Flu00] Flusser J.: On the independence of rotation moment invariants. *Pattern Recognition 33*, 9 (2000), 1405–1410. 52

[FM08] Fathi A., Mori G.: Action recognition by learning mid-level motion features. In *CVPR* (2008). 134, 135, 153, 154

[FR95] Freeman W., Roth M.: Orientation histogram for hand gesture recognition. In *Int'l Workshop on Automatic Face- and Gesture-Reczhaognition* (1995). 58

[FSV01] Foggia P., Sansone C., Vento M.: A performance comparison of five algorithms for graph isomorphism. *Proc. of the 3rd IAPR TC-15 Workshop on Graph-based Representations in Pattern Recognition* (2001), 188–199. 27

[FTG06] Ferrari V., Tuytelaars T., Gool L. J. V.: Object detection by contour segment networks. In *ECCV (3)* (2006), pp. 14–28. 63, 66, 81, 82, 83, 85

[Gav99] Gavrila D. M.: The visual analysis of human movement: a survey. *Comput. Vis. Image Underst. 73*, 1 (1999), 82–98. 100, 101

[GBS*07] Gorelick L., Blank M., Shechtman E., Irani M., Basri R.: Actions as space-time shapes. *PAMI 29*, 12 (December 2007), 2247–2253. 134, 135, 153, 154

[GIB08] Gilbert A., Illingworth J., Bowden R.: Scale invariant action recognition using compound features mined from dense spatio-temporal corners. In *ECCV* (Berlin, Heidelberg, 2008), Springer-Verlag, pp. 222–233. 116, 117, 124, 125, 134, 153

[GP03] Grigorescu C., Petkov N.: Distance sets for shape filters and shape recognition. *IEEE Transactions on Image Processing 12*, 10 (2003), 1274–1286. 59

[GR96] Gold S., Rangarajan A.: A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence 18*, 4 (1996), 377–388. 36

[GS] Greig D. M. P. B. T., Seheult A. H.: Exact maximum a posteriori estimation for binary images. *J. of the Royal Statistical Society, Series B 51*, 271–279. 31

[GS08] Ghosh P., Sen A.: Energy minimization using a greedy randomized heuristic for the voltage assignment problem in noc. In *SoCC* (2008), pp. 79–84. 78

[GSTL02] Gu J., Shu H., Toumoulin C., Luo L.: A novel algorithm for fast computation of zernike moments. *Pattern Recognition 35*, 12 (2002), 2905–2911. 80

[GVL96] Golub G. H., Van Loan C. F.: *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996. 37

[Hes02] Heskes T.: Stable fixed points of loopy belief propagation are local minima of the bethe free energy. In *NIPS* (2002), pp. 343–350. 32

[HH99] Huet B., Hancock E. R.: Shape recognition from large image libraries by inexact graph matching. *Pattern Recogn. Lett. 20*, 11-13 (1999), 1259–1269. 35

[Hof99] Hofmann T.: Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 1999), ACM, pp. 50–57. 114

[Hou62] Hough P.: Method and means for recognizing complex patterns. United States Patent Office. 54

[HR07] Hou S., Ramani K.: Calligraphic interfaces: Classifier combination for sketch-based 3d part retrieval. *Comput. Graph. 31*, 4 (2007), 598–609. 73, 85

[HRW07] Hoiem D., Rother C., Winn J.: 3d layoutcrf for multi-view object class recognition and segmentation. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on* (2007), pp. 1–8. 89

[HS80] Horn B. K., Schunck B. G.: *Determining Optical Flow*. Tech. rep., Cambridge, MA, USA, 1980. 104

[HS88] Harris C., Stephens M.: A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference* (1988), pp. 147–151. 108

[Hu62] Hu M.-K.: Visual pattern recognition by moment invariants. *Information Theory, IEEE Transactions on 8*, 2 (1962), 179–187. 51

[HW02a] Hancock E. R., Wilson R. C.: Graph-based methods for vision: A yorkist manifesto. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition* (London, UK, 2002), Springer-Verlag, pp. 31–46. 35

[HW02b] Hlaoui A., Wang S.: A new algorithm for graph matching with application to content-based image retrieval. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition* (London, UK, 2002), Springer-Verlag, pp. 291–300. 41

[Joh73] Johansson G.: Visual perception of biological motion and a model for its analysis. 201–211. 6

[JS04] Jurie F., Schmid C.: Scale-invariant shape features for recognition of object categories. In *International Conference on Computer Vision & Pattern Recognition* (2004), vol. II, pp. 90–96. 48

[JSWP07] Jhuang H., Serre T., Wolf L., Poggio T.: A biologically inspired system for action recognition. In *In ICCV* (2007), pp. 1–8. 113

[Kad03] :. *Scale Saliency: a novel approach to salient feature and scale selection* (July 2003). 109

[KC02] Kosinov S., Caelli T.: Inexact multisubgraph matching using graph eigenspace and clustering models. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition* (London, UK, 2002), Springer-Verlag, pp. 133–142. 37

[KC09] Kim T.-K., Cipolla R.: Canonical correlation analysis of video volume tensors for action categorization and detection. *PAMI. 31*, 8 (2009), 1415–1428. 153

[KH89] Kittler J., Hancock E.: Combining evidence in probabilistic relaxation. 29–51. 35

[KH90] Khotanzad A., Hong Y. H.: Invariant image recognition by zernike moments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 12*, 5 (1990), 489–497. 52

[KKK02] Kim S.-H., Kweon I.-S., Kim I.-C.: Probabilistic model-based object recognition using local zernike moments. In *MVA* (2002), pp. 334–337. 73

[KKSO00] Kim H.-K., Kim J.-D., Sim D.-G., Oh D.-I.: A modified zernike moment shape descriptor invariant to translation, rotation and scale for similarity-based image retrieval. In *IEEE International Conference on Multimedia and Expo (I)* (2000), pp. 307–310. 80

[KMS08] Kläser A., Marszałek M., Schmid C.: A spatio-temporal descriptor based on 3d-gradients. In *British Machine Vision Conference* (sep 2008), pp. 995–1004. 134, 153

[Kol06] Kolmogorov V.: Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. Pattern Anal. Mach. Intell. 28*, 10 (2006), 1568–1583. 32

[KPT10] Komodakis N., Paragios N., Tziritas G.: Mrf energy minimization and beyond via dual decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence 99*, PrePrints (2010). 32

[KSH05] Ke Y., Sukthankar R., Hebert M.: Efficient visual event detection using volumetric features. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 166–173. 113

[KSH07] Ke Y., Sukthankar R., Hebert M.: Event detection in crowded videos. In *IEEE International Conference on Computer Vision* (October 2007). 117, 140

[KTP07] Komodakis N., Tziritas G., Paragios N.: Fast, approximately optimal solutions for single and dynamic mrfs. pp. 1–8. 32

[KZ04] Kolmogorov V., Zabih R.: What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 2 (2004), 147–159. 14, 32

[LAS08] Liu J., Ali S., Shah M.: Recognizing human actions using multiple features. In *CVPR* (2008). 101, 123, 134

[LCSL07] Laptev I., Caputo B., Schüldt C., Lindeberg T.: Local velocity-adapted motion events for spatio-temporal recognition. *Comput. Vis. Image Underst. 108*, 3 (2007), 207–229. 113

[Lev66] Levenshtein V. I.: Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady 10*, 8 (1966), 707–710. 31

[LFJB*10] Lecellier F., Fadili J., Jehan-Besson S., Aubert G., Revenu M., Saloux E.: Region-based active contours with exponential family observations. *J. Math. Imaging Vis. 36*, 1 (2010), 28–45. 50

[LH05a] Leordeanu M., H. M.: A spectral technique for correspondence problems using pairwise constraints. In *In ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision* (Washington, DC, USA, 2005), pp. 1482–1489. 36, 37, 39

[LH05b] Leordeanu M., Hebert M.: A spectral technique for correspondence problems using pairwise constraints. In *ICCV '05:* (Washington, DC, USA, 2005), pp. 1482–1489. 21, 38, 40, 42, 140, 141, 144, 145

[LHS07] Leordeanu M., Hebert M., Sukthankar R.: Beyond local appearance: Category recognition from pairwise interactions of simple features. In *CVPR* (2007). 41, 74

[Li94] Li S. Z.: A markov random field model for object matching under contextual constraints. In *In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (1994), pp. 866–869. 35

[LK81] Lucas B. D., Kanade T.: An iterative image registration technique with an application to stereo vision. In *IJCAI'81: Proceedings of the 7th international joint conference on Artificial intelligence* (San Francisco, CA, USA, 1981), Morgan Kaufmann Publishers Inc., pp. 674–679. 104

[LL03] Laptev I., Lindeberg T.: Space-time interest points. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision* (Washington, DC, USA, 2003), IEEE Computer Society, p. 432. xvi, 107, 108

[LLS04] Leibe B., Leonardis A., Schiele B.: Combined object categorization and segmentation with an implicit shape model. In *ECCV'04 Workshop on Statistical Learning in Computer Vision* (Prague, Czech Republic, May 2004), pp. 17–32. 61

[LMSR08] Laptev I., Marszalek M., Schmid C., Rozenfeld B.: Learning realistic human actions from movies. In *CVPR* (2008), pp. 1–8. xvi, 110, 111, 113, 134, 153

[LN07] Lv F., Nevatia R.: Single view human action recognition using key pose matching and viterbi path searching. pp. 1–8. 103

[Low99] Lowe D. G.: Object recognition from local scale-invariant features. In *ICCV* (1999), pp. 1150–1157. 11, 12, 50, 73

[Low04] Lowe D. G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2 (2004), 91–110. xvi, 48, 58, 111

[LP05] Li F.-F., Perona P.: A bayesian hierarchical model for learning natural scene categories. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society*

Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 (Washington, DC, USA, 2005), IEEE Computer Society, pp. 524–531. 13

[LS03] Leibe B., Schiele B.: Analyzing appearance and contour based methods for object categorization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03)* (Madison, WI, June 2003). 60

[LS08] Liu J., Shah M.: Learning human actions via information maximization. In *CVPR* (2008). 116, 124, 125, 134, 139, 153

[LV02] Larrosa J., Valiente G.: Constraint satisfaction algorithms for graph pattern matching. *Mathematical. Structures in Comp. Sci. 12*, 4 (2002), 403–422. 27

[MB98] Messmer B. T., Bunke H.: A new algorithm for error-tolerant subgraph isomorphism detection. *IEEE Trans. Pattern Anal. Mach. Intell. 20*, 5 (1998), 493–504. 41

[MB99] Messmer B., Bunke H.: A decision tree approach to graph and subgraph isomorphism detection. 1979–1998. 27

[MC03] M. Jo a., C. Jo a. P.: A global solution to sparse correspondence problems. *IEEE Trans. Pattern Anal. Mach. Intell. 25*, 2 (2003), 187–199. 40

[MCB08] McAuley J. J., Caetano T. S., Barbosa M. S.: Graph rigidity, cyclic belief propagation and point pattern matching. *IEEE Trans. on PAMI. abs/0710.0043* (2008). 33

[McK81] McKay B. D.: Practical graph isomorphism. *Congressus Numerantium 30* (1981), 45–87. 27

[MDS05] Mortensen E. N., Deng H., Shapiro L. G.: A sift descriptor with global context. In *CVPR (1)* (2005), pp. 184–190. 60

[MFM04] Martin D. R., Fowlkes C., Malik J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell. 26*, 5 (2004), 530–549. 63

[MG01] Moeslund T. B., Granum E.: A survey of computer vision-based human motion capture. *Comput. Vis. Image Underst. 81*, 3 (2001), 231–268. 100, 101

[MGMR02] Melnik S., Garcia-Molina H., Rahm E.: Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *18th International Conference on Data Engineering (ICDE 2002)* (2002). 164

[MHK06] Moeslund T., Hilton A., Kruger V.: A survey of advances in vision-based human motion capture and analysis. 90–126. 100

[MI00] Miyamori H., Iisaku S.-i.: Video annotation for content-based retrieval using human behavior analysis and domain knowledge. In *FG '00: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000* (Washington, DC, USA, 2000), IEEE Computer Society, p. 320. 7

[MK99] Medasani S., Krishnapuram R.: A fuzzy approach to content-based image retrieval. In *ICMCS '99: Proceedings of the 1999 IEEE International Conference on Multimedia Computing and Systems* (Washington, DC, USA, 1999), IEEE Computer Society, p. 964. 36

[MKC01] Medasani S., Krishnapuram R., Choi Y.: Graph matching by relaxation of fuzzy assignments. 173–182. 36

[Mok95] Mokhtarian F.: Silhouette-based isolated object recognition through curvature scale space. *IEEE Trans. Pattern Anal. Mach. Intell. 17*, 5 (1995), 539–544. 49

[MP85] Mostafa A. Y. S., Psaltis D.: Image normalization by complex moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence 7*, 1 (January 1985), 46–55. 51

[MS98] Mokhtarian F., Suomela R.: Robust image corner detection through curvature scale space. *IEEE Trans. Pattern Anal. Mach. Intell. 20*, 12 (1998), 1376–1381. 49

[MS05] Mikolajczyk K., Schmid C.: A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell. 27*, 10 (2005), 1615–1630. 48

[MU] Mikolajczyk K., Uemura H.: Action recognition with motion-appearance vocabulary forest. *In CVPR, 2008*. 123

[Mun57] Munkres J.: Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics 5*, 1 (1957), 32–38. 30

[Muy01] Muybridge E.: *The Human Figure in Motion*. Dover Publications, 1901. 6

[NB07] Neuhaus M., Bunke H.: *Bridging the Gap Between Graph Edit Distance and Kernel Machines*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2007. 36

[NFF07] Niebles J., Fei-Fei L.: A hierarchical model of shape and appearance for human action classification. In *Proceedings of IEEE Intern. Conf. in Computer Vision and Pattern Recognition(CVPR).* (2007). 117

[NWFF08] Niebles J., Wang H., Fei-Fei L.: Unsupervised learning of human action categories using spatial-temporal words. *IJCV* (2008). xvi, xvii, 101, 106, 109, 114, 115, 123, 134, 142, 153

[NXGH08] Ning H., Xu W., Gong Y., Huang T.: Latent pose estimator for continuous action recognition. pp. II: 419–433. 115

[OAW99] Ozer B., Akansu A. N., Wolf W.: A graph based object description for information retrieval in digital image and video libraries. In *CBAIVL '99: Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries* (Washington, DC, USA, 1999), IEEE Computer Society, p. 79. 41

[OCKI06] Ogata T., Christmas W. J., Kittler J., Ishikawa S.: Improving human activity detection by combining multi-dimensional motion descriptors with boosting. In *ICPR (1)* (2006), pp. 295–298. 113

[OGM08] Olmos I., González J. A., México P.: Structural graph-based representations used for finding hidden patterns, 2008. xiii, 18

[OPP06] Oikonomopoulos A., Patras I., Pantic M.: Spatiotemporal salient points for visual recognition of human actions. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 36*, 3 (June 2006), 710–719. 109

[OPP09] Oikonomopoulos A., Patras I., Pantic M.: An implicit spatiotemporal shape model for human activity localization and recognition. *In CVPR 0* (2009), 27–33. 117, 119, 123, 124, 125, 134, 153

[OPZ06a] Opelt A., Pinz A., Zisserman A.: A boundary-fragment-model for object detection. In *ECCV (2)* (2006), pp. 575–588. 65, 74

[OPZ06b] Opelt A., Pinz A., Zisserman A.: Incremental learning of object detectors using a visual shape alphabet. In *CVPR (1)* (2006), pp. 3–10. xiv, 12, 61, 62

[PBB*99] Perchant A., Boeres C., Bloch I., Roux M., Ribeiro C.: Model-based scene recognition using graph fuzzy homomorphism solved by genetic algorithm, 1999. 41

[PCFSV04] P. Cordella L., Foggia P., Sansone C., Vento M.: A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Trans. Pattern Anal. Mach. Intell. 26*, 10 (2004), 1367–1372. 27

[Pea88] Pearl J.: *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988. 32, 163

[Pop10] Poppe R.: A survey on vision-based human action recognition. *Image Vision Comput. 28*, 6 (2010), 976–990. xvi, 100, 101, 102

[RA09] Ryoo M. S., Aggarwal J. K.: Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *ICCV* (2009). 117, 118, 119, 124, 125, 134, 140, 153, 154

[RAW08] Ravikumar P. D., Agarwal A., Wainwright M. J.: Message-passing for graph-structured linear programs: proximal projections, convergence and rounding schemes. In *ICML* (2008), pp. 800–807. 32

[Rei93] Reiss T. H.: *Recognizing Planar Objects Using Invariant Image Features*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1993. 49

[RHZ76] Rosenfeld A., Hummel R., Zucker S.: Scene labeling by relaxation operations. 420–433. 35

[RJM08] Ravishankar S., Jain A., Mittal A.: Multi-stage contour based detection of deformable objects. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision* (Berlin, Heidelberg, 2008), Springer-Verlag, pp. 483–496. 66, 75, 83

[RKLS07] Rother C., Kolmogorov V., Lempitsky V., Szummer M.: *Optimizing binary MRFs via extended roof duality*. Tech. rep., In Proc. CVPR, 2007. 32

[RLAB10] Revaud J., Lavoué G., Ariki Y., Baskurt A.: Scale-Invariant Proximity Graph for Fast Probabilistic Object Recognition. In *Conference on Image and Video Retrieval (CIVR)* (July 2010). Oral Presentation (oral acceptance rate: 10164, 165

[RLB09] Revaud J., Lavoué G., Baskurt A.: Improving zernike moments comparison for optimal similarity and rotation angle retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence 31*, 4 (2009), 627–636. 54, 73, 76, 80, 85, 91, 93, 94

[RM96] Rangarajan A., Mjolsness E.: A lagrangian relaxation network for graph matching. In *IEEE Trans. Neural Networks* (1996), IEEE Press, pp. 4629–4634. 36

[RM97] Ravela S., Manmatha R.: Retrieving images by similarity of visual appearance. In *CAIVL '97: Proceedings of the 1997 Workshop on Content-Based Access of Image and Video Libraries (CBAIVL '97)* (Washington, DC, USA, 1997), IEEE Computer Society, p. 67. 7

[RPAK88] Reeves A. P., Prokop R. P., Andrews S. E., Kuhl F. P.: Three-dimensional shape analysis using moments and fourier descriptors. *IEEE Trans. Pattern Anal. Mach. Intell. 10*, 6 (1988), 937–943. 51

[RYS*06] Rizon M., Yazid H., Saad P., Shakaff A. Y. M., Saad A. R., Mamat M. R., Yaacob S., Desa H., Karthigayan M.: Object detection using geometric invariant moment. *American Journal of Applied Sciences (Magazine/Journal) 3*, 6 (2006), 1876–1878. 52

[SA96] Shoukry A., Aboutabl M.: Neural-network approach for solving the maximal common subgraph problem. 785–790. 41

[SAS07] Scovanner P., Ali S., Shah M.: A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th international conference on Multimedia* (USA, 2007), ACM, pp. 357–360. xvi, 110, 111, 113, 116, 124, 125, 134, 153

[SB92] Shapiro L. S., Brady J. M.: Feature-based correspondence: an eigenvector approach. *Image Vision Comput. 10*, 5 (1992), 283–288. 36

[SBC05] Shotton J., Blake A., Cipolla R.: Contour-based learning for object detection. In *ICCV* (2005), pp. 503–510. xv, 12, 61, 62, 65, 74, 85, 87, 88

[SCH09] Sun X., Chen M., Hauptmann A.: Action recognition via local descriptors and holistic features. In *CVPR* (2009), pp. 58–65. 101, 112, 134, 153, 154

[SCSW] Shi Q., Cheng L., Smola A., Wang L.: A.: Discriminative human action segmentation and recognition using semi-markov model. In *In: CVPR (2008*. 115

[SD09] Sharif M. H., Djeraba C.: Exceptional motion frames detection by means of spatiotemporal region of interest features. In *ICIP'09: Proceedings of the 16th IEEE international conference on Image processing* (Piscataway, NJ, USA, 2009), IEEE Press, pp. 977–980. 103

[SDPNL08] Savarese S., Del Pozo A., Niebles J., Li F.: Spatial-temporal correlatons for unsupervised action classification. In *In WMVC* (2008), pp. 1–8. 101, 106, 123, 134, 142, 153

[SF83] Sanfeliu A., Fu K.: A distance measure between attributed relational graphs for pattern recognition. 353–362. 31

[SI07a] Shechtman E., Irani M.: Matching local self-similarities across images and videos. pp. 1–8. 117, 118

[SI07b] Shechtman E., Irani M.: Space-time behavior-based correlation—or—how to tell if two underlying motion fields are similar without computing them? *IEEE Trans. Pattern Anal. Mach. Intell. 29*, 11 (2007), 2045–2056. 117, 140, 154

[SK05] Sebastian T. B., Kimia B. B.: Curves vs. skeletons in object recognition. *Signal Processing 85*, 2 (2005), 247–263. 56

[SKK04] Sebastian T. B., Klein P. N., Kimia B. B.: Recognition of shapes by editing their shock graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 5 (2004), 550–571. 56

[SLC04] Schuldt C., Laptev I., Caputo B.: Recognizing human actions: a local svm approach. In *ICPR* (September 2004), vol. 3, pp. 32–36 Vol.3. xiii, 6, 101, 102, 106, 113, 123, 152

[SM83] Salton G., McGill M. J.: *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1983. 105

[SM97] Schmidt C., Mohr R.: Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19*, 5 (May 1997). 48, 50, 73

[SM09] Seo H., Milanfar P.: Detection of human actions from a single example. pp. 1965–1970. 118

[SS05] Schellewald C., Schnörr C.: Probabilistic subgraph matching based on convex relaxation. In *EMMCVPR* (2005), pp. 171–186. 40

[SSDZ98] Siddiqi K., Shokoufandeh A., Dickinson S. J., Zucker S. W.: Shock graphs and shape matching. In *ICCV* (1998), pp. 222–229. 56

[SvG08] Schindler K., van Gool L.: Action snippets: How many frames does human action recognition require? In *CVPR* (June 2008), IEEE Press. 134, 153

[SY98] Suganthan P. N., Yan H.: Recognition of handprinted chinese characters by constrained graph matching. *Image Vision Comput. 16*, 3 (1998), 191–201. 41

[SZ03] Sivic J., Zisserman A.: Video google: A text retrieval approach to object matching in videos. In *Proceedings of the International conference on computer vision (ICCV)* (2003), vol. 2, pp. 1470–1477. 105

[TC88] Teh C.-H., Chin R. T.: On image analysis by the methods of moments. *IEEE Trans. Pattern Anal. Mach. Intell. 10*, 4 (1988), 496–513. 52, 53

[TCSB06] T.S.Caetano, Caelli T., Schuurmans D., Barone D.: Graphical models and point pattern matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 10 (2006), 1646–1663. xiv, 32, 33, 34, 41, 78, 91

[TCSU08] Turaga P., Chellappa R., Subrahmanian V. S., Udrea O.: Machine recognition of human activities: A survey. *Circuits and Systems for Video Technology, IEEE Transactions on 18*, 11 (September 2008), 1473–1488. 101

[Tea80] Teague M. R.: Image analysis via the general theory of moments. *Journal of the Optical Society of America (1917-1983) 70* (August 1980), 920–930. 52

[TF79] Tsai W., Fu K.: Error-correcting isomorphisms of attributed relational graphs for pattern analysis. 757–768. 34

[TF83] Tsai W.-H., Fu K.-S.: Subgraph error-correcting isomorphisms for syntactic pattern recognition. *Trans. Systems, Man and Cybernetics SMC-13* (1983), 48–62. 34

[TH03] Torsello A., Hancock E. R.: Computing approximate tree edit distance using relaxation labeling. *Pattern Recogn. Lett. 24*, 8 (2003), 1089–1097. 35

[TKR08a] Torresani L., Kolmogorov V., Rother C.: Feature correspondence via graph matching: Models and global optimization. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision* (Berlin, Heidelberg, 2008), Springer-Verlag, pp. 596–609. 21, 42, 140, 141

[TKR08b] Torresani L., Kolmogorov V., Rother C.: Feature correspondence via graph matching: Models and global optimization. pp. II: 596–609. 33, 41, 73

[TKR08c] Torresani L., Kolmogorov V., Rother C.: Feature correspondence via graph matching: Models and global optimization. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2008), vol. 2, pp. 596–609. 164

[TSTC03] Thayananthan A., Stenger B., Torr P. H. S., Cipolla R.: Shape context and chamfer matching in cluttered scenes. In *CVPR (1)* (2003), pp. 127–133. 59

[TWLB09] Ta A. P., Wolf C., Lavoué G., Baskurt A.: 3D Object detection and viewpoint selection in sketch images using local patch-based Zernike moments. In *7th International workshop on Content-Based Multimedia Indexing (CBMI)* (June 2009), IEEE, (Ed.). 72

[TY04] Tu Z., Yuille A. L.: Shape matching and recognition - using generative models and informative features. In *ECCV (3)* (2004), pp. 195–209. 60

[Ull76] Ullmann J. R.: An algorithm for subgraph isomorphism. *J. ACM 23*, 1 (1976), 31–42. 25, 26

[Ull96] Ullman S.: *High-Level Vision: Object Recognition and Visual Cognition*, illustrated edition ed. The MIT Press, July 1996. 5

[Ume88] Umeyama S.: An eigendecomposition approach to weighted graph matching problems. *IEEE Trans. Pattern Anal. Mach. Intell. 10*, 5 (1988), 695–703. 36

[USB03] Ulrich M., Steger C., Baumgartner A.: Real-time object recognition using a modified generalized hough transform. *Pattern Recognition 36*, 11 (November 2003), 2557–2570. 56

[VH99] Veltkamp R., Hagedoorn M.: *State-of-the-art in shape matching*. Tech. Rep. UU-CS-1999-27, Utrecht University, the Netherlands, 1999. 47, 48

[VNU03] Vidal-Naquet M., Ullman S.: Object recognition with informative features and linear classification. In *ICCV* (2003), pp. 281–288. 61

[VZ02] Varma M., Zisserman A.: Classifying images of materials: Achieving viewpoint and illumination independence. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part III* (London, UK, 2002), Springer-Verlag, pp. 255–271. 13

[WB95] Wu X., Bhanu B.: Gabor wavelets for 3-d object recognition. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision* (Washington, DC, USA, 1995), IEEE Computer Society, p. 537. 50

[WBR07] Weinland D., Boyer E., Ronfard R.: Action recognition from arbitrary views using 3d exemplars. pp. 1–7. 101, 115, 123

[WC07] Wong S.-F., Cipolla R.: Extracting spatiotemporal interest points using global information. In *ICCV* (2007). 134

[WcFtH97] Wang Y.-K., chin Fan K., tzong Horng J.: Genetic-based search for error-correcting graph isomorphism. *IEEE Transactions on Systems, Man, and Cybernetics: Part B - Cybernetics 27* (1997), 588–597. 41

[WF01a] Weiss Y., Freeman W. T.: Correctness of belief propagation in gaussian graphical models of arbitrary topology. *Neural Comput. 13*, 10 (2001), 2173–2200. 32

[WF01b] Weiss Y., Freeman W. T.: On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory.* (2001), 736–744. 32

[WGLR08] Wang L., Geng X., Leckie C., Ramamohanarao K.: Moving shape dynamics: A signal processing perspective. In *CVPR* (2008). 101, 123

[WH97] Wilson R. C., Hancock E. R.: Structural matching by discrete relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19* (1997), 634–648. 35

[WH04] Wang H., Hancock E. R.: A kernel view of spectral point pattern matching. In *SSPR/SPR* (2004), pp. 361–369. 36

[WHT03] Wang L., Hu W., Tan T.: Recent developments in human motion analysis. *Pattern Recognition 36* (2003), 585–601. 101

[WJ06] Wolf C., Jolion J.-M.: Object count/Area Graphs for the Evaluation of Object Detection and Segmentation Algorithms. *International Journal on Document Analysis and Recognition 8*, 4 (2006), 280–296. 83, 89

[WJKB00] Wolf C., Jolion J., Kropatsch W., Bischof H.: Content Based Image Retrieval using Interest Points and Texture Features. In *Proceedings of the International Conference on Pattern Recognition* (2000), vol. 4, pp. 234–237. 73

[WJW05] Wainwright M. J., Jaakkola T. S., Willsky A. S.: Map estimation via agreement on trees: message-passing and linear programming. *IEEE Trans. on Information Theory 51* (2005), 2005. 32

[WKC07] Wong S.-F., Kim T.-K., Cipolla R.: Learning motion categories using both semantic and structural information. *IEEE Conf. on Computer Vision and Pattern Recognition* (2007). 115, 116, 124, 125, 134, 153

[WSM07] Wang Y., Sabzmeydani P., Mori G.: Semi-latent dirichlet allocation: a hierarchical model for human action recognition. In *Proceedings of the 2nd conference on Human motion* (Berlin, Heidelberg, 2007), Springer-Verlag, pp. 240–254. 115

[WTG08] Willems G., Tuytelaars T., Gool L.: An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV* (2008), pp. 650–663. 109, 134, 153

[WUK*09] Wang H., Ullah M. M., Kläser A., Laptev I., Schmid C.: Evaluation of local spatio-temporal features for action recognition. In *University of Central Florida, U.S.A* (2009). 112

[WW02] Wyk B. J. v., Wyk M. A. v.: Non-bayesian graph matching without explicit compatibility calculations. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition* (London, UK, 2002), Springer-Verlag, pp. 74–82. 36

[WWH02] Wyk B. J. v., Wyk M. A. v., Hanrahan H. E.: Successive projection graph matching. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition* (London, UK, 2002), Springer-Verlag, pp. 263–271. 36

[WYC90] Wong A., You M., Chan S.: An algorithm for graph optimal monomorphism. *T-SMC 20* (1990), 628–636. 34

[WZC94] Wang J., Zhang K., Chirn G.: The approximate graph matching problem. pp. B:284–288. 41

[XK01] Xu L., King I.: A pca approach for fast retrieval of structural patterns in attributed graphs. In *Humboldt University Berlin* (2001). 37

[XYCD07] Xiaodong Y., Yi L., Cornelia F., David D.: Object Detection Using Shape Codebook. In *British Machine Vision Conference (BMVC'07)* (December 2007), BMVC07. accepted. 65, 66

[YFW05] Yedidia J. S., Freeman W. T., Weiss Y.: Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory 51* (2005), 2282–2312. 32

[YOI92] Yamato J., Ohya J., Ishii K.: Recognizing human action in time-sequential images using hidden markov model. pp. 379–385. 115

[ZDS09] Zampelli1 S., Deville Y., Solnon C.: Solving subgraph isomorphism problems with constraint programming. *Constraints* (2009). 140

[Zer34] Zernike F.: Diffraction theory of the cut procedure and its improved form, the phase contrast method. *Physica 1* (1934), 689–704. 52

[ZHCC08] Zhang Z., Hu Y., Chan S., Chia L.-T.: Motion context: A new representation for human action recognition. In *ECCV (4)* (2008), pp. 817–829. 117, 124, 125, 134, 139, 153

[ZK06] Zhang W., Kosecka J.: Generalized ransac framework for relaxed correspondence problems. In *3DPVT* (2006), pp. 854–860. 78

[ZL02] Zhang D., Lu G.: Improving retrieval performance of zernike moment descriptor on affined shapes. In *ICME* (2002), vol. 1, pp. 205–208. 80

[ZL04] Zhang D., Lu G.: Review of shape representation and description techniques. *Pattern Recognition 37*, 1 (January 2004), 1–19. 47, 48, 53

[ZM03] Zhang H., Malik J.: Learning a discriminative classifier using shape context distances. In *CVPR (1)* (2003), pp. 242–247. 60

[ZMI06] Zelnik-Manor L., Irani M.: Statistical analysis of dynamic actions. *IEEE Trans. Pattern Anal. Mach. Intell. 28*, 9 (2006), 1530–1535. 117

[ZMLS07] Zhang J., Marszalek M., Lazebnik S., Schmid C.: Local features and kernels for classification of texture and object categories: A comprehensive study. *Int. J. Comput. Vision 73*, 2 (2007), 213–238. 61

[ZR72] Zahn C., Roskies R.: Fourier descriptors for plane closed curves. 269–281. 49

[ZS08] Zass R., Shashua A.: Probabilistic graph and hypergraph matching. In *CVPR* (2008). 21, 38, 39, 40, 42, 140, 141, 142

[ZWWS08] Zhu Q., Wang L., Wu Y., Shi J.: Contour context selection for object detection: A set-to-set contour matching approach. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision* (Berlin, Heidelberg, 2008), Springer-Verlag, pp. 774–787. xv, xix, 66, 75, 81, 83, 84, 85

Title: Inexact graph matching techniques: Application to object detection and human action recognition

Abstract: Object detection and human action recognition are two active fields of research in computer vision, which have applications ranging from robotics and video surveillance, medical image analysis, human-computer interactions to content-based video annotation and retrieval. At this time, building such robust recognition systems still remain very challenging tasks, because of the variations in action/object classes, different possible viewpoints, as well as illumination changes, moving cameras, complex dynamic backgrounds and occlusions. In this thesis, we deal with object and activity recognition problems. Despite differences in the applications' goals, the associated fundamental problems share numerous properties, for instance the necessity of handling non-rigid transformations. Describing a model object or a video by a set of local features, we formulate the recognition problem as a graph matching problem, where nodes represent local features, and edges represent spatial and/or spatio-temporal relationships between them. Inexact matching of valued graphs is a well known NP-hard problem, therefore we concentrated on finding approximate solutions. To this end, the graph matching problem is formulated as an energy minimization problem. Based on this energy function, we propose two different solutions for the two applications: object detection in images and activity recognition in video sequences. We also propose new features to improve the conventional Bag of words model, which is widely used in computer vision. Experiments on both standard datasets and our own datasets, demonstrate that our methods provide good results regarding the recent state-of-the-art in both domains.

Keywords: Object recognition, object localization, activity recognition, graph matching, pairwise features, multiple actions.

Titre: Mise en correspondance inexacte de graphes: application à la reconnaissance d'objets et d'activits dans la vidéo.

Résumé: La détection d'objets et la reconnaissance des activites humaines sont les deux domaines actifs dans la vision par ordinateur, qui trouve des applications en robotique, vidéo surveillance, analyse des images médicales, interaction homme-machine, annotation et recherche de la vidéo par le contenue. Actuellement, il reste encore très difficile de construire de tels systèmes, en raison des variations des classes d'objets et d'actions, les différents points de vue, ainsi que des changements d'illumination, des mouvements de caméra, des fonds dynamiques et des occlusions. Dans cette thèse, nous traitons le problème de la détection d'objet et d'activités dans la vidéo. Malgré ses différences de buts, les problèmes fondamentaux associés partagent de nombreuses propriétés, par exemple la nécessité de manipuler des transformations non-ridiges. En décrivant un modèle d'objet ou une vidéo par un ensemble des caractéristiques locales, nous formulons le problème de reconnaissance comme celui d'une mise en correspondance de graphes, dont les noeuds représentent les caractéristiques locales, et les arrêtes représentent les relations que l'on veut vérifier entre ces caractéristiques. Le problème de mise en correspondance inexacte de graphes est connu comme NP-difficile, nous avons donc porté notre effort sur des solutions approchées. Pour cela, le problème est transformé en problème d'optimisation d'une fonction d'énergie, qui contient un terme en rapport avec la distance entre les descripteurs locaux et d'autres termes en rapport avec les relations spatiales (ou/et temporelles) entre eux. Basé sur cette énergie, deux différentes solutions ont été proposées et validées pour les deux applications ciblées: la reconnaissance d'objets à partir d'images et la reconnaissance des activités dans la vidéo. En plus, nous avons également proposé un nouveaux descripteur pour améliorer les modèles de Sac-de-mots, qui sont largement utilisé dans la vision par ordinateur. Nos expérimentations sur deux bases standardes, ainsi que sur nos bases démontrent que les méthodes proposées donnent de bons résultats en comparant avec l'état de l'art dans ces deux domaines.

Mots clés: mise en correspondence de graphe, détection d'objets, localision d'objets, reconnaissance des activités, pairwises features, activités multiples.