EIT FAKULTÄT FÜR ELEKTROTECHNIK UND INFORMATIONSTECHNIK

Vision-Based Representation and Recognition of Human Activities in Image Sequences

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieur

(Dr.-Ing.)

von **M.Sc. Samy Sadek Mohamed Bakheet**

geb. am 12.12.1973 in Sohag, Ägypten

genehmigt durch die Fakultät für Elektrotechnik und Informationstechnik

der Otto-von-Guericke-Universität Magdeburg

Gutachter:

Prof. Dr.-Ing. habil. Ayoub Al-Hamadi

Prof. Dr. rer. nat. Christian Wöhler

Promotionskolloquium am: 17.06.2013

With deepest love, gratitude, and admiration, I dedicate this thesis to my parents, my wife, and my lovely kids, Abdel-Rahman and Tasneem.

S. Bakheet

Acknowledgement

There are a number of people and organizations
that I would sincerely like to extend my gratitude for
their assistance in completing this dissertation. Firstly,
I wish to express my deepest thank and appreciation to the
missionary spirit of Prof. Bernd Michaelis, Chair of Technical
Computer Science Group, whose constant support, encouragement,
and kind words in times of need truly made my doctoral program not
only extremely rewarding (both academically and personally), but also
a very educational yet touching experience that I will never ever forget and
will always cherish. I would like to acknowledge my debt and gratitude to Prof.
Ayoub Al-Hamadi who helped me on this journey in so many ways and has given
me the opportunity to pursue the doctoral degree from Otto-von-Guericke University
Magdeburg (FEIT), Germany. Profound thanks are due to Prof. Christian Wöhler from
Dortmund University of Technology for consenting to be the external examiner of the defense
session and for his insightful comments, questions, and suggestions on my work. My heartfelt
thanks go to Prof. Diedrich and Prof. Leone for being in the examination committee, and also to
Prof. Usama Sayed for his inspiring guidance throughout my research in Egypt. A special vote
of thanks goes to all colleagues and collaborators in IIKT, especially my office room-mates
E. Lilienblum and S. von Enzberg, and my colleagues M. Elzobi and A. Sayeed, for
always being so kind and helpful. Last but definitely not least, I am greatly thankful
and deeply indebted to my homeland (Egypt) represented by Sohag University
for granting me a scholarship to pursue my doctoral studies in Germany. In
a similar vein, special warm thanks go to my beloved parents for their
spiritual care and protection, and for their amazing love and unwavering
support. I am also deeply grateful to my wife for her endless patience,
endurance and understanding from the very beginning of my
academic endeavors, and for her ministry to our kids at home,
particularly during my stay abroad. Finally, but no less
dearly, thanks to my kids (Abd Al-Rahman and Tasneem),
who are the sunshine of my life for showing me that
life is much more than dissertations and studies.

Magdeburg, 28.06.2013

Samy Bakheet

Abstract

The overall objective of the research presented in this doctoral thesis is to explore and establish theories and methodologies for accurate representation and recognition of human actions in video data. For the methodological contributions of this thesis, multiple approaches involving diverse conceptualizations are developed to represent and recognize human actions from video sequences. Moreover, in Chapter 4, we investigate a variety of distinctive features (shape and motion features) for the representation and recognition of human actions. For our first approach, we present a new method for human action recognition based on interest point features. The main contribution of this approach is twofold. First, a reliable neural model as a classifier is employed for the task of action classification. Secondly, we unfold how the temporal shape variations of actions can be accurately described using fuzzy log-polar histogram descriptors. When tested on the KTH and Weizmann datasets, the method recognizes actions with average recognition rates of 94.3% and 97.8% respectively. These results compare very favorably with those of other investigators reported in the literature. Furthermore, due to its low computational demands, the approach can be integrated into real-time applications.

With the second approach, a Bayesian model for action recognition based on multiple cues is introduced. In a nutshell, this approach proceeds as follows. First, a series of silhouettes of moving body parts are extracted from a given video sequence (i.e. action snippet). Next, each action snippet is divided into several time-slices represented by fuzzy intervals. As shape features, a variety of shape descriptors both boundary-based (e.g., Fourier descriptors, curvature features, etc.) and region-based (e.g., moments invariants, moment-based features, etc.) are extracted from the silhouettes. Finally, an NB (Naïve Bayes) classifier is trained in the feature space for action classification. The recognition results achieved on KTH dataset

tie in with those well established in the literature. As for the third approach, an efficient methodology for action recognition is presented based on chord-length shape features. The contributions, in this work, are as follows. We first illustrate how an effective shape descriptor is constructed using 1-D chord-length functions. Second, we unfold how the process of feature reduction is performed by using Gaussian membership functions. On KTH dataset, this approach has been shown to produce recognition results which compare favorably to the state of the art.

In our practical approach towards action recognition in real-world video data, on the basis of motion vector distribution characteristics, we propose an innovative fuzzy framework to recognize actions in realistic videos. In this framework, a compact and computationally efficient fuzzy descriptor is constructed based on fuzzy directional features. For the training process, a set of one-vs.-all SVM classifiers capable to discriminate between intra-subject features and inter-subjects features is trained on the action descriptors to classify the action in the real-world scene. Due to their simplicity and low computational requirements, the employed features have proven to be amenable to real-time implementation. From a set of preliminary experiments on our dataset, we found that the feature representation parameters directly affect the recognition results. In addition, in terms of the holistic performance of the framework, the larger values of these parameters provide the greatest improvement in overall recognition rate. The best recognition accuracy achieved is 96.3%. This result can be regarded as "encouraging", when considering the realistic working environments, and it confirms the basic correctness of the approach. However, realizing more comprehensive experimental studies on larger real-world datasets is deemed to be necessary to validate the scalability and feasibility of the approach in a broader scope.

Index Terms—Human activity recognition, motion analysis, log-polar histogram, moment invariants, chord-length function, fuzzy directional features, video interpretation.

Zusammenfassung

Das Ziel der in dieser Dissertationsschrift vorgestellten Forschungsarbeiten ist die Erforschung und Etablierung einer Theorie und Methodik zur präzisen Repräsentation und automatischen Erkennung von menschlichen Bewegungsabläufen in Videodaten. Der methodische Beitrag dieser Arbeit besteht in mehreren neu entwickelten Ansätzen und diversen Konzepten zur Repräsentation und Erkennung von menschlichen Aktionen in Videosequenzen. Weiterhin werden in Kapitel 4 eine Vielzahl markanter visueller Merkmale (Form- und Bewegungsmerkmale) für die visuelle Repräsentation und Erkennung von Menschlichen Bewegungsabläufen untersucht. Als erster Ansatz wird eine neue Methode zur Erkennung von Bewegungsabläufen vorgestellt, die auf der Detektion markanter Bildpunkte (Interest Point features) basiert. Dieser Ansatz beinhaltet zwei wesentliche Beiträge. Zuerst wird ein robustes neuronales Modell für die Klassifikation von Bewegungsabläufen angewendet. Weiterhin wird dargestellt, wie zeitliche Formänderun-gen in Bewegungen mit Hilfe von *fuzzy logarithmisch-polaren Histogramm* Deskriptoren präzise beschrieben werden können. Anhand der KTH und Weizmann Bewegungsdatensätze konnte mit den Methoden eine durchschnittliche Erkennungsrate von 94,3% bzw. 97,8% nachgewiesen werden. Im direkten Vergleich zu anderen aus der Literatur bekannten Forschungsergebnissen schneiden die hier vorgestellten Methoden sehr gut ab. Desweiteren kann der Ansatz aufgrund seiner geringen Rechenanforderungen in Echtzeitanwendungen integriert werden.

Im zweiten Ansatz wird ein auf mehreren Merkmalen basierendes Bayes'sches Modell für die Klassifikation menschlicher Bewegungen vorgestellt. Hierbei wird zunächst eine zeitliche Folge von Umrissen der bewegten Körperteile aus dem Video extrahiert. Anschließend wird jedes Bewegungsvideo in Abschnitte mit unscharfen Intervallgrenzen zerteilt. Eine Vielzahl von Formdeskriptoren, die sowohl

konturbasiert (z.B. Fourier-Deskriptoren, Krümmungsmerkmale) als auch regionenbasiert (z.B. invariante Momente, momentenbasierte Merkmale) sind, werden aus den Umrissen extrahiert. Abschließend wird ein naiver Bayes (NB) Klassifikator für die Bewegungsklassifikation im Merkmalsraum trainiert. Die Erkennungsergebnisse auf dem KTH Datensatz sind vielversprechend und belegen die Wirksamkeit dieses Ansatzes. Für den dritten Ansatz wurde eine effiziente Methodik zur Erkennung von Bewegungsabläufen durch auf Sehnenlängen basierenden Formmerkmalen entwickelt. Hier wurden die folgenden Beiträge geleistet: Zunächst wird ein effektiver Formdeskriptor mit eindimensionalen Sehnenlängen Funktionen konstruiert. Anschließend wird eine Merkmalsreduktion mithilfe von gauß'schen Zugehörigkeitsfunktionen (Gaussian Membership Functions) durch geführt. Im Vergleich zu anderen Methoden aus dem aktuellen Stand der Technik zeigt der Ansatz bei Anwendung auf dem KTH-Testdatensatz gute Ergebnisse.

Als praxisorientierten Ansatz für die automatische Erkennung von Bewegungsabläufen in realen Videodaten schlagen wir desweiteren ein innovatives Fuzzy-Framework vor. In diesem Framework werden unscharfe Richtungsmerkmale als Deskriptoren verwendet, die kompakt und wenig rechenintensiv sind. Im Trainingsvorgang werden diese Deskriptoren mit einem Satz von „iner-gegen-Alle“ SVM Klassifikatoren trainiert, die zwischen Intra- und Inter-Klassenmerkmalen unterscheiden können. Aufgrund ihrer Einfachheit und ihres geringen Rechenaufwands sind sie für eine Echtzeitimplementation geeignet. Anhand von Voruntersuchungen an unserem realen Datensatz konnte festgestellt werden, dass die Erkennungsraten direkt von den Parametern für die Merkmalsrepräsentation abhängen. Im Sinne der ganzheitlichen Leistungsfähigkeit unseres Erkennungssystems konnte weiterhin gezeigt werden, dass mit höheren Parameterwerten die größten Verbesserungen der Gesamt-Erkennungsrate erreicht werden. Mit diesem Ansatz konnten maximale Erkennungsraten von 96,3% erzielt werden, die in Anbetracht der realen Arbeitsumgebung als vielversprechend eingeschätzt werden können, und die grundsätzliche Richtigkeit des Ansatzes beweisen. Dennoch sind umfangreichere experimentelle Studien auf größeren echten Datensätzen notwendig, um die Skalierbarkeit und Realisierbarkeit in breiterem Rahmen zu validieren.

Declaration of Authorship

I hereby declare that this dissertation entitled *"Vision-Based Representation and Recognition of Human Activities in Image Sequences"* is the result of my own research except for the references cited which I have duly acknowledged. It has not been presented anywhere in part or in full for the award of any degree.

In addition, some of the material contained in the thesis has been published in peer-reviewed journals or proceedings of international conferences/symposia. A list of these publications can be found at the end of the dissertation.

Magdeburg, 28.06.2013

Samy Bakheet

Table of Contents

List of Figures

List of Tables

Acronyms and Abbreviations

Nomenclature	Description
ML	Machine Learning
ANNs	Artificial Neural Networks
MSNN	Multi-level Sigmoidal Neural Network
SVMs	Support Vector Machines
DBNs	Dynamic Bayesian Networks
NB	Naïve Bayes
HMMs	Hidden Markov Models
CHMMs	Coupled Hidden Markov Models
CHSMMs	Coupled Hidden Semi Markov Models
SOM	Self Organizing Maps
PCA	Principal Component Analysis
LDA	Linear Discriminant Analysis
pLSA	Latent Semantic Analysis
SIFT	Scale Invariant Feature Transform
HOF	Histogram of Optical Flow
HOG	Histogram of Oriented Gradients
HCI	Human-Computer Interaction
HHI	Human-Human Interaction
MEI	Motion Energy Image
MHI	Motion History Image
MACH	Maximum Average Correlation Height
GMMs	Gaussian Mixture Models

Nomenclature	Description
EM	Expectation Maximization
MAP	Maximum A Posteriori
QP	Quadratic Programming
KTH	Kungliga Tekniska Högskolan (in Swedish)
FDs	Fourier Descriptors
ASL	American Sign Language
MoG	Mixture of Gaussians
CBIR	Content-Based Image Retrieval
DTW	Dynamic Time Warping
LTI	Linear Time Invariant
pdf	Probability Density Function
fps	frame per second
CoG	Center of Gravidity
BP	Backprogration
ROI	Region of Interest
VC	Vapnik-Chervonenkis
VA	Viterbi Algorithm

CHAPTER 1

Statement of Problem

1.1 Preface

SINCE the very birth of human civilization and rational thinking, the understanding and explanation of the behavior of everything in the physical world around the human being has been one of the most intractable long-studied problems. In the first place, the focus of the attention was primarily upon the most transcendent "physical phenomena" of the external world, such as the sun, the fire, the sky, the air, the moon, the day and the night, the directions, the water, the land, etc. Most of these phenomena were initially interpreted as divine actions. However, with the lapse of time (over many centuries), the human civilization was being highly evolved scientifically and technologically and the first and most rudimentary "dogmatism" were taken out and dusted. Therefore natural sciences so-called "exact sciences" (such as Physics and Mathematics) have been adopted as tools to explain, analyze, and aid in better understanding of these phenomena occurring around us. This not only enabled to solve many difficult problems with better and better models, but also contributed to approach their solutions flexibly.

Regarding the inanimate objects so-called "non-living" objects, how they behave inside the space and the consequences of their interactions have been understood accurately thanks to the achievements of the minds of giants, such as Newton, Euler, Einstein, among others. While on the contrary, the existence of emotions and consciousness has contributed an additional level of challenge and complexity to the process of understanding the actions of living beings. To contribute to this, and in addition to the deep investigation and detailed analysis of their physical

properties, serious efforts have been put into finding the inward electrical impulses that make these beings behaving in certain determinate ways. As a point of fact, it was found that the survival instincts most drive the "non-rational animals" (i.e., all the living beings, excluding humans) and control their behaviors and movements. Survival instinct, grossly and loosely speaking, is the most powerful instinct the animals have; it contains a series of actions (constructed or designed as an internal memory) that are responsible for how these animals feed, reproduce, and raise young, and how they protect their survival.

It is now biologically established that, in light of Darwin's theory of the evolution of species and the subsequent history of Darwinism, most actions and biological functions of animals have evolved over millions of years in both size and complexity through natural selection and adaptation to best adapt to the bad survival environment, maintaining life and continuing species. Additionally, throughout their lives, animals gather and weigh information to decide upon alternative states. For human beings, the cognitive capacity for the language communication, abstract thinking, and action planning is unique to them and not shared with the rest of animals. On one side, abstract thinking enables humans to think, conceive ideas, and and act rationally. On the other, humans can plan ahead and consider the long-term consequences of behaviors, choices, and preferences. Together, these two features pose serious roadblocks to any advance toward recognizing and/or predicting the behaviors that can be performed by human beings.

Recently, a new interdisciplinary field of study within computer science has emerged, concerned with the theoretical and empirical investigation of many of the previously mentioned capabilities. This new discipline comprises techniques from various fields such as, image processing and analysis, computer graphics, artificial intelligence, pattern recognition, robotics, etc. With this new field, an image sequence concerning human-populated scene can be modeled accurately. The rapid advances in the hardware technology of media capture, storage, and computation power have contributed to impressive developments in image understanding techniques and applications. Moreover, numerous technical contributions and software implementations have been possible by the newly emerged capabilities.

1.2 Challenges & Obstacles

In this section we briefly review the main challenges and setbacks that we face when dealing with human action recognition in video data which confront the design and implementation of any successful action recognition model. Then, we show how such challenges have a considerable impact on the ability of current recognition

approaches to effectively recognize actions. We begin our discussion by stating that, in fact, the problem of action recognition shares many challenges with other problems, such as object detection and tracking, motion recognition, etc.

1.2.1 Common challenges on activity recognition

The main common challenges in human action recognition shared with other problems in computer vision include:

- Illumination conditions
- Occlusions
- Clutter in background
- Object deformations
- Intra/inter class variations
- Pose variations
- Camera point-of-view

The next paragraphs will briefly describe each one of these challenges and setbacks to identify specific aspects of the task being tackled in the study. First, changes in illumination conditions are one of the major difficulties that current object detection/recognition and tracking techniques are confronted with. In other words, it is a serious problem that affects both holistic techniques and methods based on some spacial feature representation. This is due to the fact that the same object is differently perceived under different illumination conditions. For occlusion, it is a surprising capability of visual perception, still unmatched by computer vision algorithms. Plainly speaking, occlusion means closer objects block more distant objects from being viewed. Therefore object recognition and tracking under occlusions is a "bottleneck" problem in computer vision, not yet fully solved.

Background clutter, like occlusion and changing illumination, is also a realistic challenge for activity recognition and other pattern recognition tasks, that interferes with the capture of the vital information. As a result, human activity recognition in the presence of heavy background clutters seems to be a most difficult problem indeed. In addition to that, the probable drastic change (deformation) in the shape of object between two consecutive frames also provides additional challenge to the task of developing a robust model for activity recognition. Large "intra-class" variability arises when the visual differences amongst action instances belonging

to the same class are quite significant. That makes correct classification of actions appear more difficult to be achieved successfully. In this case, a good model for activity recognition should have the ability to learn the features that allow the different instances of action to be members of the same class.

In addition, a robust activity recognition model should be able to distinguish amongst actions of different classes, when the visual differences between them are very slight. In this case, the low "inter-class" variability arises, which makes classifying actions correctly an arduous task, as well as the large intra-class variability does. In relation to what has been stated before, pose variations provide also a significant handicap for activity recognition. Thus an action recognition model should be designed such that it can handle pose variations robustly as well as to other challenges. The camera viewpoint where the scene is taken determines the parts of body that will be visible. Furthermore, different viewpoints of the camera can cause some portions of body to be visually occluded. Hence, a robust model for action recognition should be designed to take into account different views. For deformable objects (i.e., action poses), while they are arising from the relative position of their own parts, their constitutive parts can cause occlusions to be originated. Add to that, the different appearances of articulated objects make hard for shapes to be learned correctly. In everyday visual experience, objects rarely exist in isolation. Instead, they usually appear as part of multiple objects. This raises the challenge of distinguishing the features of an object corresponding to the foreground from those of other objects corresponding to the background.

1.2.2 Specific challenges on activity recognition

While human vision has an extraordinary ability to efficiently recognize human actions from video data, with a high degree of accuracy, it is an arduous task for the computer to do such a task in a very similar manner. First, the fact that the same action is performed by different people at different velocity poses a quite significant technical problem to automatic system to achieve activity recognition task efficiently. Moreover, moving shadows generated by bad lighting conditions can also degrade tracking the motion of human body parts. Some body parts can be occluded owing to camera viewpoints that provides an additional difficulty to the human activity recognition task. Added to that, small moving objects (i.e., distractors) in background are also another problematic issue for this work. For example, in a scene of crowded street, trees swinging and/or shop advertisements blinking in background are challenging issues for motion detection and tracking.

FIG. 1.1. Sample video shots belonging to nine categories of human actions often seen in movies; from left to right, and from top to bottom: Hugging; Exiting From A Car; Beating; Playing Football; Shaking Hand; Kissing; Drinking; Answering A Phone; Smoking. These samples were collected from various websites.

1.3 Motivations & Applications

In everyday life, one carries out successfully many high-level tasks, beginning with object detection and motion recognition, ending with avtivity/event recognition and scene interpretation. For example, as human beings and not other kind of animals, we have the ability to find out where a favorite book or a pen is. When we are on the street, we have the capacity to detect easily the where of traffic signs or other signals of the police superintendent. It would be easy to argue that this capability is not limited to finding (detecting) an object of interest (a target). It also includes identifying that object. This would imply that we are able to decide which car we own, from those parked in a parking lot. On top of that, we can discriminate a close personal acquaintance from others within a large area crowded of people. Further, using a small amount of examples, we are effortlessly able to learn new categories of objects as well as new autonomous instances of them.

Since the 1950's, following consecutive advances in video technologies in both hardware (e.g., affordability of low-cost cameras and high quality video webcams) and software (e.g., video editing software), the amount of digital video data has grown rapidly and immensely for various usages in many areas, such as advertising, news video broadcasting, concerts and sporting events, personal video archive, medical video data, etc. Owing to the recent publicity of the World Wide Web (WWW) and other Internet services, immense video data are accessible online and available for sharing. For example, five years later a large number of videos are uploaded on YouTube every day, at a rate of more than 20h of video per minute. In other words, it would take nearly a full day to watch all the video posted to YouTube in a single minute. Interestingly, human actions represent the majority part of these huge voluminous videos. Fig. 1.1 shows some examples for these actions collected from different Web sites.

Recently, pose detection and human action recognition have gained more interest among video processing community because they find various applications in which the use of them could play a significantly beneficial role, both in term of productivity and quality derived from new software engineering tools. Some prominent examples of such potential applications include:

- Video search and indexing for browsing
- Human-computer interfaces
- Analysis of sport athletics and dance choreography
- Film and television archive analysis
- Real-time active object monitoring for video surveillance
- Telemonitoring of chronic patients and elderly people
- On-line pedestrian detection and tracking for smart vehicles

Examples for these applications are shown in Fig. 1.2. Current Web search engines offer convenient ways to access and to retrieve huge amounts of video data. Owing to the high complexity of video data, efficient content-based video retrieval from a large database with respect to a specific user's query calls at least for: compact and effective video representations, efficient similarity metric, and efficient indexing on the compact representations. For example, content-based image retrieval (CBIR) from large databases for medical and civilian research, and planning purposes has a broader technological impact on the society and the daily

FIG. 1.2. Various compelling applications for automatic activity recognition, for instance in the areas of automated security, financial management, robot learning and control, video surveillance, healthcare, and user interface design.

life. Likewise, the ability of describing a scene through the objects involved in it is very advantageous for manipulating and interpreting that scene in useful ways.

Also security surveillance (by installing surveillance cameras and red-light signal violation cameras at various junctions) is other emerging application that is employed to detect the presence of weapons and explosives in many strategic places such as, airports, seaports, banks, railway stations, hospitals, and government buildings. Access security systems that control access by applying an automated Turing [1] test, are widely used to exclude people of vision impairments by presenting them to a visual exam that they cannot pass. It is important to note that, for all the systems mentioned so far, they have to be designed such that they meet the requirements of robustness and real-time performance, since the performance is critical and foremost in these applications.

However, the problems such as object modeling and motion recognition and tracking that relate closely the problem of human activity recognition (i.e., the main focus of this thesis), are still challenging and pose difficulties for general real-world scenes, and thus much work remains before we will see the approaches proposed for them in a mature stage. In other words, yet there are no definitive solutions that achieve satisfactory results in order to overcome such difficulties. This suggests that it is still necessary to find more efficient solutions to solve the issues that need

[1] Turing is an English mathematician whose works explored the possibility of computers and raised fundamental questions about artificial intelligence.

further research efforts to better understand problems and to develop better and more appropriate methodologies. That is not to say that human activity recognition is a solved problem, since most of the existing approaches only address one or a few of the specific challenges or difficulties. This is why the problem is still an open problem, and, to the best of our knowledge, research work involved in this topic is still in its infancy or far from mature.

1.4 Human Action Recognition: An Overview

Human action recognition is one of the most extensively studied problems in computer vision. However, the approaches to the problem are still far from satisfactory and very specific to dataset at hand. Roughly speaking, there are three major steps involved in any approach to solve this problem:

1. **Segmentation**: A simple approach (i.e., frame difference) to achieve this objective works as follows: First, the background is estimated to be the previous frame. Then the estimated background is subtracted from the input frame. A threshold is applied to the absolute difference to obtain the foreground mask. Depending on the object structure, speed, frame rate and global threshold, this approach may or may not be useful. For example, although such a technique does not need any knowledge about background and is very adaptive to dynamic environments, it suffers from the so-called foreground aperture problem due to homogeneous color of moving object [9]. Therefore it often fails to detect all moving pixels. A reliable and robust background subtraction algorithm should handle: 1) sudden or gradual illumination changes; 2) high frequency, repetitive motion in the background (e.g., tree leaves, flags, waves,...); and 3) long-term scene changes (e.g., a car is parked for a month).

2. **Feature selection and extraction**: In this stage, the features of an activity and how they change with time are obtained and then analyzed. First,The representative descriptors of features are obtained from video sequence at each time interval. The choice of the features that make up the feature vector is an important design decision in the subsequent feature classification module of entire solution. Thus, features should be chosen such that they are quite representative of the objects of interest as well as the activity being recognized. Furthermore their descriptors have to be a quite compact, accurate and efficient way of extracting and representing these features. Regarding the features extracted from video sequence, they could be low-level primitives with very

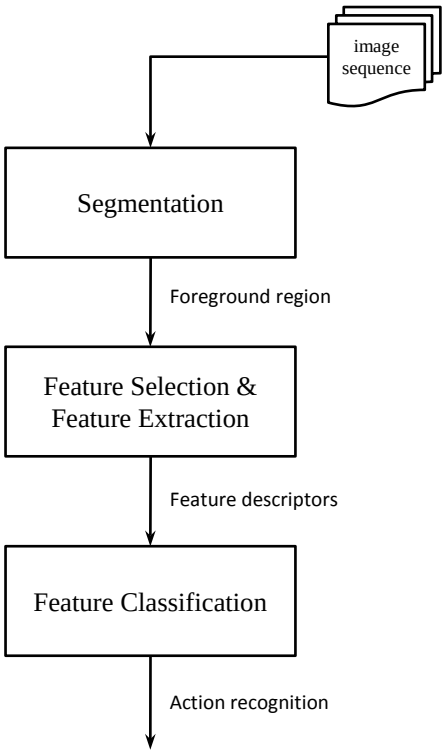


FIG. 1.3. A general overview of activity recognition system. Each of the three major blocks can in turn contain sub-blocks.

little descriptive characteristics, such as contour length, aspect ratio, etc. or high-level descriptive features, such as the position of person's hands/legs, label of the object etc. In an effort to narrow the search for an activity in the video sequence, as well as to improve the recognition accuracy of the actions themselves, Some researchers opt for making use of non-visual features such as closed caption text [10], and sound [11].

3. **Feature classification**: After doing feature extraction and selection, various machine learning models could be applied to classify the features extracted in the previous stage. Some of these models or classifiers include Neural networks [12,13], Hidden Markov Models (HMM) [14,15], Belief networks [16], or rule networks [17,18]. It is worth mentioning here that the choice of which classifier to use is a matter of personal discretion within the constraints of computational tractability and the size of the dataset being worked on.

We would like to draw the reader's attention that each of the steps outlined above will be described and discussed in more detail in the following chapters. The overall sketch of these steps is presented as a block diagram, shown in Fig. 1.3.

1.4.1 Requirements

The current development of computer vision algorithms along with the rapid advancements in hardware technologies make it possible for human activity recognition to be achieved reliably and in real-time. However, many existing activity recognition approaches have their own numerous inherent limitations, such as accuracy, speed, robustness, etc. The prime motivation of this thesis is to contribute to the state of the art in the development and improvement of vision-based modeling and recognition of human actions from video sequences. To achieve this goal, three various approaches are presented for representing and recognizing human actions in videos. With the presented approaches, we attempt to allow the following assertions to be applied to the action recognition process.

- **Reliability**: In order to assess the high success rate and reliability of action recognition performance, the evaluation results should confirm that a high recognition rate can be achieved, while maintaining low false alarm rates. This means that activities can be recognized more effectively and accurately with least confusions between each two different activities. Furthermore, that reliability or efficacy of the technique should be comparable to other well-established state-of-the-art techniques in literature in terms of recognition rate and number of false alarm properties.

- **Real-time performance**: For the real-time performance of automatic action recognition, the different steps or tasks are required to be done in real-time or near real-time. First, fast feature segmentation (i.e. background subtraction) is done using Mixture of Gaussians (MoG). In addition, subsequent higher-level tasks, such as feature extraction and/or selection, feature tracking, and classifier fusion are also performed in real-time. It is worthwhile mentioning that a real-time action recognition system is capable to provide latency guarantees for real-time applications and embedded systems.

- **Robustness**: In practical use, it is desirable for action recognition systems to be robust to illumination variations, partial occlusions, cluttered background, and noisy silhouettes. An ideal action recognition algorithm should also show tolerance to changes in scale, rotation, and viewpoint while being fast to implement. Moreover, a technique for action modeling and recognition should be able to robustly model various spatio-temporal variations .

- **Scalability**: Generally speaking, scalability of a recognition method implies that the method has the potential to learn to recognize additional pattern

vocabularies without decrement in performance. The presented approaches for action recognition can demonstrate good scalability of recognition rate with size of datasets. In addition, the ideal action recognition system should be able to deal equally with large and small action vocabularies. As conditions comes to be worse, the system performance begins to decrease gradually. The simpler out of two systems performing equally well is better.

- **Human-independence**: The ideal action recognition system should have the potential to model and recognize various human activities independently of their agents. Such a system should also deal with human activities with different shapes, trajectories and durations.

1.5 Goals and Contributions of Thesis

The main objective of this thesis is to contribute to the problem of human action recognition in video data by developing a new approaches for recognizing human actions from video sequences. Since the perceived meaning of some action highly depends on cultural factors and the specific context, the main contribution of the work is concentrated on recognizing actions from video sequences, rather than the high-level semantics of these actions. The new approaches should be robust enough for handling noisy video data which is the case of real-world data. This consequently allows the proposed approaches to improve their qualifications and ability to be most appropriate and feasible for a vast majority of real-world applications, such as surveillance and public security systems, healthcare settings, human-computer interfaces, intelligent vehicle control, etc. Furthermore, real-time processing is also a major issue and very necessary to support applications with real-time requirements. Broadly speaking, the approaches meet the following constraints and/or performance characteristics:

- **Real-time application:** As will be detailed later in the subsequent chapters, the implementations of the presented approaches are based on 2D image data, rather than 3D world data. Therefore, they require much less computing resources in processing time and memory usage than many other complicated and sophisticated approaches in the literature. In other words, the proposed approaches can provide timing guarantees to real-time applications.

- **Stationary camera:** In many video applications (such as such surveillance and health-care systems), stationary cameras are typically used to monitor critical areas to prevent access by unauthorized personnel or monitor activities in

health care facilities. All experiments of this work were carried out using this category of cameras that provides a good quality image with a low distortion. Nevertheless, the approach is not restricted by this constraint. Therefore, with an appropriate human detector, a portable camera can also be used.

- **Mono-camera application:** For realizing visual recognition and tracking of people and their actions, a monocular camera can be enough. To avoid redundancy (by installing various cameras), only one camera is manually fixed in position to record a scene from a specific viewpoint.

For better usage of the approaches, the following hypotheses are also postulated:

- The proposed approaches can recognize actions of two publicly benchmark datasets (i.e., KTH [7] and Weizmann datasets [19]). Examples of these actions include, running, walking, jogging, hand waving, etc. Furthermore, the presented methods can theoretically be extended to recognize more challenging realistic actions, such as Hollywood dataset [20].

- Based on the datasets of this study, in a scene, there is only one action performed by a single person. But, it is possible to build a person detector to detect all people in a scene. Then, a person tracker can be used to track all persons in the scene. Once persons are set apart, recognition can be applied to each person independently in the scene in order to detect relevant actions.

- The developed approaches can work well under a reasonable image resolution and frame rate. Since 2D image data have been implemented, a minimal frame-rate (e.g. 25 fps) is needed to capture most of motion. Also, the image resolution should be taken in agreement with the distance of the targeted person from the camera. Hence, the shape of body parts can be perceivable.

- The current research is focused on detecting actions performed by a single person, and not interaction actions (e.g., handshake, kiss, hug, etc.). Nevertheless, it is possible to extend the proposed techniques to detect more complex actions. Furthermore, regarding crowd scenes, once a robust person tracker is available, the methods can handle such actions as well.

The contributions of this dissertation can be summarized as follows:

- Various vision-based theories and methodologies are attempted to accurately represent and recognize human actions from video data.

- Multiple approaches based on diverse conceptualizations are suggested for the vision-based representation and recognition of human actions in videos.

- A variety of distinctive visual features are investigated and developed (i.e., shape features and motion features) for the vision-based representation and recognition of human actions.

- An innovative fuzzy recognition framework is proposed to represent and recognize human actions in real-world videos.

1.6 Overview of the Manuscript

This dissertation is structured around seven main chapters, excluding this introductory chapter, which are summarized as follows

- **Chapter 2** aims mainly at providing an up-to-date state-of-the-art picture on human action recognition techniques. The different representations and models of activities are discussed. Depending on how human activities are modeled and represented, recognition approaches are broadly categorized into one of two types: (1) spatio-temporal approaches and (2) sequential approaches. Spatio-temporal approaches fall again into three sub-categories: (1) volumes themselves, (2) trajectories, and (3) local interest-point descriptors. On the other hand, according to their recognition methodologies, sequential approaches are categorized into two major types: (1) exemplar-based approaches and (2) model-based approaches.

- **Chapter 3** discusses the major approaches for segmenting image sequences to objects that is regarded as a curial step in scene understanding and action modeling. These approaches include: frame differencing, motion segmentation, and background estimation and subtraction (e.g., mixture of Gaussians). This chapter is divided into two major parts. In the first part, we attempt to provide an overall idea about image segmentation in general. Then, a new method for image segmentation is presented based on the concept of the Rényi entropy. The second part is devoted to the video segmentation process as a main part in developing any object-based video recognition system in general, and in developing the human action recognition system in particular.

- **Chapter 4** details the process of the extraction of action descriptors. We present a detailed description of various features and descriptors developed in our works on action recognition. These features broadly include interest-point based features, shape border based features, and chord-length function features. A careful analysis/investigation of these features has suggested that

they turned out to have the potential to provide a rich source of information for the interpretation/analysis of human activities.

- **Chapter 5** concerns the description of the learning-classification framework for action recognition. We discussed three of the most widely used and most influential machine learning algorithms (i.e, ANN, SVM and NB) that were trained and tested separately for the action features described in Chapter 4. Moreover, an adaptive neural model (i.e., Multilevel Sigmoidal Neural Network) is used, which is developed to relax the restriction of the traditional neural model and to allow the neural units to generate multiple responses.

- **Chapter 6** commences the discussion by providing an overview of two publicly benchmark action datasets (i.e., KTH [7] and Weizmann [2]) on which the outputs of this written research are based. Then, we illustrate how the experiments were conducted in detail in this work. Finally, the obtained results are compared with those of similar recent state-of-the-art methods.

- **Chapter 7** begins with the description of our real-wold action dataset that we use for this research, and, followed by giving some interesting characteristics of this dataset. It proceeds to give a detailed description for our proposed framework for action recognition in real-world video data. Then, the results from a set of preliminary experiments conducted to evaluate the stability of the recognition system and its effectiveness in recognizing actions are reported. Finally, in the last section, implications of the results are discussed and conclusions drawn.

- **Chapter 8** is a short concluding chapter that contains two sections. In the first section, the key contributions of the thesis are summed up, and some conclusions from the preceding investigations and experiments are drawn. In the light of the drawn conclusions, some possible directions for future research within this area, either as an extension of the theory presented in this thesis, or as an alternative are suggested in the second section.

CHAPTER 2

State of the Art

2.1 Introduction

In this chapter, we aim to provide an overview of recent studies concerned with human action recognition. The methodologies of both simple activities and high-level activities are presented. To learn more about the advantages and disadvantages of each of these methods, we have opted to use an "approach-based" taxonomy in categorizing and displaying such methodologies. First, we present various recognition methods developed to recognize simple activities performed by a single person in video sequences. Both spatia-temporal volumetric approaches and sequential approaches to recognize such simple actions are discussed. Then, we present and discuss various hierarchical methodologies including syntactic, description-based, and statistical approaches. Moreover, various approaches that have been attempted to recognize group activities and human-object interactions are mentioned. Finally, we also show some public action datasets that have been used in the performance evaluations and comparisons of these methods.

Recognizing human actions in videos is an important area of computer vision research today, but a challenging task that has gained a lot of attention during the last decade [8]. Generally speaking, human action recognition aims at recognizing or identifying automatically the action performed in a video sequence. In the case where a video sequence includes only one action performed by a single person, the foremost task here is to correctly classify which type of action the video exhibits. For a more general case where a video clip contains continuous actions performed in sequence, the start and end times of each occurring action in the video clip should

be detected. It is very important to establish that the recognition of complex human actions from video sequences would provide significant engineering potentials for the many paramount applications. For example, the automatic recognition of abnormal or suspected actions as opposed to normal actions is urgently needed by surveillance systems located in densely populated places, such as airports, railway stations, and government buildings. In all such cases, "suspected" actions, such as 'someone putting a bag in a rubbish bin' or 'someone leaving a suitcase behind' should be detected by a security monitoring system in an airport. With successful action recognition, the real-time monitoring of patients and elderly people that represents a cornerstone of any successful treatment, would be practically achievable or viable. In addition, efficient action recognition can support the design and implementation of vision-based intelligent environments, and can also lead on step ahead to more natural Human-Computer Interactions (HCIs).

Based on their complexity, human activities are conceptually divided into four distinct categories: gestures, actions, interactions, and group activities. Basically, a gesture is a primitive movement using the body or body part that conveys information, such as "shaking a head around", "shake a head back and forth", "shrug shoulders", etc. An action is an activity composed of multiple temporally-organized gestures. Typical examples of actions include "jugging", "running", "walking", etc. Human activities that comprise two or more persons and/or objects are called "Interactions". While "Two humans fighting" is an example for a human-human interaction between two humans, "a person stealing a bag from another" is another example for human-object interaction containing two humans and one object. Finally, human activities done or managed by conceptual groups containing multiple humans and/or objects are termed "group activities". An examples of these activities is "a group holding a meeting". ln this context, it would not be irrelevant to point out that in the thesis context from now on the term "activity" is used synonymously to the term "action", as these two terms are frequently used generically and interchangeably in activity (or action) recognition community.

2.2 Literature Review

In this section, various types of methodologies in the state-of-the-art human action recognition are reviewed. In this regard, different approaches that have been developed to recognize different levels of human actions, are also discussed. In [21], Aggarwal et al. review several fundamental low-level constituents required for understanding human motion (i.e. body posture analysis and tracking). Their

review put a stress on the recognition and analysis of simple motions and gestures. For detecting and recognizing the human activities of complex structures, methodologies of motion analysis do not appear to be quite sufficient. Similar to [21], methodologies of human action recognition are hierarchically categorized into various levels. At first, action recognition methodologies fall into one of two categories: (1) hierarchical approaches and (2) non-hierarchical approaches.

Regarding hierarchical approaches, depending on the recognition methodologies used, these approaches can be broadly divided into three categories, namely, (1) statistical approaches, (2) syntactic approaches, (3) and description-based approaches. Statistical approaches attempt to recognize high-level activities by building statistical state-based models hierarchically composed of other models [22–27]. One widely publicized prominent example of such approaches is that developed by Oliver et al. [28]. In a similar way, for modeling and recognizing sequential human activities, syntactic approaches employ a grammar syntax (e.g., stochastic context-free grammar (SCFG)) [29]. With these approaches, a high-level human activity is essentially modeled by a sequence of other low-level activities [29–33]. In a not altogether dissimilar way, 'description-based' approaches attempt to model high-level activities by defining them by their spatial, temporal, and logical relationships [34,34–40]. On the other hand, non-hierarchical approaches have the potential to directly represent and recognize activities based on image sequences. By their nature, such approaches are most qualified and wellpositioned with the task of recognizing short-term activities of sequential characteristics, such as gestures and simple actions. Hierarchical approaches, on the other hand, have a layered structure where high-level activities are represented in terms of other simpler low-level actions, so-called sub-events'. Due to their nature, activity recognition models of multiple layers are well suited to analysis complex long-term activities.

In non-hierarchical approaches, activities are straightforwardly recognized from input video sequences. Essentially, a human action is deemed by these approaches to be as a specific category of image sequences. Thus the task of recognizing the action from a given video sequence is the same as classifying the video sequence into its category. Such approaches appear to be the most suitable choice when a sequential pattern of an action can be extracted from training data. In addition, single-layer approaches, due to their nature, have proved to be most appropriate and effective for analyzing relatively simple "short-term" human motions (e.g., running and walking). To enable an activity recognition system to determine reliably whether an activity is likely or unlikely to occur, multiple representation and matching algorithms have been previously suggested. As for the recognition from continuous video data, a sliding window technique is commonly adopted by

most non-hierarchical approaches in order to categorize each likely sub-sequence.

Depending on how human activities are modeled and represented, all non-hierarchical approaches are again categorized into one of two types: (1) spatio-temporal approaches and (2) sequential approaches. While a given video is regarded as a 3-D (XYT) volume by spatio-temporal approaches, it is considered as a sequence of observations by sequential approaches. Based on which features are picked from the 3-D spatio-temporal volumes, spatio-temporal approaches fall again into three sub-categories: (1) volumes themselves, (2) trajectories, and (3) local interest point descriptors. On the other hand, according to their recognition methodologies, sequential approaches can be categorized into two main types: (1) exemplar-based approaches and (2) model-based approaches.

Sequential approaches deem an action as a sequence of observations. For this view, an activity is represented by a sequence of feature vectors picked up from video data; thus by searching for such sequence, the activity can be recognized. On the other hand, for the spatio-temporal approaches, an input video sequence representing an activity is typically treated either as a 3-D spatio-temporal volume or as a collection of features picked up from the volume. The spatio-temporal volumes are created by the concatenation of consecutive frames along time line. The following sections will be devoted to reviewing and discussing the non-hierarchical approaches which are closely related and pertinent to the work presented in this dissertation. Fig. 2.1 presents a schematic overview of these approaches.

2.3 Spatio-temporal Recognition Approaches

The pioneering work dates back to 1985 when Aldelson and Bergen [41] first proposed the concept "spatio-temporal volume", in which motion models are dependent on energy and response to filters. Since then, the spatio-temporal volumes have been predominantly exploited by a wide range of image processing algorithms, not only for performing segmentation of dynamic scenes, but also as a cue for inferring depth information of static scenes. When a 3-D (XYZ) real world scene is projected onto a 2-D (XY) image plane, one dimension of scene information is lost. In this case, 2-D spatial information of the scene, such as spatial layout, shadow, and shape, are maintained. From computer vision point of view, a video sequence is viewed as a collection of consecutive 2-D images. Therefor, by stacking 2-D (XY) images along time axis (T), a specific 3-D (XYT) spatio-temporal volume can be constructed, by which a human motion visually occurring within the video sequence can be represented, and thus analyzed using a variety of strategies. The pivotal idea on which the spatio-temporal approaches are originally founded is the analysis of 3-D

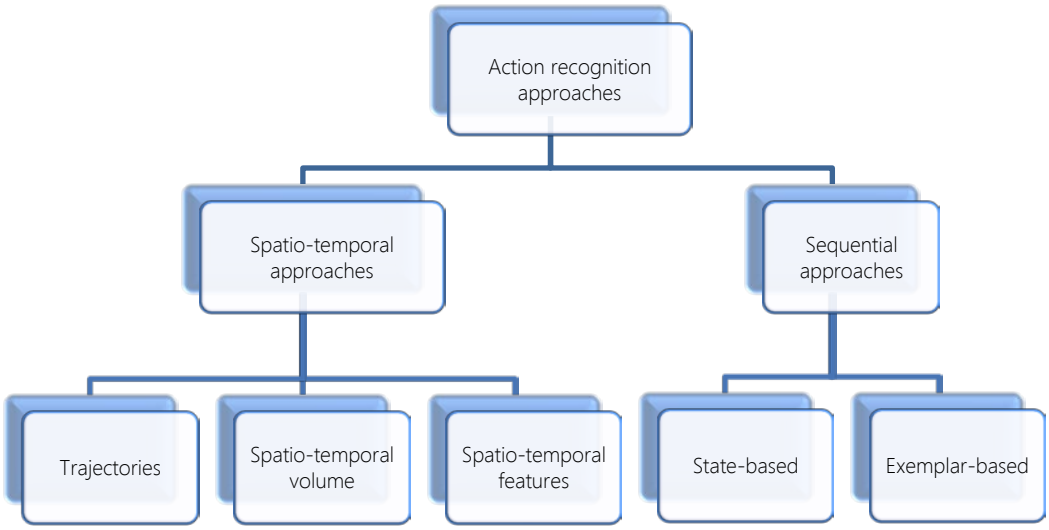


FIG. 2.1. Synopsis of various approaches to human action recognition.

spatio-temporal volumes of actions. Strictly speaking, a spatio-temporal approach for action recognition typically involves four main steps:

1. Given the video data as a training set of labeled action instances, a 3-D spatio-temporal volume (i.e. activity model or template volume) for each activity is constructed and maintained.

2. For each new unlabeled video provided, a 3-D spatio-temporal volume representing the action occurring within that video is also constructed.

3. At this time, the similarity between the action within the new video sample and each existing action action category is calculated by computing any dissimilarity measures (e.g., Euclidean distance, Pearson's correlation coefficient, etc.) between their representative template volumes.

4. The ultimate goal can now be fulfilled by assigning the action category with the highest similarity value to the video involving the considered action.

Typically, the above scheme shows a spatio-temporal methodology for recognizing human activities from videos based on the 3-D template volume and the template matching algorithm. Example 3-D XYT volumes constructed from a "running" action are demonstrated in Fig. 2.2. As a matter of fact, there are several close alternatives to the pure 3-D volume, that can be used in representing an action, such as trajectories or a set of features extracted from the volume or the trajectory of the action. For the first case, instead of representing an action with a volume, it

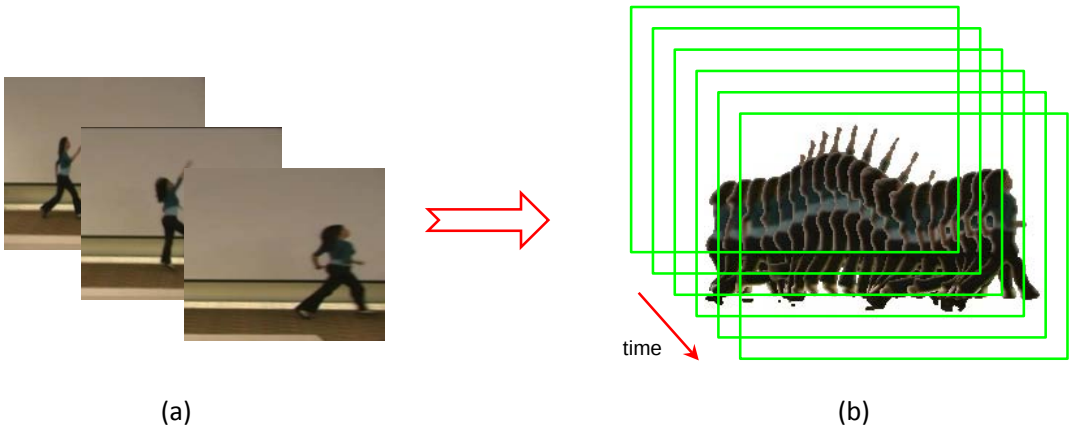


FIG. 2.2. Spatio-temporal volume for a "run-jump" action: (a) original video sequence and (b) 3-D XYT image volume.

can be represented by a set of features extracted from the volume itself. The key idea behind that is that 3-D volumes can be treated as rigid objects, hence their representations can quite be constructed by extracting general patterns from them. For the second case, a human action is represented as a set of trajectories in a spatio-temporal space or other spaces. When some feature points (e.g. estimated localities of human joints), are conveniently trackable, a set of trajectories can be used to effectively represent the occurring motion. In recent years, great efforts have been devoted to develop various saptio-temporal recognition algorithms. Much of the focus of these efforts have focused on developing approaches for matching trajectories, volumes, or a set of features extracted from each of them. Template matching is a good example of such approaches, in which a volume (i.e. representative model) for each action is created using training video sequences. An unseen action is recognized by matching its volume with a known set of volumes for each possible activity. One of the more widely used matching approaches is 'Neighbor-based Matching'. In this approach, a set of volumes or trajectories describing an activity is maintained. A new activity is recognized by matching its volume with all (or a portion) of those volumes maintained beforehand. Eventually, several statistical algorithms that specify an explicit probability distribution to model the activity have recently been used successfully in the treatment of video matching.

2.3.1 Volume based action recognition

As stated so far, detecting spatio-temporal correlations among the spatio-temporal patterns of activities is, indeed, the pivotal idea behind the action recognition from

spatio-temporal volumes. A primary task here is that humans' movements represented by saptio-temporal volumes are compared to determine how similar they are. A large number of methodologies for spatio-temporal volume representation and recognition identify these similarities correctly and efficiently, which have been around for a long time, and are very common in computer vision.

In [42], an approaches to the action recognition problem based on the MACH (maximum average correlation height) filters has been proposed. In that work, the authors generalize the conventional 2-D MACH filter for 3-D spatio-temporal volumes. A synthesized MACH filter for each action class is generated to approximate an unseen 3-D volume. When an MACH filter of action is synthesized, similar actions in an unseen video are recognized by applying the MACH filter to the video. On two popular datasets (i.e., KTH [7] and Weizmann [2]), as well as their own dataset including video clips from movie scenes, the authors performed the experiments. Further, example of recognized actions include hitting and kissing.

In [43], a spatio-temporal correlation method that detect similarity between video segments is proposed to recognize human actions. In this method, motion flows are estimated from a 3-D spatio-temporal volume. The similarities between an unseen video volume and maintained template volumes are measured by using a 3-D spatio-temporal template correlation. The local motion flows is captured by extracting small spatio-temporal patches around each point (x, y, t) of the 3-D spatio-temporal volume. The overall correlation between an input video volume and the template volume is obtained by all match scores which, in turn, are given by the correlations between all patches in the video and all patches in a template. When an unseen video is presented to the recognizer, it looks for all 3-D volume patches localized at every point (x, y, t) which are best matches to the template. Furthermore, that method showed an increasing ability to learn and recognize multiple human actions, such as pool dives, waving, and ballet movements.

In [1], the authors present an approach to the representation and recognition of human movement. In their work, a representation known as "temporal templates" are introduced to capture both motion and shape, represented as evolving silhouettes. Two 2-D images (i.e. the motion energy images (MEI) and motion history images (MHI)), instead of maintaining 3-D spatio-temporal volumes, are employed as templates for action recognition. The two images that can essentially be seen as a weighted projection of a 3-D spatio-temporal volume into 2-D XY plane, are generated from a set of foreground images. A template matching algorithm is then applied to the two image to recognize simple human actions (e.g. crouching, arm waving, sitting) from video sequences. Examples of MHIs for three actions (sit-down, arms-raise, crouch-down) are shown in Fig. 2.3. Notice that the final

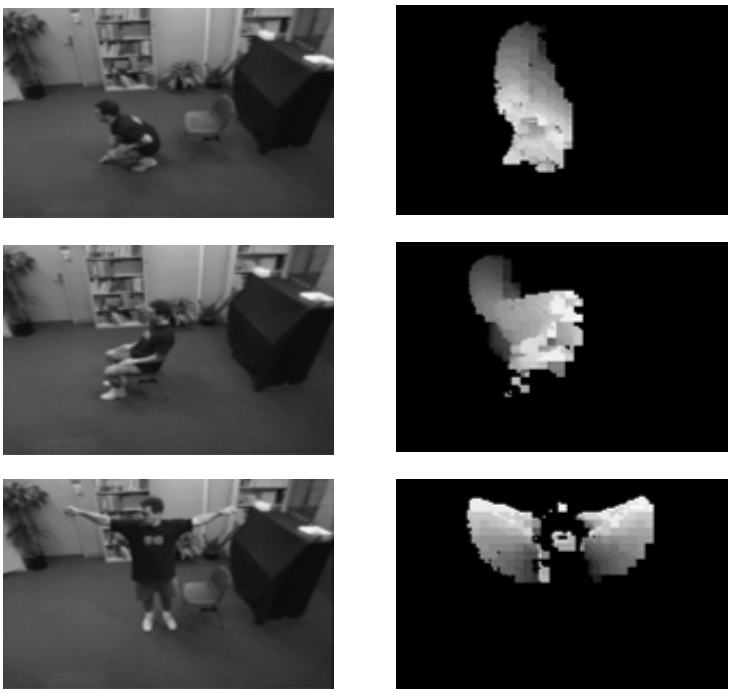


FIG. 2.3. Examples of MHIs for three actions (from up to down: crouch-down, sit-down, and arms-raise): (a) original image sequences, (b) MHIs obtained by [1].

motion locations appear brighter in the MHIs. In [2,19], the authors introduce an action model using space-time shapes. These shapes are obtained from silhouette information detected by some form of background subtraction. Various features (i.e., action dynamics, local saliency, shape structure and orientation) are extracted by using properties of the solution to the Poisson equation. A high-dimensional feature vector is then used to represent chunks of 10 frames length that are matched to space-time shapes in test sequences, in a sliding window fashion during classification. Fig. 2.4 shows three examples of these shapes appeared in [2].

In addition, in [44], Ke et al. employed segmented space-time volumes to model and recognize human actions. For finding volume segments of an action and measuring their similarity to the action model, a hierarchical mean-shift is applied to cluster similarly voxels. Support vector machines (SVMs) are then used to classify videos based on both shapes and flows of the 3-D volumes. Similar to many recognition approaches, actions are recognized by looking for a subset of over-segmented space-time volumes that best matches the shape of the action model. Experiments of that system were conducted on the popular KTH dataset [7] as well as a new tennis action database [44].

FIG. 2.4. Space-time volumes based on silhouette information presented in [2]

.

2.3.2 Trajectory based action recognition

The central idea of trajectory-based approaches is to use a set of trajectories of joint positions points on the human body to recognize actions from videos. In other words, in these approaches, an action is seen as a collection of spatio-temporal trajectories. Thus, the performer of an action can be modeled by a collection of points in 2-D or 3-D space that correspond to the joint positions. When a person performs an action, the changes in joint positions are recorded as spatio-temporal trajectories, creating 3-D (XYT) or 4-D (XYZT) representations of the action. Example trajectories are shown in Fig. 2.5. To extract the joint positions of a person, there are many methodologies for human body part estimation (e.g., the stick figure modeling). In [45], it has been stated that tracking of joint positions is quite enough for humans to visually identify actions in videos, and this paradigm has extensively been explored [46,47]. In the literature, there are several approaches making use of trajectories to learn human motion patterns and generate some high-level semantic description of such actions [48,49]. For example, in [49], a set of 4-D joint trajectories are utilized to learn actions acquired by moving cameras. In [48], each action is defined as a set consisting of 13 4-D joint trajectories. To measure the view-invariant similarity between two sets of trajectories, an affine projection is performed to obtain normalized trajectories of action.

Instead of maintaining trajectories themselves, in [50], meaningful curvature patterns are extracted from the trajectories to represent human actions. After detecting skin pixels, the authors tracked the location of a hand in 2-D space to obtain a spatio-temporal curve. Actions are then represented as a set of peaks and intervals between them. Such peak features have been verified to be view-invariant. Several prototypes representing actions are constructed, which can be seen as action

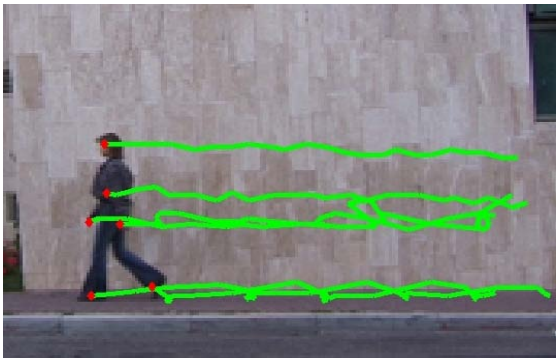


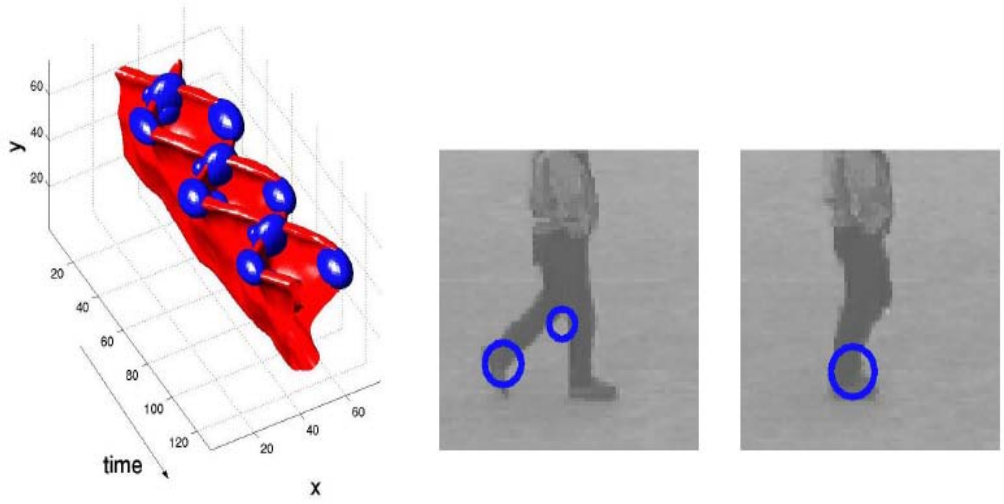
FIG. 2.5. Tracking trajectories of walk action from KTH action dataset [3].

templates. Finally, once templates are matched, actions can be easily recognized. Simple actions from office environments that were recognized include 'Picking up an object' and 'opening a cabinet'.

2.3.3 Local feature based action recognition

The intuitive idea of the approaches discussed in this subsection is to represent and recognize actions in terms of a a set of local features extracted from 3-D spatio-temporal volumes. The fact that 3-D spatio-temporal volumes can be seen as rigid 3-D objects is the main motivation behind the spatio-temporal local features approaches. This means that once appropriate features that represent the characteristics of 3-D volumes of an action are extracted, the recognition can successfully be conducted by object matching. While, many of these approaches extract sparsely distributed local features from 3-D volumes [4, 49, 51–53], we find some other approaches adopt to detect interest points at every frame and combining them temporally to describe the overall motion of human actions [2,54,55]. Fig. 2.6 shows an example of such 3-D spatio-temporal local features extracted from a video of a 'walking' action using [4]. Due to their stability and robustness to noise, illumination changes, camera jitter, and background movements, these features have been intensively used in many applications with encouraging results.

In [55], the authors present an approach that extracts spatio-temporal features at multiple temporal scales to isolate and cluster actions. To deal with the speed variations of actions, they analyze multiple temporally scaled video volumes. Then local intensity gradients are estimated and normalized for all points within a 3-D volume. For each video sequence, these spatio-temporal features of gradients are histogramed without considering locations of the extracted features, and also a histogram-based distance metric is generated, similar to [54]. To learn and recognize



(a) (b)

FIG. 2.6. Spatio-temporal local features extracted from a video of a 'walking' action; (a) 3-D plot of a leg pattern and the detected local features; (b) interest points overlaid on single frames in the sequence [4].

actions, an arbitrary clustering algorithm is used on these histograms. The approach is able to detect and recognize various actions, such as tennis and basketball plays. Added to that, there are other approaches that utilize sparse local features extracted from 3-D volumes, in order to describe and recognize actions. For example in [4], sparse 3-D (XYT) interest points are extracted to represent and recognize human actions in videos. For detecting the robust points in a 3-D volumes, the spatio-temporal Harris detector [4] extends the widely-used Harris corner detector to the time axis [56]. One advantage of this scale-invariant spatio-temporal detector is that various types of motion patterns can be robustly captured. Such features help in dealing with initially occluded backgrounds. Moreover, these features can be combined with SVMs to classify multiple actions [7]. The recognition results achieved confirmed the efficiency and applicability of these features.

A fact should be pointed out here that the paradigm of using sparse local features extracted from spatio-temporal volumes for the action recognition has been explored in depth previously in machine vision. As suggested in [4], due to the fact that local motion can be efficiently described by such sparse features, they have been intensively focused and will likely continue to focus not only on action recognition, but also on other related domains. Furthermore, many object recognition approaches utilizing sparse appearance local features (e.g., SIFT descriptors [57] and HOG descriptors [58]) have demonstrated that higher descriptiveness and robustness can be achieved by applying these descriptors to the image patches

containing interest points. That in turn provides an extra motivation to the action action recognition approaches. It might seem appropriate and cost-effective, from a technical perspective, for these features to be extracted only at a shape change or salient appearance in space-time volume. Similar to object recognition descriptors, most of these local features have been shown to achieve invariance to scale changes, rotation, affine transformation, and affine illumination changes. In [59], Niebles et al. introduce an unsupervised classification methodology utilizing the feature detector described in [51] to recognize actions. This method is regarded as a generative approach that model each action category as a set of local features. To recognize actions, the authors use a popular statistical technique, so-called probabilistic Latent Semantic Analysis (pLSA). Once the posterior probability is obtained for each action class, classification is performed by assigning the feature to the action class associated with the highest posterior probability.

Within this context, there are several approaches that attempt to develop and improve local feature extractors. For example, in [60], a recognition approach that detects sparse features (i.e. view-invariant action sketches) is presented. Also, like the cuboid features presented in the work of Dollar et al. [51], Scovanner et al. [61] propose an extension of the SIFT descriptor [57] to 3-D. In [62], a methodology for feature selection, like that of [63], is presented, as well as an improved detector for cuboid features. While only 'brightness' cuboid features have previously been used [4, 51], both color and motion information as cuboid features are utilized by [64]. For the sake of extracting more distinct and meaningful features, a method to prune cuboid features is presented in [65]. Due to their robustness to both noise and illumination changes, spatio-temporal feature-based approaches have received and still receives much interest by many researchers in computer vision and pattern recognition. Add to that, action recognition based on these features very often calls for neither background subtraction nor body-part modeling [52, 53]. Hence, such approaches are quite computationally tractable. However, when applied to more complex long-term human action recognition, these features become intractable and highly prone to limitations in performance. View-invariance is a challenging problem in action recognition, which should be handled by feature-based approaches. For a non-periodic motion, it has been shown that the relations among features are crucial and should be taken into account. Even though, describing these relations is not likely to be computationally efficient, several approaches have been developed with the hope of overcoming some limitations [20, 53, 66, 67].

2.4 Sequential Recognition Approaches

In sequential approaches, unlike most other approaches, recognition of a human action is conducted using a sequence of features. With these approaches, a video sequence containing an action is interpreted as an ordered set of observations. Hence once a specific set representing a video is observed, a decision that an action has occurred in the video can be responsibly made. To fulfill their goals, sequential approaches proceed in three main steps: first, they extract the features describing distinct temporal states (i.e. poses) of the action action contained in a video sequence. Second a likelihood used to measure how likely the feature vectors are given the probability distributions of the activity class. Eventually, when the likelihood value between the activity sequence and the activity model is greater than a given threshold, then a decision that the activity occurred is made.

Sequential recognition approaches can be generally subdivided into two major classes: state model-based approaches and exemplar-based recognition approaches. In state model-based recognition approaches, a action model is postulated, and then trained to obtain a collection of features corresponding to each activity category. Therefore, the activity recognition can be performed simply by computing the likelihood that an input sequence is generated by each model of activity. On the other hand, exemplar-based recognition approaches use training samples to describe action categories. In other words, either a set of training sequences for each activity or a representative sequence for each activity category is maintained. Then a new sequence to be recognized is matched with each representative sequence and the best match activity is considered as the most likely activity.

2.4.1 Exemplar-based recognition approaches

As stated previously, exemplar-based recognition approaches maintain either a set of sample sequences or a template sequence of action profiles to represent human actions. The basic workflow of a typical exemplar-based recognition approach proceeds as follows. As soon as a new video is supplied, its sequence of feature vectors is compared with the maintained sample sequences or template sequence. If the similarity between the sequence of the input video and a template action is above a certain threshold, then it is deduced that the input video most likely contains the action. Since different people perform the same action differently, not only in style, but also in timing, matching test and training activities is not a trivial task that has to be of interest to action recognition algorithms.

Dynamic time warping (DTW) algorithm [68], originally used in speech processing, was first presented to measure the similarity of time series of different length. In action recognition, various researchers have used the DTW algorithm to measure the similarity of action sequences with variations [69–71]. The strength and potential of the DTW-algorithm is that it can enable optimal nonlinear matching of template sequences consisting of a variable number of items. In [71], a 3-D model-based tracking method for action recognition is presented, in which an extension to the original DTW algorithm is developed. Human motion at each frame is characterized by joint angles as features. The DTW matching algorithm is employed to compare the angle sequences of an input video with a template sequence.

In [70], a DTW matching-based technique based on maintaining several models (i.e. views) of an object (e.g. a hand) constructed in different conditions to represent human dynamics. Similar to [71], DTW matching algorithm is used to match a new video with the maintained templates. Further timing variations of action performance are taken into account. In [72], Efros et al. introduce a method to recognize distant actions, where each person is around 30 pixels tall. The 2-D (XY) optical flows are detected by tracking moving person using temporal image differences, similar to [73], which are essentially utilized to obtain the final motion descriptors. For classifying the sequences of motion descriptors of actions, the basic nearest neighbor algorithm is employed. Their Experiments were conducted on three datasets (i.e., tennis plays, soccer plays, and ballet movements).

Similar to [73], in [69] a human action is represented as a time function that describes parameter changes. The intra- and inter-personal changes of action is explicitly modeled and considered in the matching of action sequences. Nonlinear characteristics of activity timing changes are also learned. For considering time warping when matching activity sequences, the original DTW algorithm was extended. In particular, the work of [74] models a human action as an LTI (Linear Time Invariant) dynamical system and estimate its model parameters to recognize activities in videos. In this work, two contour representations (i.e. Fourier descriptors and silhouette width) are extracted from the silhouette sequences. After transforming a new video to parameters of a LTI model, SVMs are used to classify these parameters, and then recognize the action contained in the input video.

2.4.2 State model-based recognition approaches

The pivotal idea the state model-based recognition approaches is to construct a statistical model composed of a set of states for each human action, which is designed such that it generates a sequence of a specific probability. The likelihood

between an observation and each action model can be obtained by computing the probability of the model that produces the feature sequence of the observation. In this regard, the classification of a given video sequence is performed by employing any statistical inference, for example, Bayesian inference based on maximum a posteriori probability (MAP) or maximum likelihood (ML). Hidden Markov models (HMMs) [75] that are essentially a member of a wider family of models (i.e. dynamic Bayesian networks (DBNs) [76]) are widely used for classifying sequential data. Furthermore, they have been demonstrated to be a potent tool for modeling time-dynamic sequences of variable lengths. HMMs (or DBNs) characterize an activity as a sequence of hidden states, each describing a status (i.e. pose) of a person performing the activity. The system then deterministically transitions to another state. A Markov chain is completely characterized by the set of all states and transition probabilities. Once all HMMs are trained, the observation probabilities in each sequence of all activities are tied with a few probabilities. Thus, activity recognition can be performed based on the computed probability of the input sequence generated by a specific state-model.

To the best of our knowledge, [77] is the first work in literature where standard HMMs is first applied on action recognitions. In this work, each binary foreground image is first converted to a series of meshes from which the features are extracted. After the HMMs are trained, activities are recognized by simply measuring the likelihoods between the HMMs and a given video containing an action. It is perhaps not irrelevant to mention that the authors have shown that HMMs have achieved a high level of success in modeling feature changes. In [1], state models are used for recognizing gestures. Each gesture is represented by a trajectory produced directly from changes in hand positions; and each trajectory is decomposed into sequential vectors. More importantly, to handle variations in speed and motion of the same gesture, each state is allowed to be fuzzy in nature. For obtaining an optimal match between an input and each prototype, a novel dynamic programming algorithm was developed. Likewise, in [78], a framework for recognizing American Sign Language (ASL) is presented using standard HMMs. In this method, the features that describe positions and shapes of the tracked hands are extracted. Further, each word (sign) of ASL, similar to [77], is modeled as an HMM that creates a sequence of features. For computing the probabilities of observations, the well-known Viterbi algorithm (VA) [79], is employed to approximate the likelihood distance efficiently.

However, the basic HMM suffers from being unable to model several interacting and complex activities, because it originally is a sequential model and at one instant in time only one state is active. In addition to the standard HMMs-based recognition methodologies discussed previously, variants of HMMs (e.g., coupled hidden

Markov model (CHMM) [80]) have also been adopted by several researchers for the modeling and recognition of human activities [81–84]. In a typical methodology using variants of HMMs [77,78,85], each action being recognized has its own model (HMM), similar to former standard HMMs-based methodologies. Visual extracted features are utilized as observations directly generated by the model. It may be pertinent to mention that with HMMs extensions, it is able to recognize and classify more complex activities more efficiently. In [81], the coupled HMM (CHMM), as a variant or extension of the standard HMM, is developed and used for modeling interaction between two persons. As the name suggests, a coupled HMM is formed from combining two or more basic HMMs, each describes only an agent's motion. In their work, two HMMs are joined for recognizing complex human interactions. In [86], a view-invariant recognition system is proposed, where a CHMM-like model called 'Action Net' is built to connect 2-D key poses of actors to represent 3-D shapes for action recognition. In addition, in Natarajan and Nevatias work [83], coupled hidden semi Markov models (CHSMMs) are introduced for modeling and recognizing human interactions. In their work, a statistical model are built, which could determine the characteristics of activities being recognized more effectively and efficiently compared to CHMMs. Similar to [81], the method was validated on various human-human interactions (HHIs).

2.5 Discussion and Conclusion

Automatically recognizing human activities in video sequences is increasingly receiving research attentions due to its great potentials for many applications in several contexts and domains. For example, robust action-based surveillance is required and essential for success in a variety of today's (and tomorrow's) technologically driven health care settings and many other settings. Likewise, action-based HCI is probably one of the most widespread applications for human action recognition where no explicit actions (e.g., keystrokes and mouse clicks) are available to capture user input; instead, interactions are more likely to occur through human actions and/or gestures. In this chapter, we have presented an update overview of the current state-of-the-art in human action modeling and recognition. The presented approaches are miscellaneous and usually provide a broader spectrum of results. Within this review chapter, previous related works have been categorized depending on which approach a particular methodology adopts. In this context, there are two main categories of approaches: hierarchical approaches developed for the analysis of high-level interactions and non-hierarchical approaches developed for recognizing short-term human activities and gestures. While the main focus

in this chapter has been upon the non-hierarchical approaches as they are immediately relevant to the topic of this thesis, the hierarchical approaches have also been referred to sparingly. Several types of methodologies developed for action recognition have been described and discussed thoroughly.

It is worth mentioning here that while the early research into human motion modeling and/or recognition dates back to the pioneering work of Johansson [45] in the mid 1970s, research on human action recognition did not shift to the forefront until the early 1990's. About a decade later, by the end of the 1999s, research in human action recognition has barely begun to get into its infancy [21]. It may be interesting to state that the past ten years or so have witnessed an increasing number of research efforts in human action recognition. However, the contributions to improved human action recognition have been modest, as well as the off-the-shelf technology solution space for human action recognition is still far form being quite mature yet. The experimental systems of action recognition are now appearing at a very limited number of locations (e.g., airports and other public places).

Noise and segmentation issues pose substantially larger burdens for the action recognition system. Further, while motion tracking can be performed almost effortlessly by humans, it remains one of the most challenging research problems in the fields of computer vision and image/video processing. When the tracking algorithm fails to correctly track the object of interest due to noise, shades, occlusion, etc., the activity recognition task becomes much more complicated. Hence, it seems to be an extremely difficult task to build an activity recognition system able to accurately compensate for such failures in those settings. Once such issues of action recognition are appropriately addressed by careful design and implementation, it is very likely that this will lead to improved recognition performance, and ultimately to more and more such systems being deployed in various applications.

CHAPTER 3

Segmentation of Image/Video Data

3.1 Introduction

In this chapter, we will discuss in detail how to detect Region-of-Interests (ROIs) within the image/video which correspond to distinct foreground objects (i.e., moving human body or limbs being tracked) in the scene. These regions do, indeed, carry much relevant information that is not only most crucial for the subsequent feature extraction and analysis, but also would affect the feature classification task of the whole human activity recognition system. This chapter is generally divided into two main parts. In the first part, first we show the overall process of image segmentation, and then our method for image segmentation based on generalized $\alpha-$entropy is described. The second part is devoted to video segmentation process as a main part in developing any object-based video recognition system in general, and in developing the human activity recognition system in particular.

Conceptually speaking, the ultimate goal of image segmentation is to extract meaningful objects from an image. In other words, image segmentation can be seen as a clustering task where image pixels are split into salient regions corresponding to natural objects or parts of objects. After this process finished successfully, it is expected to obtain a set of disjoint regions with uniform and homogeneous attributes such as color, intensity, texture, etc. which are meant to be semantic equivalence categories. Indeed, image segmentation is potentially significant but inherently challenging. It is frequently needed as a preliminary step for solving various image analysis tasks. For instance, segmentation is a vital pre-processing step in many applications, such as object recognition, image/video coding, video

editing, content-based image/video retrieval and browsing, and so forth. Over the past couple of decades or so, a significant number of segmentation methodologies with different approaches, such as spatial-domain methods, graph-based methods, and feature-space methods have been developed, which have considerably contributed to various ever evolving fields in computer vision, such as object and activity recognition [87–90]. In computer vision and pattern recognition, scene segmentation is a most fundamental and complex task. Although well posed, the scene segmentation is recognized as an ill-posed problem and remains unsolved, or at least did not receive a fully satisfactory solution. Further, despite the current extensive literature of techniques and extensions for object segmentation, there is a lack of a general-purpose approach able to handle the problem in its full generality.

3.2 Image Segmentation

Image segmentation is one of the fundamental and most studied problems in computer vision and has found its application directly or indirectly in many tasks such as object recognition, image coding, image understanding, etc. The terms image segmentation, object isolation, and object extraction are often used interchangeably or at least with potentially overlapping meanings in various contexts in computer vision, since they perform the same function and yield equivalent results. It may be worthwhile mentioning that segmentation is application dependent and its success is always measured by the needs of the application. Therefore, direct segmentation of an input image has to pragmatically consider semantic information about a particular application. As seen from what mentioned previously, segmentation methods are not universally applicable to all images.

Human vision always perceives a complex visual scene by decomposing it into its components (i.e. objects). In other words, we humans see a visual scene as a set of objects. As illustrated in [91], human visual system is able to perceive the global structure of a set of elements. All ideas associated with the process of visual grouping was first introduced and explained by the Gestalt theory of perception [92], as a first step in the analysis of a visual scene. While, segmentation is unconsciously achieved by the human vision system, it is still time-consuming and quite challenging for machine vision system.

3.2.1 Brief overview

Broadly speaking, image segmentation is viewed as a process in which an image is split into nonoverlapping constituent regions. These regions are homogeneous with

respect to some visual properties like intensity, color, texture, shape, etc. Formally, let R be the domain of a given image, then the segmentation problem is to determine the sets $\{R_k, k = 1, 2, \ldots, K\}$ which satisfy the following condition:

$$R = \bigcup_{k=1}^{K} R_k \tag{3.1}$$

where $R_i \bigcap R_j = \emptyset,\ i \neq j$, and indeed each R_i is connected. Ideally, a segmentation approach looks for the sets that correspond to distinct objects (or regions of interest) in the image. Essentially, the segmented image is a 2d function s defined on the same domain of the original image, I, but its range of values differs from one case to another. The following three cases are well-known:

case 1: A binary segmentation (foreground and background):

$$s(x, y) = i,\ i \in \{0, 1\}$$

case 2: A multiclass segmentation (the case of C different segments):

$$s(x, y) = i,\ i \in \{0, 1, 2, \ldots, C\}$$

case 3: A boundary image:

$$s(x, y) = i,\ i \in \{0, 1\}$$

where the value of 1 means that there is a boundary at the position (x, y), whereas the value of 0 means that there is no boundary at that position. This case might apparently seem to be very similar to the first one, or more correctly they are variants to each other. The segmentation problem can be generally approached in two steps. First the original image is transformed into a higher dimensional feature space in which the boundaries between different classes are determined. Then a unique class-label is assigned to each pixel such that all the pixels having the same attribute are given the same label. Finally, the segmented image is produced. Fig. 3.1 shows an example of this two-step segmentation process mentioned earlier. As seen in this figure, an image of brain tissue basically consists of three major regions: caudate, putamen, and thalamus.

The two-step segmentation procedure can be applied to segment the three regions as follows: first the original image is transformed to the feature space, where the boundaries between the three regions or classes are defined. Then a unique class-label is now assigned to all pixels belonging to each class of these three classes. Two basic questions arise here: first what the representative features

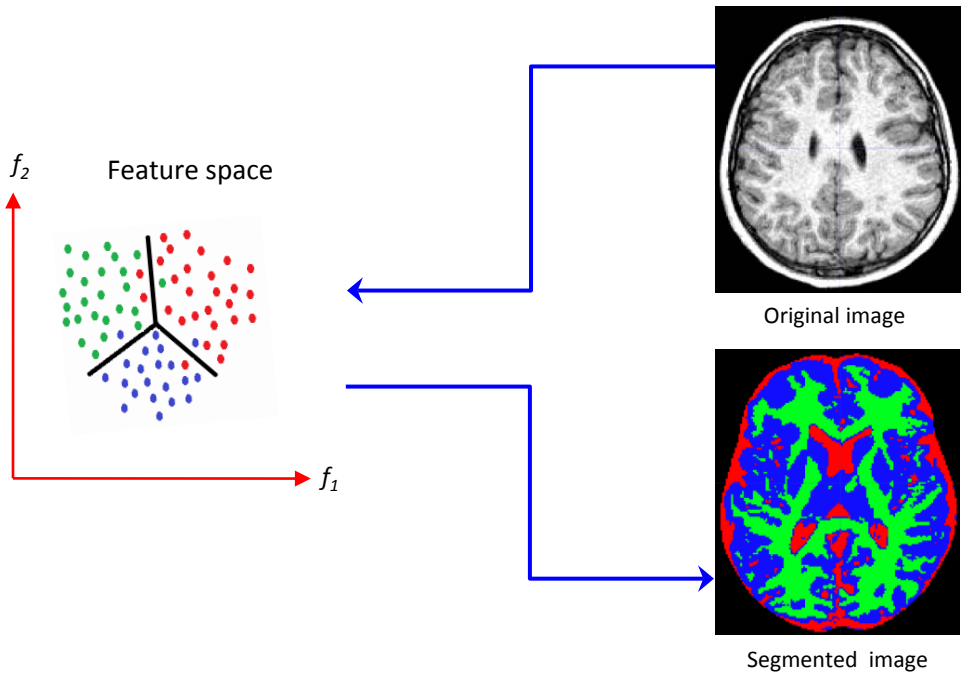


FIG. 3.1. An example for the two-step image segmentation procedure.

that can be used to efficiently characterize the relationship of the pixels belonging to a specific class, and second how to distinguish such pixels from other pixels belonging to another class. In most cases, the pixel similarity are measured based on the consistency of some features, i.e., color, intensity, shape, texture, or combination of those attributes. This implies that the feature vector of each pixel involves the information of one or more of those attributes.

Presently, several segmentation algorithms exist in literature, these algorithms are broadly categorized into: edge-based methods, looking explicitly for boundaries between segments, and region oriented techniques, where some homogeneity criterion is applied to each pixel within a specific segment. Within the popular region based methodologies, there are two well-known approaches: the region growing approaches in which the segmentation process begins from seed pixels, then pixels are added to regions as long as homogeneity is adequately preserved. The region merging approaches recursively merge adjacent regions that are similar enough. In [93], segmentation algorithms are loosely divided into three categories: region-based scheme, edge-based scheme, and pixel-based scheme, an extra category is provided by [94] to include the techniques of texture segmentation.

In literature, there are several segmentation approaches based on different methodologies, which try to segment an image by adaptively selecting proper threshold values. In this case, the segmentation process involves the analysis of

the grey-level histogram of the image. The Otsu method [95] is one of the most popular global thresholding methods of image segmentation in literature, which depends on statistical and spatial information and requires the histogram consists of only Gaussian distributions. In Otsu method, the normalized histogram is a basic concept, and the valley is searched for based on a statistical sense in order to select threshold automatically. More formally speaking, the Otsu's algorithm supposes the image contains two classes of pixels (e.g., foreground and background), and then attempts to find the optimum threshold separating these two classes, such that their intra-class variance is minimal. This variance is defined as a weighted sum of variances of these two classes [95]:

$$\sigma_w^2 = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t) \tag{3.2}$$

where the weights ω_i are the probabilities of the two classes separated by a threshold t and σ_i^2 variances of such classes. It has been proved that minimizing the intra-class variance is equivalent to maximizing the inter-class variance,

$$\sigma_b^2 = \sigma^2 - \sigma_w^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2 \tag{3.3}$$

which is expressed in terms of class probabilities μ_i and class means σ_i that in turn can be updated iteratively. While the Otsu method considers only the information of the image as a whole in all cases, it is distinguished by its good adaptation and simple calculation, and by choosing an ideal threshold. In addition to that, this method still serves as a reference method for comparing a large number of segmentation and binarization techniques.

3.2.2 Image segmentation based on generalized α-entropy

Since the pioneering work by Shannon [96], entropy appears as an attention-grabbing tool in many areas of data processing. In [97], Rènyi introduced a wider class of entropies known as α-entropies. The functionalities of new α-entropies share the major properties of traditional Shannon's entropy. Furthermore, α-entropies can be easily estimated using a kernel estimate that makes their use attractive in many areas of image processing [98]. Within this section, our segmentation method based on generalized Rènyi entropy is introduced.

Entropy of generalized distributions

Entropy first appeared in thermodynamics as an information theoretical concept, which is intimately related to the internal energy of the system. Then, it has been

applied across physics, information theory, mathematics and other branches of science and engineering. When given a system whose description is not known precisely, the entropy is defined as the expected amount of information required to specify the state of the system. Formally, let $P = \{p_1, p_2, \ldots, p_n\}$ be a finite discrete probability distribution satisfying the conditions $p_k \geq 0, k = 1, 2, \ldots, n$ and $\sum_{k=1}^{n} p_k = 1$. The amount of uncertainty of the distribution P is called the entropy of the distribution. Shannon entropy [96] of the distribution P, as a measure of uncertainty and denoted by $H(P)$, is defined as

$$H(P) = -\sum_{k=1}^{n} p_k \log_2 p_k \tag{3.4}$$

Note that Shannon entropy defined by Eq. (3.4) is additive, i.e. it satisfies:

$$H(A + B) = H(A) + H(B) \tag{3.5}$$

for any two distributions A and B. Eq. (3.5) states an important property of entropy, namely, its additivity; the entropy of a combined experiment consisting of the performance of two independent experiments is equal to the sum of the entropies of these two experiments. The formalism defined by Eq. (3.4) has been shown to be restricted to the Boltzmann-Gibbs-Shannon (BGS) statistics. However, for nonextensive systems, some kind of extension appears to be necessary. Rènyi entropy [97] that appropriately describes the nonextensive systems, is defined as:

$$H_\alpha(P) = \frac{1}{1-\alpha} \log_2 \left(\sum_{k=1}^{n} p_k^\alpha \right) \tag{3.6}$$

where $\alpha \geq 0$ and $\alpha \neq 1$. The real number α is called an entropic order that characterizes the degree of nonextensivity. This expression reduces to Shannon entropy in the limit $\alpha \longrightarrow 1$. We shall see that in order to get the fine characterization of Rànyi entropy, it is advantageous to extend the notion of a probability distribution, and define entropy for the generalized distributions. The characterization of measures of entropy (and information) becomes much simpler if we consider these quantities as defined on the set of generalized probability distributions.

Suppose $[\Omega, P]$ be a probability space that is, Ω an arbitrary nonempty set, called the set of elementary events, and P a probability measure, that is, a nonnegative and additive set function for which $P(\Omega)$. Let us call a function $\xi = \xi(\omega)$ which is defined for $\omega \in \Omega_1$, where $\Omega_1 \subset \Omega$. If $P(\Omega_1) = 1$ we call ξ an ordinary (or complete) random variable, while if $0 < P(\Omega_1) \leq 1$ we call ξ an incomplete random variable. Evidently, an incomplete random variable can be interpreted as a quantity describing the

result of an experiment depending on chance which is not always observable, only with probability $P(\Omega_1) < 1$. The distribution of a generalized random variable is called a generalized probability distribution. Thus a finite discrete generalized probability distribution is simply a sequence $p_1, p_2, ..., p_n$ of nonnegative numbers such that setting $P = \{p_k\}_{k=1}^{n}$ and taking

$$\varpi(P) = \sum_{k=1}^{n} p_k \tag{3.7}$$

where $\varpi(P)$ is the weight of the distribution and $0 < \varpi(P) \leq 1$. A distribution that has a weight less than 1 is termed an incomplete distribution. By using Eq. (3.6) and Eq. (3.7), Rànyi entropy for the generalized distribution [87] can be written as

$$H_\alpha(P) = \frac{1}{1-\alpha} \log_2 \left(\frac{\sum_{k=1}^{n} p_k^\alpha}{\sum_{k=1}^{n} p_k} \right) \tag{3.8}$$

Note that Rànyi entropy has a nonextensive property for statistical independent systems, defined by the following pseudo additivity entropic formula:

$$H_\alpha(A+B) = H_\alpha(A) + H_\alpha(B) + (\alpha - 1) \cdot H_\alpha(A) \cdot H_\alpha(B) \tag{3.9}$$

Segmentation procedure

Image segmentation problem is considered to be one of the most holy grail challenges of computer vision field especially when done for noisy images. Consequently it has received considerable attention by many researchers in computer vision community. There are many approach for image segmentation, however, these approach are still inadequate. In this section, we propose an entropic method that achieves the task of segmentation in a novel way. This method not only overcomes image noise, but also utilizes time and memory optimally. This wisely happens by the advantage of using the Rànyi entropy of generalized distributions to measure the structural information of image and then locate the optimal threshold depending on the postulation that the optimal threshold corresponds to the segmentation with maximum structure (i.e., maximum information content of the distribution). The working scheme of the proposed segmentation approach is shown as a block diagram in Fig. 3.2. The following steps are involved.

1. **Pre-processing** Pre-processing ultimately aims at improving the image in ways that increase the opportunity for success of the other ulterior processes. In this step, we apply a Gaussian filter to the input image prior to any process in order to reduce the amount of noise in an image.

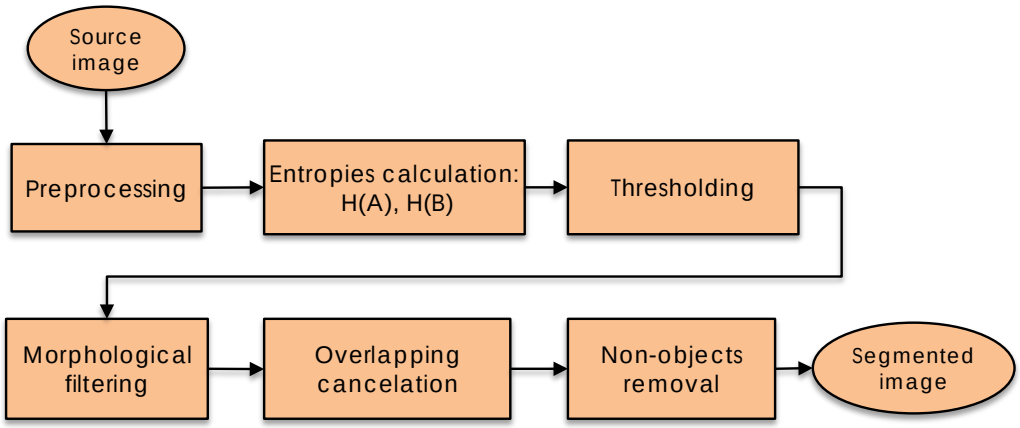


FIG. 3.2. Block diagram of the proposed segmentation approach.

2. **Entropy calculation** Let $\{p_i\}_{i=1}^{n}$ be the probability distribution of intensity in the image. By applying a threshold t, the distribution is divided into two sub-distributions; one corresponding to the foreground objects (class f) and the other to the background (class b), and denoted $P^f = \{p_i\}_{i=1}^{t}$ and $P^b = \{p_i\}_{i=t+1}^{n}$, respectively. Thus, the generalized Rànyi entropies for the two distributions as functions of t can be written as:

$$H_\alpha^f(t) = \frac{1}{\alpha-1}\log_2\left(\frac{\sum_{k=1}^{t} p_k^\alpha}{\sum_{k=1}^{t} p_k}\right) \quad (3.10)$$

$$H_\alpha^b(t) = \frac{1}{\alpha-1}\log_2\left(\frac{\sum_{k=t+1}^{n} p_k^\alpha}{\sum_{k=t+1}^{n} p_k}\right) \quad (3.11)$$

3. **Image thresholding** Thresholding is the most often used technique to distinguish objects from background. In this step an input image is converted by threshed into a binary image so that the objects in the input image can be easily separated from the background. To get the desired optimum threshold value t^*, we have to maximize the total entropy,$H_\alpha^{f+b}(t)$. When the function $H_\alpha^{f+b}(t)$ is maximized, the value of parameter t that maximizes the function is believed to be the optimum threshold value [99]. Mathematically, the problem can be formulated as

$$t^* = argmax[H_\alpha^{f+b}(t)] = argmax[H_\alpha^f(t) + H_\alpha^b(t) + (1-\alpha)\cdot H_\alpha^f(t)\cdot H_\alpha^b(t)] \quad (3.12)$$

4. **Morphological operations:** In image processing, dilation, erosion, closing and opening are all well-known as morphological operations. In this step we aim at improving the results of the previous thresholding step. Due to the

inconsistency within the color of objects, the resulting binary image perhaps includes some holes inside. By applying the closing morphological operation, we can get rid of the holes form the binary image. Furthermore Opening operation with small structure element can be used to separate some objects that are still connected in small number of pixels [100].

5. **Overlapping cancelation:** In this step we attempt to remove the overlapping between objects that perhaps happened through extensively applying the previous morphological operations. To perform this, we first get the Euclidean Distance Transform (EDT) of the binary image. Then we apply the well-known watershed algorithm [101] on the resulting EDT image. The EDT ultimately converts the binary image into one where each pixel has a value equal to its distance to the nearest foreground pixel. The distances are measured in Euclidean distance metric. The peaks of the distance transform are assumed to be in the centers of the objects. Then the overlapping objects can be yet easily separated.

6. **Non-objects removal:** This step helps in removing incorrect objects according to the object size. Sizes of objects are measured in comparison to the total size of image. Each tiny noise object of size less than a predefined minimum threshold can be discarded. Also each object whose size is greater than the maximum threshold size can be removed as well. Note that thresholds of size used herein are often dependent on the application, and so they are considered as user-defined data.

Segmentation results

We began the experiments by using different histograms, each describes two classes (objects and background). To investigate the influence of using generalized Rènyi entropy on segmentation quality, other formula of entropy (Tsallis entropy [98]) was also attempted, which is defined by:

$$H_\alpha = \frac{1 - \sum_{k=1}^{n} p_k^\alpha}{\alpha - 1} \tag{3.13}$$

The obtained results highlight the usefulness of the proposed approach and demonstrate its capability, especially when generalized Rènyi entropy is used. Fig. 3.3 shows an example of segmentation results. As seen in the figure, an image of a medical domain with a spatial background scattering noise is shown; a stained brain cell shows branching of cell dendrites-fibers receiving input from other brain cells. Several values of α were attempted, but best result were achieved at $\alpha = 0.9$.

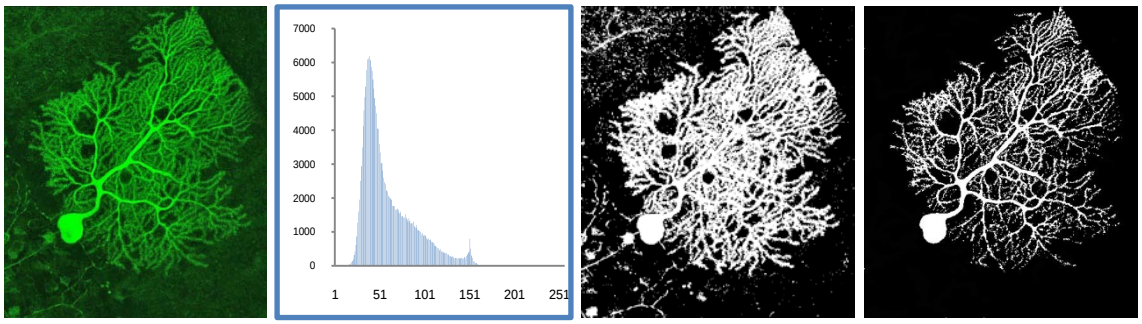


FIG. 3.3. Entropic segmentation of a brain cell image with a spatial noise around.

Extension of the approach

In [87], the authors have presented an extension of the approach (described in the previous subsection) via Fuzzification of the Rènyi entropy of generalized distributions. The basic idea of the fuzzification of entropy at this juncture involves the process of incorporating fuzzy memberships into the relations of entropy described by (3.10) and (3.11). Therefore fuzzy image segmentation is basically based on considering that how strongly an intensity (pixel value) belongs to the background or the targets can be depicted by the fuzzy memberships. Indeed, the farther away an intensity of a pixel is from a given threshold τ, the greater is its ability to belong to a particular category. Consequently, for any pixel that is ℓ levels below or ℓ levels above the threshold τ, the membership values are given by

$$\mu_f(\tau-\ell) = \frac{1}{2}+\frac{\sum_{i=0}^{\ell} p(\tau-i)}{2p(\tau)} \tag{3.14}$$

$$\mu_b(\tau+\ell) = \frac{1}{2}+\frac{\sum_{i=1}^{\ell} p(\tau+i)}{2(1-p(\tau))} \tag{3.15}$$

which measure the degrees of pixel's belongingness to the class f and class b (i.e., foreground and background) respectively (see Fig. 3.4). The equations (3.14) and (3.14) clearly show that the uncertainty associated with the belongingness of the pixel to a specific class increases dramatically as its intensity value approaches to the threshold, and it achieves its maximum when the intensity value becomes equal to the threshold, and then $\mu_f(\tau) = \mu_b(\tau) = 0.5$. Now by considering the membership functions given by equations (3.14) and (3.15) above, the entropic

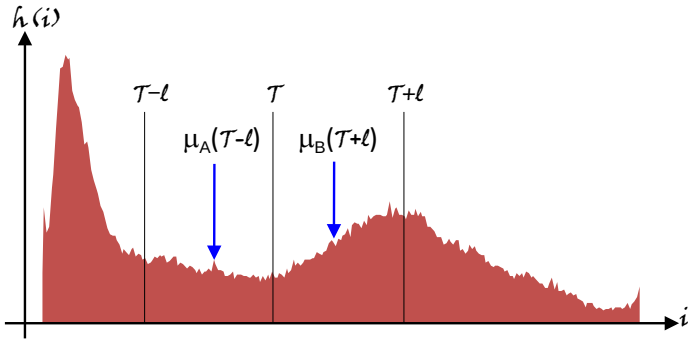


FIG. 3.4. Fuzzy membership as an indication of how strongly a pixel belongs to its region.

formulas in equations (3.10) and (3.11) can be written in a fuzzy form as follows

$$H_\alpha^f(t) = \frac{1}{\alpha-1}\log_2\left(\frac{\sum_{k=1}^{t}(\mu_f(k))^\alpha}{\sum_{k=1}^{t}\mu_f(k)}\right) \tag{3.16}$$

$$H_\alpha^b(t) = \frac{1}{\alpha-1}\log_2\left(\frac{\sum_{k=t+1}^{n}(\mu_b(k))^\alpha}{\sum_{k=t+1}^{n}\mu_b(k)}\right) \tag{3.17}$$

By virtue of the fact that the value of the parameter t that maximizes the total entropy functional meets the optimum threshold, the optimum threshold t^* can be obtained by solving Eq. (3.12). For color image, Eq. (3.12) is rewritten as:

$$\vec{t}^* = \arg\max\left(H_\alpha^f(\vec{t}) + H_\alpha^b(\vec{t}) - (1-\alpha)H_\alpha^f(\vec{t})H_\alpha^b(\vec{t})\right) \tag{3.18}$$

where $\vec{t} = (t_R, t_G, t_B)$ and the absolute of the optimum threshold $\vec{t}^*$ is given by

$$\|\vec{t}^*\| = \sqrt{(\omega_R t_R)^2 + \omega_G t_G)^2 + \omega_B t_B)^2} \tag{3.19}$$

where ω_R, ω_G, and ω_B are the weights of the components R, G, and B respectively, which satisfy

$$\omega_R + \omega_G + \omega_B = 1 \tag{3.20}$$

Thereafter, some post-processing operations similar to those of the main segmentation algorithm described so far (e.g., blob analysis with morphological operators) need to be performed on the segmented image to refine the results of initial segmentation. Some examples of image segmentation obtained by the proposed segmentation technique are shown in Fig. 3.5.

3.2.3 Summary and conclusion

To sum up this section, we can draw the following conclusions. We have presented an efficient method for image segmentation based on a generalized α-entropy.

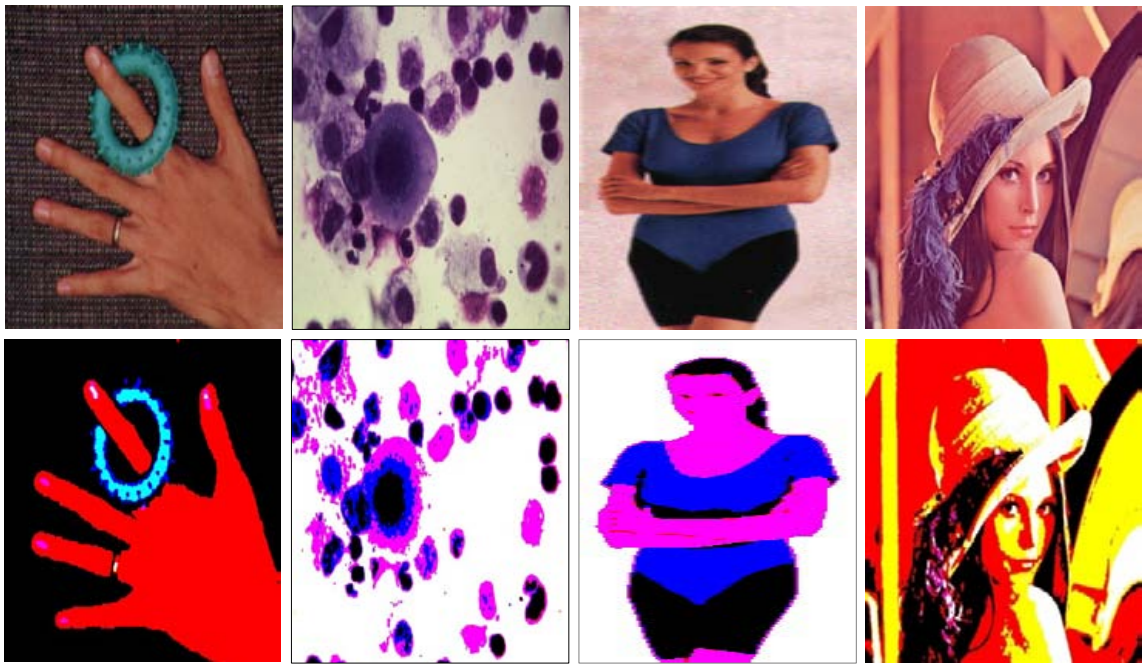


FIG. 3.5. Image segmentation examples; the top row shows the original source images, while the bottom row presents the segmented images.

This method has achieved the task of segmentation in a novel way; it could yield good results in many cases and perform well when applied to noisy images. The segmentation results conform that using generalized Rènyi formalism of entropy is more viable than using Tsallis counterpart in segmenting cell images. Important advantages of the proposed method are its simplicity, computational efficiency, and its relative immunity to noise.

In the second part of this section, an extension to the basic segmentation technique based on fuzzification of Rènyi entropy has been presented. It has been found that the extended algorithm is able to suppress noise strongly within regions while preserving luminance transitions (i.e. edges) between regions. Furthermore the algorithm can perform well when applied to both grayscale and color images, without the need for any a priori knowledge on the distortions, and any color model can be used. The results obtained show clearly that using the formalism of fuzzy Rènyi entropy is more viable than using the entropy alone in the image segmentation task. It was also found that the proposed algorithm is fast and robust. One last point worthy of mention here is that although the presented algorithm has been applied to still images, it appears there are no technical or apparent theoretical limitations that would prevent it to be applied to video sequences.

3.3 Video Segmentation

The past two decades or so have witnessed a great explosion in the amount of the distributed visual information by the explosive growth of the Internet and the great advances in hardware technologies and software developments. This trend is expected not only to continue in the future, but to grow very rapidly due to the media fusion of television and several web services. This fact and the existing and continuously emerging new applications strongly advocate the significant and even urgent need for the development of a large number of new techniques for distributing and processing the huge amount of available visual data.

3.3.1 Brief overview

Broadly speaking, video segmentation involves the process of partitioning video into a set of spatial, temporal, or spatio-temporal regions that are homogeneous and disjoint in some feature space. In this context, the task basically aims to isolate those portions of a video sequence that constitute objects or regions of interest (ROI) and separate them from the sequence. Indeed, the task is a preliminary and crucial step in many high-level tasks in computer vision, such as object localization and identification, tracking and recognition. Moreover, it plays a critical role as an integral component of a variety of video-based applications, including:

1. Motion estimation in scenes of multiple moving objects
2. 2D dense motion/optical flow estimation
3. Video monitoring and interpretation
4. Advanced video coding and adaption
5. Video summarization, browsing and semantic indexing
6. Video editing and authoring

There are some leading factors that determine which segmentation methods are most appropriate to be employed in a particular video-based application, such as

- *Segmentation accuracy:* In some cases, such as object-based video editing or shape similarity matching, the estimated borders should quite align with borders of actual object; a single pixel error in alignment will be visible. Hence segmentation results must essentially be perfect. In other cases, for example when segmentation is used in improving the compression efficiency or rate

control, some misalignment between the estimated boundaries and borders of actual object may be acceptable and tolerable, to certain degree of errors.

- *Real-time efficiency:* In case the application requires segmentation to be performed in real-time (e.g., rate control in video-telephony), a simple fully automatic segmentation algorithm are most appropriate to be employed. In many offline applications, however, (e.g. video indexing or offline video coding), semi-automatic segmentation algorithms are most able to work well.

- *Scene complexity:* Structurally speaking, the scene complexity is usually measured by the number of geometric primitives of which the scene is composed. In fact, one can model video content complexity in terms of amount of camera motion, motion smoothness of objects, contrast among objects, objects entering and leaving the scene, color and texture homogeneity inside objects, etc. In more complex scenes, it appears clearly that more sophisticated segmentation techniques would be necessary to get the most out of the contents.

Motion segmentation is strongly associated with two other tasks of equal interest, namely, motion detection, and motion estimation. Essentially motion detection is a binary labeling problem in which each pixel of the image at a time is attributed either as moving or stationary. While, motion detection in the case of a static platform has been studied intensively in the literature, and is considered as computationally efficient, the same task for a moving camera that may require a specific type of global or local motion estimation [102], still remains as computationally intractable. These two cases of motion detection differ only in how to create the background model. For example, in a stationary platform, all considerable variations of a given video sequence are first estimated at the pixel level, and then statistical techniques can be used to construct a background model for each pixel [103]. Such an approach can be easily extended to moving cameras using motion compensated pixel differencing before estimating the background model [104, 105].

In general, video segmentation can be achieved by three main approaches: optical flow, frame (i.e. temporal) differencing and background subtraction. Optical flow is considered to be one of the most robust techniques for video segmentation, that is able to work well even in non-stationary camera platforms. However this approach is relatively computationally demanding, and thus not always practical for operation on embedded real-time systems. Temporal differencing technique is a simplest background modeling technique, which not only is very adaptive to dynamic environments, but also can be employed without any priori knowledge about background [106]. However, temporal differencing scheme is prone to the

serious aperture problem of foreground due to the color uniformity of moving objects; this leads to inconsistent detections. Having perfect background modeling, a background subtraction scheme can reliably detect all moving pixels. Though background subtraction is very sensitive to scene changes due to changes in lighting and movement of background objects.

3.3.2 Frame differencing

Frame differencing is one of the most common techniques for background segmentation, which, as its name suggests, involves taking the difference between two frames and using this difference to detect the moving objects in the scene. Formally speaking, frame differencing can be explained as follows. First, let us assume that $f(x, y, t)$ represents the intensity value at pixel location (x, y), in frame t. By comparing $f(x, y, t)$ with $f(x, y, t - z)$ and $f(x, y, t + z)$, a binary motion map $M(x, y, t)$ can be defined such that it is equal to 1 if and only if motion took place at pixel location (x, y) in frame t, where $z > 0$ is an arbitrary integer. Then pixel-wise AND operation is performed to determine the motion map $M(x, y, t)$ as follows:

$$M(x, y, t) = d_1(x, y, t) \otimes d_2(x, y, t) \tag{3.21}$$

where $\otimes$ denotes the pixel-wise AND operator, and the functions $d_1(x, y, t)$ and $d_2(x, y, t)$ are determined, respectively, from the following two equations:

$$d_1(x, y, t) = \begin{cases} 1, & |f(x, y, t) - f(x, y, t - z)| > \tau \\ 0, & otherwise \end{cases} \tag{3.22}$$

$$d_2(x, y, t) = \begin{cases} 1, & |f(x, y, t) - f(x, y, t + z)| > \tau \\ 0, & otherwise \end{cases} \tag{3.23}$$

where τ is an application-specific threshold. An optimal threshold determination algorithm can be used to find out the optimum value of τ. One or more post-processing steps are then employed to filter out the isolated labels that do not correspond to actual moving objects in the scene. Such post-processing operations may involve: (i) the use of smoothing filters (e.g., Gaussian smoothing, Mean or Median filters), (ii) the use of morphological operations (e.g., openings, closings, dilations, and erosions), and (iii) the removal of labels with less than a predefined number of entries. As a consequence, false segmentations associated with the isolated labels are more likely to be eliminated or substantially discouraged. Potential advantages of such post-processing steps are that isolated labels will be removed producing a smoother, less noisy segmentation results with smoother boundaries.

Once post-processings mentioned before are successfully applied on $M(x, y, t)$, the largest connected component at each location (x, y) associated with $M(x, y, t) = 1$ can be identified by using connect component analysis. To ensure spatio-temporal continuity of the changed regions, adding memory to motion detection seems be intrinsically advantageous for this purpose [107]. Temporal integration of luminance values across multiple frames before thresholding might be one of the most straightforward methodologies to fulfill this objective, which can be described as follows. The frame difference with memory is frequently treated as some variation of the consecutive frame difference and normalized frame difference. This difference can be derived from the difference between the current frame $f(x, y, t)$ and a weighted average of all previous frames $\tilde{f}(x, y, t)$:

$$\tilde{M}(x, y, t) = f(x, y, t) - \tilde{f}(x, y, t) \tag{3.24}$$

where,

$$\tilde{f}(x, y, t) = \begin{cases} \alpha f(x, y, t) + (1 - \alpha)\tilde{f}(x, y, t - 1), & t = 1, 2, \dots \\ 0, & t = 0 \end{cases} \tag{3.25}$$

where α is an arbitrary filter constant (i.e., a weighting factor) which is a fraction between 0 and 1. After processing a number of frames, the changed regions in $\tilde{f}(x, y, t)$ become blurred, while the unchanged regions preserve their sharpness with a relatively low noise. As is the case in two-frame methods, a global or a spatially adaptive threshold can be employed to $\tilde{M}(x, y, t)$. It was found that the major effect of the temporal integration is to make the likelihood getting rid of spurious labels higher; consequently spatially contiguous regions are more likely to be preferentially identified.

Generally speaking, in comparison with several state-of-the-art methods for motion segmentation, frame differencing not only has lower computational requirements, but also can yield a relatively substantial improvement in temporal coherence. However, there are also some shortcomings. For example, a serious problem facing frame differencing is the difficulty to update the background intelligently and frequently without containing the foreground inside. In other words, it shows a particular tendency to only highlight the the leading and trailing edges of a moving object in the foreground, while interior pixels with uniform intensity are not contained in the set of moving pixels, and there are also ghost pixels and objects. Such problems might make the subsequent process of motion analysis more difficult or even impossible to be achieved properly in some cases.

3.3.3 Optical flow

Generally speaking, optical flow can be viewed as the distribution of apparent velocities of motion of brightness patterns in a video sequence, which can emerge from relative movement of objects and the observer. As a result, optical flow can serve as a valuable source of voluminous and significant information not only about the spatial arrangement of the observed objects but also about the rate of change of such an arrangement [108]. Furthermore, the analysis of discontinuities of optical flow can be a great aid to the segmentation of a video sequence into its distinct regions corresponding to different moving objects inside the sequence. Optical flow attempts to approximate the local image motion based on local derivatives of a given image sequence. Particularly, it is to specify how much each pixel in the sequence moves between adjacent images. It is presumed that the movements of patterns cause some temporal changes in the image brightness, and all these changes are due to motion only. This assumption (so-called brightness constancy) is the central basis on which any estimation process of optical flow must be raised. Brightness constancy assumption is generally true, with isolated exceptions.[1]

With optical flow estimators, the image derivatives are often calculated by recursively applying a succession of low-pass and high-pass filters [109]. Hence the estimation process of optical flow usually involves a two-step procedure:

1. Measuring the spatio-temporal brightness derivatives (i.e., equivalently the same as measuring the velocities normal to the local intensity structures).

2. Integrating normal velocities into full velocities either locally via least squares regression [110], or globally via a regularization [111]

In the literature, there are a variety of approaches to estimate optical flow, which can be generally summarized in three categories, namely feature-based, correlation-based, and gradient-based. Among them, due to their mathematical simplicity and low computational demands, gradient-based algorithms have been rated as more attractive, and thus they have received and continue to receive much attention by many researchers in various fields and applications of computer vision.

[1]While the brightness constancy is often hypothesized by researchers, it may be violated in some limited cases. In such cases, the resulting flow field would be most likely to be a very poor approximation to the 2-D motion field. For instance, consider a uniform sphere rotating around its own central axis with a stationary light source. In this case, the intensity remains constant (i.e., no optical flow), so that no motion will be perceived. On the other hand, in a case of a static sphere with a moving light source, drifting intensities will be produced, hence some flow field will erroneously arise. The aforementioned example is visually depicted in Fig. 3.6.

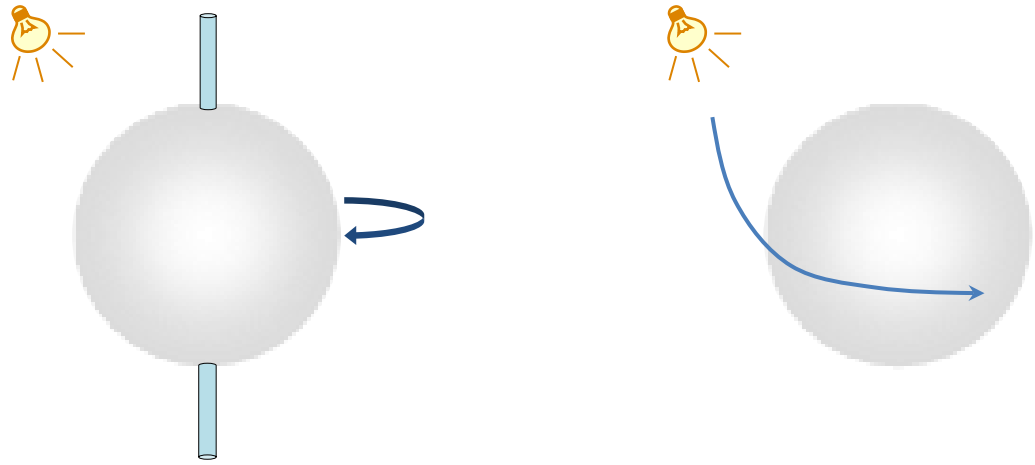


FIG. 3.6. Optical flow differs from actual motion field: (a) intensity remains constant, so that no motion is perceived; (b) no object motion exists, however moving light source produces shading changes.

Optical flow estimation

The point here is to show how the estimation process of optical flow field of a given video sequence can be modeled. First let $E(x, y, t)$ be the brightness of an image patch at location (x, y) in the image plane at time t. Let us again consider the patch is allowed to move a distance δx in the x-direction and δy in the y-direction in time δt (see Fig. 3.7). Based upon the hypothesize of brightness constancy, the brightness of the patch will remain unchanged, thus

$$E(x, y, t) = E(x + \delta x, y + \delta y, t + \delta t) \tag{3.26}$$

Expanding r.h.s. of Eq. (3.26) around (x, y, t) using first order Taylor's series gives

$$E(x, y, t) = E(x, y, t) + \delta x \frac{\partial E}{\partial x} + \delta y \frac{\partial E}{\partial y} + \delta t \frac{\partial E}{\partial t} + \varepsilon(\delta x, \delta y, \delta t) \tag{3.27}$$

where $\varepsilon(\delta x, \delta y, \delta t)$ is the remainder term of the Taylor expansion that indicates to the neglected second and higher order terms. Now subtracting $E(x, y, t)$ from both sides and then dividing both sides by δt we have

$$\frac{\delta x}{\partial t}\frac{\partial E}{\partial x} + \frac{\delta y}{\partial t}\frac{\partial E}{\partial y} + \frac{\partial E}{\partial t} + \mathcal{O}(\delta t) = 0 \tag{3.28}$$

where $\mathcal{O}(\delta t)$ is of order δt. It may be convenient, assuming that both δx and δy vary as δt). Taking the limit as $\delta t \to 0$ yields

$$\frac{\partial E}{\partial x}\frac{dx}{dt} + \frac{\partial E}{\partial y}\frac{dy}{dt} + \frac{\partial E}{\partial t} = 0 \tag{3.29}$$

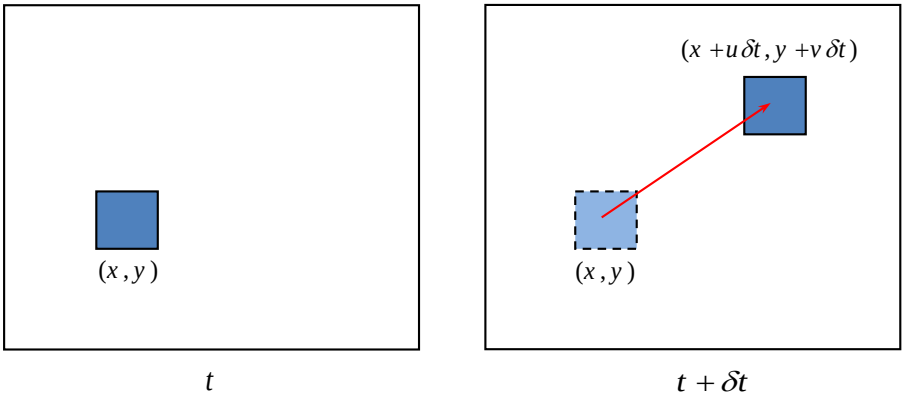


FIG. 3.7. Brightness constancy assumption: the brightness at image location (x, y) at time t is identical to that at location $(x + \delta x, y + \delta y)$ at time $t + \delta t$.

Let us take the abbreviations: $u = \frac{dx}{dt}$, $v = \frac{dy}{dt}$, then it is obvious that we have a single linear equation in the two unknown parameters:u and v,

$$E_x u + E_y v + E_t = 0 \tag{3.30}$$

where E_x, E_y, and E_t are the partial derivatives of image brightness with respect to x, y and t, respectively. The brightness invariance constraint expressed by Eq. (3.30) is visually depicted in Fig. 3.8. This equation can be rewritten in vector form as:

$$\vec{\nabla} E \cdot \vec{\mathbf{v}} = -E_t \tag{3.31}$$

where $\vec{\nabla} = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$ and $\vec{\mathbf{v}} = (u, v)$. As it is followed from the last equation, the component of motion in the direction of the brightness gradient is given by

$$\frac{-E_t}{\sqrt{E_x^2 + E_y^2}}.$$

It can be seen clearly that the component of the movement in the direction of the iso-brightness contours, at right angles to the brightness gradient cannot be determined. Subsequently, introducing additional constraints appear to be necessary at this time to calculate the flow velocity (u, v).

The smoothness constraint proposed by Horn and Schunck [111] is one of most popularly used constraints imposed in the determination of optical flow. As the name implies, this constraint forces flow vectors to vary smoothly. Mathematically, the smoothness constraint is imposed in optical flow determination by minimizing the square of the Laplacians of the x and y components of flow.

$$\arg\min_{u,v}(\nabla^2 u + \nabla^2 v) = \arg\min_{u,v}(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}) \tag{3.32}$$

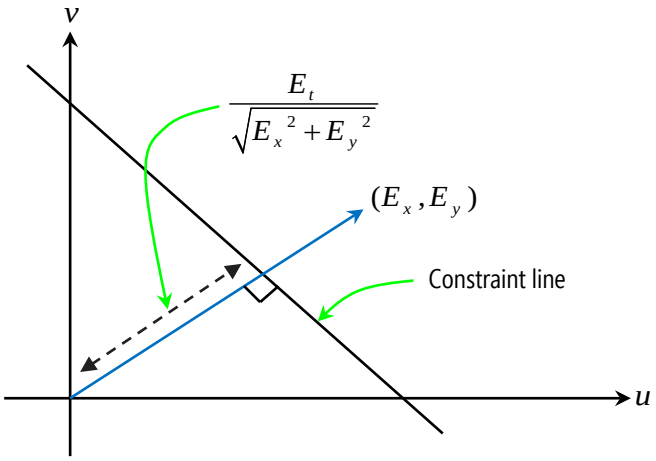


FIG. 3.8. Brightness invariance constraint.

To determine optical flow using the two previous constraints, first the partial derivatives (E_x, E_y, and E_t in Eq. (3.30)) and also the Laplacian of the ($\nabla^2 u$ and $\nabla^2 v$ in Eq. (3.32)) are calculated. Then, a weighted sum of the errors in the two constraints is minimized:

$$\varepsilon = \iint \left((E_x u + E_y v + E_t)^2 + \alpha^2 (\nabla^2 u + \nabla^2 v) \right) dxdy \tag{3.33}$$

where α is a weight between the two types of errors. Small values for α yield smoother flow fields, while high value indicate that the values of flow (i.e. u and v) are attracted strongly to the line (u, v) in Fig. 3.8, where probable solutions for the smoothness constraint of Horn & Schunck are resided. Finally optical flow can be calculated iteratively from the estimated derivatives and the average of the flow velocity estimates using the Gauss-Seidel iteration method [112].

$$\begin{aligned} u^{n+1} &= \bar{u}^n - E_x(E_x\bar{u}^n + E_y\bar{v}^n + E_t)/(\alpha^2 + E_x^2 + E_y^2) \\ v^{n+1} &= \bar{v}^n - E_y(E_x\bar{u}^n + E_y\bar{v}^n + E_t)/(\alpha^2 + E_x^2 + E_y^2) \end{aligned} \tag{3.34}$$

where n is the iteration number. It is worth mentioning that this method is categorized as a global approach, which is known to be more susceptible to noise.

Unlike to Horn and Schunck method, Lucas and Kanade [110] presented a non-iterative method that assumes a locally constant flow within a neighborhood $\{q_i\}_{i=1}^n$ of a fixed point p. Thus Eq. (3.30) can be assumed to hold for all pixels within a window centered at p. Thereby, a constrained system of n equations can be

established and written in matrix form as $A\mathbf{v} = b$, where

$$A = \begin{bmatrix} E_x(q_1) & E_y(q_1) \\ E_x(q_2) & E_y(q_2) \\ \vdots & \vdots \\ E_x(q_n) & E_y(q_n) \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} u \\ v \end{bmatrix}, \quad b = \begin{bmatrix} -E_t(q_1) \\ -E_t(q_2) \\ \vdots \\ -E_t(q_n) \end{bmatrix}$$

It appears clearly that this equation system is usually over-determined, as there more equations than unknowns. Based on the least square principle, the Lucas-Kanade algorithm obtains a compromise solution:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_i w_i E_x(q_i)^2 & w_i \sum_i E_x(q_i)E_y(q_i) \\ \sum_i w_i E_x(q_i)E_y(q_i) & w_i \sum_i E_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_i w_i E_x(q_i)E_t(q_i) \\ \sum_i w_i E_y(q_i)E_t(q_i) \end{bmatrix} \quad (3.35)$$

where w_i, $i = 1 \dots n$ are the weights of the weighted version of the least squares equation, which give more weight to the pixels that are closer to the central pixel; each w_i is usually set to a Gaussian function of the distance between q_i and p.

In a traditional approach, a dense flow field is first estimated, and then the scene is segmented based on the obtained motion information, where adjacent video components are merged together to form semantically meaningful object or video content of interest if they obey the same Hough or affine transformation motion model. However, dense field motion vectors are known to be susceptible to noisy data. Change detection is often used as a tool to exclude noisy optical flow; though it may induce some holes in the uniform regions [113].

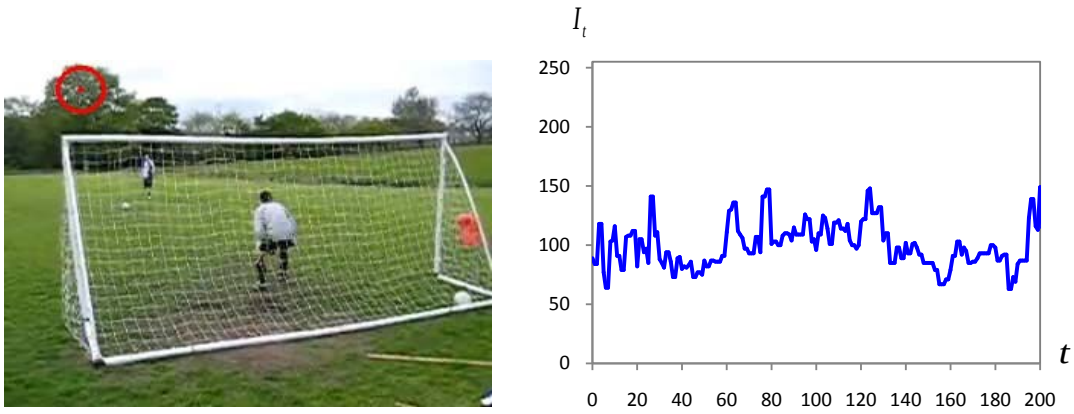
As in [114], we present a framework for a real-time automated traffic accident detection system using the HOF (Histogram of Optical Flow). In their approach, two major steps are performed. First, after estimating the flow fields based on the algorithm of [111], HOF-based features are extracted from video shots. Second, logistic regression is employed to develop a model for the probability of occurrence of accidents by fitting data to a logistic curve. In a case of occurrence of an accident, the trajectory of the vehicle by which this accident has been occasioned is determined. The presented HOF algorithm is structurally similar to that of the HOG (Histogram of Oriented Gradient) that was first introduced by [58] and essentially is a feature descriptor used for the purpose of pedestrian detection in static imagery, but they differ in that the HOF runs locally on optical flow field in motion scenes. Moreover, the HOF shows to be conceptually more simple and less time consuming to derive the the feature descriptors than the corresponding HOG.

3.3.4 Background modeling

For achieving highest sensitivity with lowest false alarms in the detection of moving objects, the detection of unusual motion is always desirable. Background subtraction is a widely used approach for detecting the unusual motion in a scene, which involves comparing each new frame to a designed model of the scene background. In a fully stationary scene, it would be very reasonable to model the intensity value of a pixel over time with a Gaussian distribution $N(\mu, \sigma^2)$, assuming the image noise is Gaussian mean zero $N(0, \sigma^2)$. Most often, the Gaussian distribution model of the intensity value of a pixel constitutes the underlying model for many background subtraction algorithms. For instance, a simple background subtraction technique involves first computing an average image of the scene of no moving objects, then subtracting each new frame from this image, and finally thresholding the result.

One major challenge that is commonly faced in many visual surveillance and monitoring applications dealing with outdoor scenes is that the background of the scene often contains many non-static noisy objects. The source of such a noise in the background could be the swaying of trees and the movement of grass due to the breeze of the wind in the scene. This would make the values of the pixel intensity change significantly over time. For instance, one pixel can be image of tree leaf at one frame, tree branch at another frame, the sky on a third frame and some mixture subsequently; that exhibits totally different color properties in each situation. Fig. 3.9 illustrates how the gray level of a vegetation pixel from a soccer scene changes over a short period of time (200 -frames).

Gaussian models have been widely applied for solving estimation problems in a variety of application areas [115]. It is a remarkable fact that Gaussian distributions remain Gaussian distributions after any linear transformation. This might serve as an explanation for why such models are important, and justify their success in many applications. Gaussian models are commonly used with many adaptive systems. For example, in surveillance applications, a Gaussian distribution is assumed to allow the system to be adaptive to different perturbations and changes, such as illumination changes, color inconsistencies, etc. It is worth mentioning that Gaussian mixture models are an example of a larger class of density models that have several functions as additive components [115]. A Gaussian mixture is essentially a Point Distribution Function (PDF), which can be normally expressed as a weighted sum of Gaussian densities. Let X_t be a pixel in the current frame I_t, where t is the frame index, and K is the number of distributions. Thus, each pixel



(a) (b)

FIG. 3.9. Temporal variation in the gray level of a vegetation pixel in a soccer scene: (a) a soccer video sequence where the center of the red circle is the location of the pixel of interest, (b) a plot for the intensity value of the pixel over time.

can be modeled separately by a mixture of K Gaussians as follows,

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} \eta(X_t; \mu_{i,t}, \Sigma_{i,t}) \tag{3.36}$$

where $\mu_{i,t}$ and $\Sigma_{i,t}$ are the i^{th} mean and covariance at time t, respectively, and $\omega_{i,t}$ is an estimate of the weight of the i^{th} Gaussian in the mixture at time t, where

$$\sum_{i=1}^{K} \omega_{i,t} = 1 \tag{3.37}$$

η denotes a Gaussian probability density function that is given by:

$$\eta(X_t; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t-\mu)^T \Sigma^{-1} (X_t-\mu)} \tag{3.38}$$

All the functions previously mentioned are joined together to create a combined density function, which is then employed to model colors of a dynamic scene or object. During constructing the model, all probabilities are evaluated for each pixel. The overall mean of the mixture is given by

$$\mu_t = \sum_{i=1}^{K} \omega_{i,t} \mu_{i,t} \tag{3.39}$$

which is the weighted sum of the means of the component densities of the mixture. For example, in [116–118] Gaussian mixture models are used as a basis to

model background distribution. In [118], the authors adopt a number of Gaussian functions as an approximation of a multimodel distribution in color space. While conditional probabilities are evaluated for all color pixels, probability densities are estimated from the background colors, clothing, heads, hands, etc. In their approach, two assumptions are proposed by the model. The first one states that a spatially contiguous region in the image plane is assumed to be generated by an object of interest. The second assumes that the set of colors for either the foreground or the background are relatively distinct. This implies that the pixels belonging to the foreground objects can be handle as a statistical distribution in the image plane.

In addition, in [119], an adaptive technique using Gaussian mixture model is proposed to build the tracker of a surveillance system. In this technique, each background pixel is modeled as a mixture of Gaussians. To determine which Gaussians are most likely to be portion of the background process, simple heuristics are used to approximate the Gaussians. Specifically, each pixel is modeled by a mixture of K Gaussians as stated previously in Eq. (3.36), where K is the number of distributions. In practice, the parameter K is normally chosen to be 3, 4 or 5. Before the foreground is detected, the background is updated, as follows: The value of each new pixel X_t is compared with all existing K Gaussian distributions. If X_t matches component *i* (i.e., X_t is within λ standard deviations of $\mu_{i,t}$), where λ is a parameter often chosen to be 2 or 2.5, the i^{th} component is then updated as follows:

$$\left.\begin{aligned} \omega_{i,t} &= (1-\alpha)\omega_{i,t-1} + \alpha \\ \mu_{i,t} &= (1-\rho)\mu_{i,t-1} + \rho X_t \\ \sigma^2_{i,t} &= (1-\rho)\sigma^2_{i,t-1} + \rho(X_t-\mu_{i,t})^{\mathsf{T}}(X_t-\mu_{i,t}) \end{aligned}\right\} \tag{3.40}$$

where $\rho = Pr(X_t|\mu_{i,t-1}, \Sigma_{i,t-1})$. α is a predefined learning parameter and $\sigma^2_{i,t}$ is the variance of the i^{th} Gaussian in the mixture at time t. On the other hand, for the remaining unmatched distributions, the parameters leave unchanged; however the corresponding weights $\omega_{i,t}$ are only updated as follows

$$\omega_{i,t} = (1-\alpha)\omega_{i,t-1} \tag{3.41}$$

If X_t matches none of the K components, then the least probable component (i.e., the component with the lowest weight) is replaced with a new one that has $\mu_{i,t} = X_t\Sigma_{i,t}$ large, and $\omega_{i,t}$ low [118]. After the updates, the weights $\omega_{i,t}$ are normalized. Strictly speaking, Gaussian distributions that have the most supporting evidence and the least variance can be specified to solve the background estimation problem. Since, the moving pixels generally have a higher variance than background pixels, the Gaussians are first ranked in decreasing order based on the value of $\omega_{i,t}/\|\Sigma_{i,t}\|$ in

(a) (b)

FIG. 3.10. Background estimation using MoG model with $K = 5$, $\tau = 0.5$: (a) An example snapshot from an original image sequence of a soccer scene, (b) Extracted foreground objects are shown in red.

order to represent background processes. By applying a threshold τ, the background distribution remains on top with the lowest variance, where

$$B = \arg\min_{b} \left(\frac{\sum_{i=1}^{b} \omega_{i,t}}{\sum_{i=1}^{K} \omega_{i,t}} > \tau \right) \tag{3.42}$$

In a case of proper normalization, the denominator in Eq. (3.42) is expected to tend to 1. Finally, all pixels X_t that match none of the components are best candidates to be marked as foreground. An example of the results of background estimation using MoG model with $K = 5$ and $\tau = 0.5$ is shown in Fig. 3.10. Finally, it may be worthwhile mentioning here that there is widespread agreement among researchers that flexibility is one of the comparative merits of the background estimation models based on mixture of Gaussians (MoG). It is claimed to be most advantageous for handling various variations in the background.

3.3.5 Summary and conclusion

In the second part of this chapter, we have discussed three main approaches for segmenting video sequences to objects which is the first step towards scene understanding and activity modeling. Frame differencing is one of the simplest and most convenient techniques for finding movement as a change between successive frames in a series of images. Computationally, it involves subtracting each incoming frame from background motion compensated previous frame and then thresholding the result. Even though frame differencing may possess some serious drawbacks,

such as its high sensitivity to noise, its often tendency to extract undesired regions from the background, and its low capability in detecting the inner parts of large objects, it is most robust to illumination changes.

On the other hand, The motion segmentation methods based on optical flow can provide valuable information not only about the size and locations of moving objects, but also about the velocities and directions. In addition, with these methods, substantial degree of precision can be achieved by applying an appropriate acceleration or velocity threshold to exclude the insignificant motions occurring in the background, such as moving trees/brush that are not interesting for the tracking or monitoring process. However, the motion segmentation methods based on optical flow are relatively imprecise and consume too much processing power as the neighborhood for each pixel should be sought to calculate the movement vector. Moreover, these methods tend to be quite sensitive to occlusion due to its dependence on brightness smooth changes.

The third category of the motion segmentation methods we have discussed previously are those that are based on background estimation and subtraction that play gigantic role in activity recognition and may affect the quality of the recognition outcome. Essentially, background estimation and subtraction is a critical step in moving object segmentation for video understanding and activity interpretation in which each pixel in a static scene is classified either as a part of the background or foreground, with the aim of distinguishing moving objects in the scene. Such techniques are based on maintaining a model for the background that models each pixel as a mixture of Gaussians. This model must be initialized and updated where a good initialization is crucial and a prerequisite for a correct segmentation. For example, small values for the update parameters may lead to integration of the moving objects that are inactive or no longer necessary for a while as a part of the background, while large values for the update parameters would create a need for a relatively long period of time for a stable estimation of the background and might explicitly account for failure in adaptation to sudden changes in illumination. This suggests that the update parameters and the initial background should be set carefully before running the updating process and for every new scene. The motion detection techniques that are based on maintaining a background model (e.g., MoG techniques) have the potential to learn the multimodal backgrounds automatically and adaptively and yield promising results in terms of adaptation and precision. Moreover, these techniques are independent of the velocity of moving objects and not prone to the common foreground aperture problem, but they are likely to be sensitive to dynamic scene changes due to lighting and extrinsic events.

CHAPTER 4

Features for Activity Recognition

4.1 Introduction

THE, ultimate goal of this current research is the recognition of human activities form video sequences. While this task appears intuitively as a trivial task for humans, yet it still remains very challenging for computers that usually fail to reach the accuracy of humans although this task has been tackled by numerous researchers during the past two decades or so. The main difficulty associated with activity recognition may lie in how to accurately determine proper tractable features that can reliably describe activities. The other common difficulty inherent in such a task is attributed to changes in pose, scale, orientation, location, imaging and lighting conditions, occlusions, and within-class shape variations. Feature extraction is a crucial step and important task in activity recognition to achieve the desired recognition performance, since subsequent classification and analysis processes depend greatly on the detected features.

Different researchers use different methods to extract robust features for the recognition of human activities and/or gestures in video sequences. Most of these methods often depend on using distinct features, such as shape and motion features that can be robustly extracted from human detection and tracking. In most cases, these features are extracted on shape and/or texture of segmented objects or based on a 3-D model constructed for each activity class, thus they depend heavily on the quality of the segmentation process, certain geometric constraints, and other heuristics. One of the major difficulties facing most of these methods of feature extraction is that it would be very difficult or almost impossible to achieve an

optimal segmentation of images that is still a complicated and error-prone problem. In addition, errors by a mismatched model or constraint can contribute to the malfunction or failure of the whole recognition system.

Unlike to the approaches mentioned above, appearance-based approaches (e.g., principal component analysis (PCA) and linear discriminant analysis (LDA)) use the whole image as features instead of considering local features. These approaches are very attractive and impressive in their ability to cope with real objects in real images, as they do not need image features or geometric primitives to be detected and matched. However, a major inherent limitation of these approaches lies in that they are essentially global approaches and thus can neither handle local variations nor deal robustly with partial occlusion, changes in illumination, and extraneous noise. In the subsequent sections of this chapter, the features that are covered by our approaches in this research will be explained and discussed in detail.

4.2 Interest Point-based Action Features

Due to their high compactness representation of video data and robustness to occlusions, background clutter, significant scale changes, and high activity irregularities, local features based on salient interest points have been successfully used for a wide variety of recognition tasks [7,51,59,120]. In in [8], we detect salient spatio-temporal interest points based on Harris detector [56]. Then a fuzzy log-polar histogram that comprises of the distribution of the interest points is constructed at each time slice. A similarity matrix reflecting the temporal similarities of the fuzzy histograms is constructed for each video clip (i.e., action snippet).

4.2.1 Space-time interest point detection

The process of detecting space-time interest points is basically built on the idea of the space interest point operators of Harris and Förstner [56]. Space-time Local structures in image where the image values have significant local variations in both space and time are detected. Hence, maximizing a normalized space-time Laplacian operator over space and time scales would allow the space-time extents of the detected events to be detected.

4.2.1.1 Interest points in space domain

Harris interest point detector [56] still retains superior performance to that of many competitors [121], whose pivotal idea is that the interest points are found at spatial locations where the appearance of a given image $f(x, y)$ changes significantly and

abruptly in both directions. Formally, let an image $f : \mathbb{R}^2 \to \mathbb{R}$ be modeled by its linear scale-space representation $L : \mathbb{R}^2 \times \mathbb{R}_+ \to \mathbb{R}$, as:

$$L(x, y; \sigma_l^2) = g(x, y; \sigma_l^2) * f(x, y) \tag{4.1}$$

where $*$ denotes the convolution operator and $g(x, y; \sigma_l)$ are Gaussian kernels of variance σ_l^2 and given by

$$g(x, y; \sigma_l^2) = \frac{1}{2\pi\sigma_l^2} \exp(-(x^2 + y^2)/2\sigma_l^2) \tag{4.2}$$

Consequently, for a given scale of observation σ_i^2, the second moment matrix identifying these points can be written as:

$$\begin{aligned} \mu(\cdot; \sigma_l^2, \sigma_i^2) &= g(.; \sigma_i^2) * ((\nabla L(.; \sigma_l^2))(\nabla L(.; \sigma_l^2))^T) \\ &= g(.; \sigma_i^2) * \begin{pmatrix} L_x^2(\cdot; \sigma_l^2) & L_x(\cdot; \sigma_l^2) L_y(\cdot; \sigma_l^2) \\ L_y(\cdot; \sigma_l^2) L_x(\cdot; \sigma_l^2) & L_y^2(\cdot; \sigma_l^2) \end{pmatrix} \end{aligned} \tag{4.3}$$

where σ_i is a variance of a Gaussian window over which the second moment matrix is integrated and L_x and L_y are the partial derivatives of $L(.; \sigma_l)$ with respect to x and y directions, respectively. The local derivatives are computed at local scale using Gaussian kernels as follows,

$$\begin{aligned} L_x(.; \sigma_l^2) &= \partial_x(g(x, y; \sigma_l^2) * f(x, y)) \\ L_y(.; \sigma_l^2) &= \partial_y(g(x, y; \sigma_l^2) * f(x, y)) \end{aligned} \tag{4.4}$$

The second moment descriptor can be viewed as the covariance matrix of a 2-D distribution of image orientations within the local neighborhood around a point. Therefore, the eigenvalues λ_1, λ_2, ($\lambda_1 \leq \lambda_2$) of the matrix $\mu(\cdot; \sigma_l^2, \sigma_i^2)$ can be used to describe the variations in $f(x, y)$ along both directions of image (see Fig. 4.1). Specifically, the sufficiently large values of the eigenvalues λ_1, λ_2 indicate the presence of an interest point. Clearly the larger the values of λ_1 and λ_2, the more likely of a point to be interest point. To detect such points, positive maxima of the *cornerness* functional Ω at each point in the given image is examined:

$$\begin{aligned} \Omega &= det(\mu(\cdot; \sigma_l, \sigma_i)) - \alpha\, trace^2(\mu(\cdot; \sigma_l, \sigma_i)) \\ &= \lambda_1\lambda_2 - \alpha(\lambda_1 + \lambda_2)^2 \end{aligned} \tag{4.5}$$

[1] where α is a tunable parameter that is commonly set to 0.04 in literature. Interest points are generally located at positive local maxima of Ω in a 3×3 neighborhood.

[1] In linear algebra, the *trace* (or character) of a square matrix is defined as the sum of the elements along the main diagonal.

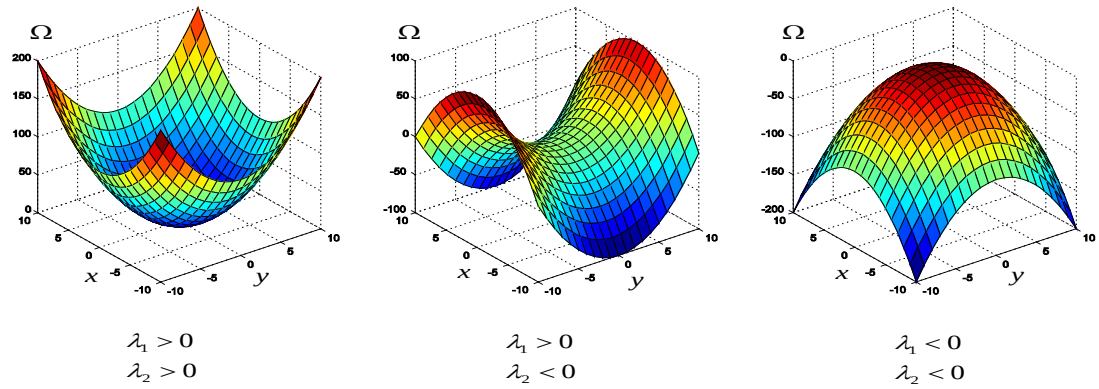


FIG. 4.1. The eigenvalues λ_1, λ_2 are proportional to the principal curvature.

Hence, it is easy to see that the ratio $r = \frac{\lambda_2}{\lambda_1}$ should be high at the locations where interest points reside. Further, Eq. (4.5) suggests that the ratio r must satisfy $\alpha \leq r/(1+r)^2$ for the positive local maxima of Ω. In this case, the positive maxima of Ω only agrees with ideally isotropic interest points with $\lambda_1 = \lambda_2$ (i.e., $\alpha = 0.25$). Consequently, low values of α would lead to the detection of interest points with more lengthened shape. Additionally it seems reasonable to get rid of unstable and weak maxima points. Hence only the maxima points of values greater than predetermined threshold are eligible to be nominated for being interest points.

4.2.1.2 Interest points in space-time domain

The idea of developing an operator responding to temporal dynamics in video sequences at specific locations was initially proposed by Laptev and Lindeberg [4], which is mainly based on extending the traditional notion of spatial interest points to include large variations along the time direction. Such points of these properties will match interest points in space domain with distinct locations in time that correspond to local space-time neighborhoods with non-stationary motion. Formally speaking, let us assume that for a given image sequence $f^{st} : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}$ its linear scale-space representation $L^{st} : \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R}^2_+ \to \mathbb{R}$ is constructed by convolving f with a Gaussian kernel with distinct space variance σ_l^2 and time variance τ_l^2:

$$L^{st}(x, y, t; \sigma_l^2, \tau_l^2) = g^{st}(x, y, t; \sigma_l^2, \tau_l^2) * f(x, y, t) \tag{4.6}$$

where the space-time separable Gaussian kernel is given by

$$g^{st}(.; \sigma_l^2, \tau_l^2) = \frac{1}{\sqrt{(2\pi)^3 \sigma_l^4 \tau_l^2}} \exp(-(x^2 + y^2)/2\sigma_l^2 - t^2/2\tau_l^2) \tag{4.7}$$

The space and time domains are generally independent that might explain why a sperate scale parameter τ_l^2 is used for the time domain. Furthermore, the dependence of the events detected by this type of points on both the space and time scales of observation justifies the need for a sperate treatment of such parameters σ_l^2 and τ_l^2. In the space-time domain, the second-moment matrix that is 3-by-3 matrix composed of first-order space-time derivatives of $L^{st}(.;\sigma_l^2,\tau_l^2)$ is given by

$$\mu = g^{st}(.;\sigma_i^2,\tau_i^2) * \begin{pmatrix} (L_x^{st})^2 & L_x^{st}L_y^{st} & L_x^{st}L_t^{st} \\ L_x^{st}L_y^{st} & (L_y^{st})^2 & L_y^{st}L_t^{st} \\ L_x^{st}L_t^{st} & L_y^{st}L_t^{st} & (L_t^{st})^2 \end{pmatrix} \tag{4.8}$$

where $g^{st}(.;\sigma_i^2,\tau_i^2)$ is a Gaussian weighting function with the integration scales σ_i^2 and τ_i^2 that relate to the previous local scales by these linear relations: $\sigma_i^2 = s\sigma_l^2$ and $\tau_i^2 = s\tau_l^2$, where s is an arbitrary constant. Similar to the space domain, the first-order partial derivatives are given by:

$$\begin{aligned} L_x^{st}(.;\sigma_l^2,\tau_l^2) &= \partial_x(g^{st} * f) \\ L_y^{st}(.;\sigma_l^2,\tau_l^2) &= \partial_y(g^{st} * f) \\ L_t^{st}(.;\sigma_l^2,\tau_l^2) &= \partial_t(g^{st} * f) \end{aligned} \tag{4.9}$$

To detect the space-time interest points, similar to spatial domain, regions in f that have considerable eigenvalues $\lambda_1, \lambda_2, \lambda_3$ of μ are searched for. One direct way to achieve this is to extend the Harris corner function given in Eq. (4.5) to the space-time domain as follows:

$$\begin{aligned} \Omega^{st} &= det(\mu) - \alpha trace^3(\mu) \\ &= \lambda_1\lambda_2\lambda_3 - \alpha(\lambda_1+\lambda_2+\lambda_3)^3 \end{aligned} \tag{4.10}$$

Yet, it is fairly easy to see that points with large values of $\lambda_1, \lambda_2, \lambda_3 (\lambda_1 \leq \lambda_2 \leq \lambda_3)$ correspond to positive local maxima of Ω^{st}. Hence, the finding of local positive space-time maxima of Ω^{st} would ensure the detection of space-time interest points of f. Fig. 4.2 provides an example for space-time interest point detection in a sample image sequence that shows a person performing "drinking" action.

The primary contribution of our work in [8] has been the proposal of an effective fuzzy approach that is mainly based on the structural information of spatio-temporal interest points. It is highly expected that the distribution of the interest points to be very compact and representative features for a particular action and thus for action recognition. Therefore, the shapes of such points are most likely to be very similar for actions belonging to the same class and they able to distinguish these actions from other actions of different classes. Nevertheless, the 3-D

FIG. 4.2. Example space-time interest point detection. The image sequence shows a human subject performing "drinking" action.

distributions of interest points often appears to be not sufficiently discriminative to distinguish actions clearly due to the high inter-action similarity and the high intra-action variability. The projection of point clusters to a number of planes of lower dimensional can offer a simple and efficient solution to resolve these types of interference issues, which enables us to clearly explain the ambiguity between different types of actions. It is expected that this would lead to the achievement of more discriminative motion representation that can handle different viewpoints.

In several applications of object recognition, for simplicity, three perpendicular planes (e.g., x-y plane, t-x plane, and t-y plane) are frequently used to which the detected 3-D (x-y-t) interest points can be projected. For example, a 3-D (x-y-t) scatter plot for the point clusters of the image sequence of the drinking person given in Fig. 4.2 is visually shown in the top row of Fig. 4.3, while the projections of these point clusters to x-y plane, y-t plane, and y-t plane are shown in the second row of this figure (form left to right in order). As can be seen in the figure, after projection the shapes of the interest points of action are more clearly defined in 2-D than in 3-D; accordingly, the structural relations among these points would enable to get the feature representing space-time structure of action shapes form the information of projected points. The shape context algorithm [122] is then applied on each projection plane regraded as a 2-D image. In the original shape context approach, a

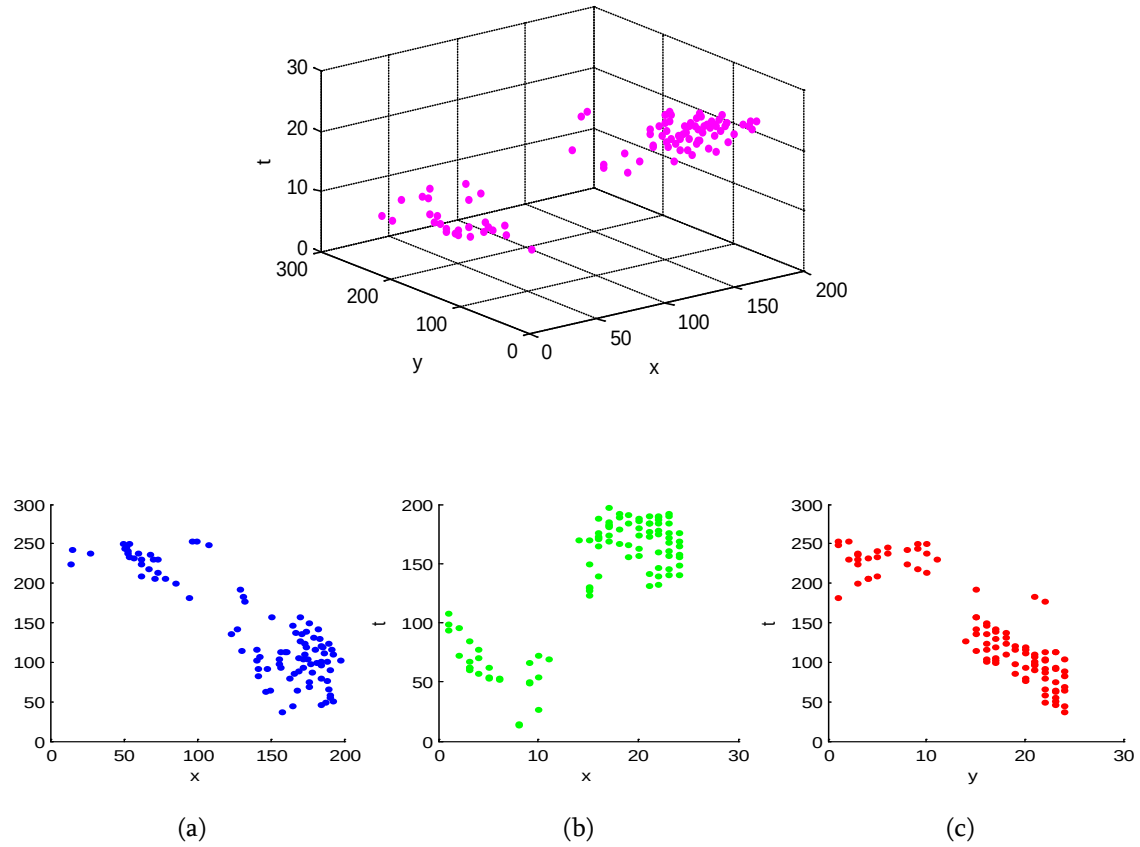


FIG. 4.3. Space-time interest point distribution in x-y-t space (top row) of the sequence of "drinking" action given in Fig. 4.2 and its projections (bottom row) to (a) x-y plane, (b) t-x plane, and (c) t-y plane.

discrete set of detected edge points is used for representing an object shape. For each edge point, a shape context descriptor is calculated on the remaining points.

In our work of [8], the original concept of shape context is fuzzified based on so-called fuzzy log-polar histograms. In contract to the original shape context, the new fuzzy version of shape context is applied on the detected interest points rather than on the edge points. Then each fuzzy histogram at each temporal state of action is flattened to be a feature vector representing the action pose at this state. This method highlights the information of temporal shape variations that intuitively appear to provide a crucial cue for action modeling and recognition.

4.2.2 Fuzzy log-polar histogram

As stated before, the key idea of a fuzzy log-polar histogram is essentially ground on the division of a given video clip (i.e., action snippet) into several time-slices.

These slices are defined by fuzzy intervals. Gaussian membership functions appear to be most appropriate to represent such intervals, which can be given as

$$\mathcal{G}_j(t;\varepsilon_j,\sigma,\gamma) = e^{-\frac{1}{2}\left|\frac{t-\varepsilon_j}{\sigma}\right|^{\gamma}},\ j=1,2,\ldots,m \tag{4.11}$$

where ε_j, σ, and γ are the center, width, and fuzzification factor of time slice, respectively, while m is the total number of all time slices. The membership functions defined above are chosen to be of identical shape on condition that their sum is equal to one at any instance of time. In this approach, it is expected that the use of such fuzzy membership functions would lead not only to a most reduced performance decline resulting from time warping effects, but also to efficient extraction of local features of action shape. To extract the local features of the shape representing action at an instance of time, temporal localized shape context is developed, inspired by the original idea of shape context. Compared with the shape context [122], this localized shape context differs in meaningful ways. The key idea behind such a modified shape context is based on computing rich descriptors form fewer interest points (i.e., keypoints). The shape descriptors presented here calculate the log-polar histograms on condition that they are invariant to simple transforms like scaling, rotation and translation. The histograms are normalized for all affine transforms as well. Furthermore the shape context is reasonably extended by combining local descriptors with fuzzy memberships functions.

An human action can be distinctly viewed to be composed of a series of body poses over time. Reasonable estimate of a pose can be constructed using a small set of keypoints. Ideally, such points are distinctive, persist across minor variation of shapes, robust to occlusion, and do not require segmentation. Let $\mathcal{S}$ be a set of n detected keypoints that represents the status of a given action at an instance time:

$$\mathcal{S} = \{p_i = (x_i, y_i) \in \mathbb{R}^2 | i = 1, 2, \ldots, n\} \tag{4.12}$$

Then, for each keypoint $p_i \in \mathcal{S}$, the log-polar coordinates (i.e. radial distance ρ_i and angle η_i) are, respectively, given by

$$\begin{aligned} \rho_i &= \log\left(\sqrt{(x_i - g_x)^2 + (y_i - g_y)^2}\right) \\ \eta_i &= \arctan\left(\tfrac{y_i - g_y}{x_i - g_x}\right) \end{aligned} \tag{4.13}$$

where g_x and g_y are the two spatial components of the center of mass (i.e., centroid) of the keypoint set $\mathcal{S}$, which are simply given by

$$g_x = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad g_y = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{4.14}$$

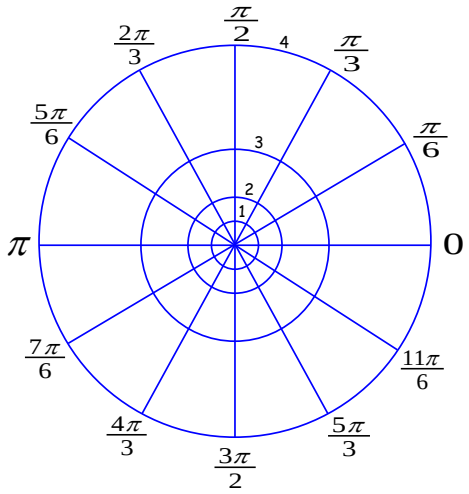


FIG. 4.4. A log-polar histogram with 4 and 12 bins for orientation and magnitude.

Fig. 4.4 shows an example of a log-polar histogram with 32 bins in total (12 bins for orientation, 4 bins for magnitude), which is centered at the centroid of the set $\mathcal{S}$. It is pertinent to mention here that the centroid of the point set is invariant to linear transformations, including translation, scaling, and rotation. For this, the angle η_i is computed with respect to a horizontal line passing through the center of mass. Now, in order to compute a modified version of shape context, a log-polar histogram is overlaid on the keypoint set $\mathcal{S}$ that represent the action shape. Thus the fuzzy log-polar 2-D histograms for the modified shape context of action can be constructed at each time-slice j as follows,

$$\hbar_j(k_1, k_2) = \sum_{\substack{\rho_i \in bin(k_1),\\ \eta_i \in bin(k_2)}} \mu_j(t_i),\ j = 1, 2, \ldots, m \tag{4.15}$$

where k_1 and k_2 are two indices for the point magnitude and orientation respectively and t_i is the frame number to which the point belong. For better comparison purposes, the previous 2-D histograms are then converted into 1-D histograms by applying a simple linear transformation on the histogram indices k_1 and k_2:

$$h_j(k) = \hbar_j(k_1 d\eta + k_2),\ k = 0, 1, \ldots, d\rho d\eta - 1 \tag{4.16}$$

where $d\rho$ and $d\eta$ are the total number of bins for magnitude and orientation, respectively. Fig. 4.5 presents a visual depiction of the temporal division of a snippet of a running action (into m time-slices) and its corresponding fuzzy log-polar histograms, one corresponding to each time-slice. The resulting 1-D histograms are then normalized to achieve robustness to scale variations. These normalized histograms are used as shape contextual information for classification and matching.

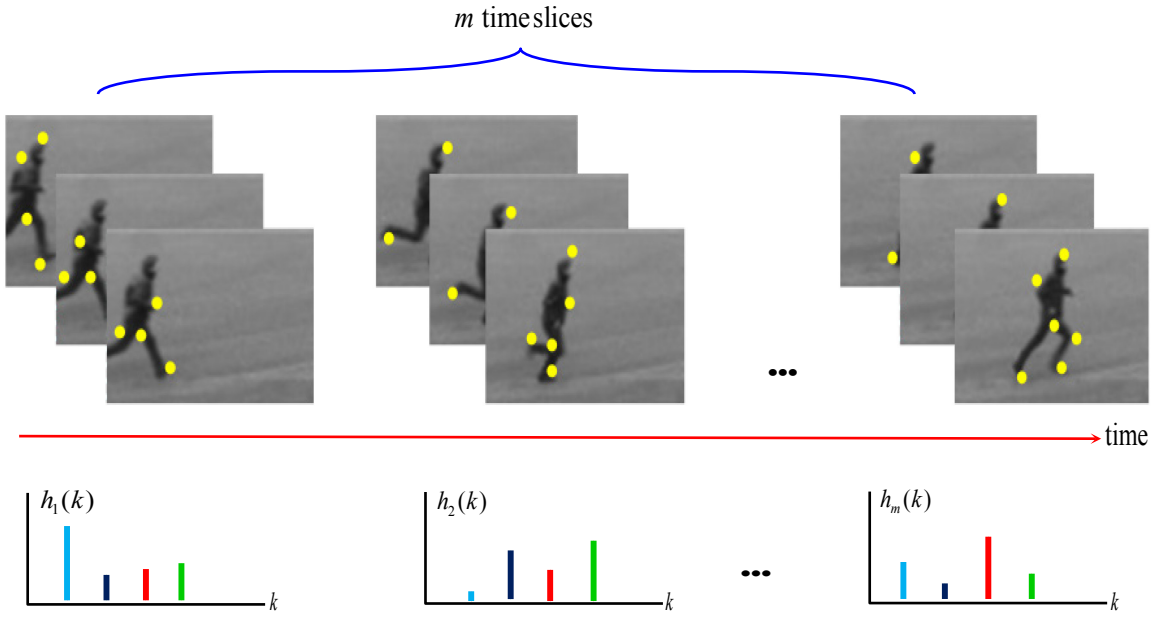


FIG. 4.5. Illustrative visualization for temporal slicing of a video sequence of running action and the corresponding log-polar histograms representing the spatio-temporal shape contextual information of that action.

Finally, the normalized histograms are concatenated to to obtain one descriptor per video. Subsequently, any classification algorithm, such as ANNs, SVMs, HMMs, etc. can be trained on the final descriptors in order to learn and recognize actions.

4.3 Invariant Shape-based Features

As stated previously, the process of feature extraction and selection is deemed to be a core component of any activity recognition system, but is also very challenging and time consuming. Interestingly, shape cues tend to be extracted more efficiently and are more robust to appearance variations, so that shape-based features have shown to be very successful for many recognition tasks such as object and scene recognition. It may seem obvious that using a variety or combination of distinct shape features helps in ensuring robust human pose estimation and thus leads to distinguish between different action categories best. This would definitely affect the overall performance of any proposed human action recognition system. In particular, in [123, 124], we have utilized the shape features that can be extracted from the segmented silhouettes of moving human body parts in order to represent the action poses. Such shape features have the potential to provide a rich source of information for the interpretation/analysis of human motion.

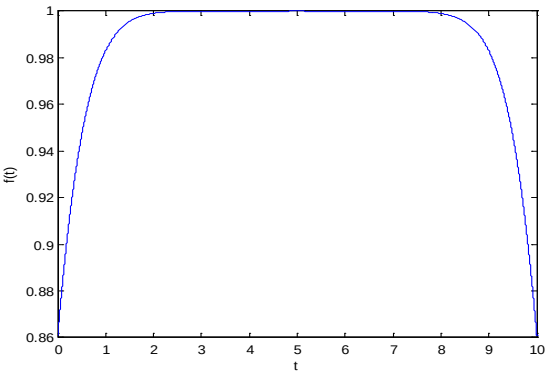


FIG. 4.6. An example of a membership function used to represent the temporal interval, with $\alpha = 5$, $\beta = 6$, and $\gamma = 10$.

Moreover, the motion information that can be extracted by following the trajectory of the motion centroid can be integrated with shape features, as will described later by end of this chapter. Such a combination between these two types of information can significantly increase the discriminative power for action recognition. Before starting the feature extraction procedure, we first temporally split each video snippet into several time-slices. Similarly to what done in the previous section, these time-slices are defined by fuzzy intervals. Each of these intervals is described by a fuzzy membership function defined as follows:

$$f(t;\alpha,\beta,\gamma) = \frac{1}{1 + (|\frac{t-\alpha}{\beta}|)^{\gamma}} \tag{4.17}$$

where α, β, and γ are the center, width, and fuzzification factor of the interval, respectively as shown in Fig. 4.6. We have opted to allow all the membership functions to be of identical shape on condition that their sum is equal to one at any instance of time t. It is experimentally observed that using this type of functions allows not only the probable degradation in recognition performance caused by time warping effects much more tolerable, but also could enable local shape features to be extracted more reliably. Regarding shape features, we consider here a variety of invariant descriptors, such as Fourier descriptors, curvature features, invariant shape moments, etc. In the subsequent subsections, we illustrate in a little more detail how these features and descriptors are defined and extracted.

4.3.1 Fourier descriptors

The application capabilities of Fourier Descriptors (FDs) for object recognition have been successfully demonstrated through a number of case study examples. For example, several studies based on FDs have been successfully conducted to

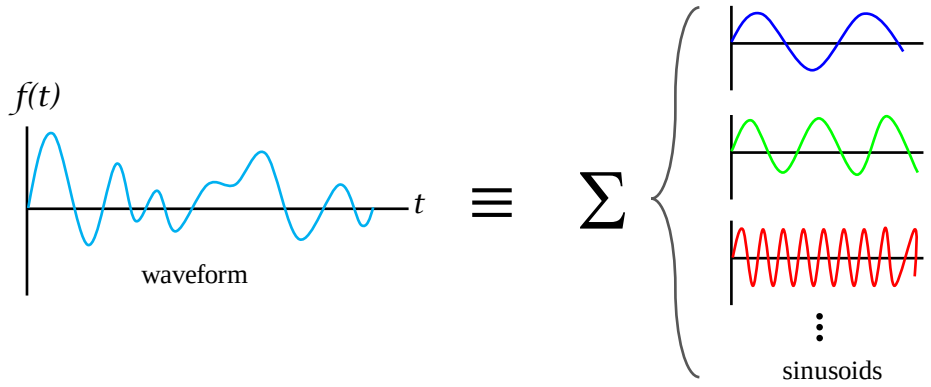


FIG. 4.7. Fourier analysis to show how an arbitrary periodic function $f(t)$ can be written in terms of a linear combination of sinusoids with different frequencies and amplitudes.

recognize different types of marine life, product deformations, tree leaves, etc. The pivotal idea of FDs is to use the Fourier transformed outline as the shape feature. Further, the idea of Fourier transforms is, in turn, viewed as a natural extension of the concept of Fourier series that involves expressing any waveform $f(t)$ in terms of a linear combination of simpler functions (i.e., sinusoids) of different frequencies and amplitudes (see Fig. 4.7):

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \sin(n\omega t) + b_n \cos(n\omega t) \tag{4.18}$$

where $\omega = 2\pi/\tau$; τ is the period of the waveform. The coefficients a_0, a_n, and b_n are called trigonometric coefficients. Such simple functions are often thought of as building blocks. Using the well-known Euler's equality: $\exp(i\phi) = \cos(\phi) + i\sin(\phi)$, it is not difficult to derive the exponential or the complex form of the Fourier series from Eq. (4.18) as follows,

$$f(t) = \sum_{n=-\infty}^{\infty} c_n \exp(in\omega t),\ i = \sqrt{-1} \tag{4.19}$$

The last equation that is seen as a much shorter formula for Fourier series relates directly to the sinusoidal form; however the new coefficients c_n are complex in general and can easily be determined by the following equation:

$$c_n = \frac{1}{2\tau} \int_{-\tau}^{\tau} f(t) \exp(-in\omega t) dt \tag{4.20}$$

The above equation provides a conceptual base on which FDs as a metric characterizing the boundary shape of object of interest can be established. In essence, FDs depend on the notion of the shape signature that is one dimensional function

derived from the shape function [125, 126]. More formally, given a shape whose outline (i.e., shape contour $\mathcal{C}$) is defined by a periodic complex function as follows,

$$\mathcal{C} = \{z_n \in \mathbb{C} : z_n = x_n + iy_n,\ 0 \leq n < N\} \tag{4.21}$$

where $\mathbb{C}$ is the set complex number and $x_n, y_n \in \mathbb{R}$ are the spatial coordinates of the outline points of the shape. Hence, one simple shape signature (i.e., centroid distance function) derived from the complex coordinates of the outline points of shape can be defined as follows,

$$r_n = |z_n - \tilde{z}|,\ 0 \leq n < N \tag{4.22}$$

where $\tilde{z} = \tilde{x} + i\tilde{y}$ is the centroid of the shape whose coordinates are given by

$$\begin{aligned} \tilde{x} &= \tfrac{1}{6A}\textstyle\sum_{n=0}^{N-1}(x_n + x_{n+1})(x_n y_{n+1} - x_{n+1}y_n) \\ \tilde{y} &= \tfrac{1}{6A}\textstyle\sum_{n=0}^{N-1}(y_n + y_{n+1})(x_n y_{n+1} - x_{n+1}y_n) \end{aligned} \tag{4.23}$$

where A denotes the total shape's area defined by:

$$A = \frac{1}{2}\left|\sum_{n=0}^{N-1}(x_n y_{n+1} - x_{n+1}y_n)\right| \tag{4.24}$$

The location of the shape centroid is deemed to be fixed with different points distribution on a contour (i.e., this location is fixed no matter how the distribution of boundary points is). It is pertinent to mention that other examples of shape signatures widely used for shape representation and recognition may include: complex coordinates, tangent angle, curvature, area, arc length, etc. The most exciting prospect of shape signature is its potential ability to capture the most perceptual feature of the shape [127]. To obtain FDs for a given shape, first the shape is represented by a shape signature that is a 1-D function derived from the outline coordinates of the shape. Then, discrete Fourier transform is applied to the signature to obtain Fourier transformed coefficients. More formally, the coefficients $c_k,\ 0 \leq k < N$ of the discrete Fourier transform can be calculated as follows,

$$c_k = \frac{1}{N}\sum_{n=0}^{N-1} r_n \exp\left(-\frac{2\pi i}{N}nk\right),\ 0 \leq k < N \tag{4.25}$$

where $r_n, 0 \leq n < N$ is a shape signature at the n-th boundary point, which is defined by Eq. (4.22). Perhaps, one of the crucial inherent traits of Fourier series analysis is that the original signal (i.e., shape signature) can be perfectly reconstructed using the inverse discrete Fourier transform:

$$r_n = \frac{1}{N}\sum_{k=0}^{N-1} c_k \exp\left(\frac{2\pi i}{N}nk\right),\ 0 \leq n < N \tag{4.26}$$

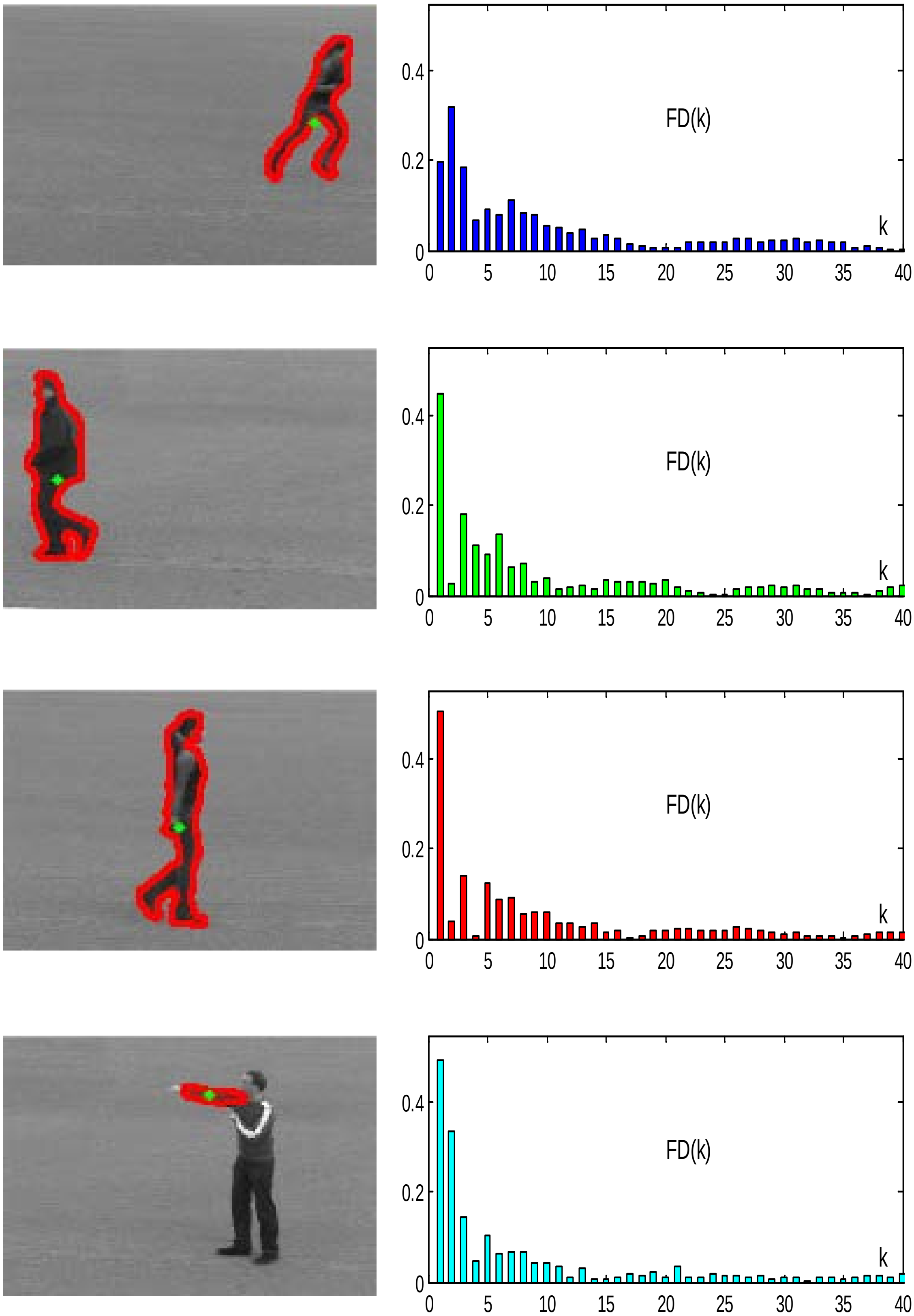
What would be interesting to point out here is the fact that FDs obtained using different shape signatures are likely to vary significantly in terms of their overall performance. For instance, in [125, 128], it has argued that FDs obtained using the centroid distance function are superior to those obtained using other shape signatures in their overall performance and their relative accuracy on shape recognition. As the centroid distance function r_n defined by Eq. (4.22) is just invariant under rotation and translation, hence the Fourier coefficients given in Eq. (4.25) should be further normalized in order to be also invariant under scaling and change of the starting point on the contour. Strictly speaking, based on Fourier transform theory, shape descriptors can be derived from Fourier coefficients as follows. The first two coefficients (i.e., c_0 and c_1) are first truncated from the Fourier coefficients c_n. The phase information are then ignored and only magnitude of the remaining coefficients are used after dividing each of them by c_1. To summarize, the Fourier shape descriptors are formally written as:

$$\mathcal{FD} = \left\{ d_n = \frac{|c_{n+1}|}{|c_1|},\ 0 < n < N-1 \right\} \tag{4.27}$$

where $|\cdot|$ is the modulus operator. It is easy to verify that such a choice of the coefficients ensures that the resulting shape descriptors are invariant to shape translation, rotation and scaling, and they are independent of the choice of the starting point on the contour. Fig. 4.8 shows bar plots visualizing the FDs based features extracted from the motion shapes for six different sequences of running, jogging, walking, boxing, waving, and clapping actions from top to bottom, respectively. As a final remark here, it is perhaps interesting to mention that FDs possess several desirable properties (e.g., simple derivation, simple normalization, simple to do matching, and their robustness to noise). All the aforementioned advantages allowed these descriptors to be very popular in both scientific and the industrial societies with high potential for many applications. Furthermore, as convincingly argued in [128], for efficient shape retrieval, 10 Fourier coefficients have been shown to suffice to describe adequately the shape information of detected features.

4.3.2 Moment invariants

In many applications, image objects can be efficiently recognized from imagery independently of their scale, position, and orientation by representing each of these objects of interest with a set of measurable descriptors. For efficient object representation and comparison, such descriptors are typically defined by invariant features extracted from various imagery types and any a priori knowledge available. There is a rich literature of various feature recognition techniques that utilize spatial


FD(k)
k
0.4
0.2
0
0 5 10 15 20 25 30 35 40
FD(k)
k
0.4
0.2
0
0 5 10 15 20 25 30 35 40
FD(k)
k
0.4
0.2
0
0 5 10 15 20 25 30 35 40
FD(k)
k
0.4
0.2
0
0 5 10 15 20 25 30 35 40

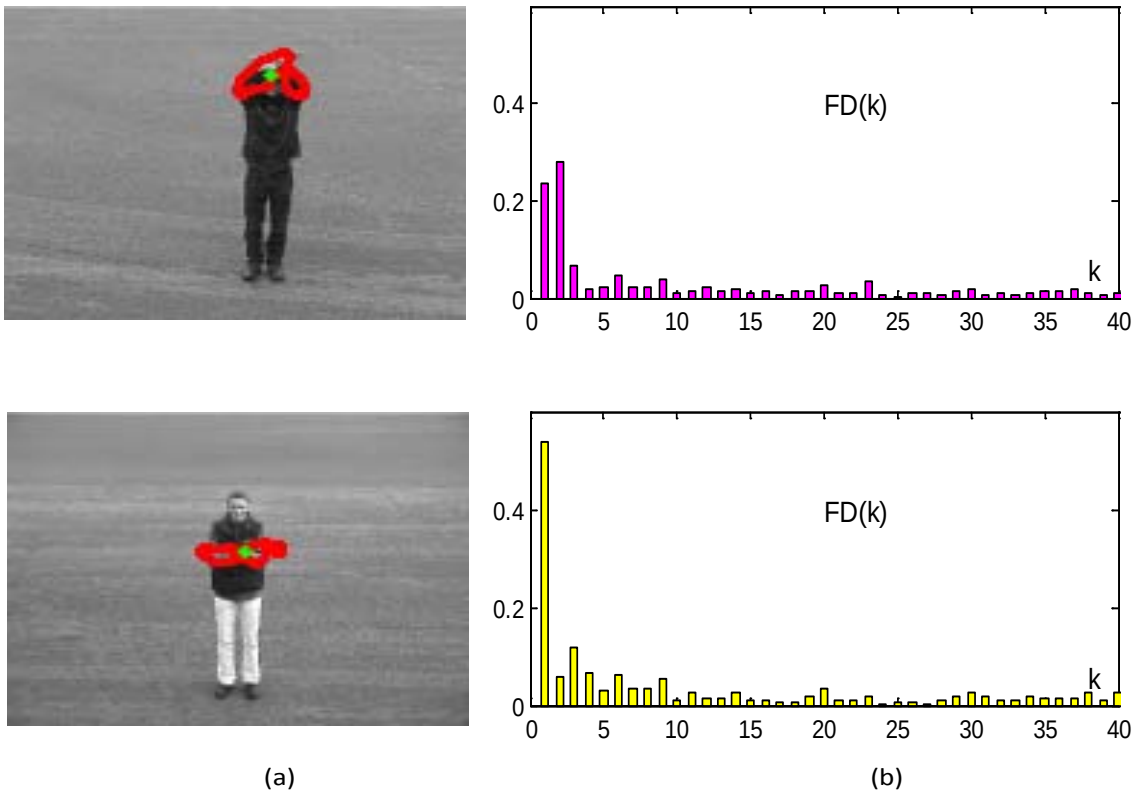


FIG. 4.8. FDs for motion shape description: (a) original sequences with motion shapes for six different actions of running, jogging, walking, boxing, waving and clapping from top to bottom respectively; the green circle within each shape locates the shape centroid, (b) the corresponding FDs obtained for the shapes shown in (a).

moments to construct such invariant features [129–131]. Many of these techniques are essentially based on the general moment theory widely known and applied in research in several areas of statistics and mechanics.

In particular, geometric moments have vast practical applications in many area of computer vision and invariant pattern recognition, ranging from lower-level recognition such as pose estimation to higher-level recognition such as activity recognition and analysis. When applied to images, they were identified to be most descriptive of the image contents (i.e., intensity distribution) with respect to its axes. Once such moments are properly defined, both global and detailed geometric information of image contents can be reasonably expected to be detected robustly. In such a scenario, moments would be able to characterize various image objects such that the properties with analogies in statistics or mechanics are extracted, and thus the shape of all objects of interest can be described well. Formally speaking, in continuous domain, an image is viewed as a 2-D Cartesian density distribution function $f(x, y)$. Under such a continuity assumption, the general form of the

geometric moments of order $(p+q)$ for the function $f(x,y)$, evaluated over the entire plane Ω is defined by the following double integrals:

$$M_{pq} = \int\int_{\Omega} \psi_{pq}(x,y) f(x,y) dxdy, \quad p,q = 0,1,2,\ldots\infty \tag{4.28}$$

where ψ_{pq} is a basis function or weighting kernel by which a weighted description for the image function $f(x,y)$ across the entire plane Ω is generated. Several advantageous properties of the basis functions are expected to be provided to the moments, which can potentially enable extracted features and descriptions to be invariant to image scaling, translation, and rotation, and partially invariant to illumination changes. Now, to apply this concept of moments to digital images, Eq. (4.28) needs to be transformed to the discrete domain. Note that the probability density function $f(x,y)$ (of a continuous distribution) is different from that of the probability of a discrete distribution. For convenience, it seems intuitively plausible, assuming that the plane ξ is partitioned into small squared regions of size 1×1 pixels, with fixed intensity $I(x,y)$ over each squared region. So if we let that P_{xy} is a discrete pixel value at a spatial location (x,y), then we can write:

$$P_{xy} = I(x,y) \triangle A \tag{4.29}$$

where $\triangle A$ denotes to the region area (that is equal to unity in this case). Hence, analyzing over the entire discrete intensity plane of image would eventually yield the following discrete form for Eq. (4.28):

$$M_{pq} = \sum_{y} \sum_{x} \psi_{pq}(x,y) I(x,y), \quad p,q = 0,1,2,\ldots\infty \tag{4.30}$$

It is perhaps worthwhile to point out here that the choice of above basis functions ψ_{pq} greatly depends on the application of use, and on the invariant properties desired. Furthermore, it is expected that choosing a specific basis function results in some constraints, such as to restrict the range of the image coordinates, x and y, enable the image and its descriptors to be translated to other coordinates (e.g., polar coordinates), etc. In [129,132], Hu stated that the 2-D Cartesian moment of order $(p+q)$ for an $m \times n$ discretized image, $I(x,y)$ can be defined by taking the basis function in Eq. (4.30) as a monomial of power $p+q$ (product of powers of the variables x and y, i.e., $\psi_{pq}(x,y) = x^p y^q$) as follows,

$$M_{pq} = \sum_{y=0}^{n-1} \sum_{x=0}^{m-1} x^p y^q I(x,y), \quad p,q = 0,1,2,\ldots\infty \tag{4.31}$$

The full moment set of order k that includes all moments, M_{pq}, such that $p+q \geq k$ compromises of exactly $\frac{1}{2}(k+1)(k+2)$ elements. Ever since the pioneering work

of Hu [129] on moment functions that has explored quite thoroughly the use of moments for image analysis and object representation, a broad range of new applications utilizing moment invariants in image analysis and pattern recognition fields has started to evolve. It is clear that the Cartesian moments given by Eq. (4.31) are not invariant to geometric transformations. To achieve invariance under translation, these moments are calculated with respect the center of mass as follows,

$$\mu_{pq} = \sum_{y=0}^{n-1} \sum_{x=0}^{m-1} (x - \bar{x})^p (y - \bar{y})^q I(x, y), \quad p, q = 0, 1, 2, \ldots \infty \tag{4.32}$$

where $\bar{x}$ and $\bar{y}$ are the coordinates of the centroid and given by:

$$\bar{x} = \frac{M_{10}}{M_{00}}, \qquad \bar{y} = \frac{M_{01}}{M_{00}} \tag{4.33}$$

After a bit tedious but straightforward manipulation, equations (4.32) and (4.31) lead to the following relation between the Cartesian and centralized moments

$$\mu_{pq} = \sum_{i}^{p} \sum_{j}^{q} \binom{p}{i} \binom{q}{j} (-\bar{x})^{p-i} (-\bar{y})^{q-j} M_{ij} \tag{4.34}$$

However, it should be emphasized that the expression in Eq. (4.32) suggests that the centralized moments are only invariant to translation. To enable invariance under scale changes, the 2-D centralized moments μ_{pq} need to be normalized to obtain scale-normalized centralized moments η_{pq} as follows,

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \tag{4.35}$$

where the exponent γ is given in terms of p and q as follows,

$$\gamma = \frac{p + q}{2} + 1, \quad p + q \geq 2$$

On the basis of what has been stated before the non-orthogonal centralized moments are translation invariant, and it was easy to normalize them to changes in scale. However, to consider them appropriately as features for action recognition, they also need to be calculated irrespective to rotation variances. To enable these moments to be rotation-invariant, they has to be reformulated. Strictly speaking, two methods are used to allow the scale-normalized centralized moments to be rotation-invariant. The first one depends mainly on so-called principle axes, which has been shown to be most prone to serious stability problems, especially when applied to images lacking unique principle axes (i.e., rotationally symmetric images). The second one is that of absolute moment invariants. In this latter method, the derivation of

expressions are made from applying algebraic invariants to the moment function under a rotation transformation. As a result, a collection of nonlinear expressions of centralized moments can be found, which generates a family of sets of absolute moment invariants. As an example, a set of moment invariants [129], can be derived based on the following nonlinear expressions of centralized moments:

$$\phi_r = |I_{p-r,r}|^2, r = 1, 2, \ldots, p - 2r \tag{4.36}$$

where,

$$I_{p-r,r} = \sum_{l=0}^{r} (-j)^l \binom{p-2l}{l} \sum_{k=0}^{r} \binom{r}{k} \mu_{p-2k-l,2k+l} \tag{4.37}$$

where $p - 2r > 0, j = \sqrt{-1}$. In a simple experiment of invariant object recognition, this set has proven to be able to efficiently recognize several typed characters.

4.3.3 Moment-based features

Besides the moment invariants discussed above, a set of other features derived from the moments of second order can also be extracted and added to the final feature vector [124]. The existent analogy between image moments and mechanical moments would contribute to a deeper understanding of the central moments of second order, i.e. $\{\mu_{20}, \mu_{11}, \mu_{02}\}$, known as the moments of inertia. The features extracted here can be derived from the covariance matrix defined as:

$$\mathcal{J} = \begin{bmatrix} \mu'_{20} & \mu'_{11} \\ \mu'_{11} & \mu'_{02} \end{bmatrix} \tag{4.38}$$

where the matrix elements are explicitly given by:

$$\mu'_{20} = \frac{\mu'_{20}}{\mu'_{00}}, \quad \mu'_{11} = \frac{\mu'_{11}}{\mu'_{00}}, \quad \mu'_{02} = \frac{\mu'_{02}}{\mu'_{00}}$$

From the covariance matrix of moments given above, several features can be derived from the central moments of second order. First, the principal axes can be determined by calculating the eigenvalues of the covariance matrix as follows,

$$\lambda_{1,2} = \frac{1}{2}\left[\mu'_{20} + \mu'_{02} \pm \sqrt{4(\mu'_{11})^2 + (\mu'_{20} - \mu'_{02})^2}\right] \tag{4.39}$$

Notably the covariance matrix that corresponds to the inertial tensor defines an inertially equivalent approximation of the considered object, referred to as the image ellipse. This ellipse is a constant intensity elliptical disk with the same mass as the original image, which is defined with semi-major axis, λ_1 along the x-axis

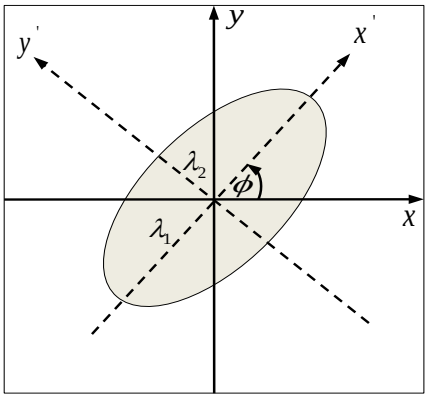


FIG. 4.9. Image ellipse as an approximation of the considered object.

and semi-minor axis, λ_2, along the y-axis, as shown in Fig. 4.9. The orientation of the object defined as the tilt angle between the x-axis and the x'-axis (semi-major axis) around which the object can be rotated with minimal inertia is calculated by:

$$\phi = \frac{1}{2}\arctan\left(\frac{2\mu'_{11}}{\mu'_{20} - \mu'_{02}}\right) \tag{4.40}$$

where the angle ϕ is picked such that $-\frac{\pi}{4} \leq \phi \leq \frac{\pi}{4}$. In addition, other parameters such as the roundness κ and eccentricity ε appear to be very closely related to our task, because both can provide rich information about the shape of the object of interest. Formally, given the area A and the perimeter p of an object, then the roundness κ of the object can be determined by simply dividing the square of the perimeter by the area of the object. As for the simple geometric fact that the circle has the maximum area for a given perimeter, then κ can be scaled and given by:

$$\kappa = \frac{p^2}{4\pi A} \tag{4.41}$$

It is perhaps worth noting that $\kappa = 1$ for a circle, while for other objects $\kappa > 1$. Furthermore, using the eigenvalues λ_1 and λ_2 of the covariance matrix, the eccentricity ε can be calculated by:

$$\varepsilon = \sqrt{1 - \frac{\lambda_2}{\lambda_1}} \tag{4.42}$$

Another approach to compute the eccentricity ε involves directly using the central moments of second order as follows,

$$\varepsilon = \frac{(\mu'_{20} - \mu'_{02})^2 - 4(\mu'_{11})^2}{(\mu'_{20} + \mu'_{02})^2} \tag{4.43}$$

From Eq. (4.43), it can be seen apparently that the value of eccentricity always lies within the unit interval, i.e. $\varepsilon \in [0, 1]$. For a perfectly round object, this value is equal

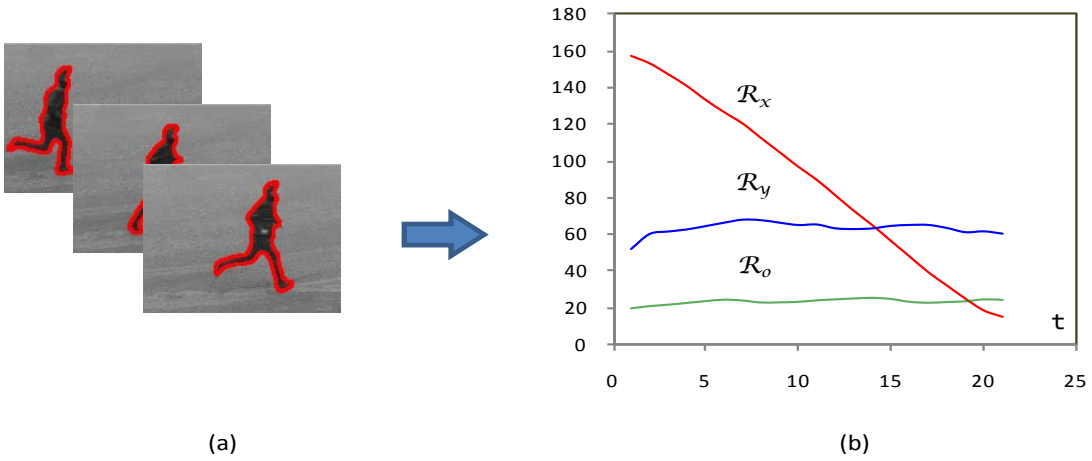


FIG. 4.10. Temporal variation in radii of gyration: (a) the person's silhouette sequence of a running action, (b) a plot reflects that the temporal changes in the radii of gyration, (i.e., $\mathcal{R}_x$, $\mathcal{R}_y$ and $\mathcal{R}_o$) of the silhouette sequence given in (a).

to zero, while it tends to one for a line shaped object. Due to such a clearly defined range, eccentricity can be compared much better than roundness. Consequently, as a measure, the eccentricity is often seen to be more appropriate than the roundness for shape matching and comparison.

Radii of gyration are another important feature that can be derived from the second order moments of an object. Technically speaking, the radius of gyration is a purely geometric parameter that is defined as the radial distance from a given axis at which the mass of an object could be concentrated without altering the second moment (i.e.rotational inertia) of the object about that axis. More formally, in terms of Cartesian moments, the values of the radii of gyration $\mathcal{R}_x$ and $\mathcal{R}_y$ about the x and y axes respectively are given by:

$$\mathcal{R}_x = \sqrt{\frac{M_{20}}{M_{00}}}, \quad \mathcal{R}_y = \sqrt{\frac{M_{02}}{M_{00}}} \tag{4.44}$$

Similarly, the radius of gyration about the origin $\mathcal{R}_o$ is defined as the radius of a circle centered at the origin where all the mass may be concentrated without change to the moments about the origin. Hereby, the value of $\mathcal{R}_o$ is given, in terms of central moments of second order by:

$$\mathcal{R}_o = \sqrt{\frac{\mu_{20} + \mu_{02}}{\mu_{00}}} \tag{4.45}$$

It is pertinent to mention here that its property of being inherently invariant to orientation might explain why $\mathcal{R}_o$ is frequently employed as a rotationally invariant feature for many object representation and detection tasks. Fig. 4.10 is an example that shows different temporal variations in the radii of gyration of a running action.

4.3.4 Curvature features

In a way similar to the extraction of the Mel Frequency Cepstral Coefficients (MFCC) features from voice signals [133], a set of other shape descriptors based on the cepstrum of the shape curvature can be also extracted. It may be worthwhile to mention that the name "cepstrum" was originally derived by reversing the first four letters of "spectrum". Briefly described, the scheme used for extracting such features works as follows. First, the curvature of a given shape is encoded by using a chain coding scheme. The cepstrum of the curvature signal (i.e., spectrum) is then obtained based on discrete Fourier transform. Finally, as shape features, a specific number of the largest coefficients can be chosen to be added to the final feature vector. In a little bit more details, the basic chain code that was first proposed in [134] by Freeman is essentially used to describe the motion on a sequence of boundary points (i.e., a digital curve). A numbering scheme is used to encode the movement direction between contiguous points along the shape border as follows,

$$\{n \,|\, n = 0, 1, 2, \ldots, N-1\} \tag{4.46}$$

Technically, there are basically, at least, two numbering schemes widely used in the literature, namely the 4-connectivity and the 8-connectivity for which the value of N is equal to 4 and 8 respectively. Thus, each coding number n will correspond to a contraclockwise angle of $\frac{\pi}{2}n$ or $\frac{\pi}{4}n$ (for the case of 4- or 8-connectivity, respectively) that is measured with respect to the positive x-axis, as shown in Fig. 4.11.

Now, let $s(t), t = 0, 1, \ldots, \tau - 1$ be the signal or spectrum formed by a numbering scheme (4- or 8-connectivity). There are various types of cepstrums widely used in practice (e.g., real, complex, power, and phase cepstrum) that can be obtained for a given spectrum of shape. For simplicity and convenience, we show here how to extract the power cepstrum of the signal $s(t)$ by simply taking the Fourier transform (FT) of the log spectrum. Strictly speaking, the power cepstrum of a given signal $s(t)$ can be verbally defined as the square of the magnitude of the Fourier transform of the logarithm of the squared magnitude of the Fourier transform of the signal. Formally, the functional of the power cepstrum of $s(t)$ is defined by:

$$\mathcal{C}\{s(t)\} = \left|\mathcal{F}\{\log(|\mathcal{F}\{s(t)\}|^2)\}\right|^2 \tag{4.47}$$

where $\mathcal{F}(\cdot)$ denotes the Fourier transform operator. A finite number of the cepstral coefficients directly calculated using Eq. (4.47) can be efficiently used as features to quantitatively model an object's shape for recognition and classification. Fig. 4.12 provides example result of the cepstrum coefficients extraction. As can be seen in the figure, six video sequences of different persons performing six types of

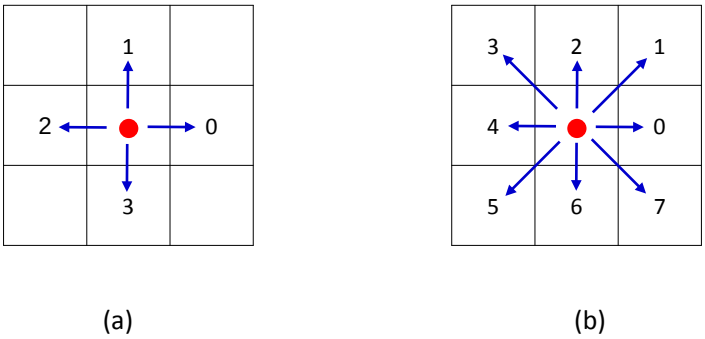


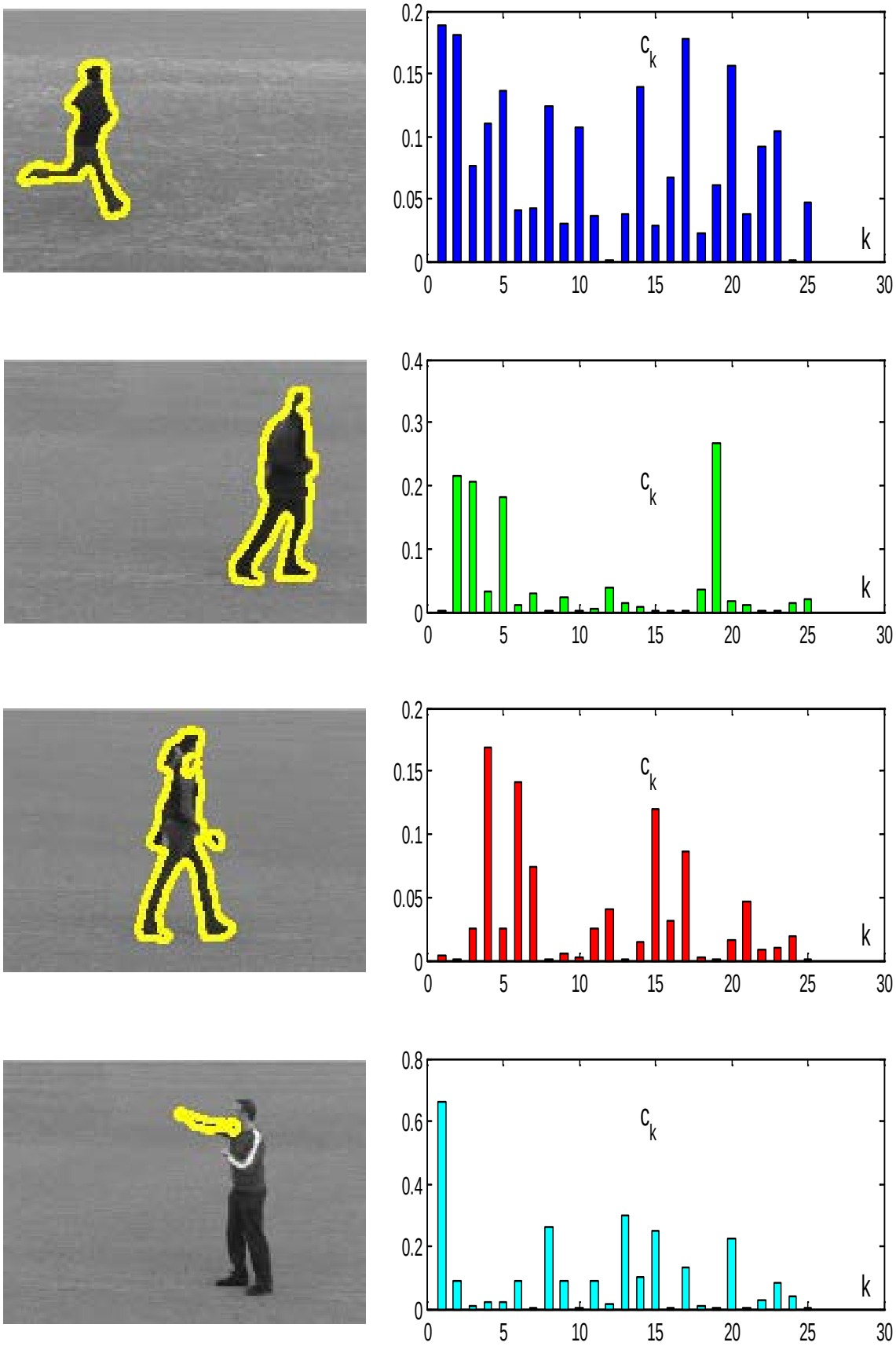
FIG. 4.11. Basic chain code direction: (a) 4-connectivity; (b) 8-connectivity.

actions (i.e., walking, jogging, running, boxing, waving, and clapping from top to bottom, respectively) are shown in Fig. 4.12(a), while the corresponding cepstral coefficients extracted from the motion shapes of these sequences are illustrated by bar plots in Fig. 4.12(b). In continuation of the above, it is important to point out that the experiments revealed that a small set of cepstrum coefficients are able to sufficiently approximate the signal, and also to reconstruct its curvature function, with a compression ratio of up to 10:1 in the original signal length [123].

The shape border-based features (described in detail in this section) including Fourier descriptors, Moments invariants, Moment-based features, and curvature features are eventually concatenated to form the feature vector of a given action at a time instance. All the feature vectors of an action snippet are then normalized to fit a zero-mean and a unit variance distribution. The normalized vectors obtained can be used as shape contextual information for classification and matching. Many approaches in various object recognition applications directly combine these vectors to get one final vector per video and classify it using any classification algorithm. It is worth mentioning that concatenating all the feature vectors extracted from all frames of an action snippet would result in a very large feature vector that might be less likely to be classified correctly. To resolve this problem and to reduce the dimensionality of the resulting vector, all feature vectors of a given action snippet at a time-slice can be weighted and averaged as follows,

$$\vec{\mu} = \frac{1}{\tau} \sum_{t=1}^{\tau} w_t \vec{x}_t \tag{4.48}$$

where $w_t = f(t; \alpha, \beta, \gamma)$ is the weighting factor and τ is the number of the feature vectors at the time-slice. Then all the vectors resulting at each of the time-slices are


0.2
0.15
0.1
0.05
0
c_k
k
0 5 10 15 20 25 30
0.4
0.3
0.2
0.1
0
c_k
k
0 5 10 15 20 25 30
0.2
0.15
0.1
0.05
0
c_k
k
0 5 10 15 20 25 30
0.8
0.6
0.4
0.2
0
c_k
k
0 5 10 15 20 25 30

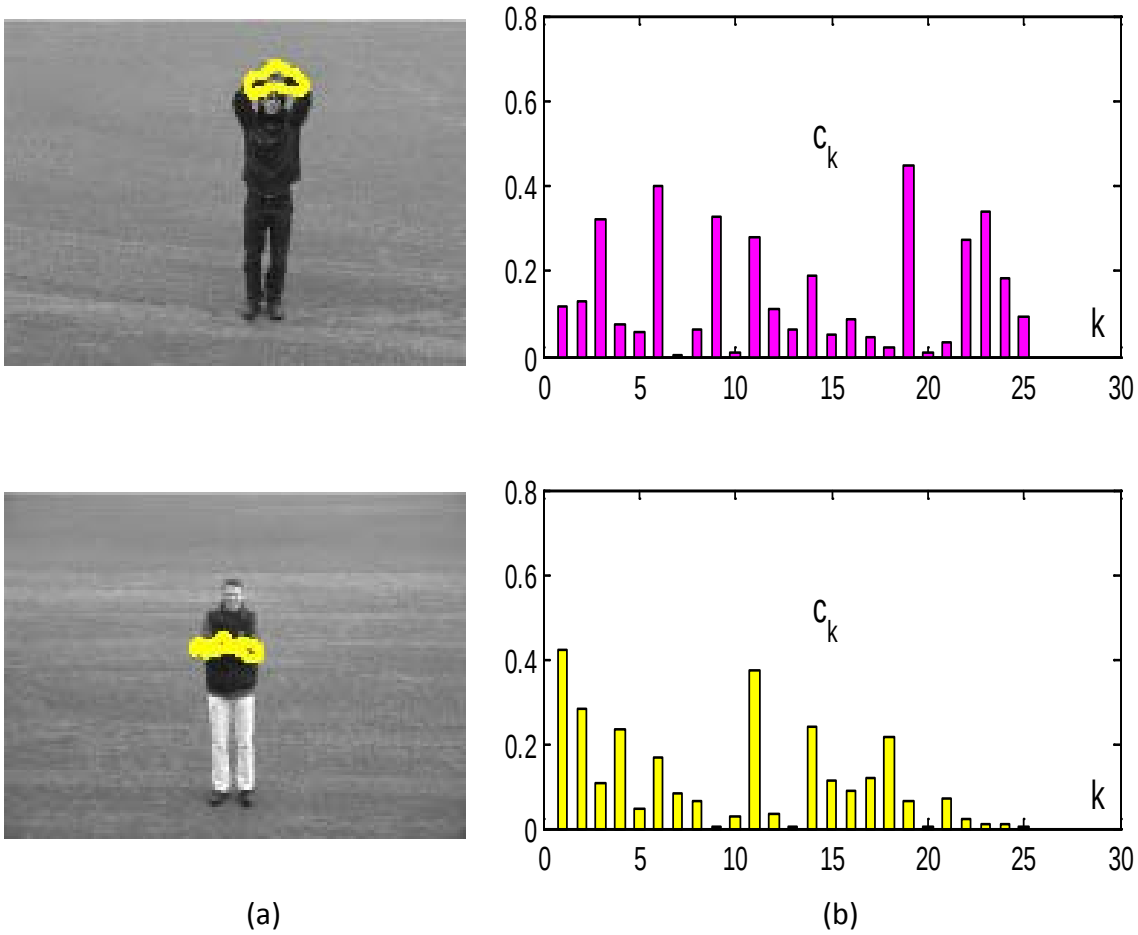


FIG. 4.12. Extraction of cepstral coefficients: (a) sample sequences of different actions; (b) bar plots for the cepstral coefficients extracted from the sequences in (a).

catenated to yield the final feature vector for a specific action snippet.

4.4 Chord-Length Features

Despite their stability and compactness, chord-length shape features have received relatively very little attention in human activity recognition literature. In this section, we first show how the chord-length functions are defined. Then, we describe how a compact computationally-efficient shape descriptor; the chord-length shape features is constructed using 1D chord-length functions.

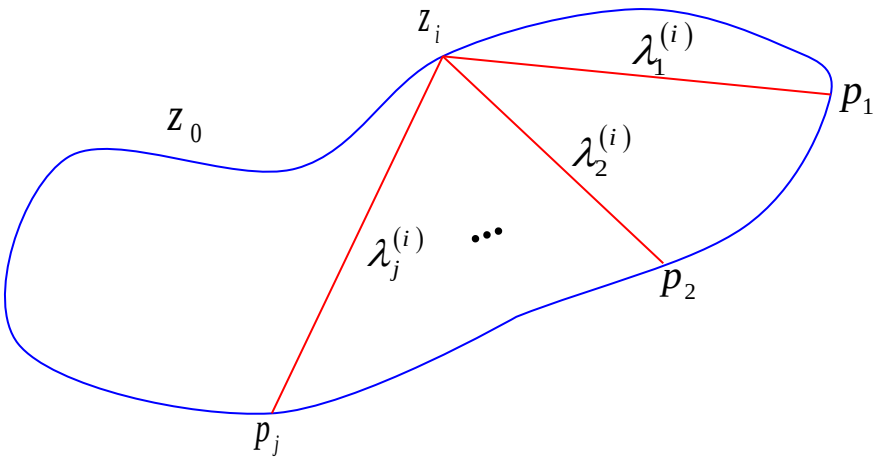


FIG. 4.13. Chord-length functions (CLFs) obtained through the division of a shape border into a finite number of arcs of equal length.

4.4.1 Chord Length Functions

A shape border, i.e. contour, is an inalienable property of every object and can be defined as a simply-connected sequence consisting of n 2D points:

$$\mathcal{C} = \{z_i \in \mathbb{R}^2,\ 0 \leq i < n\} \tag{4.49}$$

where $z_{i+n} = z_i$, as $\mathcal{C}$ is closed. The diameter ℓ of the shape is given by:

$$\ell = \max_{i,j=0}^{n-1} \|z_i - z_j\|, \quad i \neq j \tag{4.50}$$

where $\|\cdot\|$ is defined as the Euclidean distance between two points z_i and z_j. Taking as an initial point $z_i \in \mathcal{C}$, let the contour $\mathcal{C}$ be traversed anti-clockwisely and partitioned into $k > 1$ arc segments, i.e., $\widehat{z_i p_1}, \widehat{p_1 p_2}, \ldots, \widehat{p_{k-1} z_i}$ of equal length, where p_j is the jth division point and $j = 1, 2, \ldots, k-1$. Thus, we have $k-1$ chords $\overline{z_i p_1}, \overline{z_i p_2}, \ldots, \overline{z_i p_{k-1}}$, and $k-1$ lengths $\lambda_1^{(i)}, \lambda_2^{(i)}, \ldots, \lambda_{k-1}^{(i)}$, where $\lambda_j^{(i)}$ is the length of the chord $\overline{z_i p_j}$ measured as the Euclidean distance between the two points p_j and z_i as shown in Fig. 4.13.

Let us now show that while the point z_i travels along the contour, then the chord lengths $\lambda_j^{(i)}$ will vary accordingly. This implies that $\lambda_j^{(i)}$ is a function of z_i. Such a function is called the Chord-Length Function (CLF), and shortly denoted as λ_j. Therefore we can obtain $k-1$ CLFs, i.e., $\lambda_1, \lambda_2, \ldots, \lambda_{k-1}$. Since these functions are obtained from splitting the contour evenly and from moving the initial point z_i, along the contour, so that they guarantee to be invariant to translation and

rotation. However, the chord length itself is not scale invariant, but it can be made to be invariant to scale by normalization using the contour diameter ℓ. Now, it is apparently that CLFs meet all requirements for being good shape descriptors (see Fig. 4.14), including invariance to scale, translation, and rotation .CLFs might need to be scaled to be within the same range (e.g., $[0, 1]$). By their definition, CLFs are derived by segmenting the contour evenly, so that it is easy to deduce that only half of the CLFs, $\lambda_1, \lambda_2, \ldots, \lambda_{k/2}$ are enough to describe the shape adequately. It is germane to point to the fact that both global and local features of shape can be captured by using chord-lengths of different level. The local features are likely to be captured by the CLFs of the partition points closer to the initial point z_i, while the global features are captured by those of farther points. This is viewed as a distinct competitive advantage of the CLF-based descriptor over other shape descriptors.

4.4.2 Chord-length shape features

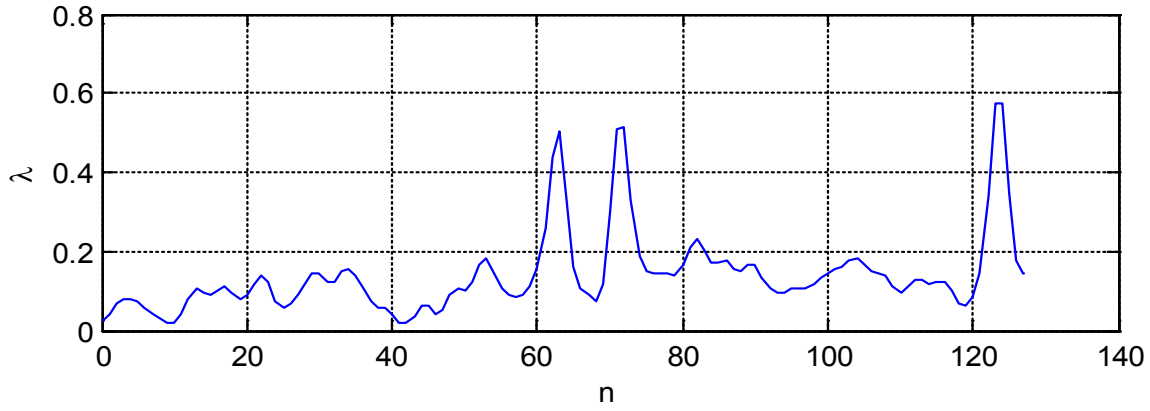
As described previously in Section 4.4.1, given a shape border (i.e., contour), $k/2$ chord-length functions can be defined by dividing the shape border into k arcs of equal length. These functions are shown to be invariant with respect to translation, rotation, and scaling. Nevertheless, as other shape descriptors, these descriptors appear to be not compact enough. In addition, they may constantly depend on a reference point whereby the shape border is parameterized. This dependence is simply because the contour is closed and any point on the contour can be used as a reference point, so that the chord-length functions may be changed. In order to avoid such problems and for convenience, the mean μ_j and variance σ_j of chord-length functions $\lambda_j, j = 1, 2, \ldots, \frac{k}{2}$, are adopted, which are given by

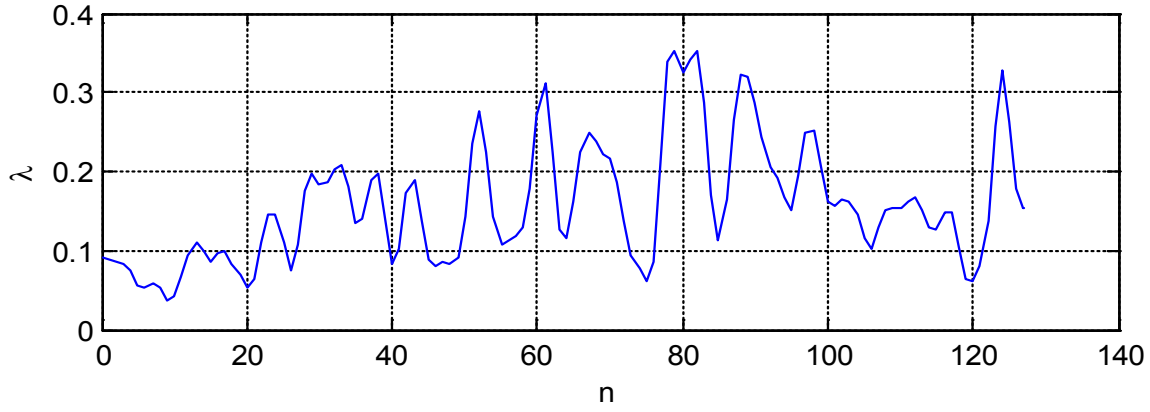
$$\mu_j = \frac{1}{n}\sum_{i=0}^{n-1} \lambda_j^{(i)}, \quad \sigma_j = \frac{1}{n-1}\sum_{i=0}^{n-1} (\lambda_j^{(i)} - \mu_j)^2 \tag{4.51}$$

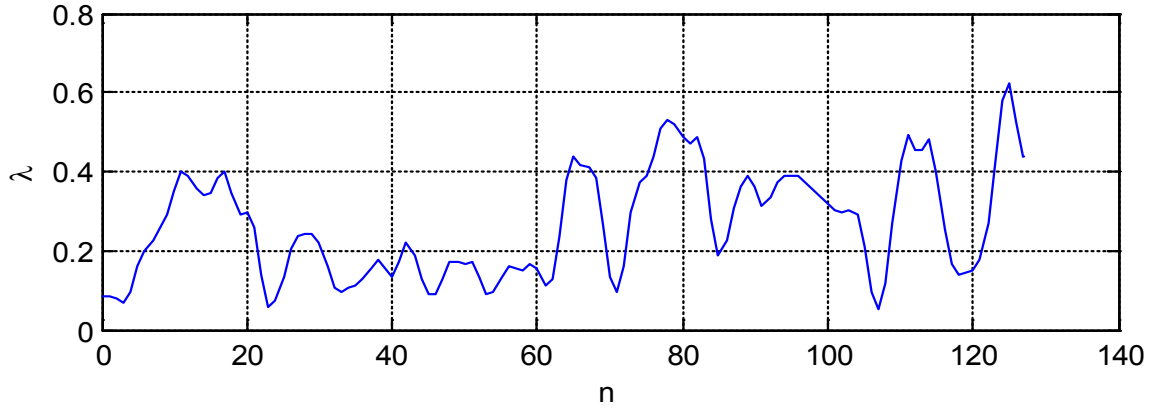
Therefore, the chord-length features that are used as a shape descriptor can be arranged in a 2D matrix of size $\frac{k}{2} \times 2$ as follows,

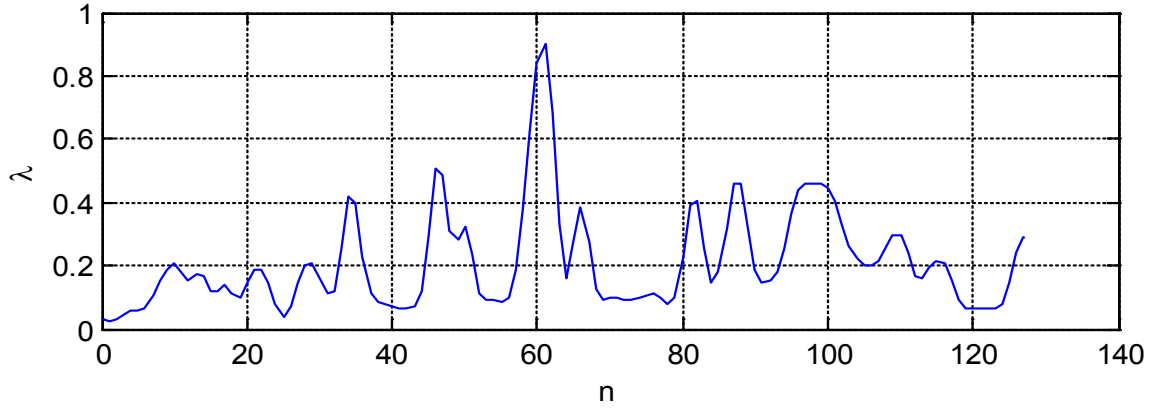
$$F = \begin{pmatrix} \mu_1 & \sigma_1 \\ \mu_2 & \sigma_2 \\ \vdots & \vdots \\ \mu_{\frac{k}{2}} & \sigma_{\frac{k}{2}} \end{pmatrix} \tag{4.52}$$

It should be noted that the prior formation of the chord-length features can be easily converted into a 1D feature vector simply by "row-scanning", i.e., concatenating the rows to obtain the feature vector of length k.


0.8
0.6
0.4
0.2
0
λ
0
20
40
60
80
100
120
140
n


0.4
0.3
0.2
0.1
0
λ
0
20
40
60
80
100
120
140
n


0.8
0.6
0.4
0.2
0
λ
0
20
40
60
80
100
120
140
n


1
0.8
0.6
0.4
0.2
0
λ
0
20
40
60
80
100
120
140
n

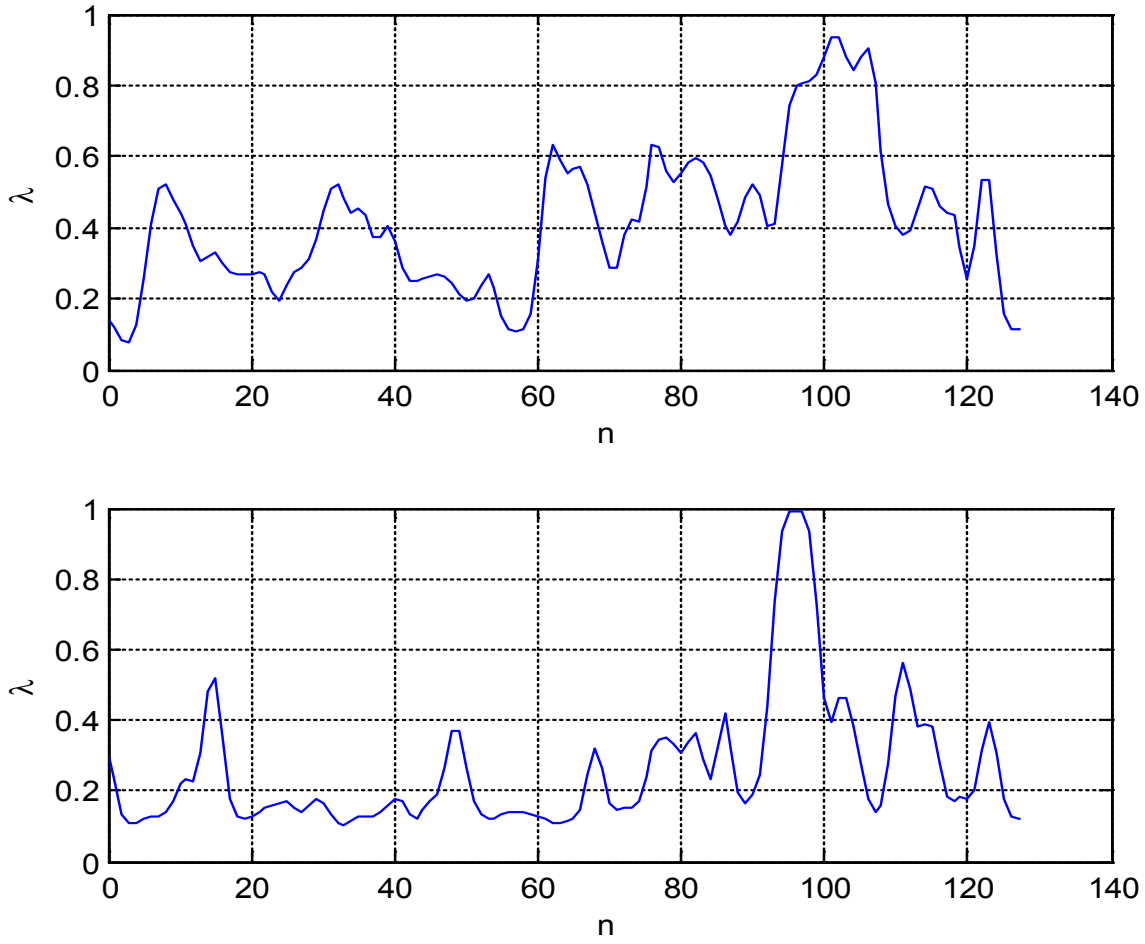


FIG. 4.14. Plots of chord-length functions (CLFs) for sample shape borders (normalized to 128 points) extracted from actions of walking, jogging, running, boxing, waving, and clapping, from top to bottom, respectively.

Now, to obtain the final chord-length features of a given human action, we have to first obtain the chord-length features of all poses of that action. Since each action snippet was temporally divided into a number of fuzzy states, each represents a pose of the action, then the chord-length features of an action pose is obtained by:

$$P_j = \frac{1}{n_j} \sum_{t=1}^{n_j} \mathcal{G}_j(t) F_t, \quad j = 1, 2, \ldots, m \tag{4.53}$$

where $\mathcal{G}_j$ is the fuzzy membership function that defines the temporal slice j, n_j is the total number of the chord-length feature vectors of the pose j, and m is the total number of time-slices. Note that Eq. (4.53) implies that the final chord-length features of an action pose is approximated as the averaged weighted sum of all features of that pose. The weight factor here is the Gaussian membership function ($\mathcal{G}_j \in [0, 1]$) defined previously at the beginning of this chapter. At least this time,

we have all the chord-length descriptors of the poses that an action has. Next, the resulting feature vectors are normalized to the integral value of unity to achieve robustness to scale variations and to reduce the influence of illumination. The normalized feature vectors obtained can now be exploited as shape descriptors for action classification and recognition.

Generally, the normalized vectors can be directly combined to obtain the resultant feature vector per video clip that in turn can be classified by any machine learning algorithm (such as, SVM, ANN, NB, decision trees, etc.). Accordingly, the final feature vector of a given action can be constructed by catenating all the descriptors of its temporal poses, and given as:

$$\vec{F}_{action} = \bigcup_{j=1}^{m} \vec{P}_j = [\vec{P}_1, \vec{P}_2, \ldots, \vec{P}_m] \tag{4.54}$$

where $\bigcup$ is the catenation operator and m is the number of poses. As seen from Eq. (4.54), the temporal information of action are retained.

4.5 Discussion and Conclusion

In this chapter, we have presented a detailed description of various features and descriptors developed in our works on activity recognition, that broadly include interest-point based features, shape border based features, and chord-length features. At the beginning of the chapter, the features extracted based on spatio-temporal salient interest-points have been detailed. Such features are characterized by their high compactness representation of video data and robustness to occlusions, background clutter, significant scale changes, and high activity irregularities, so that they have been successfully employed for a wide variety of recognition tasks. In addition, the extraction of these features is relatively straightforward, rather computationally efficient, and more importantly eliminates the requirement for any prior segmentation or other pre-processing steps.

Thereafter, we have shown a set of other features that could be extracted from the segmented silhouettes (or their boundaries) of moving human body parts in order to represent action poses. These features include both local and global properties, such as Fourier descriptors, moment invariants, moment-based features, and cepstrum descriptors. A careful analysis/investigation of such features has suggested that they turned out to have the potential to provide a rich source of information for the interpretation/analysis of human activities.

Finally, at the end of the chapter, we have shown how the chord-length functions are defined form a finite set of boundary points of a shape. Then, we have described

how a compact computationally-efficient shape descriptor; the chord-length shape features could be constructed using 1D chord-length functions. Despite their stability and compactness, these features have received little attention in the literature. One distinct competitive advantage of the chord-length descriptors over other shape descriptors is that both global and local features of shape can be captured by using chord-lengths of different level. The local features are expected to be extracted by the chord-length functions of partition points closer to the initial point, while the global features are extracted by those of farther points.

CHAPTER 5

ML Models for Activity Feature Classification

5.1 Introduction

ROUGHLY speaking, feature classification is viewed as a crucial step in various image/video analysis applications such as, higher-level image understanding, scene interpretation, event retrieval and activity modeling. The task of feature classification that has been successfully tackled by multi-label learning algorithms is still a subject of intense research effort the machine learning and computer vision communities, where having effective feature extraction is of extreme importance for reliable classification. More specifically, in current action recognition, such a task involves the attempt to correctly categorize different types of unknown human actions through a machine learning process involving training some machine learning method (e.g., ANN, SVM, HMM, decision tree, etc.) on known instances of action features. In this chapter, we illustrate how the action features described in Chapter 4 can be classified. Within the scope of this dissertation, three machine learning (ML) models (i.e., ANN, SVM, and Bayesian network) have been employed in this task of feature classification.

In essence, Artificial Neural Network (ANN) or simply neural network [135] is an information processing paradigm inspired by the way biological nervous systems, e.g. the brain, process information. To the best of our knowledge, an ANN model has many useful characteristics over conventional modeling techniques in some aspects, including strong self-adaptive, robustness, fault tolerance and storage memory capabilities. Furthermore, it has been found to yield good results, particulary when the response variable is highly nonlinear. Hence, ANNs have found

(and still find) their wide range of applications in many areas, such as image/video representation, pattern recognition, fault diagnosis, etc. However, an ANN model is likely to suffer from a number of limitations regarding generalization capabilities, such as over-fitting, fixed topology and slow convergence, and further the choice of the network parameters (e.g., hidden layer size, learning rate, momentum, etc.) is fully found on the experience or knowledge of researcher [136].

Support Vector Machine (SVM) [137] originally developed as an alternative to the ANN paradigm is a relatively new ML paradigm based on the statistical learning theory framework. The pivotal idea of SVM is to generalize data sets of limited size efficiently by implementing the structural risk minimization inductive principle. In more simple words, an SVM model tries to apply a nonlinear mapping to the feature space, and then use a linear model to form the decision boundary. Due to their several distinguished characteristics, including powerful model generalization capability, strong nonlinear processing ability and usefulness in convex optimization problem, SVMs have been successfully applied (and still being applied) to various problems of pattern recognition and machine learning, such as speaker identification, face detection, and text recognition. However, the SVM paradigm essentially is a hard-margin classifier, so that it is unlikely to admit vague outputs that are a very desirable property in a wide range of practical applications. Moreover, such a paradigm has an intrinsic limitation that it can be hardly applied to very large training datasets due to the high computational time involved in solving the quadratic programming (QP) problem that is at least quadratic to the number of training samples. Therefore, unlike some other machine learning paradigms (e.g., decision trees and neural networks), SVMs have not impressed with the high rate of adoption by communities working with huge datasets [138].

Naïve Bayesian (NB) [139] is one of the earliest content-based machine learning models, which has achieved wide popularity as a simple yet consistently performing probabilistic paradigm based on the theory of Bayesian networks and built on the assumption of conditional independence between the attributes given the class. It is designated 'naïve' due to the assumption of independence among features. Due to its high efficiency in handling a large number of features that other machine learning models cannot, NB is still competitive, even though it ignores dependencies between features. In addition, in many classification tasks, an NB classifier turned out to be highly scalable, easy to implement, relatively robust, and able to achieve considerable accuracy. However, since the traditional NB classifier essentially operates under the independence assumption, so that it is expected that its performance suffers in domains involving correlated features, unlike other well-known classifiers (e.g., SVM and k-nearest neighbor). But, using compound-risk

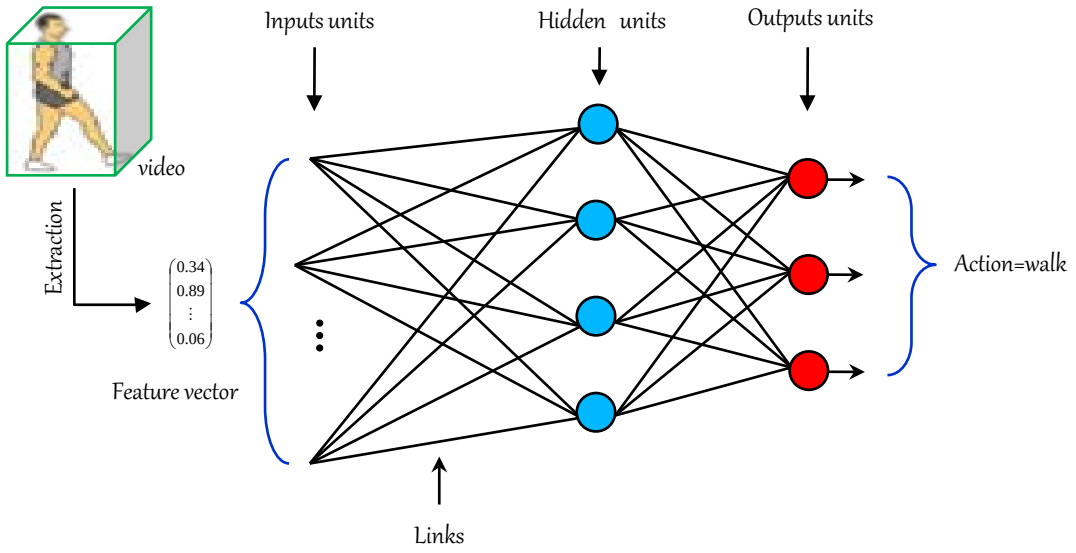


FIG. 5.1. ANN for human activity recognition.

factors is very likely to be able to tackle these problems effectively and thus allow classification results to be more accurate than those of the traditional NB. Each of the machine learning paradigms briefly mentioned above that we have employed for features classification in the work of this thesis will be described in greater detail in subsequent sections of this chapter.

5.2 Artificial Neural Network

Artificial Neural Networks (ANNs) have been defined in a wide spectrum of ways. For instance, in relation to their biological origins, ANNs are said to be relatively crude electronic models established on the neural structure of the brain that basically learns from experience. The typical architecture of ANN model (i.e., very similar to feedforward model) that we have used in this study for the task of activity classification constitutes three distinct types of layers of neurons (or nodes) connected in a layer-to-layer manner, as shown in Fig. 5.1. As seen in the figure, the neurons are arranged in layers, each layer having full interconnection to the next layer. Links between nodes are going in only one direction from input layer through hidden layers to output layer. It is pertinent to point out that the choice of a specific architecture usually depends on the properties of the architecture and also the unique requirements of the application being developed. The model should be configured such that the application of a set of input feature vectors of actions produces the desired set of action categories. In literature, there are several approaches and techniques which we can use to set the strengths of the connections of

the neural network. One approach that we have pursued in one component of this study is to train the neural model by feeding it, teaching activities and allowing it alter its weights according to some learning rule.

5.2.1 Training and learning of ANN

Like any supervised ML classifier, the neural classifier needs to be trained on a training set of manually labeled activities before it can be used for activity recognition. This process involves adjusting the weights of each unit in such a manner that the error between the actual category of an activity and the desired output is minimized. To achieve this purpose, the popular Back Propagation (BP) algorithm is employed to iteratively adjust the link weights using the steepest descent technique, in which the global error is backward propagated to the networks units, and the weights are modified proportional to their contribution. More specifically, the error is first defined as the difference between target and actual outputs, and then the mean square error is used as the training error to be minimized:

$$E = \frac{1}{2}\sum_{j}(t_j - o_j)^2 \tag{5.1}$$

where o_j and t_j is the actual and target outputs of node j, respectively. However, in the BP algorithm, it is the rate of change of error which is the most important feedback through the network:

$$\Delta w_{ij} = -\eta\frac{\delta E}{\delta w_{ij}} \tag{5.2}$$

Now the objective is to compute the quantity $\frac{\delta E}{\delta w_{ij}}$ for all w_{ij} (i.e. weights form node i to node j). This process involves computing how fast error changes as each of these four factors is varied (i.e., output of node j, total input to node j, weight w_{ij} coming into node j, and output of node i in previous layer). The overall structure of the learning algorithm used to construct the classifier to learn activities is given by Algorithm 5.1. It is worthy to mention that the weight update loop may be iterated thousands of times in our application, so that choice of termination condition seems to be of overwhelming importance as too few iterations can fail to reduce error sufficiently. Further, too many iterations can lead to overfitting the training dataset. The termination criteria can be either a fixed number of iterations (epochs) or until the error falls below some predetermined threshold. In addition, it has been shown that adding momentum α to the original weight update rule for the learning algorithm would help to escape a small local minima in the error in error

Algorithm 5.1: Outline of network training algorithm.

Input: Feedforward neural network with n_{in} inputs, n_{hid} units in hidden layers, and n_{out} output units and learning rate η.
Output: Updated network weights $w_{ij}\ \forall i, j$
Initialize all weights w_{ij} to small random numbers;
while *convergence criterion not reached* **do**
  foreach *training action example* **do**
    Fed the training example into the network and compute the outputs;
    foreach *output unit* k **do**
      $\delta_k \leftarrow o_k(1 - o_k)(t_k - o_k)$;
    foreach *hidden unit* h **do**
      $\delta_h \leftarrow o_h(1 - o_h) \sum_k w_{hk}\delta_k$;
    Update each network weight w_{ij}:
      $\Delta w_{ij} = \eta \delta_j x_{ij}$;
      $w_{ij} \leftarrow w_{ij} + \Delta w_{ij}$;

surface and also speeds up the convergence. Hence, the modified weight update rule can be redefined as:

$$\Delta w_{ij} \leftarrow \eta \delta_i x_{ij} + \alpha \Delta w_{ij},\ 0 < \alpha < 1 \tag{5.3}$$

As stated previously, the learning process has to terminate at the point where the error $E(\vec{\mathbf{w}})$ is minimum. Here, we argue that in general it is found that the error squared versus the weight graph is a paraboloid in higher dimensional space, and the vertex of this paraboloid represents the point where the error is minimized, i.e., there is only one global minimum point. The weight vector that corresponds to this point is then the ideal weight vector, $\vec{w}_{opt}$.

5.2.2 Multi-level neural networks

The ANN classifier offers several advantages over other competitive ML classifiers. Some of these advantages include the high rapidity, easiness of training, realistic generalization capability, high selectivity, and great capability to create arbitrary partitions of feature space. However, the neural model, in the standard form, may have low classification accuracy and poor generalization properties because its neural units usually employ a standard bi-level function that results in only two values (i.e., binary responses) [140]. To relax this restriction and allow the neural units to generate multiple responses, a new functional extension for the standard sigmoidal functions should be developed [8]. This extension is termed

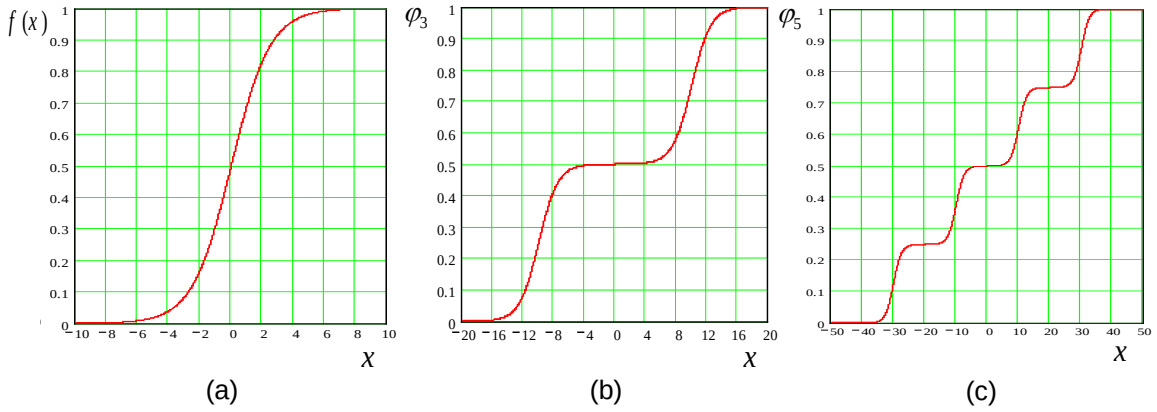


FIG. 5.2. Standard sigmoidal function and its Multi-level versions:(a) Sigmoidal function; (b) Multi-level function for $r = 3$; (c) Multi-level function for $r = 5$.

Multilevel Activation Function (MAF), and therefore the neural model employing this functional extension is termed 'Multilevel Sigmoidal Neural Network', or simply MSNN. There are several multi-level versions corresponding to several standard activation functions. It is straightforward to derive a multi-level version from a given bi-level standard sigmoidal activation function. Formally, let the general form of a standard sigmoidal function $f(x)$ (in Fig. 5.2(a)) is given as,

$$f(x) = \frac{1}{1 + e^{-\beta x}} \tag{5.4}$$

where $\beta > 0$ is an arbitrary constant, known as steepness factor. The multilevel version of activation functions are straightforwardly derived from Eq.(5.4) as:

$$\varphi_r(x) \leftarrow f(x) + (\lambda - 1)f(c) \tag{5.5}$$

where λ is an index running from 1 to $r - 1$; r is the number of levels, and c is an arbitrary constant. Multi-level sigmoidal functions for $r = 3$ and 5 are depicted in Fig. 5.2(b) and Fig. 5.2(c), respectively. As a final comment here, it is noteworthy that, in [141], the authors have experimentally reported that the neural classifier employing multilevel activation functions exhibits a superior performance over its neural counterpart employing conventional sigmoidal activation functions.

5.3 Support Vector Machines (SVMs)

In this section, we describe Support Vector Machines (SVMs) as an activity classifier we used in most of the experimental work presented in this thesis. SVMs are seen as a relatively new supervised ML methodology developed by Cortes & Vapnik [142], which were first applied as an alternative to multi-layer neural networks. The

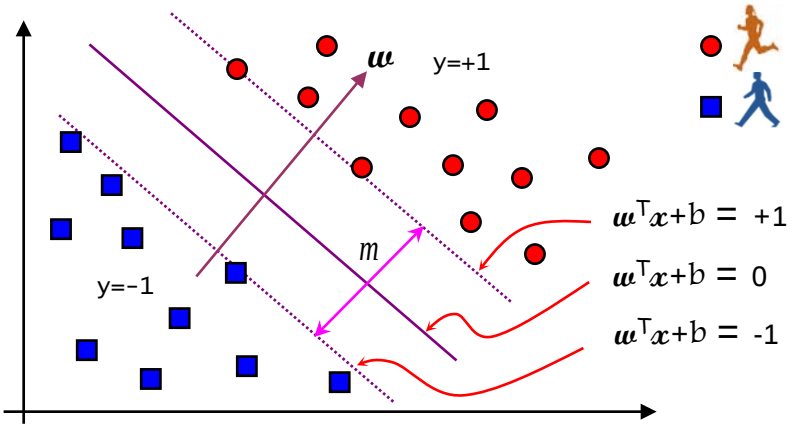


FIG. 5.3. Large-Margin linear decision boundary.

standard SVM was originally designed for binary classification, but recently several variants of SVMs based on the same principles have been introduced to extend the original version of SVM into multi-class problems. SVMs are based upon the principle of structural risk minimization, i.e., they depend on the maximum-margin principle of maximizing the margin between the decision hyperplane and the closest training examples, and also support nonlinear separation.

Let us first consider the simplest case of a linearly separable binary classification problem. To obtain the optimum decision boundary, SVM attempts to maximize the minimal distance from the decision boundary to the labeled data. Once this decision boundary is decided, a given unseen activity can be checked on which side of the decision boundary it lies. Formally, let $\mathcal{S} = \{\boldsymbol{x}_i \in \mathbb{R}^d | \, i = 1, \ldots, n\}$ be the training samples (i.e., feature vectors of actions), and $y_i \in \{+1, -1\}$ be the class label of $\boldsymbol{x}_i$, thus two parallel separating hyperplanes can be formed such that:

$$y_i = \begin{cases} +1, & \boldsymbol{w}^\top \boldsymbol{x}_i + b \geq 1 \\ -1, & \boldsymbol{w}^\top \boldsymbol{x}_i + b \leq -1 \end{cases} \tag{5.6}$$

where $\top$ denotes the transpose operator, $\boldsymbol{w}$ is a perpendicular vector to the two hyperplanes and b is the bias, as shown in Fig. 5.3). Thus, the separating decision boundary (i.e. the optimal hyperplane) that maximizes the margin between the two classes is created by solving the following constrained optimization problem:

$$\begin{aligned} &\text{Minimize :} && \tfrac{1}{2}\|\boldsymbol{w}\|^2 \\ &\text{subject to} && y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 \; \forall i \end{aligned} \tag{5.7}$$

By Lagrange duality, after some lengthy but straightforward calculations, the dual

problem of the primal problem in Eq.(5.7) is given as:

$$\begin{aligned} &\text{Maximize :} \quad \mathcal{W}(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j \\ &\text{subject to} \quad \alpha_i \geq 0, \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned} \tag{5.8}$$

where $\alpha_i \geq 0$ are the lagrangian multipliers. Since Eq. (5.8) describes a QP problem, and a global maximum always exists for α_i, $\boldsymbol{w}$ can be deduced as:

$$\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i \tag{5.9}$$

An interesting characteristic of this solution of the dual problem in Eq. (5.9) is that many of α_i are zero. The feature vectors $\boldsymbol{x}_i$ corresponding to $\alpha_i > 0$ are termed *support vectors* that lay on the hyperplanes, hence the decision boundary can be adequately determined by them alone. Formally, let $t_j (j = 1, \ldots, \ell)$ be the indices of ℓ support vectors, then Eq. (5.9) can be rewritten as follows,

$$\boldsymbol{w} = \sum_{j=1}^{\ell} \alpha_{t_j} y_{t_j} \boldsymbol{x}_{t_j} \tag{5.10}$$

For testing with a feature vector $\boldsymbol{z}$ of an unknown activity, we first evaluate this function: $f(\boldsymbol{z}) = \boldsymbol{w}^\top \boldsymbol{z} + b = \sum_{j=1}^{\ell} \alpha_{t_j} y_{t_j} (\boldsymbol{x}_{t_j}^\top z) + b$. It is then decided that $\boldsymbol{z}$ belongs to the first activity class if $f(\boldsymbol{z}) > 0$ and to the second activity class otherwise. It may be pertinent to mention here that the QP problem has long been the focus of attention by many scientific communities and come under extensive investigation [143]. Accordingly, for solving QP problems numerically, recently there is a wide range of software packages (e.g., CPLEX, LINDO, LOQO, MINQ, etc.) implementing different approaches [144]. For SVM, the sequential minimal optimization (SMO) approach is frequently adopted for training, in which initially the simple case of the original QP problem containing only two variables is solved. Then, in each subsequent iteration, a pair of (α_i, α_j) is picked and used to solve the original QP problem. This iteration process continues until a convergence criteria is met.

5.3.1 Soft-Margin classification

To enable SVM to handle nonlinearly separably classification problems, it has been shown [142] that this type of problems is effectively approached by allowing some examples to violate the margin constraints (see Fig. 5.4). These potential violations can be formulated using some positive slack variables ξ_i and a penalty parameter $C \geq 0$ that penalize the margin violations. The slack variables that approximate the number of misclassified examples basically depend on the output

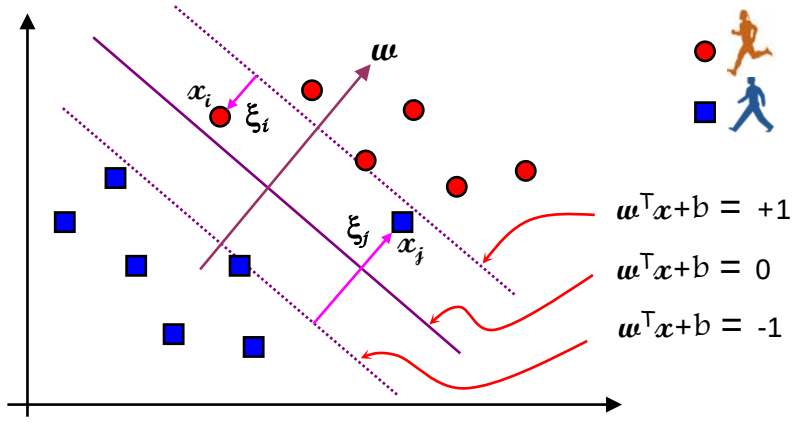


FIG. 5.4. Soft-Margin decision boundary.

of the discriminant function $\boldsymbol{w}^\top \boldsymbol{x} + b$. Formally, the optimization problem, in this case, can be written as:

$$\begin{aligned} &\text{Minimize :} && \tfrac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ &\text{subject to} && y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0 \end{aligned} \tag{5.11}$$

After tedious but elementary calculations very similar to those performed for the linearly separable case, we obtain the dual constrained optimization problem as:

$$\begin{aligned} &\text{Maximize :} && \mathcal{W}(\boldsymbol{\alpha}) = \sum_{i=1}^{n}\alpha_i - \tfrac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j \\ &\text{subject to} && 0 \leq \alpha_i \leq C, \quad \sum_{i=1}^{n}\alpha_i y_i = 0 \end{aligned} \tag{5.12}$$

The dual optimization problem in (5.12) is very similar to that of the linear separable case, but here there is an upper bound C on the coefficients α_i. Likewise, by using the same formula in (5.10), the weight vector $\boldsymbol{w}$ can be recovered. Once more, the coefficients α_i can be obtained by using any QP solver. The solution algorithm attempts to keep $\boldsymbol{\xi}$ null, while maximizing the margin. It does not minimize the number of misclassifications, but minimizes the sum of distances from the margin hyperplanes. When C increases, the number of error decreases and the number of support vectors drops; further as C tends to ∞, the number of errors tends to 0.

5.3.2 Extension to non-linear decision boundary

Yet, only large-margin SVM with a linear decision boundary has been discussed. In order to be able to generalize SVM from linear to nonlinear case, the key idea is to use a mapping function $\phi(\cdot)$ that transforms all data points $\boldsymbol{x}_i$ from the input space $\boldsymbol{X}$ into a high dimensional feature space $\boldsymbol{F}$, as shown in Fig. 5.5. With such a proper transformation, a linear operation in the feature space is equivalent to a nonlinear

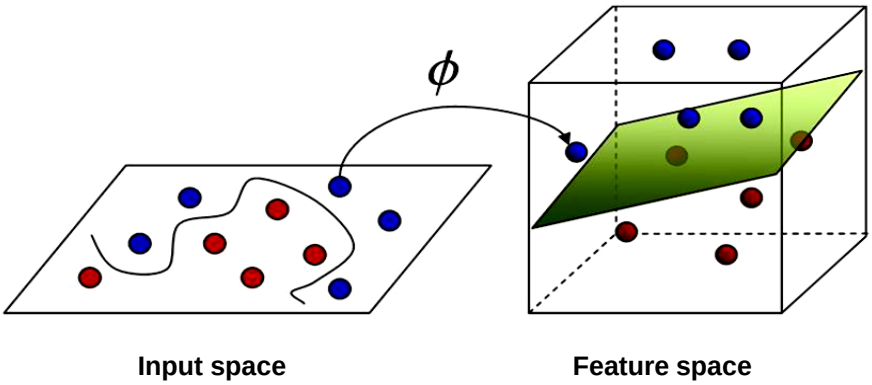


FIG. 5.5. A nonlinear mapping from input space to feature space [5].

operation in the input space. This, in turn, makes the classification problem easier to handle and ultimately solve, as the original nonlinearly separable problem will become linearly separable. It is important to note that, in practice, the feature space has a higher dimensionality than that of the input space, so that the computational requirements in the feature space are expected to be substantially higher than those in the input space. This is where the so-called '*kernel trick*' comes to rescue; it is to replace all costly computations in the feature space by inexpensive computations in the input space that give the same results. In other words, evaluating the so-called kernel function will allow all expensive computations in the feature space to be achieved implicitly. Recalling the expression for the SVM optimization problem given by Eq. (5.8), we can see that the data points only appear as inner product (i.e., $\boldsymbol{x}_i^\top \boldsymbol{x}_j$). Hence, the kernel function is defined such that it calculates the inner product in the feature space. Such a definition is given as:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\phi}(\boldsymbol{x}_i)^\top \boldsymbol{\phi}(\boldsymbol{x}_j) \tag{5.13}$$

It is really fascinating to observe that the definition of the kernel function in Eq. (5.13) eliminates the necessity of knowing the mapping function $\boldsymbol{\phi}(\cdot)$ explicitly. This use of the kernel function to avoid carrying out $\boldsymbol{\phi}(\cdot)$ explicitly is famously known as the kernel trick. More generally, given a mapping: $\boldsymbol{\varphi} : \boldsymbol{X} \rightarrow \boldsymbol{F}$, then a kernel can be defined as the inner product of the elements of the input space, as follows

$$K(\boldsymbol{x}, \boldsymbol{y}) \leftarrow \sum_i \boldsymbol{\varphi}(x_i)\boldsymbol{\varphi}(y_i) \quad \forall\, \boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{X} \tag{5.14}$$

Note that it is required that the kernel function fulfills the Mercer's condition [145]:

$$\forall g(\boldsymbol{x}) \text{ such that } \int g^2(\boldsymbol{x}) d\boldsymbol{x} \geq 0 \Rightarrow \iint K(\boldsymbol{x}, \boldsymbol{y}) g(\boldsymbol{x}) g(\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} \geq 0 \tag{5.15}$$

Now, by substituting every occurrence of the inner product in Eq. (5.8) with the kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, the dual problem is rewritten as:

$$\begin{aligned} &\text{Maximize :} \quad \mathcal{W}(\boldsymbol{\alpha}) = \textstyle\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ &\text{subject to} \quad\ 0 \leq \alpha_i \leq C, \quad \textstyle\sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned} \tag{5.16}$$

For testing a given activity $\boldsymbol{z}$, the following function is evaluated:

$$f(\boldsymbol{z}) = \langle \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{z}) \rangle + b = \sum_{\boldsymbol{x}_i \in \mathcal{S}} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{z}) + b \tag{5.17}$$

where $\mathcal{S}$ is the set of support vectors. Likewise, as said before, it is decided that the activity $\boldsymbol{z}$ belongs to the first class if $f(\boldsymbol{x}) \geq 0$ and to the second class otherwise. It is worthwhile to point out that the dependence of the training process of SVM only on the value of the kernel function implies that there is no restriction imposed upon the form of the data points (i.e., $\boldsymbol{x}_i$ can be a sequence or a tree, instead of a feature vector). In practice, there are several commonly used kernel functions, such as:

- Polynomial kernel of degree d:

$$K(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^\top \boldsymbol{y} + 1)^d$$

- Radial Basis Function (RBF) kernel (Gaussian kernel) with width σ:

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\|\boldsymbol{x} - \boldsymbol{y}\|^2 / 2\sigma^2)$$

- Sigmoidal kernel with parameters κ and θ:

$$K(\boldsymbol{x}, \boldsymbol{y}) = \tanh(\kappa \boldsymbol{x}^\top \boldsymbol{y} + \theta)$$

Note that the sigmoid kernel only satisfies the Mercer's condition for certain values of κ and θ. Finding the optimal parameters for the kernel function represents one of the most computationally challenging problems in SVM design, which requires extensive training before the SVM classifier can be set up. The proper choice of a kernel function is very crucial, but often the trickiest part in building up SVM, as the kernel function creates the kernel matrix by which all data are summarized. In practice, an RBF kernel with a reasonable width or a lower degree polynomial kernel is a good initial try. As a final note on this point, it may be worth pointing out that SVM models with RBF kernel is closely associated with RBF neural networks, with the centers of the radial basis functions automatically chosen for SVM.

5.3.3 SVM multi-class classification

As stated at the outset of this section, SVM originally is a typical binary classifier, which can only separate two classes. However, in the current application of activity recognition, data set contains more than two classes (e.g. six class of activities). The important question that poses itself at this juncture is: how can SVM be extended to the general case of multi-class classification problems? Unfortunately, this question is still an open issue, being a hot research subject. A direct solution based on a single SVM formulation for multi-class classification problems is considered to be computationally intractable and usually avoided due to several complexities associated with objective functions, constraints, and the QP formulation. Nevertheless, a few attempts have been made to generalize SVM to deal with multiclass problems. In these attempts, the pivotal idea is to use a combination of several binary SVMs to solve a given multi-class classification problem. Loosely speaking, there are two popular strategies for achieving this task of multi-class classification. One is the 'one-versus-all' (or one-vs.-rest) strategy, where each SVM is trained to distinguish one class from the rest of the classes, while the other is the 'one-vs.-one' strategy in which all possible combinations of pairs of classes are considered; there is a single SVM to classify each pair. However, in our view, the optimal technique of extending binary SVMs to the multi-class classification problem is still an open question requiring much further investigation by machine learning community.

One-vs-all SVM for activity classification

To construct an m class SVM classifier ($m = 6$ in our experiments) using the one-vs.-all strategy, a set of m binary SVMs:

$$\left\{f^{(1)}, f^{(2)}, \ldots, f^{(m)}, \quad f^{(j)} = \text{sgn}(g^{(j)}),\ g^{(j)} : \boldsymbol{X} \rightarrow \mathbb{R}\right\}$$

is constructed, each trained to separate one activity class from the remaining $(m-1)$ classes. To perform the multi-class classification task, these binary classifiers are put together according to the maximal output before applying the sign function:

$$\underset{j=1\ldots m}{\arg\max}\, g^{(j)}(\boldsymbol{x}), \text{ where } g^{(j)}(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i^{(j)} y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b^{(j)} \tag{5.18}$$

Note that the signed real-valued value produced by the function $g^{(j)}(x)$ is seen as the distance from the separation hyperplane to the activity $\boldsymbol{x}$. This value can also be considered as a 'confidence' value. The higher the value, the more confident the class label that the activity $\boldsymbol{x}$ belongs to the positive class. Consequently, the activity $\boldsymbol{x}$ is assigned to the specific class with the highest confidence value.

One-vs.-one SVM for activity classification

Analogous to the one-vs.-all SVM approach, to construct an m class SVM classifier using the one-vs.-one scheme, we also require a set of binary SVMs, but the number of SVMs required in this case is larger (i.e., exactly equal to $\ell = \binom{m}{2} = \frac{m(m-1)}{2}$). An SVM is then trained for each possible pair of classes ignoring the activity examples that do not belong to the classes in question. To classify an unknown activity $\boldsymbol{z}$, all discriminant functions of the ℓ learned SVMs are applied:

$$f^j(\boldsymbol{z}) = \sum_{i=1}^{n} \alpha_i^j y_i K(\boldsymbol{x}_i, \boldsymbol{z}) + b^j, \; j = 1, 2 \ldots, \ell \tag{5.19}$$

Finally, we count the number of times the activity $\boldsymbol{z}$ was assigned to each class label. The activity is simply assigned to the class whose label has the highest count.

5.4 Naïve Bayes (NB) Classifier

Generally speaking, there are three approaches whereby a classifier can be established. The first is to construct a classification rule directly (e.g., SVMs, K-NN, decision trees, perceptron, etc.). The second is to create a probability model for the class memberships from the training data (e.g., MLP with cross-entropy cost), while the third one attempts to build a probabilistic model of data within each class. The classifier that is generated by either of the first two approaches is referred to as a 'discriminative' classifier, while that generated by the third approach is so-called a 'generative' classifier. Additionally, the last two approaches are seen as typical instances of probabilistic classification. Naïve Bayesian (NB) [146] is a simple probabilistic model based on Bayes' theorem [147] with strong (naïve) independence assumptions, or more specifically, independent feature model.

Probabilistic classifiers, such as Naïve Bayes, basically depend upon turning data into probabilities for classification. As an illustrative example, in Fig. (5.6), the measurements of some feature $\boldsymbol{x}$ for two classes ω_1 and ω_2 are depicted. As can be seen, the members of the first class show a clear tendency to have larger values than those of the second class; however some overlap between the classes exists. While, at extremes of the range, it is a straightforward task to predict the correct class for a value of the feature $\boldsymbol{x}$, performing the same task in the middle of the range seems to be challenging or problematic. Statistically speaking, let $\mathcal{D}$ be a training activity dataset containing pre-classified examples:

$$\mathcal{D} = \{(\boldsymbol{x}, y) \in \mathbb{R}^n \times \{\omega_1, \ldots, \omega_m\}\}$$

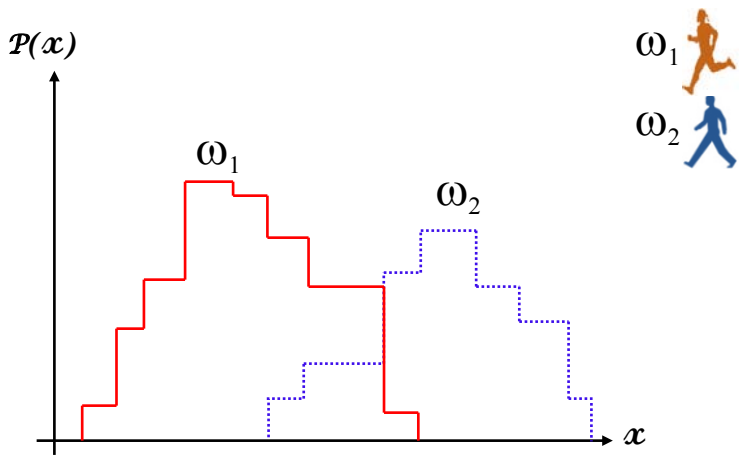


FIG. 5.6. Feature values ($\boldsymbol{x}$) along with their probabilities for two classes [6].

where $\boldsymbol{x}$ and y is a feature vector of a specific activity and its associate class, respectively. In a simple setting of classification learning, the ultimate goal is to assign the activity represented by the feature vector $\boldsymbol{x} = (x_1, \ldots, x_n)^\top$, to the most probable of the available classes $\boldsymbol{\omega} = \{\omega_1, \ldots, \omega_m\}$, where x_i is the value of the i-th attribute. In general, the classification error can be minimized by computing the Maximum A posterior (MAP) corresponding to the optimal class:

$$\omega^*_{\text{MAP}} \equiv \underset{\omega_j \in \boldsymbol{\omega}}{\arg\max}\ p(\omega_j|\boldsymbol{x}) \tag{5.20}$$

The MAP classification rule above shows that the feature vector $\boldsymbol{x}$ is assigned to class ω^* where $p(\omega^*|\boldsymbol{x}) > p(\omega_j|\boldsymbol{x})$, $\omega_j \neq \omega^* \in \boldsymbol{\omega}$. To identify this class, we need to estimate the conditional probabilities $p(\omega_j|\boldsymbol{x})$. Bayes' theorem states that:

$$p(\omega|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega)p(\omega)}{p(\boldsymbol{x})} \tag{5.21}$$

where the previous probabilities are defined as follows
$p(\omega)$: independent probability of ω (i.e. prior probability)
$p(\boldsymbol{x})$: independent probability of $\boldsymbol{x}$ (i.e. evidence)
$p(\boldsymbol{x}|\omega)$: conditional probability of $\boldsymbol{x}$ given ω (i.e. likelihood)
$p(\omega|\boldsymbol{x})$: conditional probability of ω given $\boldsymbol{x}$ (i.e. posterior probability)
Based on Bayes theorem, the MAP class in (5.20) can be computed as follows,

$$\begin{aligned} \omega^*_{\text{MAP}} &\equiv \underset{\omega_j \in \boldsymbol{\omega}}{\arg\max}\ p(\omega_j|\boldsymbol{x}) \\ &= \underset{\omega_j \in \boldsymbol{\omega}}{\arg\max}\ \frac{p(\boldsymbol{x}|\omega_j)p(\omega_j)}{p(\boldsymbol{x})} \\ &= \underset{\omega_j \in \boldsymbol{\omega}}{\arg\max}\ p(\boldsymbol{x}|\omega_j)p(\omega_j) \end{aligned} \tag{5.22}$$

Note that the quantity $p(\boldsymbol{x})$ is eliminated from the denominator in (5.22), as the probability of the data is constant and independent of the class label. Our main interest is in the optimal class given observed training data $\boldsymbol{x}$. Assuming a uniform prior, that is all classes are equally probable a priori, i.e. $p(\omega_j) = p(\omega_k)\ \forall\, \omega_j, \omega_k \in \boldsymbol{\omega}$, then the computation of the posterior is simplified and given by:

$$\omega^*_{\mathrm{ML}} = \underset{\omega_j \in \boldsymbol{\omega}}{\arg\max}\ p(\boldsymbol{x}|\omega_j) \tag{5.23}$$

In this case, we obtain the so-called the Maximum Likelihood (ML) estimate of the most probable class, which maximizes the likelihood of obtaining the training data. Recalling the MAP rule in (5.22), we see that Bayes classification depends on the joint probability (i.e. likelihood) and the prior probability:

$$p(\omega|\boldsymbol{x}) \propto p(\boldsymbol{x}|\omega)p(\omega) = p(x_1, \ldots, x_n|\omega)p(\omega) \tag{5.24}$$

Intuitively, the inherent difficulty arising here lies in learning the joint probability $p(x_1, \ldots, x_n|\omega)$. In order to circumvent this difficulty and make the computations more tractable, the so-called "Naive Bayes independence assumption" that assumes the probabilities of each attribute value are conditionally independent given the class is employed. Hence, the joint probability can be computed efficiently as a sequential product of conditional probabilities:

$$\begin{aligned}
p(x_1, \ldots, x_n|\omega) &= p(x_1|x_2, \ldots, x_n; \omega)p(x_2 \ldots, x_n|\omega) \\
&= p(x_1|x_2, \ldots, x_n; \omega)p(x_2|x_3, \ldots, x_n; \omega)p(x_3, \ldots, x_n|\omega) \\
&= \vdots \\
&= p(x_1|x_2, \ldots, x_n; \omega)p(x_2|x_3, \ldots, x_n; \omega) \ldots p(x_{n-1}|x_n; \omega)p(x_n|\omega) \\
&\approx p(x_1|\omega)p(x_2|\omega) \ldots p(x_{n-1}|\omega)p(x_n|\omega) \\
&= \prod_{i=1}^{n} p(x_i|\omega)
\end{aligned} \tag{5.25}$$

It should be emphasized here that the conditional independence assumption is almost always violated in practice; however NB classifier often performs surprisingly well anyway, even when the assumption of attribute independence does not strictly hold. Upon the substitution of the joint probability from Eq. (5.25) in Eq. (5.22), Naïve Bayes model, depicted in Fig. 5.7, is defined by:

$$\omega_{NB} = \underset{\omega_j \in \boldsymbol{\omega}}{\arg\max} \left(p(\omega_j) \prod_{i=1}^{n} p(x_i|\omega_j) \right) \tag{5.26}$$

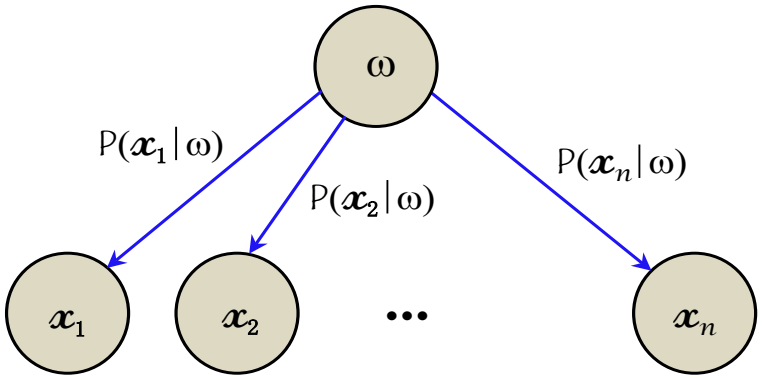


FIG. 5.7. Naïve Bayes model with the assumption of conditional independence.

Estimating probabilities:

Conditional probabilities can be estimated directly as relative frequencies (i.e., by the fraction of times the event is observed to n_c occur over the total number of opportunities n). This way may provide poor estimates even when n_c is very small. To get around this problem, conditional probabilities can be estimated by:

$$\hat{P}(x_i|\omega_j) = \frac{n_c + m\alpha}{n + m} \tag{5.27}$$

where n_c is the number of examples for which $\omega = \omega_j$ and $x = x_i$, while n is the number of training examples for which $\omega = \omega_j$. The parameter α is a prior estimate (usually, $\alpha = \frac{1}{t}$ for t possible values of x_i), and $m \geq 1$ is a weight given to prior. And as to concerns about the priori probabilities of classes $p(\omega_j)$, they can be directly estimated by their relative frequencies in the training dataset:

$$\hat{P}(\omega_j) = \frac{\#(\omega_j)}{\sum_j \#(\omega_j)} \tag{5.28}$$

where # denotes the count (or frequency) with which the bracketed class occurs in the training dataset. Consequently, the naïve MAP decision rule can be written as:

$$\omega_{NB} = \arg\max_{\omega_j \in \boldsymbol{\omega}} \left(\hat{P}(\omega_j) \prod_{i=1}^{n} \hat{P}(x_i|\omega_j) \right) \tag{5.29}$$

In the case of continuous-valued real features (i.e. numberless values for feature attributes), the class-conditional probabilities $p(\boldsymbol{x}|\omega)$ can be appropriately modeled by the normal (Gaussian) distribution (Fig. 5.8) $\mathcal{N}(\mu, \sigma^2)$:

$$\hat{P}(x_i|\omega_j) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-\frac{1}{2}\left(\frac{x_i - \mu_{ij}}{\sigma_{ij}}\right)^2} \tag{5.30}$$

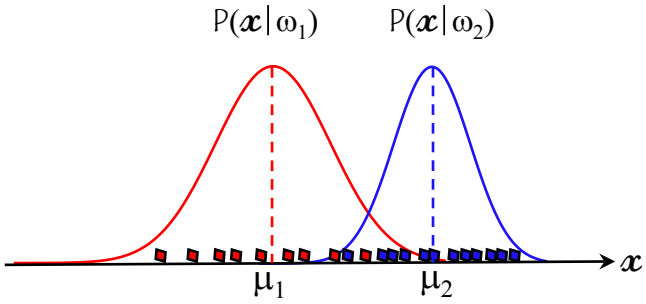


FIG. 5.8. Class-conditional probability distributions of features.

where μ_{ij} and σ_{ij}^2 are the mean (average) and variance of the feature value x_i of activity instance associated with the class ω_j, respectively. Note that, in general, both the mean and the variation of distributions depend on class. Conversely, If we allow the same variance for all classes, the classification rule would be easier. According to above analysis and derivations, the main steps involved in the NB classification algorithm can be simplified as listed in Algorithm 5.2 bellow.

Algorithm 5.2: Naïve Bayes classification algorithm

1. *Naïve_Bayes_Learn*(action_examples):
foreach *target value* ω_j *of action instances* **do**
  $\hat{P}(\omega_j) \leftarrow$ estimate $p(\omega_j)$;
  foreach *feature value* x_i *of a feature* $\boldsymbol{x}$ **do**
    $\hat{P}(x_i|\omega_j) \leftarrow$ estimate $p(x_i|\omega_j)$;

2. *classify_new_activity*($\boldsymbol{z}$)

$$\omega_{NB} = \arg\max_{\omega_j \in \boldsymbol{\omega}} \hat{P}(\omega_j) \prod_{z_i \in \boldsymbol{z}} \hat{P}(z_i|\omega_j)$$

Regarding the parameter dimensionality of the NB model, it is not hard to see that given N data points (i.e. feature vectors of activities) and a model with r parameters for the probabilities $p(\boldsymbol{x}_i)$, the NB model would have a set of $mrN + (m-1)$ parameters, where m is the total number of classes.

5.5 Discussion and Conclusion

In this chapter, we have presented and discussed three of the most widely used and most influential machine learning algorithms (i.e, ANN, SVM and NB) that were trained and tested separately for the activity features presented in the previous

chapter. Among the other traditional machine learning models, ANN has been argued to offer not only a computing schema, but also a conceptual model for non-parametric modeling of data in human activity recognition and in a variety of other contexts in which the output and the inputs are related by a non-linear function. However, the ANN model in its standard form, suffers from a serious inherent limitation that severely reduces its classification accuracy and its generalization properties, as its neural units usually employ a standard bi-level function that produces only two values (i.e. binary responses). The MSNN (Multilevel Sigmoidal Neural Network) model has been developed to relax this restriction and to allow the neural units to generate multiple responses. Nevertheless ANN, as a classifier, generally still has some restrictions, such as the over-fitting phenomena, local minima, relatively slow convergence during training and the network structure and parameters are very likely to be problem dependent.

As a relatively new alternative to the existing linear and non-linear machine learning paradigms, SVM that is based on structural risk minimization of statistical learning theory has a more rigorous theoretical and mathematical foundation and there is no local minimum problem. Unlike a lot of traditional ML algorithms, SVM has been found to possess several prominent characteristics including high generalization capability, excellent properties in learning limited samples and small training error; and further it can potentially avoid over-fitting phenomena. However, when working with large datasets, the learning algorithm tends to be more complex and quite demanding in terms of computational and memory resources required to solve the QP problem. In addition, extending SVM to directly handle the multi-class setting remains an ongoing research goal in machine learning.

An NB is a simple probabilistic classifier based upon applying Bayes theorem with strong naive independence assumptions and turned out to be an appropriate choice, especially when the dimensionality of the input space is sufficiently large, or the amount of data available for training is limited or incomplete. Some of the strengths of NB (i.e., the main reasons for its popularity) are: it is rapid, sort of robust to irrelevant features, efficient for applications with numerous equally important features, and most importantly, it has been shown to be theoretically optimal when the independence assumptions hold. While NB is presumed to suffer from poor performance when its basic independence assumption is violated, it was empirically found that its performance remains favorably comparable to that of its counterparts, even with the violation of the independence assumption.

CHAPTER 6

Datasets and Experiments

6.1 Introduction

VIDEO datasets for the evaluation of systems of vision-based human activity recognition consist of a large collection of videos (or video clips) about the activities of interest. Each video sequence includes an individual performing a single action or a series of successive actions. All video sequences belonging to the same action category can be annotated with a categorical label describing the type of the activity performed within them. As observed from the literature, researchers have generally taken different perspectives about the dataset used to evaluate recognition systems. At the very beginning, each research group has been interested in creating its own datasets for the evaluation of its techniques and methods. The major problem that arises here is that the comparison of results obtained with different datasets can be difficult. For this reason and to avoid this problem, many other researchers [7, 51, 67, 120, 148–150], including us [8, 123, 124, 151, 152], have preferred to use some common datasets to evaluate their systems effectiveness. In this case, the comparison with other recognition methods turns out to be very meaningful and just fair, as all techniques use the same public dataset and the same experimental settings.

In the literature, there is a variety of benchmark datasets (e.g., KTH [7], Weizmann [2], etc.) commonly used to evaluate activity recognition algorithms. These datasets differ notably from one to another in many aspects (e.g., the number of action categories, the number of actions per category, the number of subjects performing actions, camera viewpoints, illumination, occlusion, etc.). In the course of

this chapter, we will proceed with the experiments based upon the theory from the preceding chapters, particularly the last three chapters. After this short introduction, it is appropriate to commence our discussion in the forthcoming section by providing an overview of two of the most popular action datasets (i.e. KTH [7] and Weizmann [2]) on which the outputs of this written research are based. Thereafter, we will explain how the experiments were conducted in details in this work.

6.2 Human Activity Recognition Datasets

An increasing number of datasets are now being made available for object recognition research with an open licence [153, 154], while the situation for action recognition appears to be slightly different. Unfortunately these datasets are chiefly designed to be specialized to some extent and focused upon a specific recognition objective (e.g., the Tulips1 dataset [155] for visual speech recognition and Georgia-Tech dataset [156] for gait recognition, where scenes involve actions performed by a total of 20 individuals in different environments). One more example is the CMU Mobo dataset [157] for gait recognition that contains 25 persons performing four types of walking action captured from different viewpoints.

6.2.1 KTH action dataset

For testing efficacy of the proposed approaches, we made a decision to use two of action datasets (i.e., KTH and Weizmann datasets) publicly available free of restrictions on use for action recognition research. For KTH[1] action dataset, it was first provided by Schuldt *et al.* [7] in 2004 and has been frequently cited as one of the most largest datasets in human action recognition literature, so that it has intensively been used by many authors for the purpose of evaluation and comparison of their own recognition algorithms [8, 44, 51, 59, 123, 151]. There are a total of six classes of actions involved in KTH dataset; three "leg actions " (i.e., walking, jogging and running) and three arm actions (i.e., boxing, hand-waving, and hand-clapping). All videos were taken over homogeneous backgrounds (as close to homogeneous as possible) with a static camera at a 25fps rate. Each action is performed by 25 subjects under four different scenarios including:

[1]KTH is an acronym for the Swedish expression "**K**ungliga **T**ekniska **H**ögskolan"; the Royal Institute of Technology in Stockholm, Sweden and is one of the top engineering schools in Europe.

s1 – *Outdoors*

All outdoor sequences were captured at the Östermalms IP sports-field and the gravelly field served as a homogeneous background.

s2 – *Outdoors with scale variation*

In videos involving locomotor activities (i.e., walking, jogging and running), scale variation was presented by allowing the action subject to begin at a distance and move diagonally closer toward the camera. By means of camera zooming, the same effect was created in videos containing non-locomotor activities (i.e., boxing, hand-waving and hand-clapping).

s3 – *Outdoors with different clothes*

This effect was produced by asking the action actors to wear a variety of clothing items, such as a long coat, a backpack, or a scarf fluttering at the back of action subject. This effect is expected to make recognizing actions more challenging.

s4 – *Indoors*

In indoor sequences, a bright monochromatic wall was set to be background. All sequences (in all scenarios) were typically acquired by a 3CCD DV-camera, and then downsampled to a resolution of 160×120 pixels represented in 256 grayscale levels. The authors [158] of the KTH dataset have argued that this low resolution may be sufficient to reduce the high impact of the camera artifacts on the recognition results, since the data are internally stored in a lossy MPEG-format by their camera. All sequences are short and of slightly different length; the average length of a sequence is approximately four second.

As there are 25 subjects performing six types of activities under four scenarios and each combination of subject-action-condition was acquired such that to produce four sequences, numerically a total of 2400 sequences are expected to be produced. However, the authors have shown that due to several reasons, some sequences were lost, causing the total number of video sequences to decline to 2391. At this point, it is worth mentioning that the KTH dataset, to the best of our knowledge, is considered to be one of the largest datasets in action recognition, which has an advantage of acquiring action sequences over multiple scenarios. An illustrative example for each action performed under four different scenarios is given by Fig. 6.1. As shown in this figure, some variations can be notably observed in each condition,

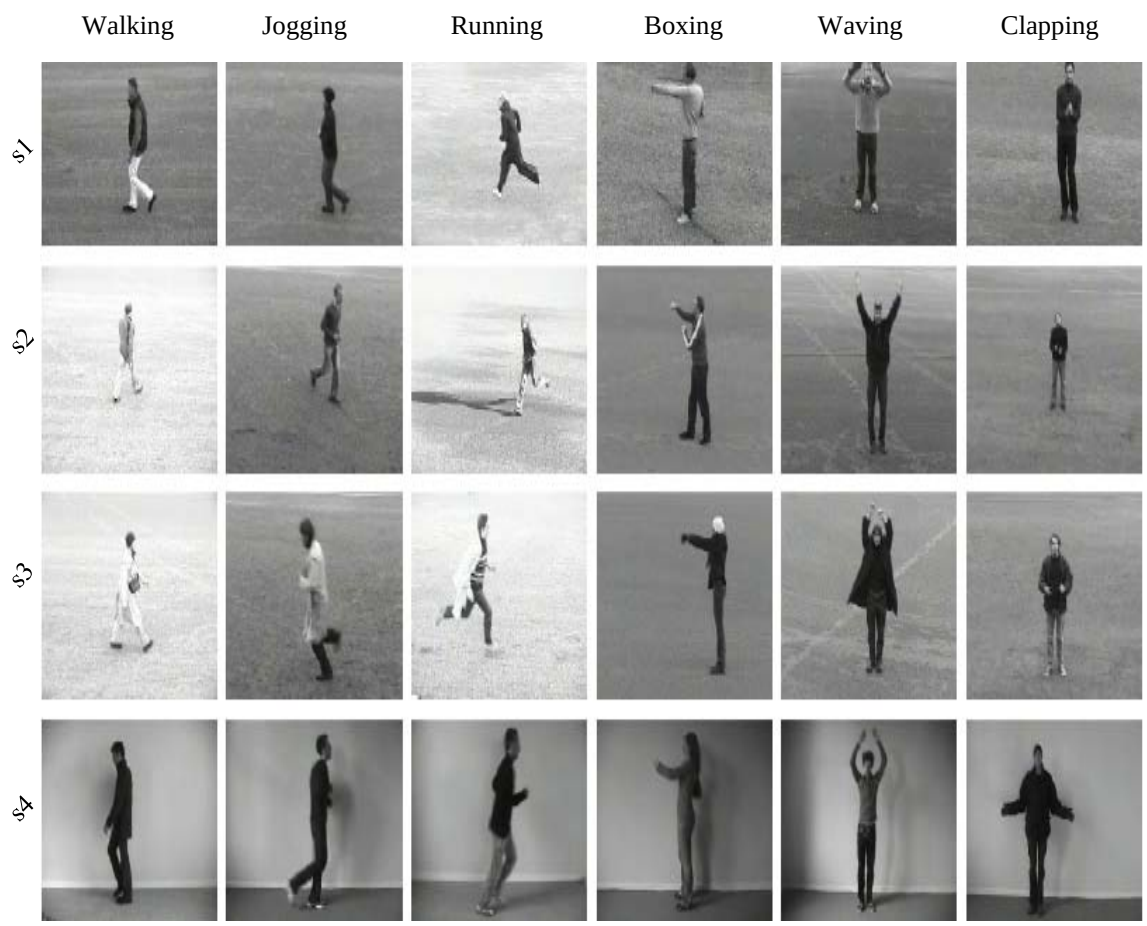


FIG. 6.1. Examples from the KTH action recognition dataset [7].

such as apparent footprints of the action subject in wet conditions, or moving shadows on the ground in sunny conditions.

6.2.2 Weizmann action dataset

As one of the most widely used activity video dataset, the Weizmann dataset presented by Blank et al. [2] has emerged in 2005 and then was made publicly available to activity recognition researchers without a restriction or other access charge. Weizmann dataset contains relatively simple action-level activities; each scene involves a single subject performing only one action. This dataset includes 10 different action categories, namely *'walk'*, *'run'*, *'jump-forward-on-two-legs' (or shortly 'jump')*, *'jumping-in-place-on-two-legs' (or 'p-jump')*, *jumping-jack (or 'jack')*, *'gallop-sideways' (or 'side')*, *'bend'*, *'skip'*, *'wave-one-hand' (or 'wave1')* and *'wave-two-hands' (or 'wave2')*. Each of these categories is performed by nine subjects, that results in a total of 90 sequences (or video clips) contained in the dataset. The sequences were

FIG. 6.2. Sample frames form action sequences in the Weizmann dataset [2].

acquired with static camera over static background at a rate of 25 frames/sec (fps), with a spatial resolution of 180×144 pixels, 24 bits per pixel. The sequences are very short; each lasts only about couple seconds. Fig. 6.2 shows a sample frame for each action involved in the Weizmann[2] action recognition dataset.

6.3 Experiments and Results

Over the course of the forthcoming sections of this chapter, we will describe in detail a set of experiments conducted using different action representation methods (i.e. features) described earlier in Chapter 4, with a special emphasis on the obtained results to confirm the performance of the developed approaches. In all of these

[2]The Weizmann Institute of Science, one of the worlds leading multidisciplinary research centers, is a university and research institute located in Rehovot, Israel, which offers only graduate and post-graduate studies in the sciences (i.e., mathematics, computer science, physics, chemistry, biological chemistry and biology) [159].

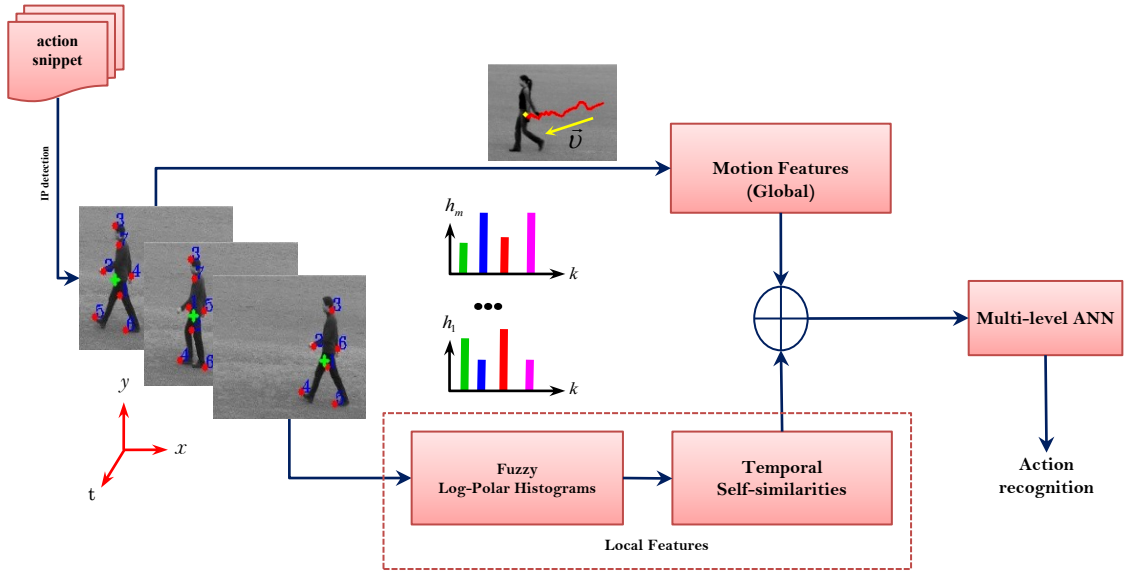


FIG. 6.3. Main components of the human action recognition framework [8].

experiments, the learning sequences from the action dataset were randomly divided into two independent subsets, namely, a training set and a test set. The former set was used to train the classifier, while the latter set was used to test the classifier and to obtain recognition results. Furthermore, in order to demonstrate the efficiency of the developed approaches, the results obtained were compared with those of other similar state-of-the-art methods in the literature.

6.3.1 Activity recognition using fuzzy log-polar histograms and temporal self-similarities

In this section, we provide description of the implementation and experimental results of our recognition approach based on fuzzy log-polar histograms and temporal self-similarities [8] for feature extraction. The working principle of this approach is schematically described in Fig. 6.3. As shown from the block diagram, the process of recognition systematically runs as follows. For each action snippet, spatio-temporal interest points (i.e. keypoints) are first detected using the modified Harris detector described in Chapter 4. Fig. 6.4 shows an example of spatio-temporal keypoints detected from different sequences, each showing a person performing a specific action. To make the recognition process more robust, action snippets are divided into a number of time-slices defined by Gaussian membership functions. Local features are then extracted based on fuzzy log-polar histograms. Since motion features tend to be relevant to the current recognition task, they are integrated into

the final feature vector fed into the MSNN classifier [160].

6.3.1.1 Pre-processing and keypoint detection

Preprocessing generally aims to prepare the representative features desirable for knowledge generation. The frames of each video clip containing a certain action are smoothed by Gaussian convolution with a kernel of size 3×3 and variance $\sigma = 0.5$ to wipe off noise and weaken image distortion. A set of spatio-temporal keypoints is then detected from the video clip using the adaptive Harris detector (cf. Chapter 4). The obtained keypoints are thereafter filtered such that under a certain amount of additive noise only stable and more localized keypoints are retained. This is done in two steps: first low contrast keypoints are discarded, and second isolated keypoints not satisfying the spatial constraints of feature points are excluded (as they are out of the spatial scope of a target object).

6.3.1.2 Extracted local features

In this section, the features based on log-polar histograms (discussed in detail in the first part of Chapter 4) are used to describe the local spatio-temporal shape characteristics of actions. Initially, each video sequence is partitioned into several time-slices. These slices are defined by linguistic intervals. Gaussian membership functions (see Fig. 6.5) are used to describe such intervals. To extract the local features representing action at an instance of time, the basic idea of the shape context [122] is modified. The idea behind a modified shape context is based on computing rich descriptors for fewer keypoints. As described in Chapter 4), to compute the modified shape context of action pose, a log-polar histogram is overlaid on the shape of action, as shown in Fig. 6.6. Thus, the fuzzy log-polar histogram representing action at a time-slice j can be constructed using membership functions:

$$h_j(k_1, k_2) = \sum_{\substack{\rho_i \in bin(k_1), \\ \theta_i \in bin(k_2)}} \mu_j(t_i),\ j = 1, 2, \ldots, m \tag{6.1}$$

Each of these histograms is a 2d array of $d\rho \times d\eta$ dimensional, where $d\rho$ and $d\eta$ are the radial and angular dimensions, respectively. By applying a simple linear transformation on the indices k_1 and k_2, the 2d histograms can be converted into one dimensional. The resulting histograms are then normalized to the integral value of unity to achieve robustness to scale variations. The normalized histograms can be used as shape contextual information for classification (see Fig. 6.7). Now, we can directly concatenate these normalized histograms to obtain one feature vector per video clip. In contrast, in this work, we aim to enrich these histograms

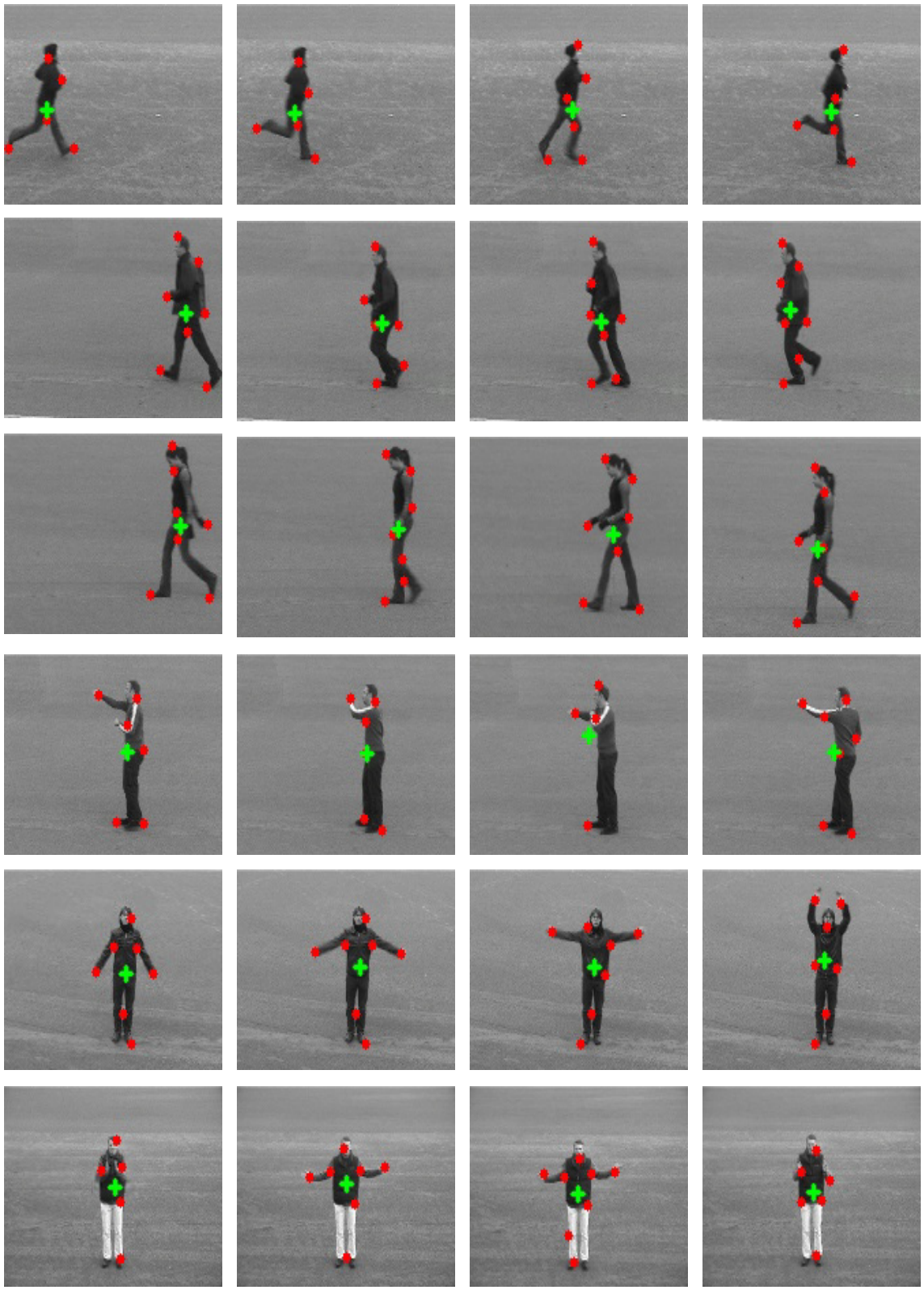


FIG. 6.4. Sparse feature points (marked in red) extracted form sample sequences containing actions of running, jogging, walking, boxing, waving and clapping, from top to bottom respectively; the green cross in each sequence locates the centroid of the extracted points.

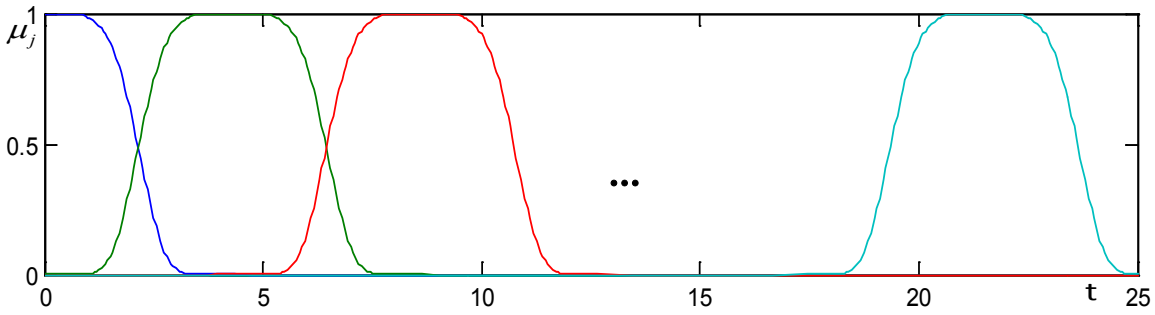


FIG. 6.5. Gaussian membership functions[3] used to represent the temporal intervals, with $\varepsilon_j = \{0, 4, 8, \ldots\}$, $\sigma = 2$, and $\gamma = 3$.

with the self-similarity analysis after using a suitable distance function to measure similarity (more precisely dissimilarity) between each pair of these histograms. This is significant to trim down the dimensionality of feature vectors.

Similarity measure

Video analysis is seldom carried out directly on row video data. Instead feature vectors extracted from small portions of video (i.e., so-called frames) are used. Thus the similarity between two video segments is measured by the similarity between their corresponding feature vectors. For comparing the similarity between two vectors, one can use several metrics such as Euclidean metric, Cosine metric, Mahalanobis metric, etc. Whilst such metrics may have some intrinsic merit, they have some limitations to be used with our approach because we might care more about the overall shape of expression profiles rather than the actual magnitudes, which is of main concern in applications such as action recognition. Therefore, we use a different similarity metric in which the relative changes are considered. Such a metric is based on Pearson Linear Correlation (PLC), and given by:

$$\rho(\vec{u}, \vec{v}) = \frac{\sum_{i=1}^{K}(u_i - \bar{u})(v_i - \bar{v})}{\sqrt{\sum_{i=1}^{K}(u_i - \bar{u})^2 \sum_{i=1}^{K}(v_i - \bar{v})^2}} \tag{6.2}$$

where $\bar{u} = \frac{1}{K}\sum_{i=1}^{K} u_i$ and $\bar{v} = \frac{1}{K}\sum_{i=1}^{K} v_i$ are the means of $\vec{u}$ and $\vec{v}$ respectively. The expression profiles are shifted down (by subtracting the means) and scaled by the standard deviations (i.e., the data have $\mu = 0$ and $\sigma = 1$). Note that Pearson linear correlation (PLC) is a measure that is invariant to scaling and shifting of the expression values. The value of PLC is constrained between -1 and $+1$ (perfectly anti-correlated and perfectly correlated). This is a similarity measure, but it can be

[3] Note that in Fig. 6.5, each membership function representing a temporal interval is plotted in a different color to enable visual discrimination.

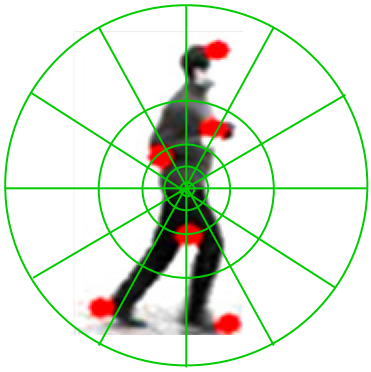


FIG. 6.6. Log-polar histogram representing shape contextual information of actions.

easily enforced to be a dissimilarity measure by:

$$s(\vec{u},\vec{v}) = \frac{1-\rho(\vec{u},\vec{v})}{2} \tag{6.3}$$

Temporal self-similarities

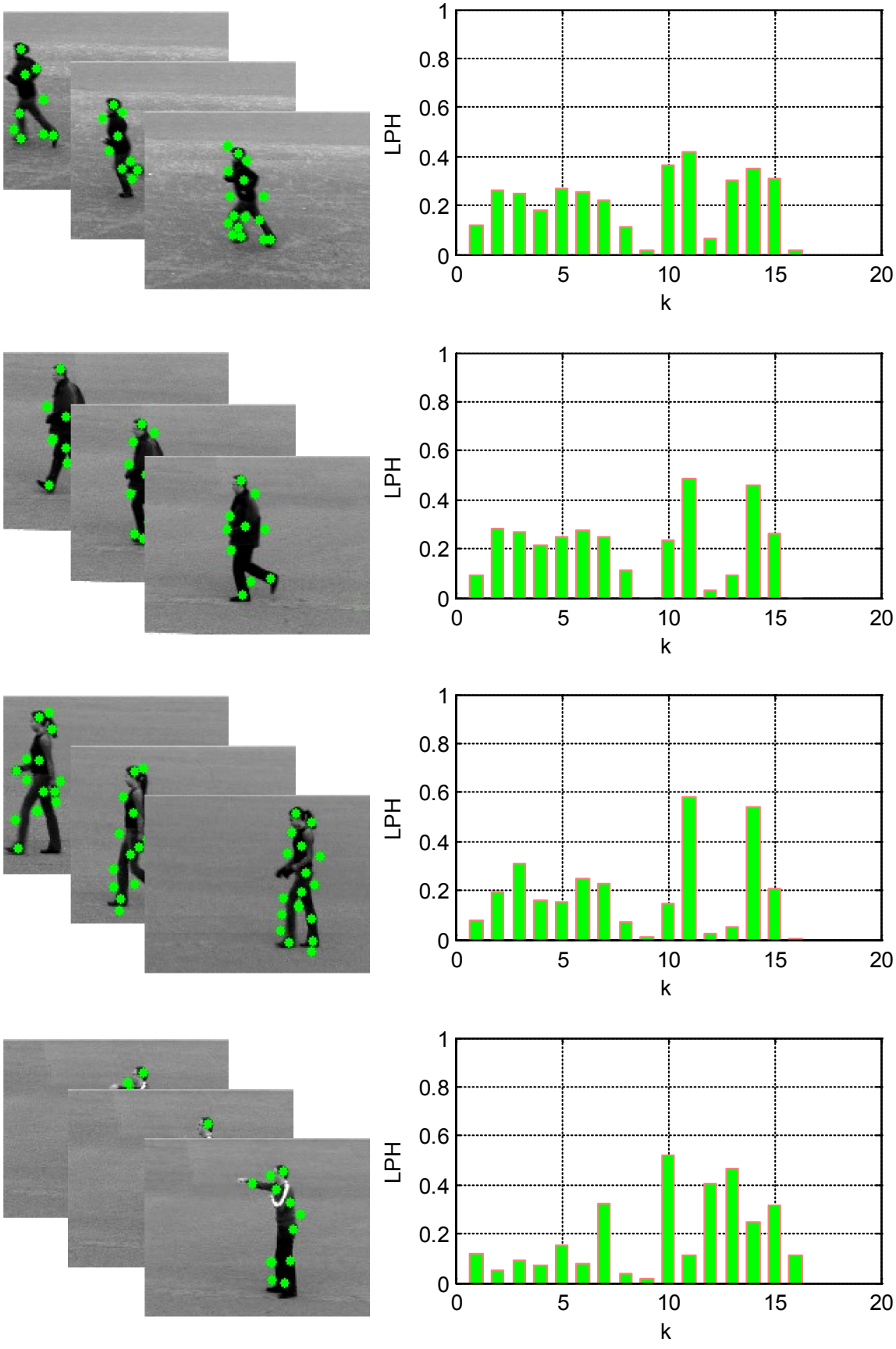
To reveal the inner structure of a human action in a video clip, second statistical moments (i.e., mean and variance) are not enough. Instead, self-similarity analysis seems to be a much more appropriate paradigm, which can formally formulated as follows. Given a histogram series $H = \langle \hbar_1, \hbar_2, \ldots, \hbar_m \rangle$ representing m time-slices of an action, then the self-similarity matrix is defined by:

$$S = [s_{ij}]_{i,j=1}^{m} = \begin{pmatrix} 0 & s_{12} & \cdots & s_{1m} \\ s_{21} & 0 & \cdots & s_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m1} & s_{m2} & \cdots & 0 \end{pmatrix} \tag{6.4}$$

where $s_{ij} = s(\hbar_i, \hbar_j)$. The diagonal entries are zero, as $s(\hbar_i, \hbar_i) = 0$. In addition, since $s_{ij} = s_{ji}$, S is a symmetric matrix.

6.3.1.3 Fusing motion information

It follows from the previous sections that the local features extracted based on using fuzzy log-polar histograms have been emphasized. The use of motion information has proven to be very beneficial in many applications of object recognition. This may motivate us to fuse motion information and local features to form the final MSNN classifier. The motion features extracted here are based on calculating the


LPH
k
0
0.2
0.4
0.6
0.8
1
0
5
10
15
20

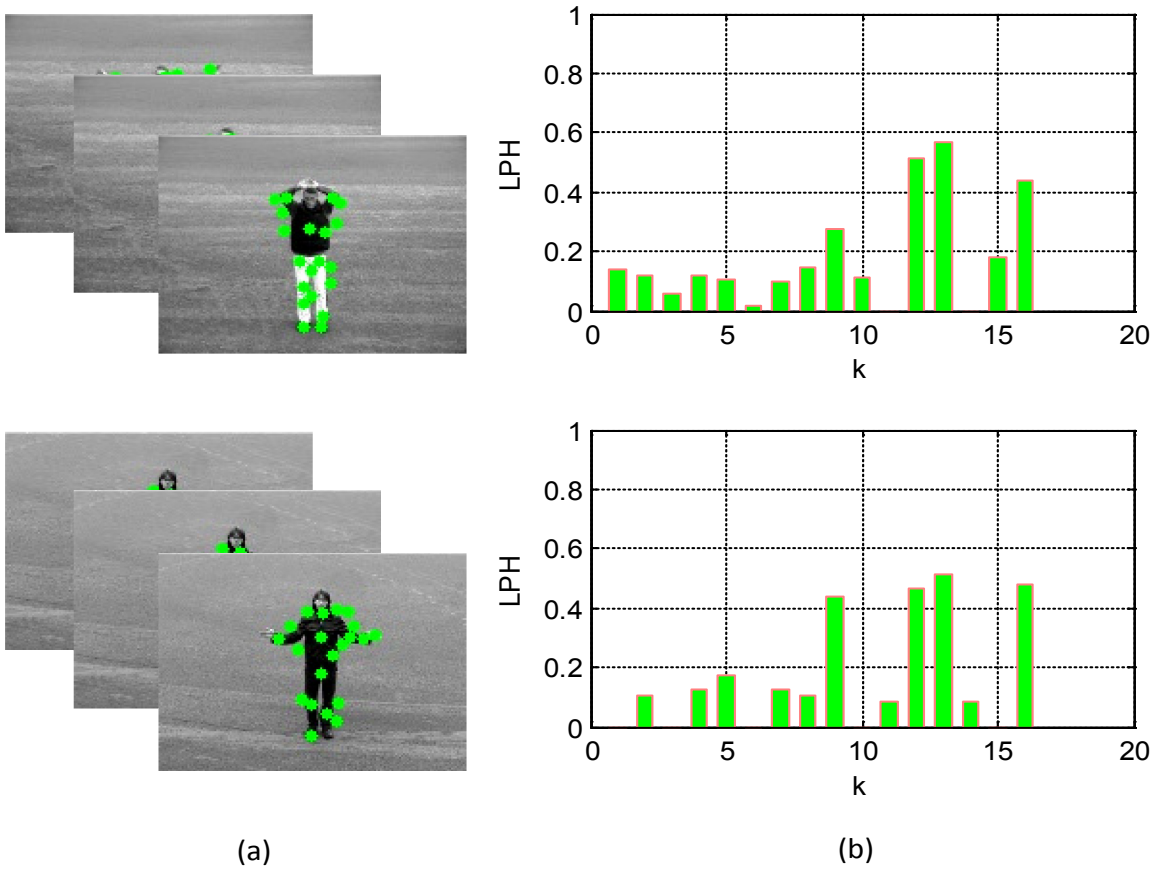


FIG. 6.7. Fuzzy log-polar histograms for motion description: (a) sample sequences with dense detected interest points for six different actions of running, jogging, walking, boxing, waving and clapping from top to bottom respectively, (b) the corresponding Fuzzy log-polar histograms obtained for the actions in (a).

centroid $\vec{c}(t)$ that delivers the center of motion (see Fig. 6.8). Therefore, the features $\vec{v}(t)$ describing the general distribution of motion are given by

$$\vec{v}(t) = \frac{\delta \vec{c}(t)}{\delta t} \tag{6.5}$$

where the spatial coordinates of $\vec{c}(t)$ are given by:

$$\begin{aligned} c_x &= \frac{1}{6\lambda} \sum_{i=1}^{n} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \\ c_y &= \frac{1}{6\lambda} \sum_{i=1}^{n} (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \end{aligned} \tag{6.6}$$

where $\lambda = \frac{1}{2} \left| \sum_{i=1}^{n} (x_i y_{i+1} - x_{i+1} y_i) \right|$. Such features have profound implications, not only about the type of motion (e.g., translational or oscillatory), but also about the rate of motion (i.e. velocity). With these features, it would be able to distinguish, for example, between an action where motion occurs over a relatively large area (e.g., running) and an action localized in a smaller region, where only small parts

FIG. 6.8. Center of gravity (marked in red) delivering the center of motion in various video sequences containing actions of running, walking, siding, waving, bending, and p-jumping form top to bottom, respectively; the green line in each sequence is to visualize the trajectory of motion centroid over time.

of the body are in motion (e.g., boxing). It is worth mentioning that fusing motion information with regular local features consistently boosts action recognition (i.e., leads to an overall increase in recognition rates).

6.3.1.4 MSNN action classification

In order to learn human actions, action recognition is modeled as a multi-class classification task where there is one class for each action, and the goal is to assign an action to an individual. There are various supervised learning algorithms by which the action recognizer can be trained. The MSNN classifier described in detail in Chapter 5 is used for the current classification task due to its outstanding generalization capability and reputation of a highly accurate paradigm. The basic model of MSNN is a multi-layer feedforward network with two hidden layers of 20 neurons each, which is most similar to the traditional MLP network structure but with improving in the hidden-unit adaptive activation functions (i.e., Multi-level Activation Functions, see Fig. 5.2 in Chapter 5). In the experiments, six categories of actions are defined and the objective is to classify each of these actions into one of the categories. Before the training phase, the classifier begins with random weights at the connections between the neurons. The learning procedure followed by the MSNN classifier is very similar to the popular backpropagation procedure presented in Chapter 5. During the learning stage, the classifier is trained using the features extracted from the action snippets in the training dataset. The up diagonal elements of the similarity matrix representing the local features are first transformed into plain vectors, and then concatenated with the global features of motion. All feature vectors are finally fed into the MSNN classifier to distinguish all action classes. After the learning stage is finished, the system is able to recognize and identify unseen actions. In fact, the MSNN classifier produces a real value between 0 and 1 that can be easily binarized by using a specific threshold.

6.3.1.5 Recognition results on KTH dataset

In this experiment, the proposed approach for action recognition is evaluated on the KTH dataset. To illustrate the effectiveness of our recognition approach, the obtained results are compared with those of other similar state-of-the-art methods in the literature. In order to prepare the simulation and to provide an unbiased estimation of the generalization abilities of the classification process, the sequences, for each action, were divided into a training set (two thirds) and a test set (one third). This was done such that both sets contained actions form all persons. The MSNN classifier is trained on the training set, while the evaluation of the recognition

Table 6.1. Confusion matrix obtained on KTH dataset.

ACTION	Walking	Running	Jogging	Waving	Clapping	Boxing
Walking	0.99	0.00	0.01	0.00	0.00	0.00
Running	0.00	0.96	0.04	0.00	0.00	0.00
Jogging	0.07	0.06	0.87	0.00	0.00	0.00
Waving	0.00	0.00	0.00	0.95	0.00	0.05
Clapping	0.00	0.00	0.00	0.00	0.94	0.06
Boxing	0.00	0.00	0.00	0.00	0.02	0.98

performance is performed on the test set. The confusion matrix depicting the results of action recognition obtained with this method and the comparison of our results with those of other related studies in the literature are shown in Table 6.1 and Table 6.2 respectively. As follows from the figures tabulated above, most of actions are correctly classified. Furthermore there is a high distinction between arm actions and leg actions. Most of the mistakes where confusions occur are between ”jogging” and ”running” actions and between ”boxing” and ”clapping” actions. This is intuitively plausible due to the fact of high similarity between each pair of these actions. From the comparison in Table 6.2, it turns out that the

Table 6.2. Comparison with other state-of-the-art methods on KTH dataset.

Method	Recognition rate
Our method	**93.6%**
Ke *et al.* [161]	63.0%
Dollár *et al.* [51]	81.2%
Rapantzikos *et al.* [64]	88.3%
Rodriguez *et al.* [42]	88.6%
Jhuang *et al.* [148]	91.7%
Wang *et al.* [162]	92.5%
Liu *et al.* [65]	92.8%
Kim *et al.* [163]	95.3%

proposed method performs competitively with state-of-the-art methods and its results compare favorably to those reported in the literature. Notably, the methods we compare with use similar experimental setups, except the method of Kim *et al.* [163] that achieved an impressive accuracy and the spatio-temporal alignment of video sequences was manually carried out. Hence, the comparison seems to be fair. Furthermore, using self-similarity analysis trims down the dimensionality of features, which enables the method to be applicable in real-time implementation.

Table 6.3. Confusion matrix obtained on Weizmann dataset

ACTION	wave2	wave1	walk	skip	side	run	pjump	jump	jack	bend
wave2	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
wave1	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
walk	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
skip	0.00	0.00	0.00	0.89	0.00	0.00	0.00	0.11	0.00	0.00
side	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
run	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
pjump	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
jump	0.00	0.00	0.00	0.00	0.00	0.11	0.00	0.89	0.00	0.00
jack	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
bend	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

6.3.1.6 Recognition results on Weizmann dataset

We conducted this second experiment was on the popular benchmark Weizmann action dataset [2]. Again, in order to provide an unbiased estimate of the generalization abilities of the method, the leave-one-out cross-validation technique was used in the validation process. As the name suggests, this involves using a group of sequences from a single subject in the original dataset as the testing data, and the remaining sequences as the training data. This is repeated such that each group of sequences in the dataset is used once as the validation. More specifically, the sequences of 8 subjects were used for training and the sequences of the remaining subject were used for validation data. Then the SVM classifiers with Gaussian radial basis function kernel are trained on the training set, while the evaluation of the recognition performance is performed on the test set. In Table 6.3, the recognition results obtained on the Weizmann dataset are summarized in a confusion matrix, where correct responses define the main diagonal.

From the figures in the matrix, a number of points can be drawn. The majority of actions are correctly classified. An average recognition rate of 97.8% is achieved with our proposed method. What is more, there is a clear distinction between arm actions and leg actions. The mistakes where confusions occur are only between *skip* and *jump* actions and between *jump* and *run* actions. This is also due to the high closeness or similarity among the actions in each pair of these actions. Once more, in order to quantify the effectiveness of the proposed method, the obtained results are compared qualitatively with those obtained previously by other investigators. The outcome of this comparison is presented in Table 6.4.

In light of this comparison, one can see that the proposed method is competitive

Table 6.4. Comparison with other similar methods on Weizmann dataset

Method	Recognition rate
Our method	**97.8%**
Kläser *et al.* [149]	84.3%
Dollár *et al.* [51]	85.2%
Niebles *et al.* [59]	90.0%
Zhang *et al.* [164]	92.8%
Bregonzio *et al.* [62]	96.6%
Fathi *et al.* [165]	100%

with other state-of-the-art methods. It is worthwhile to mention that all the methods [51,59,62,149,164] with which our method is compared, except the method proposed in [165], used similar experimental setups, thus the comparison seems to be most fair. A final remark concerns the computational burden of the approach that determines whether or not the proposed method has a potential for real-time application. In both experiments, the action recognizer comfortably runs at 22 fps on average in Microsoft Visual Studio 2008 and OpenCV Library (using a 2.8 GHz Intel dual core machine with 4 GB of RAM, running Microsoft Windows 7 Professional). This might lend support to the expectation that the method would be viable in real world settings and amenable to working with real-time applications.

6.3.2 Activity recognition using multiple cues

In this section, we first describe briefly our second approach developed for human action recognition. Then the experimental design details and the evaluation results of this approaches are presented. A schematic block diagram depicting the major components of the approach is shown Fig. 6.9. As shown in the block diagram, the backgrounds are first subtracted form each video clip by using a Gaussian mixture background model to extract the silhouettes of the moving human body parts. For this method to be more robust against time warping effects, action snippets are temporally divided into a number of overlapping segments defined by fuzzy membership functions. Then local features are extracted from each temporal segment based on a variety of shape descriptors. As the motion features intuitively appear to be more relevant and appropriate to the current action recognition task, the final features fed into classifiers are constructed using both shape and motion features. These steps are explained in more detail below.

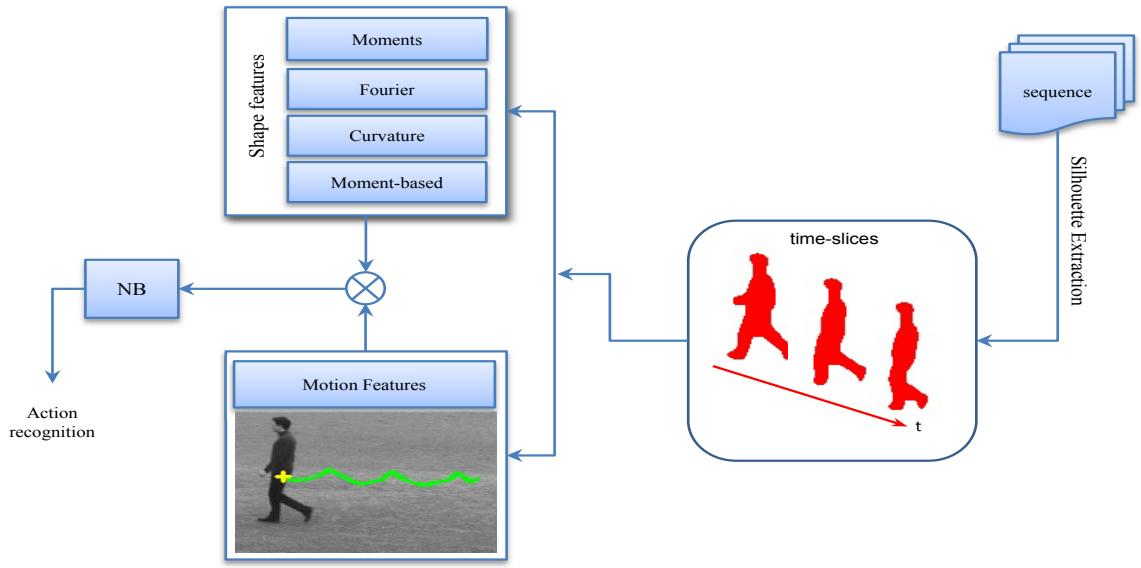


FIG. 6.9. Overview of the proposed approach for action recognition.

6.3.2.1 Preprocessing and background subtraction

As well-known, foreground segmentation (or background subtraction) is prone to noise. Therefore, before segmentation, preprocessing should be done to smooth action sequences and to get rid of impulse noise and irrelevant data from each frame of the action sequence. In this work, as a preprocessing stage, a Gaussian smoothing is used for blurring. Specifically, we use a 3×3 Gaussian filter with $\sigma = 1.4$ to reduce noise within each frame of input sequence. The basic principle of background subtraction in image sequences involves initializing and maintaining a statistical model to estimate the background of the scene. Moving objects (often called foreground objects) in the scene are then detected by checking the pixels in the scene that deviate significantly from the estimated background model [119]. Due to variation in lighting condition, multiple surfaces often appear in the view of a particular pixel (Fig. 6.10). Therefore, the mixture of gaussians is regarded as the most sufficient approximation to practical pixel process [166].

In this approach, we use Gaussian mixture model (GMM) to model background. In this model, each pixel in the scene is modeled using a mixture of K (usually 3-5) Gaussian distributions; we used $\mathrm{K} = 3$ in our experiments. The persistence and the variance of each gaussian of the mixture are used to determine which Gaussian probably corresponds to background colors. Pixels whose color values do not fit the background distributions are detected as foreground or moving object. More formally, let $\{X_1, \ldots, X_t\}$ be the history associated with a pixel at time t, where $X_i (i = 1, \ldots, t)$ are measurements of the RGB values at time i. The recent

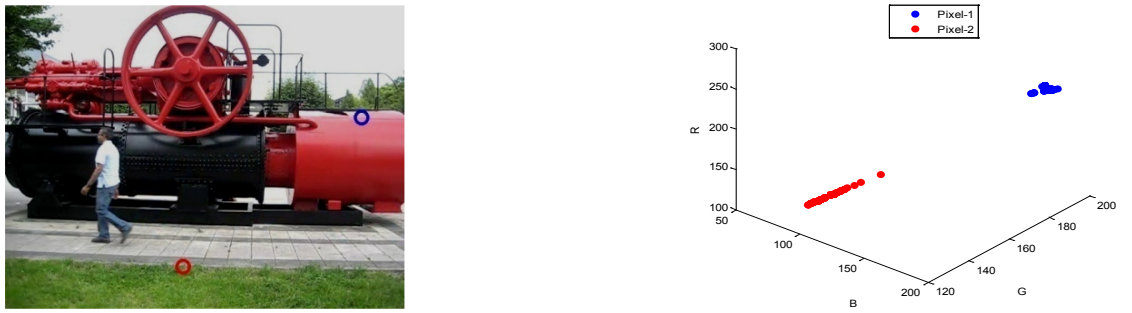


FIG. 6.10. Multi-modal distributions caused by illumination variations over time. Left: original sequence with two pixels circled in red and blue. Right: scatter plot for color distributions of the two pixels.

history of each pixel can be modeled reasonably well with a mixture of K Gaussian distributions. Thus the probability of observing the current pixel value is given by

$$P(X_t) = \sum_{i=0}^{\mathrm{K}} \omega_{i,t}\, \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \tag{6.7}$$

where $\omega_{i,t}$, t. $\mu_{i,t}$ and $\Sigma_{i,t}$ are an estimate of the weight, the mean value, and the covariance matrix of the i-th gaussian in the mixture at time t, respectively. η is a Gaussian probability density function:

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t-\mu_t)^T \Sigma^{-1} (X_t-\mu_t)} \tag{6.8}$$

Assuming the independence of the color channels, $\Sigma_{i,t}$ can be expressed as: $\Sigma_{i,t} = \sigma_i^2 \cdot I$. Thereafter, an on-line approximation is used to update the model in an iterative manner as follows. At each pixels all parameters of the most matched Gaussian are updated via an on-line K-means approximation, while only the weight parameters of others are updated while their means and variances remain unchanged (for further details, refer to Chapter 3). Sample results of applying the GMM background subtraction technique to various video sequences of different persons performing actions of walking, jogging, and running are shown in Fig. 6.11. From the sequences provided in the figure, one can observe that the GMM background learning model can be applied to scenes with backgrounds of different luminances.

6.3.2.2 Feature extraction

In this approach, a variety of features are used to describe the segmented silhouettes of moving human body parts. Such features are thought to play a primary role regarding the interpretation of human motion and labeling of human actions. Furthermore, the information of motion are also used, which are extracted by following

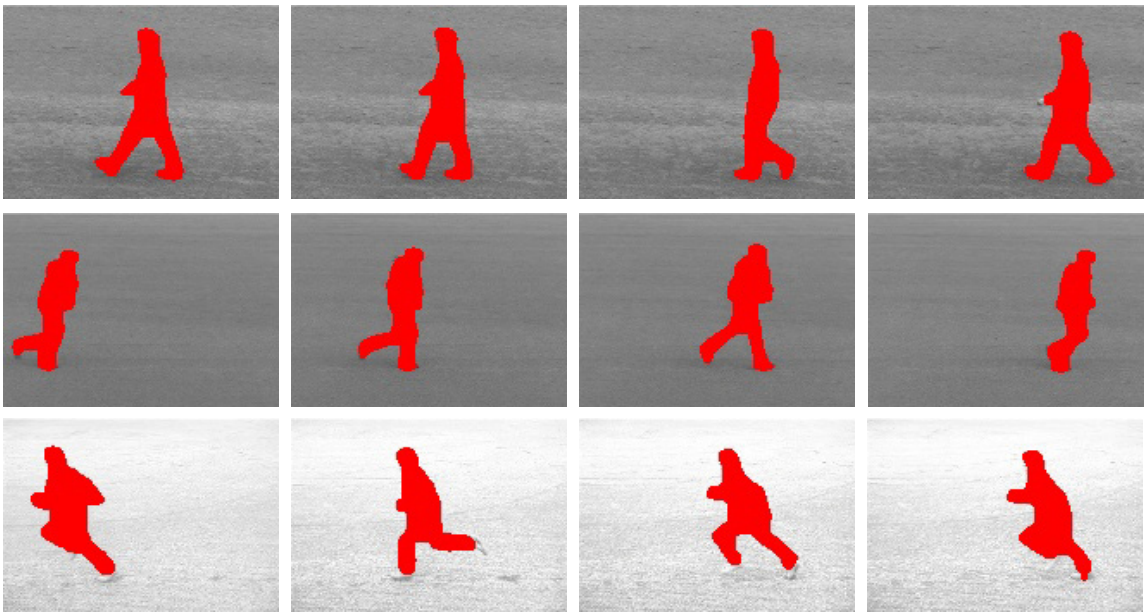


FIG. 6.11. Example silhouette sequences resulted from applying GMM to three sequences including actions of walking, jogging, and running from top to bottom, respectively.

the trajectory of the motion centroid, as described by end of this section. As in the first approach, before starting the feature extraction process, all action snippets are temporally split into several time-slices. The time-slices are defined by linguistic intervals. A Fuzzy membership function is used to describe each of these intervals (refer to Fig. 4.6 in Chapter 4). All the membership functions are chosen to be of identical shape on condition that their sum is equal to one at any instance of time t. For shape features, we consider a variety of invariant descriptors such as Fourier descriptors, curvature features, invariant shape moments, etc. Below we describe in more detail how such features are defined and extracted.

Fourier descriptors:

In this work, Fourier descriptors for action silhouettes are obtained based on the notion of the shape signature z_i (cf. Chapter 4) as follows:

$$c_k = \frac{|a_{k+2}|}{|a_1|}, k = 0, 1, \ldots, n-3 \tag{6.9}$$

where n is the number of the points of the shape boundary and Fourier transform coefficients $\{a_k\}_{k=0}^{n-1}$ are given by

$$a_k = \frac{1}{n}\sum_{i=0}^{n-1} z_i \, \exp(-\frac{j2\pi ik}{n}), \; k = 0, 1, \ldots, n-1 \tag{6.10}$$

From Eq. (6.9), it can be verified that this choice of coefficients guarantees that the resulting shape descriptors are invariant to shape translation, rotation and scaling, and they are independent of the choice of the starting point on the contour.

Moment invariants:

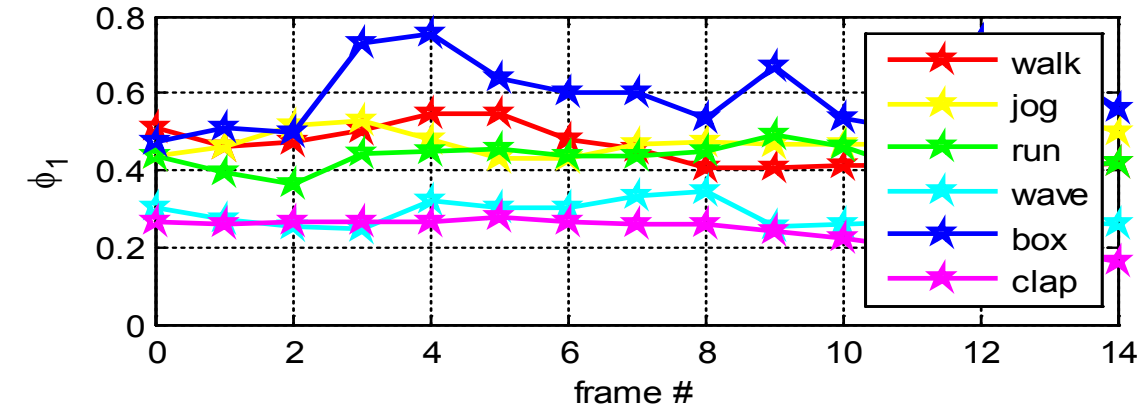
As discussed earlier in Chapter 4, relative and absolute combinations of moments which are invariant with respect to translation, rotation, and scaling changes are derived based on the theory of algebraic invariants. Rigorously speaking, the set of absolute moment invariants contains a set of nonlinear combinations of central moments invariant under rotation. Specifically, in this work, the following set of functions that is invariant with respect to object scale, translation and rotation are chosen as shape feature candidates:

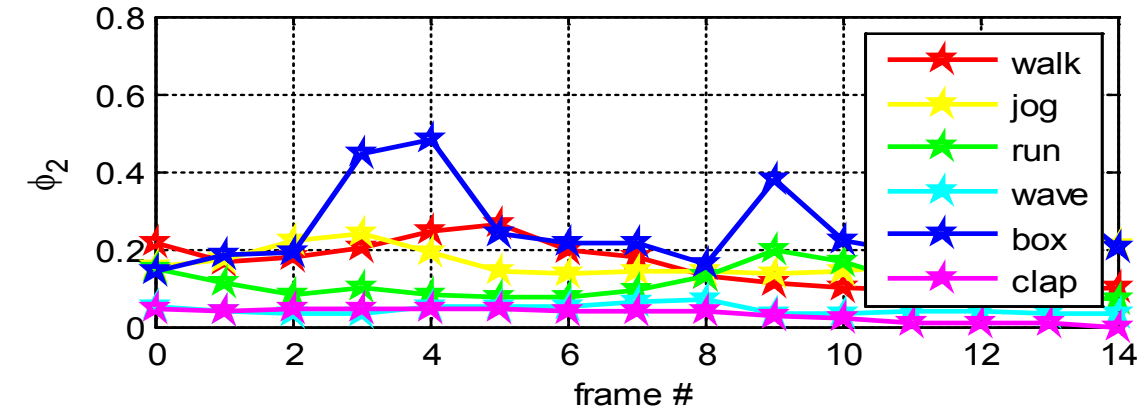
$$\begin{aligned}
\phi_1 &= \mu_{20} + \mu_{02} \\
\phi_2 &= (\mu_{20} - \mu_{02})^2 + (2\mu_{11})^2 \\
\phi_3 &= (\mu_{30} - 3\mu_{12})^2 + (3\mu_{03} - \mu_{21})^2 \\
\phi_4 &= (\mu_{30} + \mu_{12})^2 + (\mu_{03} + \mu_{21})^2 \\
\phi_5 &= (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{03} + \mu_{21})^2] \\
&\quad +(3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{03} + \mu_{21})^2] \\
\phi_6 &= (\mu_{20} - \mu_{02})[(\mu_{30} + \mu_{12})^2 - (\mu_{03} + \mu_{21})^2] + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{03} + \mu_{21}) \\
\phi_7 &= (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{03} + \mu_{21})^2] \\
&\quad +(\mu_{30} - 3\mu_{12})(\mu_{03} + \mu_{21})[3(\mu_{30} + \mu_{12})^2 - (\mu_{03} + \mu_{21})^2]
\end{aligned} \tag{6.11}$$

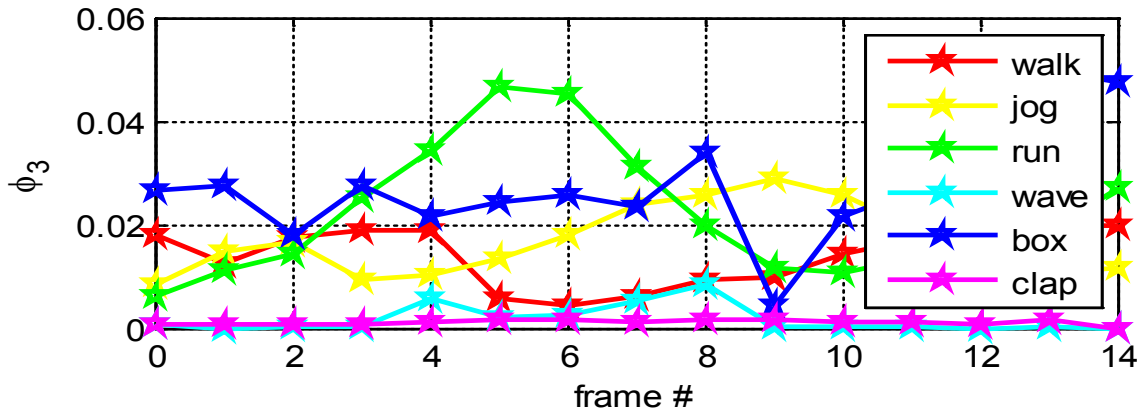
It should be noted that the above functions can be implicitly deduced by normalizing central moments up to order three. It is not difficult to verify that the functions ϕ_1 through ϕ_6 are invariant with respect to rotation and refection, while ϕ_7 changes sign under refection. In Fig. 6.12, the moment invariants $\{\phi_i, i = 1, 2, 3, 4, 5, 7\}$ for actions of walking, jogging, running, waving, clapping are shown.

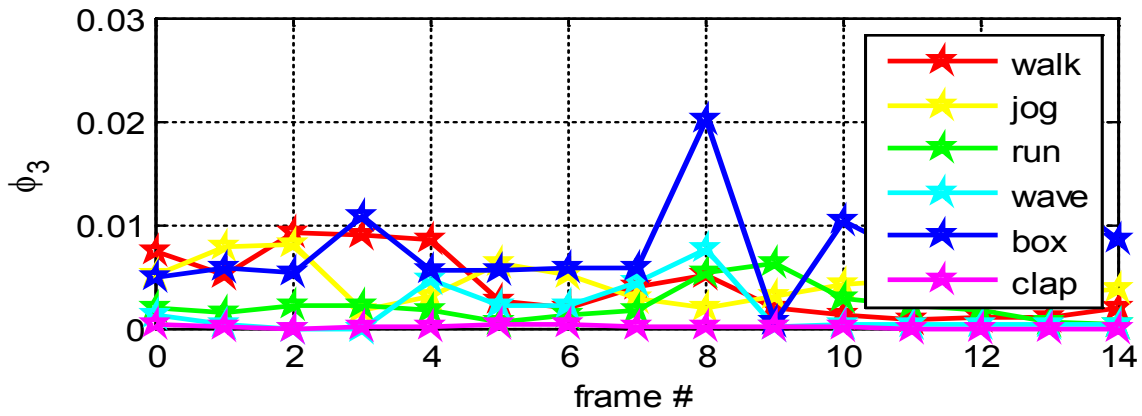
Curvature features:

In a way similar to the extraction of the Mel Frequency Cepstral Coefficients (MFCC) features from voice signals, a set of other shape descriptors based on the cepstrum of the shape curvature can be also extracted. The mechanism works as follows. First we extract the shape curvature by using Freeman chain code [134]. Then, the cepstrum of the curvature signal is obtained, and finally a certain number (e.g. $n = 10$) of the largest coefficients can be chosen to be added to the feature vector. These steps have been described in detail in Chapter 4.


0.8
0.6
0.4
0.2
0
ϕ_1
0
2
4
6
8
10
12
14
frame #
walk
jog
run
wave
box
clap


0.8
0.6
0.4
0.2
0
ϕ_2
0
2
4
6
8
10
12
14
frame #
walk
jog
run
wave
box
clap


0.06
0.04
0.02
0
ϕ_3
0
2
4
6
8
10
12
14
frame #
walk
jog
run
wave
box
clap


0.03
0.02
0.01
0
ϕ_3
0
2
4
6
8
10
12
14
frame #
walk
jog
run
wave
box
clap

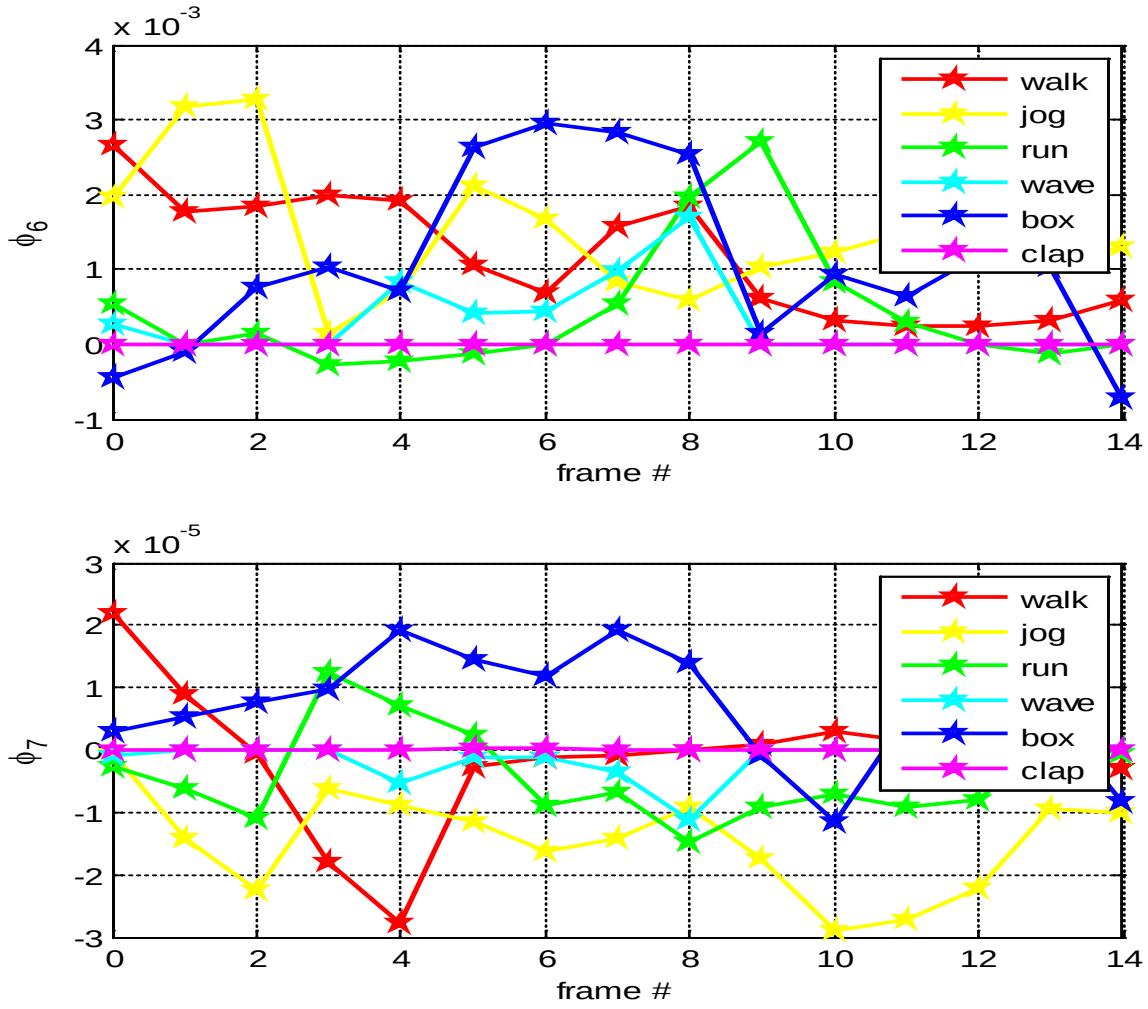


FIG. 6.12. Moment invariants values for different actions (i.e., walking, jogging, running, boxing, waving, and clapping).

Moment-based features:

Besides the moment invariants, a set of other features can be derived from the central moments. More specifically, given the central moments of second order μ_{11}, μ_{02} and μ_{20}, the covariance matrix (corresponding to inertial tensor, refer to Chapter 4) containing information about the object's orientation is defined as

$$\Sigma = \begin{bmatrix} \acute{\mu}_{20} & -\acute{\mu}_{11} \\ \acute{\mu}_{11} & \acute{\mu}_{02} \end{bmatrix} \tag{6.12}$$

where $\acute{\mu}_{20} = \frac{\mu_{20}}{\mu_{00}}, \acute{\mu}_{02} = \frac{\mu_{02}}{\mu_{00}}$, and $\acute{\mu}_{11} = \frac{\mu_{11}}{\mu_{00}}$. Thus, the eigenvalues of the covariance matrix proportional to the squared length of the eigenvector axes is given by

$$\lambda_{1,2} = \frac{\acute{\mu}_{02} + \acute{\mu}_{20} \pm \sqrt{4\acute{\mu}_{11}^2 + (\acute{\mu}_{02} - \acute{\mu}_{20})^2}}{2} \tag{6.13}$$

In this way, the orientation can be determined from the angle of the eigenvector associated with the largest eigenvalue:

$$\phi = \frac{1}{2}\arctan\left(\frac{2\acute{\mu}_{11}}{\acute{\mu}_{20}-\acute{\mu}_{02}}\right) \tag{6.14}$$

Other parameters such as the roundness κ and eccentricity ε seem to be very close. The scaled roundness κ can be determined by

$$\kappa = \frac{\ell^2}{4\pi\Lambda} \tag{6.15}$$

where Λ and ℓ denote the area and perimeter of the shape, respectively. ε can be readily calculated from the second-order central moments (or eigenvalues) using:

$$\varepsilon = \frac{(\acute{\mu}_{20}-\acute{\mu}_{02})^2 - 4\acute{\mu}_{11}^2}{(\acute{\mu}_{20}+\acute{\mu}_{02})^2} = \sqrt{1-\frac{\lambda_2}{\lambda_1}} \tag{6.16}$$

where $\lambda_1 > \lambda_2$. The variation of orientation φ, roundness κ, and eccentricity ε along time for different actions are shown in Fig. 6.13 form top to bottom, respectively.

Thereafter, the feature values of each action snippet are normalized to fit a zero-mean and a unit variance distribution. The normalized vectors obtained can now be used as shape contextual information for classification and matching. Many approaches in various object recognition applications directly combine these vectors to get one final vector per video and classify it using any classification algorithm. It is worth mentioning that concatenating all the feature vectors extracted from all frames of one action snippet would result in a very large feature vector that might be less likely to be classified correctly. To circumvent this problem and to reduce the dimensionality of extracted feature vectors, the feature vectors of each action snippet in a time-slice are weighted and averaged:

$$\vec{\mu} = \frac{1}{\tau}\sum_{t=1}^{\tau} w_t \vec{x}_t \tag{6.17}$$

where $w_t = f(t;\alpha,\beta,\gamma)$ is the weighting factor, $f(.)$ is Gaussian membership function, and τ is the number of the feature vectors in the time-slice. Finally, the feature vectors resulting at each of the time-slices are concatenated to yield the final feature vector for the action snippet.

6.3.2.3 Motion features

From the discussion in the previous subsections, it follows that the local shape features obtained at each time-slice are emphasized. In the other hand, motion information have proven to be powerful cues for recognition in many applications

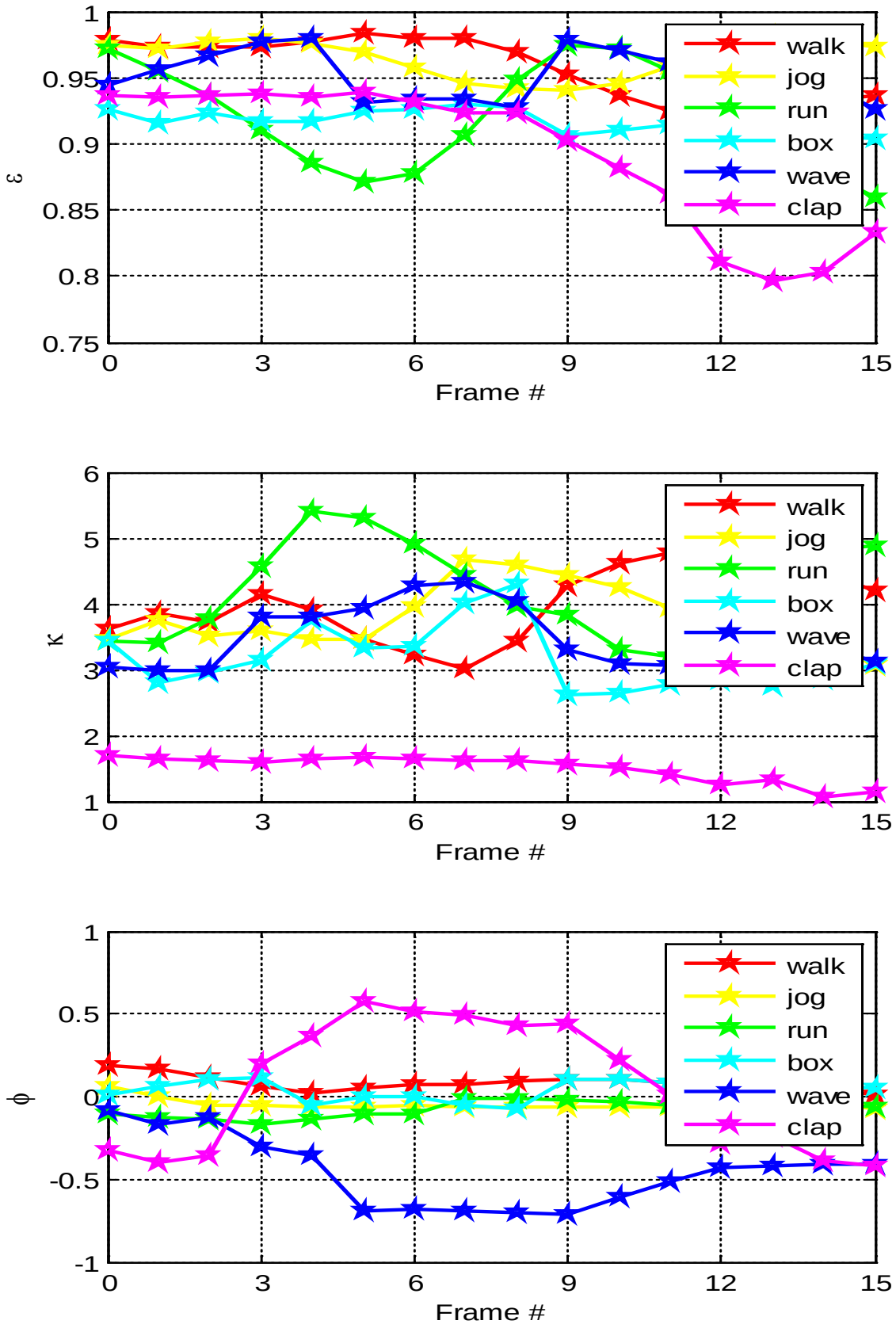


FIG. 6.13. Moment-based features for different categories of actions

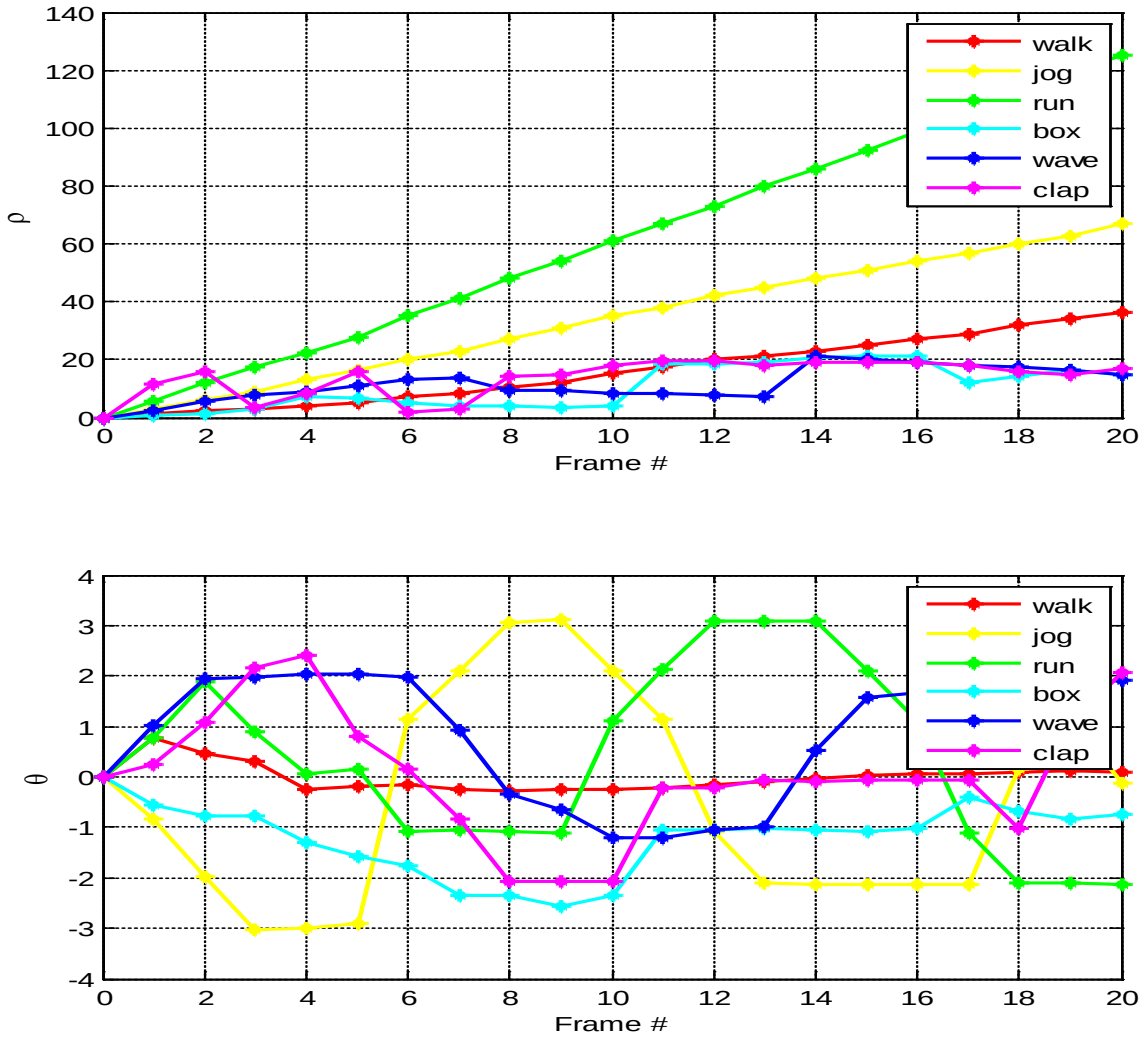


FIG. 6.14. Results for motion features: magnitude ρ (top row) and phase θ) (bottom row) extracted from action sequences.

of object recognition. This motivate us to enrich the shape features by fusing the motion information to form the final action vector fed to the NB classifier. All the motion features extracted here are based on calculating the center of mass $\vec{z}$ that delivers the center of motion. Thus, the temporal features describing the distribution of motion are given from:

$$\vec{v} = \lim_{\Delta t \to \infty} \frac{\Delta \vec{z}(t)}{\Delta t} \tag{6.18}$$

where $\vec{z} = (\frac{1}{n}\sum_{i=1}^{n} x_i, \frac{1}{n}\sum_{i=1}^{n} y_i,)$ and n is the total number of moving pixels in the frame. Alternatively, by using spatial moments, $\vec{v} = (M_{10}/M_{00}, M_{01}/M_{00})$, as described in detail in Chapter 4. These features are very informative not only about the type of motion (e.g., translational or oscillatory), but also about the rate of motion (i.e., velocity). With these features, it would be able to distinguish, for

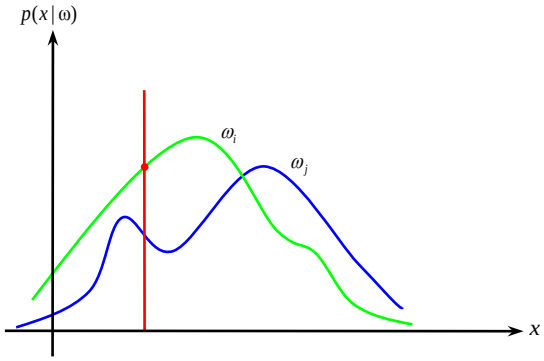


FIG. 6.15. Class-conditional probability density functions (pdfs).

example, between an action in which motion occurs over a relatively large area (e.g., running) and an action localized in a smaller region, where only small parts are in motion (e.g., boxing). Hence significant improvements in recognition performance are expected to be achieved by fusing shape and motion features. Fig. 6.14 shows sample results for motion features extracted from several action sequences, namely walking, jogging, running, boxing, waving, and clapping.

6.3.2.4 Naïve Bayes classification

In order to classify human actions, the task of action recognition is formulated as a multi-class learning problem where there is one class for each action and the main goal is to assign an action to an individual in each video sequence. There are various supervised learning algorithms by which an action recognizer can be trained. With this approach, Naïve Bayesian (NB) classifier is used.

As stated in Chapter 5, the main advantage of the NB classifier is that it requires a relatively small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. In spite of its naive design and apparently over-simplified assumptions, NB classifier has been shown to work quite well in many complex real-world situations [167]. Roughly speaking, given a final feature vector $\mathbf{x}$ extracted from an action sequence, posteriori probabilities can be calculated directly from training action snippets by using Bayes rule. Strictly speaking, Bayes' rule is a fundamental formula in decision theory, which is deduced straightforwardly from the definition of conditional probability:

$$p(\omega_i|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_i)p(\omega_i)}{p(\mathbf{x})} \tag{6.19}$$

where $p(\mathbf{x}) = \sum_{i=1}^{K} p(\mathbf{x}|\omega_i)p(\omega_i)$. $p(\omega_i|\mathbf{x})$ is the posteriori probability of observing the class ω_i given the feature vector $\mathbf{x}$. $p(\omega_i)$ is the priori probability of observing the

Table 6.5. Confusion matrix of the recognition results

ACTION	Walking	Jogging	Running	Boxing	Waving	Clapping
Walking	0.99	0.01	0.00	0.00	0.00	0.00
Jogging	0.07	0.80	0.13	0.00	0.00	0.00
Running	0.01	0.08	0.91	0.00	0.00	0.00
Boxing	0.00	0.00	0.00	0.94	0.01	0.05
Waving	0.00	0.00	0.00	0.00	0.99	0.01
Clapping	0.00	0.00	0.00	0.02	0.00	0.98

class ω_i, $p(\mathbf{x}_i|\omega_i)$ is the conditional density and K is the total number of classes. For this recognition task, it is assumed that each action snippet is uniquely described by the value of its a posteriori probability. Furthermore, all the priori probabilities are assumed to be equal, and thus find the density functions for each of the classes, where each class refers to an action. Thus, K such densities are found, corresponding to the K different actions. Having obtained these K values for each of the classes, the most likely action is given by

$$P = \max[p_1, p_2, \ldots, p_K] \tag{6.20}$$

where P is the probability of the most likely class and $p_1, p_2, \ldots, p_K$ are the probabilities of K different action categories (i.e., $K = 6$ in this experiment). As a simple but concrete illustrating example, Fig. 6.15 shows two hypothetical probability density functions (pdfs) depicting the probability density of measuring a particular feature value x given the video sequence is in action class ω_i. The two curves in the Fig. describe the difference in x of populations of two types of action.

6.3.2.5 Numerical results and comparison with competitors

This approach has been experimentally evaluated on KTH benchmark action dataset. To assess the reliability of the approach, the obtained results were also compared with those reported in the literature [42, 51, 64, 65, 148, 161, 162]. In order to prepare the experimentation and to provide an unbiased estimation of the generalization abilities of the action classification process, the action sequences were divided with respect to the subjects into a training set and a test set. This was done such that both sets contained actions form all persons. The NB classifier was trained on the training set, while the evaluation of the recognition performance was performed on the test set. The confusion matrix depicting the results of action recognition achieved by the proposed method is shown in Table 6.5.

From the figures in this Table, a number of points can be drawn. The majority of actions are correctly classified. Additionally, there is a clear distinction between

Table 6.6. Comparison with some well-known studies in the literature.

Method	Recognition rates
Our method	**93.5%**
Liu *et al.* [65]	92.8%
Wang *et al.* [162]	92.5%
Jhuang *et al.* [148]	91.7%
Rapantzikos *et al.* [64]	88.3%
Dollár *et al.* [51]	81.2%
Rodriguez *et al.* [42]	88.6%
Ke *et al.* [161]	63.0%

arm actions and leg actions. Most of the mistakes where confusions occur are between ”jogging” and ”running” actions and between ”boxing” and ”clapping” actions. This is intuitively plausible due to the fact of high similarity between each pair of these actions. To assess the efficiency of the proposed method, the obtained results have been compared with those of other previously published studies in the literature, as shown in Table 6.6. From this comparison, it turns out that our approach performs competitively with other state-of-the-art approaches and its results compare favorably with previously published results. Notably all the methods that we compared our method with have used similar experimental setups, thus the comparison is meaningful.

6.3.3 Action recognition from chord-length-function features

The schematic diagram of the proposed approach for action recognition based on chord-length function features is shown in Fig. 6.16, while the details of the inner workings of each rectangle in that figure are given in the subsequent subsections.

6.3.3.1 Preprocessing and background subtraction

For later successful feature extraction and classification, the reliable features need to be emphasized. To assist in achieving this goal and to obtain accurate and representative features suitable for recognition and analysis, video sequences are initially preprocessed to reduce noise and prepare them for further processing (e.g., segmentation). More specifically, upon receipt of action sequences, the frames of each sequence are first smoothed with a 2D Gaussian filter of size 3×3 and standard deviation equal to 0.5 to reduce noise and weaken image artifacts. For background substraction, in a manner similar to that described in the previous approach, a Gaussian mixture model (GMM) is used to estimate the background.

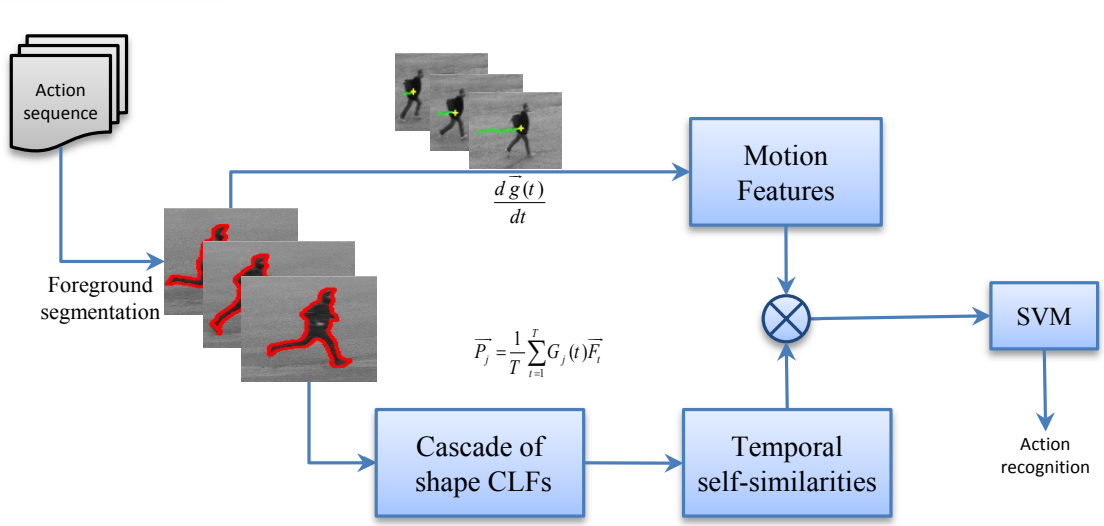


FIG. 6.16. Workflow of the proposed approach for action recognition.

6.3.3.2 Finding shape border

A shape border (or contour) is simply an outline representing or bounding the shape of object of interest. Therefore, the ability of extracting contours out of segmented body parts (i.e., silhouettes) plays a key role in the process of tracking pose features and recognizing human motion, action, and events in video sequences. This subsection is to briefly describe the method we used for the extraction and further manipulation of shape borders. Broadly speaking, let $F = \{f_{ij}\}$ be a silhouette segmented out of a sequence of action. Initially, set NBD to 1, where NBD stands for the sequential number of the current shape border). Then, the silhouette is scanned with a TV raster and the steps given as pseudocode in Algorithm 6.1 [168] are performed for each pixel such that $f_{ij} \neq 0$. Each time a new row in the silhouette is scanned, we reset LNBD[4] to 1. Fig. 6.18 shows sample shape borders (shown in red) extracted from action sequences.

6.3.3.3 Feature extraction

Similarly to the approaches described before, the feature extraction is initialized by dividing each video sequence into several time stages in order to reduce feature dimensionality. These states are defined by vague, linguistic intervals. Gaussian membership functions (see Fig. 6.5) are used to describe the temporal intervals. Note that the membership functions are chosen to be of identical shape on condition that their sum is equal to one at any instance of time. By using such fuzzy functions,

[4]In Algorithm 6.1, the LNBD stands for the sequential number of the shape border encountered most recently and pixels with densities 0 is called the 0-pixel.

Algorithm 6.1: Finding shape border algorithm.

Input : A binary picture (or silhouette) $F = \{f_{ij}\}$.
Output: A set of shape borders (or contours) of the silhouette.
while *bottom-right corner of the silhouette not reached* **do**
  if $f_{ij} = 1$ *and* $f_{i,j-1} = 0$ **then**
    decide the pixel $(i, j) \in$ OB (outer border);
    NBD $\leftarrow$ NBD $+1$;
    $(i_2, j_2) \leftarrow (i, j - 1)$;
  else
    if $f_{ij} = 0$ *and* $f_{i,j+l} = 0$ **then**
      decide $(i, j) \in$ HB (hole border);
      NBD $\leftarrow$ NBD $+1$;
      $(i_2, j_2) \leftarrow (i, j + 1)$;
      if $(f_{ij} > 1)$ **then**
        LNBD $\leftarrow f_{ij}$;
    Go to L;
  From (i_2, j_2), search clockwise for a nonzero pixel in $N_4(i, j)$ or $N_8(i, j)$ (see Fig. 6.17);
  if *nonzero pixel found* **then**
    $(i_1, j_1) \leftarrow$ first found nonzero pixel;
  else
    $f_{ij} \leftarrow -$ NBD;
    Go to L2;
  $(i_2, j_2) \leftarrow (i_1, j_1)$;
  $(i_3, j_3) \leftarrow (i, j)$;
L1: From the next element of (i_2, j_2), look counterclockwise for a nonzero pixel in $N_4(i_3, j_3)$ or $N_8(i_3, j_3)$;
  if *nonzero pixel found* **then**
    $(i_4, j_4) \leftarrow$ 1st nonzero pixel;
  if $(i_3, j_3 + 1)$ *is 0-pixel* **then**
    $f_{i_3,j_3} \leftarrow -$NBD;
  else
    if $(i_3, j_3 + 1)$ *is not 0-pixel and* $f_{i_3,j_3} = 0$ **then**
      $f_{i_3,j_3} \leftarrow$ NBD;
  if $(i_4, j_4) = (i, j)$ *and* $(i_3, j_3 + 1) = (i_1, j_1)$ **then**
    Go to L2;
  else
    $(i_2, j_2) \leftarrow (i_3, j_3)$;
    $(i_3, j_3) \leftarrow (i_4, j_4)$;
    Go back to L1;
L2: **if** $f_{ij} \neq 1$ **then**
    LNBD $\leftarrow |f_{ij}|$;
    Resume the raster scan form $(i, j + 1)$;

4-Neighborhood(N_4) 8-Neighborhood(N_8)

FIG. 6.17. Pixel connectivity. Left:4-Neighborhood, Right: 8-Neighborhood.

not only can temporal information be easily extracted, the performance decline due to time warping effects can also be nullified. As discussed in Chapter 4, given a shape, $k/2$ CLFs (i.e. chord-length-functions) can be defined by dividing the shape border into k arcs. These functions are invariant to translation, rotation, and scaling. However, like other shape descriptors, these descriptors are not sufficiently compact. In addition, they depend constantly on a reference point whereby the shape border is parameterized. This dependence is simply because the contour is closed and any point on the contour can be used as a reference point, thus the CLFs might be changed. In order to avoid these problems and for convenience, the mean μ_r and variance σ_r of the CLFs are used

$$\mu_r = \frac{1}{n}\sum_{i=0}^{n-1}\lambda_r^{(i)} \quad \sigma_r = \frac{1}{n-1}\sum_{i=0}^{n-1}(\lambda_r^{(i)} - \mu_r)^2 \tag{6.21}$$

As an example the CLFs based features (as a shape descriptor) obtained form several video sequences of persons performing different actions are shown in Fig. 6.19. The CLF descriptor of shape (at time t) can be written as

$$F_t = \langle \mu_1, \sigma_1, \mu_1, \sigma_1, \ldots, \mu_{\frac{k}{2}}, \sigma_{\frac{k}{2}} \rangle \tag{6.22}$$

To obtain the CLFs descriptor of an action, we first obtain the CLFs descriptor for all poses of the action. As each action snippet was divided into a number of time-slices. Thus, the CLFs descriptor of an action pose is given by:

$$p_j = \frac{1}{\tau}\sum_{t=1}^{\tau}\mathcal{G}_j(t)F_t, \quad j = 1, 2, \ldots, m \tag{6.23}$$

where τ is the length of temporal state. Accordingly the final CLFs descriptor of a given action can be constructed by catenating all the CLFs descriptors of its temporal poses. The resulting feature vectors (i.e., CLFs descriptors) are then normalized

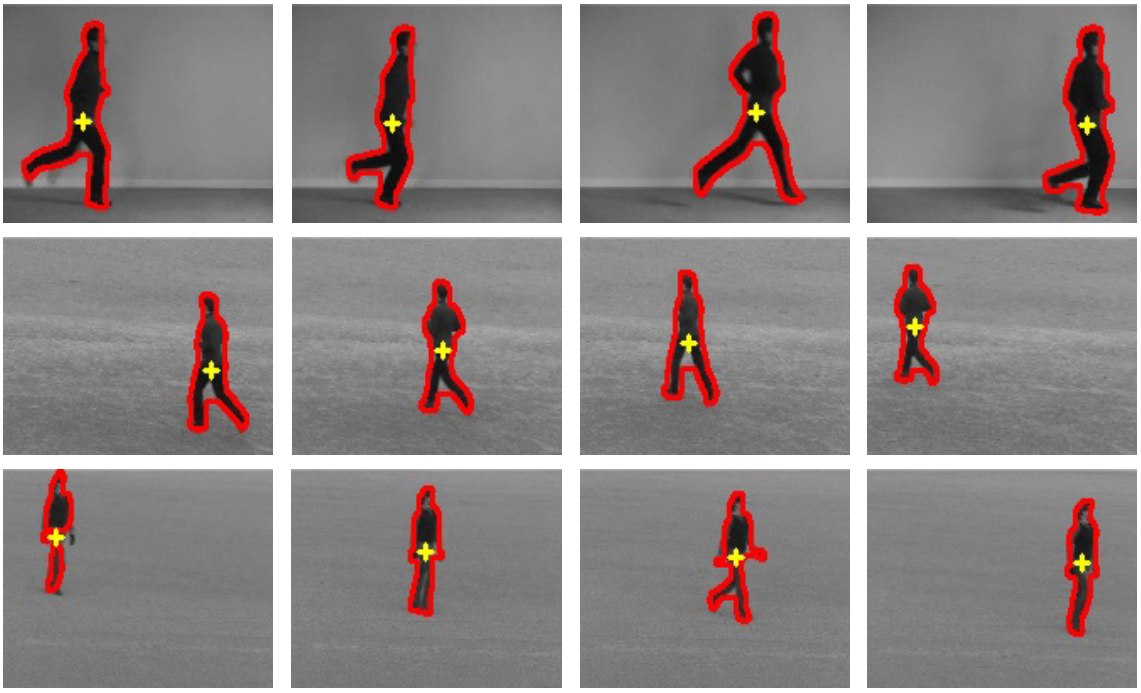


FIG. 6.18. Sample shape border (outlined in red) used in the experiments; the yellow cross within each shape indicates the centroid of the shape border.

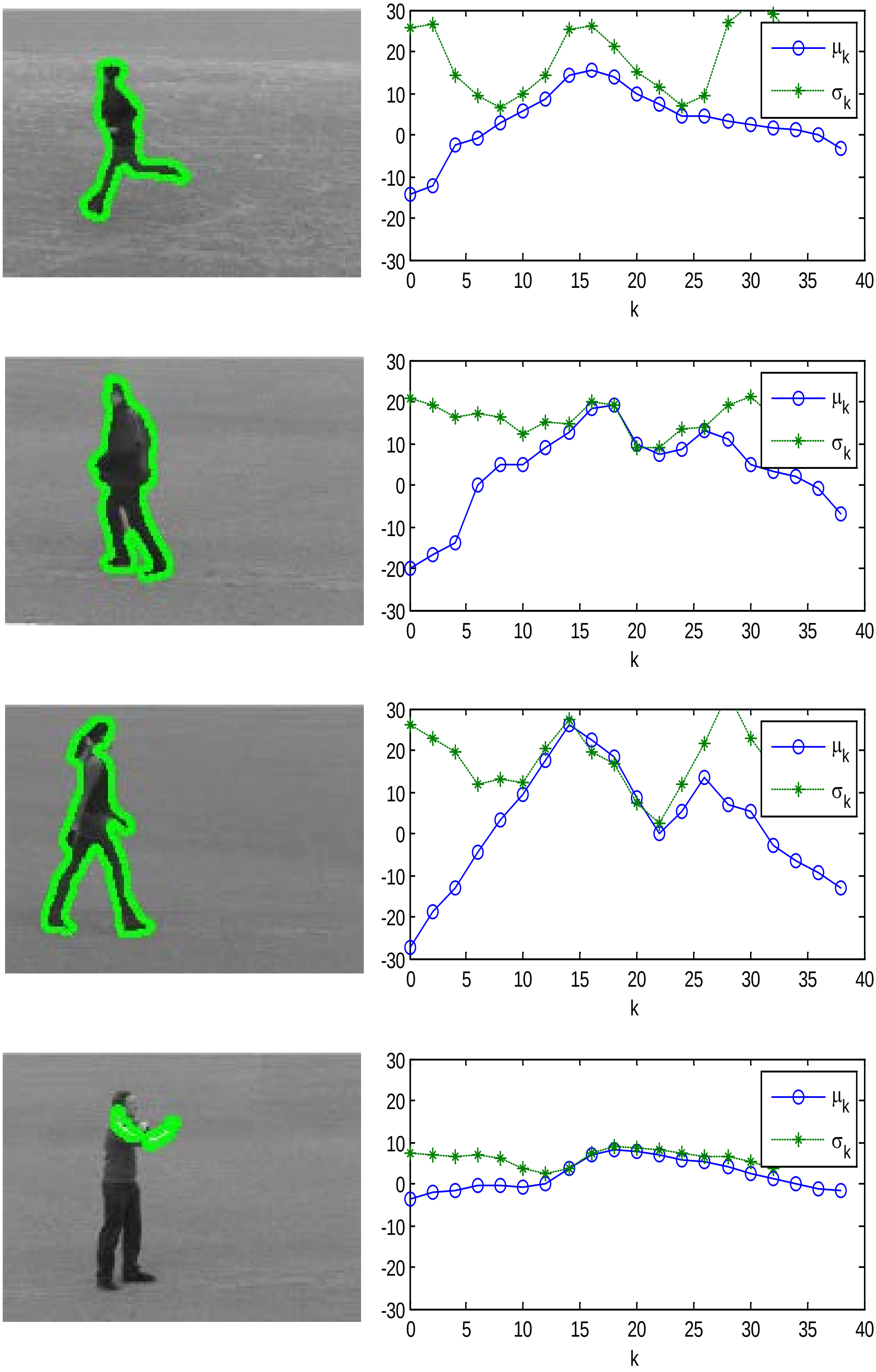
to the integral value of unity. The normalized feature vectors can be exploited as shape descriptors for classification and matching.

6.3.3.4 Adding motion features

Global features of motion have proven to be advantageous in many applications of object recognition. This encourage us to extend the idea and fuse motion features and CLFs features to form the final SVM model. The motion features are based on calculating the center of of gravity (i.e., centroid) that delivers the center of motion and is given by Eq. (6.18). It has experimentally been established that the motion features provide significant information not only about the type of motion, but also about the rate of motion (i.e., velocity). With these features, it would be able to distinguish, for example, between an action in which motion occurs over a relatively large area (e.g., running) and an action localized in a smaller region, where only small parts of the body are in motion (e.g., boxing).

6.3.3.5 Action classification using SVM

In this section, we formulate the action recognition task as a multi-class learning problem, where there is one class for each action, and the goal is to assign an action to an individual in each video sequence. There are various supervised learning



μk
σk
k

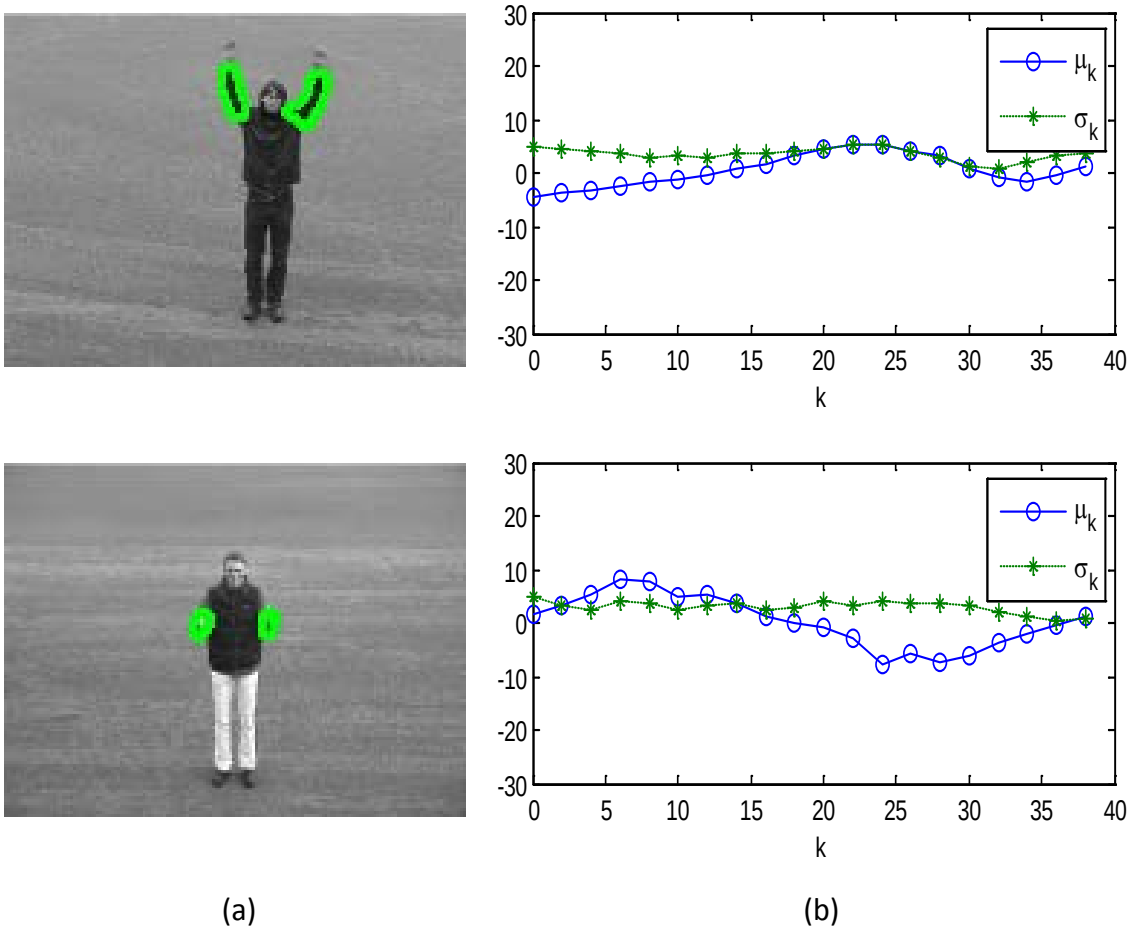


FIG. 6.19. Chord-length functions (CLFs) descriptors: (a) sample video sequences of persons performing different actions; (b) CLFs descriptors obtained for the sequences in (a).

algorithms by which an action recognizer can be trained. Support Vector Machines (SVMs) [137] are used in our framework due to their outstanding generalization capability and reputation of a highly accurate paradigm.

In this approach, six classes of actions are defined. Several one-vs-all SVM classifiers are trained using the features extracted from action snippets in the training dataset. The up diagonal elements of the temporal similarity matrix representing the shape features are first transformed into plain vectors based on the element scan order. The motion feature vectors are then catenated with the shape features vectors to generate the hybrid feature vectors. Finally, the final feature vectors are fed into SVM classifiers for the final decision.

Table 6.7. Confusion matrix of the proposed method

ACTION	walking	running	jogging	boxing	waving	clapping
walking	0.95	0.01	0.04	0.00	0.00	0.00
running	0.00	0.96	0.04	0.00	0.00	0.00
jogging	0.03	0.08	0.89	0.00	0.00	0.00
boxing	0.00	0.00	0.00	0.93	0.03	0.04
waving	0.00	0.00	0.00	0.01	0.96	0.03
clapping	0.00	0.00	0.00	0.00	0.03	0.97

6.3.3.6 Numerical results and comparison with the state-of-art

The proposed method has been evaluated on the KTH dataset. To illustrate the effectiveness of this approach, the obtained results have been compared with those of other similar state-of-the-art methods. In order to prepare the experiments and to provide an unbiased estimation of the generalization abilities of the classification process, the sequences for each action, were divided into two subsets viz. training set (two thirds) and test subset (one third). The training subset is used to train the SVM classifier, while the test subset is used to evaluate the performance and generalization ability of the classifier in the activity recognition task. The confusion matrix showing the recognition results achieved by the proposed method is given in Table 6.7, while the comparison of the obtained results with those obtained by other widely quoted methods in the literature is shown in Table 6.8.

Table 6.8. Comparison with state-of-the-art methods

Method	Recognition rates
Our method [169]	**94.3%**
Liu et al. [65]	92.8%
Wang et al. [162]	92.5%
Jhuang et al. [148]	91.7%
Rodriguez et al. [42]	88.6%
Rapantzikos et al. [64]	88.3%
Dollár et al. [51]	81.2%
Ke et al. [161]	63.0%

As follows from the figures tabulated in Table 6.7, most actions are correctly classified. Most of the mistakes where confusions occur are between ”jogging” and ”running” actions and between ”boxing” and ”clapping” actions. This is intuitively plausible due to the fact of high similarity between each pair of these actions. From the comparison given by Table 6.8, it turns out that our method performs competitively with other state-of-the-art methods and its results compare favorably with previously published results. It may not be irrelevant to mention here that the

state-of-the-art methods with which we compare our method have used the same dataset and the same experimental conditions, therefore the comparison seems to be quite fair. Finally, the experiments were performed utilizing a 2.8 GHz Intel dual core machine with 4 GB of RAM.

6.4 Discussion and Conclusion

In this chapter, two publicly benchmark action recognition datasets (i.e., KTH and Weizmann datasets) have first been described. The first one includes video sequences of 25 subjects performing six different actions under four different scenarios, while the second one consists of a total of 90 video sequences showing nine different persons, each performing 10 actions. These datasets have been used for the evaluation of all the proposed approaches for activity recognition, whose results have been presented in the later sections of this chapter. The obtained results on these datasets have shown that the presented approaches achieve good performances with respect to the state-of-the-art methods. Consequently, it may be concluded that the approaches are likely to be practicable to achieve the goal of this research. In the first approach, a fuzzy framework for representing and recognizing human activities in video sequences has been introduced. Temporal shape variations are accurately captured based on fuzzy log-polar histograms. In addition, a reliable neural model, the MSNN (Multi-level Sigmoidal Neural Network) as a classifier is used for the task of activity recognition. On the KTH and Weizmann action datasets, this approach could retrieve activities with average recognition rates of 94.3% and 97.8% respectively.

As a second approach, we have developed a method for human activity recognition based on multiple cues. As shape features, a variety of shape descriptors both boundary-based (e.g., Fourier descriptors, curvature features, etc.) and region-based (e.g., invariant Moments, Moment-based features, etc.) have been employed. The well-known NB (Naïve Bayes) classifier has been trained automatically in the feature space for activity classification. The simplicity and computational efficiency of the employed features allow this approach to be more amenable for real-time implementation. On the basis of the third approach, a new methodology for human activity recognition has been introduced, based on chord-length shape features. In this work, a compact and computationally efficient shape descriptor; the chord-length shape features is constructed using 1-D chord-length functions. The results obtained with this approach have also compared favorably with the best reported in the literature, while maintaining real-time guarantees.

CHAPTER 7

Towards Recognizing Actions in Real-world Videos

7.1 Introduction

In the action recognition literature, there is a variety of benchmark datasets that have been created for the purpose of experimentation and evaluation. As mentioned in the previous chapter, most of these datasets are chiefly designed to be specialized to some extent and focused upon a specific recognition objective, such as, Tulips1 dataset [155] for visual speech recognition, Georgia-Tech [156] and CMU Mobo [157] datasets for gait recognition, and KTH [7] and Weizmann [2] datasets for action recognition described in detail in the previous chapter.

Recognizing human actions in unconstrained settings is a longstanding and extremely challenging problem in computer vision and many of its related applications, due to a variety of challenging real-world conditions, including partial occlusion, substantial background clutter, drastic illumination variation, large intra-class variability within each class, extreme pose variation, and changes in scale, viewpoint, and appearance. Specifically, this chapter focuses on the recognition of human actions in real-world scenarios which is an important but challenging problem with prosperous applicability into human-computer interactions and security industry. Real-world datasets for the evaluation of human action recognition systems generally consist of a large collection of real-world video streams (or video clips) about the actions of interest. Each video stream includes an individual (i.e. action subject) performing a single action or a series of successive actions. All videos belonging to the same action category can be annotated with a categorical label describing the type of action performed within them.

In this chapter, we are interested in creating our own dataset for the purposes of experimentation and evaluation. It is very important to keep in mind that in this case a direct literature-based comparison of our results reported here with those of other action recognition approaches turns out to be not possible, due to difference in employed action datasets and experimental settings. During the course of the chapter, we will also proceed with our experiments based upon the theories and concepts presented in previous chapters (particularly Chapters 4 and 5).

The rest of the chapter is organized as follows. Section 7.2 introduces the new dataset that we use for this research, and present some interesting characteristics of this dataset. In Section 7.3, a detailed description for our proposed framework for action recognition in real-world streams is provided, and the results of some preliminary experiments conducted to evaluate the stability of the recognition system and its effectiveness in recognizing actions are presented and discussed. Finally, in Section 7.4, we summarize our results and draw conclusions.

7.2 Dataset

After the brief introduction above, it is time now to commence our discussion, in this section, with a description of the action dataset on which the experiments reported in this chapter are conducted. Thereafter, in the forthcoming sections, we present detailed descriptions of how the experiments were carried out and what their results show. To evaluate the performance of the proposed approach for action recognition in real world scenarios, we decided to crate our own realistic action recognition dataset (hereinafter called as IIKT[1] action dataset) which is going to be publicly available free of restrictions on use for action recognition research on the Web very soon. Analogous to the KTH [7] action dataset, a total of six action categories are contained in the IIKT action dataset; three "leg actions" (i.e., walking, jogging, and running) and three "arm actions" (i.e., boxing, hand-waving, and hand-clapping). The video sequences were typically acquired by a Canon IXUS 65 digital camera and stored in a resolution of 640×480 pixels represented in 256 grayscale levels. We believe that this resolution will likely be sufficient to reduce the high impact of the camera artifacts on the recognition results, since the data are internally stored in a lossy MPEG-format by the camera. Contrary to the KTH dataset, the sequences in IIKT dataset were taken over various non-homogeneous backgrounds at 30 fps frame rate. Within the sequences, actions are performed

[1] **IIKT** is an acronym for the German expression: "**I**nstitut für **I**nformations-und **K**ommunikations**t**echnik"; the Institute for Information Technology and Communications at OvG University Magdeburg, Germany and is one of the largest engineering schools in Germany.

by six subjects, each subject was asked to wear a different clothing item. This is expected to make recognizing actions slightly more challenging. Each action sequence was then segmented into shorter video clips of 53sec duration which we termed 'action snippets'. Fig. 7.1 shows example frames from action sequences of different categories represented in the IIKT dataset.

7.3 Action Recognition via Fuzzy Directional Features

In this section, we present a new approach for action recognition, based on a modified fuzzy version of HOF (Histogram of Optical Flow), so-called fuzzy histogram of optical flow as a new motion descriptor to model action in a realistic scene as a time-series of fuzzy directional features. A set of one-vs.-all SVM classifiers are trained on these features for the action classification. This approach was evaluated on our dataset which incorporates a collection of real-world video data.

7.3.1 Motion estimation

To detect moving objects (i.e., action subjects), we use an algorithm that works based on the same principles as the two-frame motion estimation algorithm presented by Farnebäck in [170] that computes the optical flow based on polynomial expansion. The key idea of the algorithm is to approximate a neighborhood of each pixel in a frame by a quadratic polynomial:

$$f(\mathbf{x}) \sim p(\mathbf{x}) = \mathbf{x}^\top \mathbf{A}\mathbf{x} + \mathbf{b}^\top \mathbf{x} + c \tag{7.1}$$

where $\mathbf{A}, \mathbf{b}$, and c are the expansion coefficients that are determined using a Gaussian-weighted least-squares fitting of the signal f by the polynomial p. The new frame can be thus constructed from the previous one by a global translation $\mathbf{d}$:

$$\begin{aligned} \tilde{f}(\mathbf{x}) &\sim p(\mathbf{x}-\mathbf{d}) \\ &= (\mathbf{x}-\mathbf{d})^\top \mathbf{A}(\mathbf{x}-\mathbf{d}) + \mathbf{b}^\top(\mathbf{x}-\mathbf{d}) + c \\ &= \mathbf{x}^\top \tilde{\mathbf{A}}\mathbf{x} + \tilde{\mathbf{b}}^\top \mathbf{x} + \tilde{c} \end{aligned} \tag{7.2}$$

It is easy to see that these two sets of expansion coefficients are related by

$$\begin{aligned} \tilde{\mathbf{A}} &= \mathbf{A}, \\ \tilde{\mathbf{b}} &= \mathbf{b} - 2\mathbf{A}\mathbf{d}, \\ \tilde{c} &= c + \mathbf{d}^\top \mathbf{A}\mathbf{d} - \mathbf{b}^\top \mathbf{d}. \end{aligned} \tag{7.3}$$

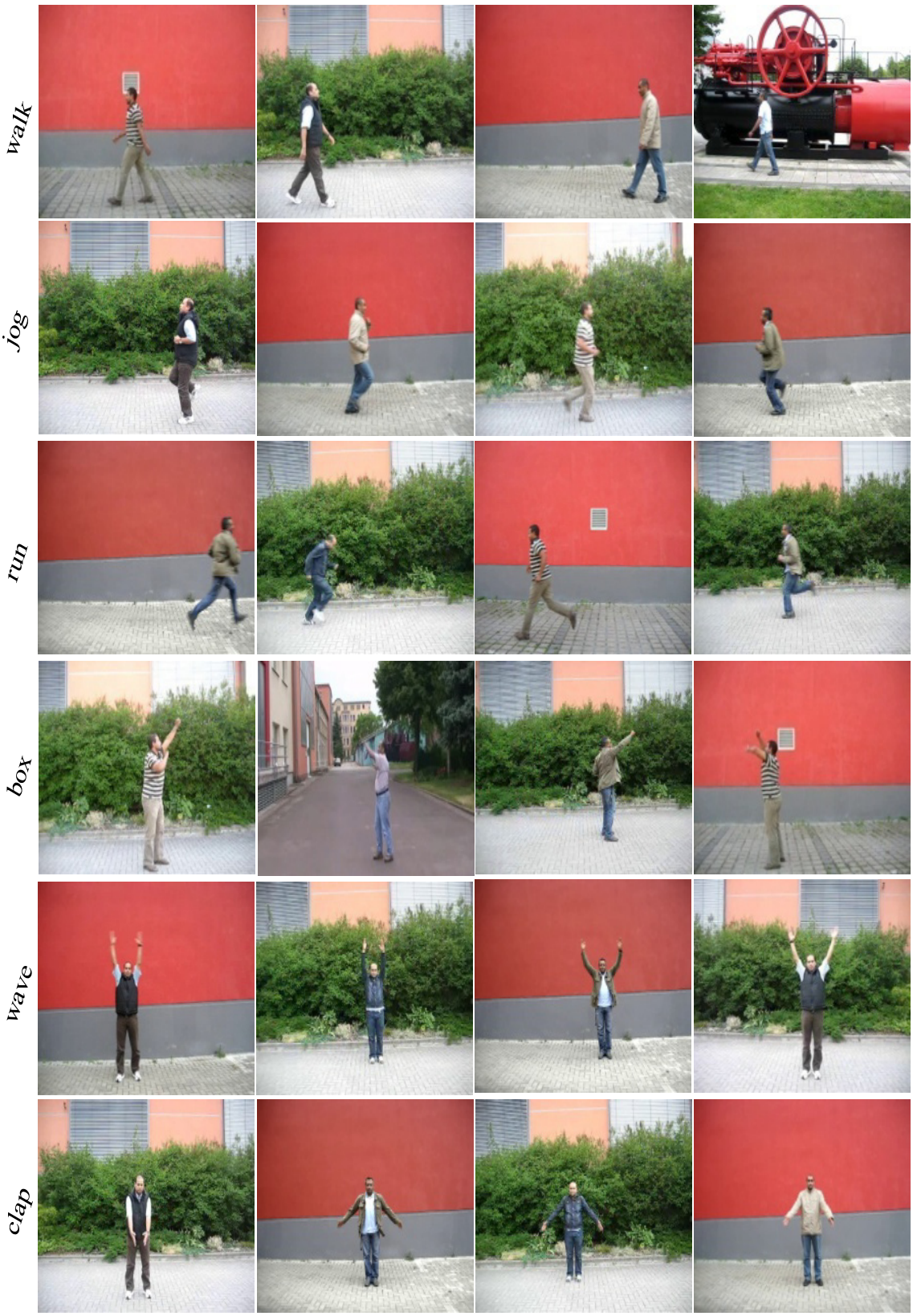


FIG. 7.1. Sample frames from the action sequences in IIKT action dataset.

Looking at Eq. (7.3), one realizes that a solution for the translation $\mathbf{d}$ exists only if

$$\mathbf{d} = \frac{1}{2}\mathbf{A}^{-1}(\tilde{\mathbf{b}} - \mathbf{b}) \tag{7.4}$$

For practical considerations, the global polynomial in Eq. (7.4) are replaced with local polynomial approximations. Thus, giving two sets of expansion coefficients $\{\mathbf{A}_1(\mathbf{x}), \mathbf{b}_1(\mathbf{x}), c_1(\mathbf{x})\}$ and $\{\mathbf{A}_2(\mathbf{x}), \mathbf{b}_2(\mathbf{x}), c_2(\mathbf{x})\}$ for the first and second image frames respectively, it is possible to do a polynomial expansion of both frames. Ideally, this yields $\mathbf{A}_1 = \mathbf{A}_2$, however, in practice one is forced to settle for the approximation:

$$\mathbf{A}(\mathbf{x}) = \frac{\mathbf{A}_1(\mathbf{x}) + \mathbf{A}_2(\mathbf{x})}{2} \tag{7.5}$$

and further the following assumption

$$\Delta\mathbf{b}(\mathbf{x}) = -\frac{1}{2}(\mathbf{b}_2(\mathbf{x}) + \mathbf{b}_1(\mathbf{x})) \tag{7.6}$$

is made, which leads to the primary constraint

$$\mathbf{A}(\mathbf{x})\mathbf{d}(\mathbf{x}) = \Delta\mathbf{b}(\mathbf{x}) \tag{7.7}$$

where $\mathbf{d}(\mathbf{x})$ implies that the global displacement in Eq. (7.2) is replaced with a spatially varying displacement field. Under the assumption that the displacement field is only slowly varying, information over a neighborhood Ω of each pixel can be integrated. Consequently, $\mathbf{d}(\mathbf{x})$ satisfying Eq. (7.7) and minimizing

$$\sum_{\mathbf{\Delta x}\in\Omega} w(\mathbf{\Delta x})\|\mathbf{A}(\mathbf{x}+\mathbf{\Delta x})\mathbf{d}(\mathbf{x}) - \Delta\mathbf{b}(\mathbf{x}+\mathbf{\Delta x})\|^2 \tag{7.8}$$

can be found, where $w(\mathbf{\Delta x})$ is a Gaussian weight function. Therefore, the minimum value is given by

$$e(\mathbf{x}) = \left(\sum w\mathbf{\Delta b}^\top\mathbf{\Delta b}\right) - \mathbf{d}(\mathbf{x})^\top \sum w\mathbf{\Delta A}^\top\mathbf{\Delta b}, \tag{7.9}$$

which is obtained for

$$\mathbf{d}(\mathbf{x}) = \left(\sum w\mathbf{\Delta A}^\top\mathbf{\Delta A}\right)^{-1} \sum w\mathbf{A}^\top\mathbf{\Delta b} \tag{7.10}$$

It was shown, in [170], that in many cases it might be advantageous to introduce a certainty weight $\mathbf{c}(\mathbf{x}+\mathbf{\Delta x})$ to equation (7.8) that can be most conveniently achieved by scaling $\mathbf{A}$ and $\mathbf{\Delta b}$. Now, to detect moving objects, particularly people (i.e., action subjects), the displacement field should be parameterized according to some motion model (e.g., affine motion model or eight-parameter model). For the eight-parameter model in 2D, the motion field can be expressed as,

$$\mathbf{d} = \mathbf{Sp} \tag{7.11}$$

where,

$$\begin{aligned}\mathbf{S} &= \begin{pmatrix} 1 & x & y & 0 & 0 & 0 & x^2 & xy \\ 0 & 0 & 0 & 1 & x & y & xy & y^2 \end{pmatrix}, \\ \mathbf{p} &= \begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \end{pmatrix}^\top \end{aligned} \tag{7.12}$$

Substituting from Eq. (7.11) into Eq. (7.8) yields the weighted least squares problem:

$$\sum_i w_i \|\mathbf{A}_i \mathbf{S}_i - \boldsymbol{\Delta}\mathbf{b}_i\|^2 \tag{7.13}$$

which in turn has the solution

$$\mathbf{p} = \left(\sum_i w_i \mathbf{S}_i^\top \mathbf{A}_i^\top \mathbf{A}_i \mathbf{S}_i \right)^{-1} \sum_i w_i \mathbf{S}_i^\top \mathbf{A}_i^\top \boldsymbol{\Delta}\mathbf{b}_i \tag{7.14}$$

The actual solution involves the accumulation of the coefficients of the 8×8 system of equations (7.14) over all points and then solving for the parameters. To improve the chances for a better displacement estimate in the algorithm, it is crucial to exploit some a priori knowledge about the displacement field that allow comparing the polynomial at $\mathbf{x}$ in the first signal to the polynomial at $\mathbf{x} + \tilde{\mathbf{d}}(\mathbf{x})$ in the second signal, where $\tilde{\mathbf{d}}(\mathbf{x})$ is the a priori displacement field. In this case, $\mathbf{A}(\mathbf{x})$ and $\Delta\mathbf{b}(\mathbf{x})$ introduced in Eq. (7.5) and Eq. (7.6) are substituted by

$$\mathbf{A}(\mathbf{x}) = \frac{\mathbf{A}_1(\mathbf{x}) + \mathbf{A}_2(\tilde{\mathbf{x}})}{2} \tag{7.15}$$

$$\Delta\mathbf{b}(\mathbf{x}) = -\frac{1}{2}(\mathbf{b}_2(\tilde{\mathbf{x}}) + \mathbf{b}_1(\mathbf{x})) + \mathbf{A}(\mathbf{x})\tilde{\mathbf{d}}(\mathbf{x}) \tag{7.16}$$

where $\tilde{\mathbf{x}} = \mathbf{x} + \tilde{\mathbf{d}}(\mathbf{x})$.

7.3.2 Optical flow pruning

It has to be admitted that despite over two decades of intensive research, most existing methods for the extraction of optical flow still lack robustness, and optical flow estimates are relatively inaccurate, particularly with respect to flow magnitude. This might be attributed to the large residual error in solving the equations for optical flow. Therefore, pruning of computed flow values appears to be a clue to accurate flow fields which in turn allows for better motion estimation. To tackle this problem, we introduce a particular kind of filter that straightens up noisy vectors in the flow field, while maintaining significant ones.

In our work, we perform this type of pruning stepwise. In other words, it involves two passes, each based on the magnitude (Euclidean length) of optical

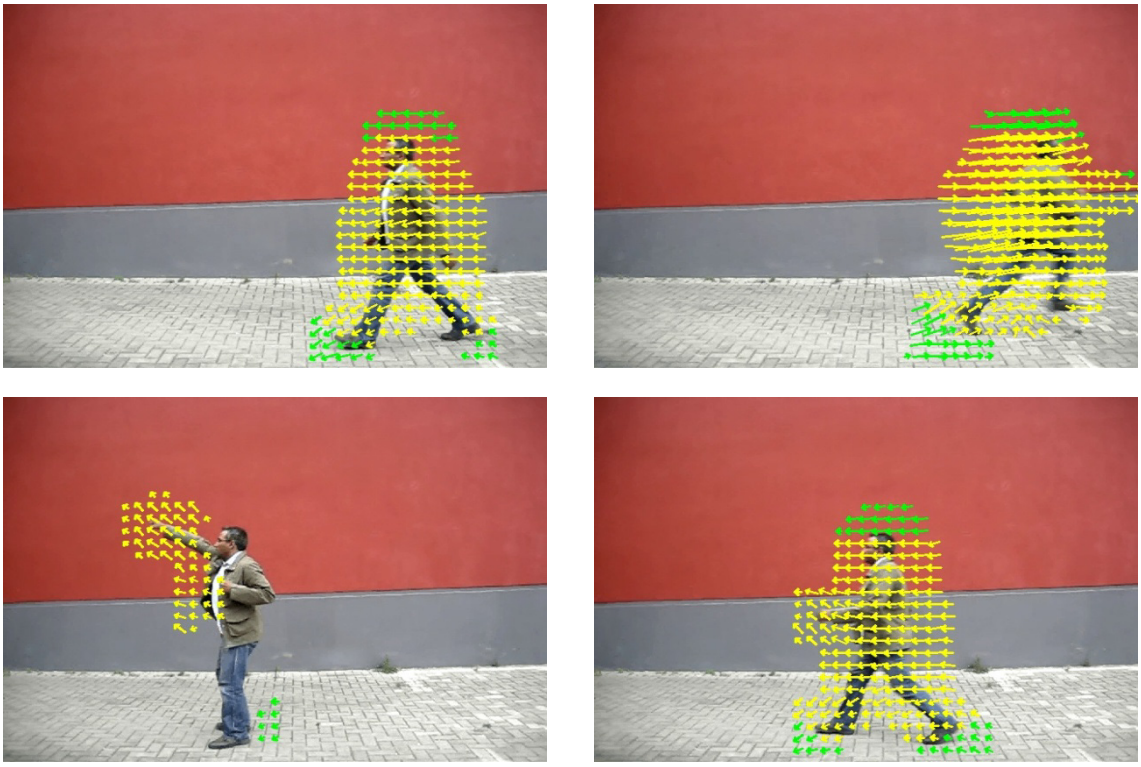


FIG. 7.2. Sample pruning results for a setup with $\lambda = 0.25\ell$; the vectors labeled in yellow are accepted as valid flow components, while the vectors labeled in green are considered as noisy flow components and thus filtered out.

flow vectors to separate relevant from irrelevant flow vectors. In the first pass, we attempt to remove all flow vectors whose magnitudes are either relatively very small or very large. For this purpose, two predefined thresholds (i.e., minimum and maximum thresholds) are used that control the filtering of flow vectors in this step. Formally speaking, given two thresholds ρ_1 and ρ_2, a flow vector $\vec{v} = [x, y]^\top$ is only accepted as valid if it satisfies the validity constraint: $\rho_1 < \|\vec{v}\| < \rho_2$, where $\|\cdot\|$ denotes the magnitude of the flow vector with respect to the Euclidean metric; otherwise it is assumed to be a noisy flow component and thus removed. In our experiments, when ρ_1 and ρ_2 are given 5 and 20 respectively, satisfactory results can be achieved. We go then with a second pass of our pruning based on the Euclidean distance between the centroid (center of mass) of flow field and the flow points. Therefore, in this pass of pruning, a vector $\vec{v}$ is treated as a valid flow component if the Euclidean distance between the center of flow and the vector being analyzed does not exceed a specific threshold λ. Formally, this is expressed as:

$$\|\vec{v} - \vec{c}\| < \lambda \tag{7.17}$$

where $\vec{c}$ is motion regions's centroid. From our experiments, we see that setting the

value of λ at 25% of the average of image width and height, $\ell = (w + h)/2$ gives an overall good pruning performance (see Fig. 7.2 for visual examples).

7.3.3 Directional feature extraction

As demonstrated in the literature review in Chapter 2, several existing theoretical approaches to action recognition tend to put much more emphasis on providing practical methods which are consistently applicable only to various joint angles acquired from motion capture data. However, when applying these approaches to video data, we are regularly faced with the complex problem of segmenting and tracking of human joints. This problem is considerably more challenging and error-prone, particularly in dynamically complex environments where the tracking objects frequently undergo large changes in pose, scale, and lighting conditions.

Motivated by the potential benefits in performance of histogram of features (e.g. HOG [58]) for object recognition, in this work, we propose to compute a new motion-related descriptor based on optical flow analysis. However, most optical flow computations turn out to be most sensitive to background noise, and changes in scale and/or directionality of motion. Furthermore, the number of moving pixels is subject to change with time. Due to these restrictions, raw values of optical flow would likely be less suitable or unsuitable as features for motion analysis. In order to overcome these difficulties, we can here use the characteristics of distribution of optical flow as features to describe motion. As a matter of fact, one can see that the motion activity of an individual moving in a scene with a static background can be characterized fully by its own self-induced optical flow profile. In Fig. 7.3, sample optical flow patterns for a sequence showing a person performing actions of walking, jogging, running, boxing, waving, and clapping are shown.

The main thrust of our work is to develop a new descriptor based on improved optical flow measurements over a spatio-temporal volume centered on a human figure to represent actions. An SVM classifier is trained on these descriptors to classify actions. To generate a robust and discriminative motion descriptor invariant to pose variation and directionality of motion, two aspects should be kept in mind, one referring to the dependency of the observed flow profile on the scale of motion activity, the other relating to the dependency of the orientation of optical flow on the directionality of motion. Moving from these considerations and requirements, we propose here the FHOF (Fuzzy Histogram of Optical Flow). A formal definition and implementation scheme of this descriptor are as follows. Given an estimate for optical flow field at each frame, the magnitude and the orientation of each flow

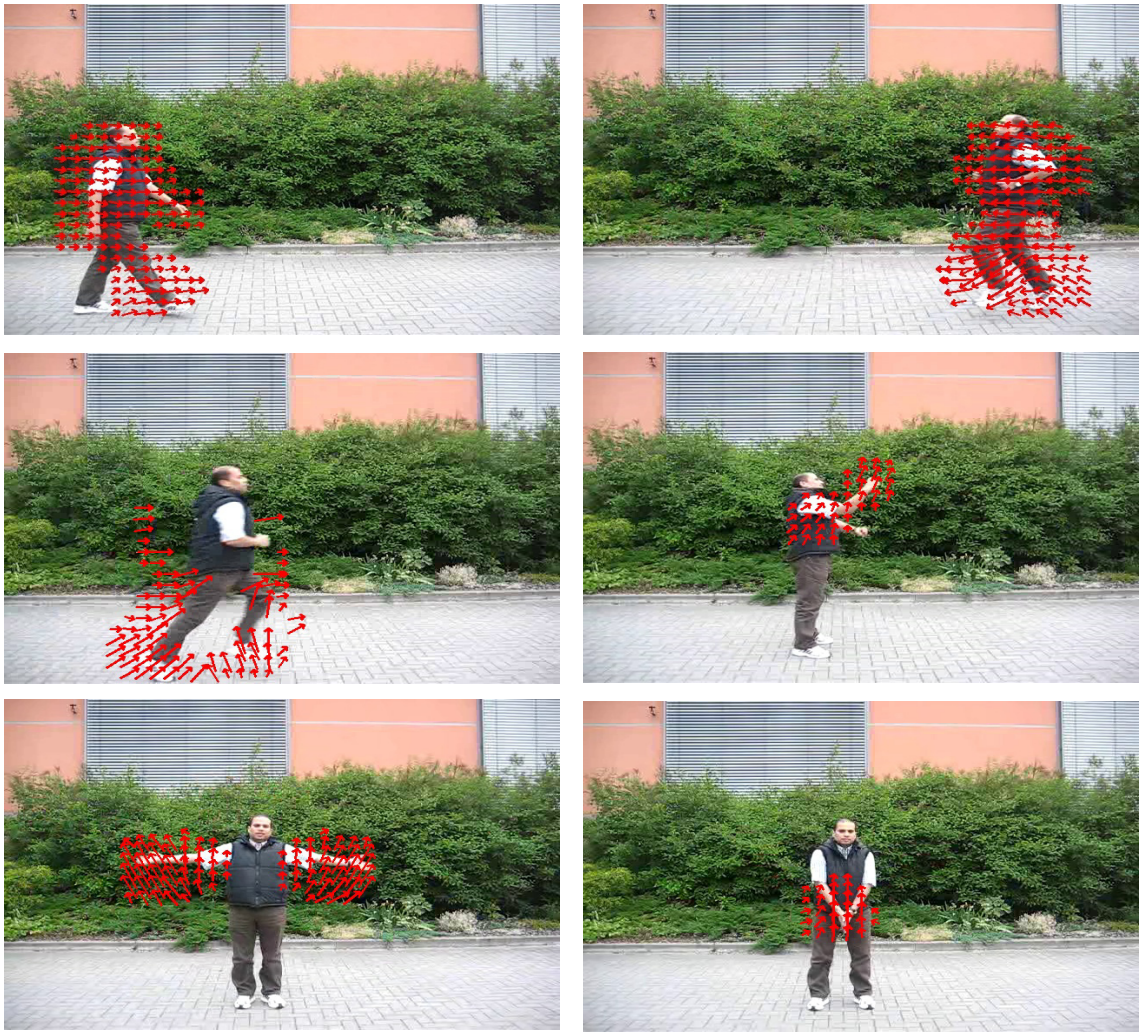


FIG. 7.3. Optical flow estimation results for a real-world video sequence showing a single person performing various actions, i.e. walking, jogging, running, boxing, waving, and clapping from left to right and top to bottom, respectively.

vector $\vec{v} = [x, y]^\top$ are specially defined[2] as follows,

$$\begin{aligned} \rho &= \sqrt{x^2 + y^2} \\ \varphi &= \text{atan2}(y, |x|) \end{aligned} \tag{7.18}$$

where $|\cdot|$ denotes the ordinary absolute value, and $\frac{-\pi}{2} < \varphi \leq \frac{\pi}{2}$ that gives the smallest angle between the x-axis and $\vec{v}$ axis, as shown in Fig. 7.4. It should be noted that the orientation angle φ in Eq. (7.18) has been defined so as to allow our histogram representation to be independent of the directionality of movement.

[2]The two-argument function atan2 is a variation of the arctangent function, which is defined as $\text{atan2}(y, x) = \arctan((\sqrt{x^2 + y^2} - x)/y)$.

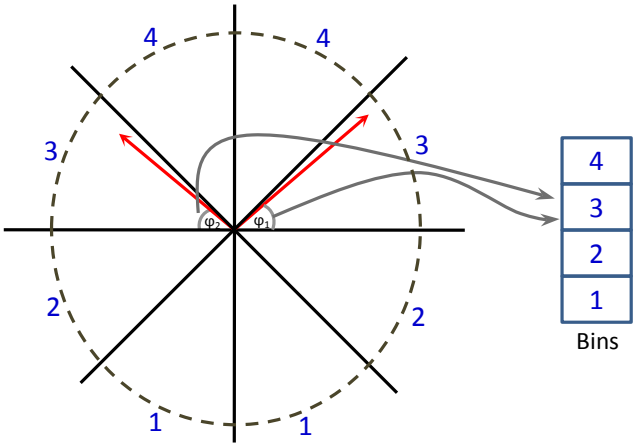


FIG. 7.4. An example for orientation histogram with four bins ($K = 4$).

Now a histogram at each frame can be built by binning the flow vectors into a fixed number of bins based on their primary angles and magnitudes. Formally, the histogram is created where each flow vector $\vec{v}$ with direction φ in the range:

$$-\frac{\pi}{2} + \pi\frac{k-1}{K} \leq \varphi < -\frac{\pi}{2} + \pi\frac{k}{K} \tag{7.19}$$

gives a contribution proportional to ρ to its corresponding bin k, $1 \leq k \leq K$ where K is the number of bins. As seen in Fig. 7.4, the resulting histogram representation is invariant to direction of motion. To achieve invariance to scale changes, the histogram is normalized by the overall magnitude of flow vectors, so that the bins integrate to unity. Moreover, as flow vectors contribute to the histogram proportionally to their magnitudes, the resulting descriptor would be more robust to noisy optical flow measurements. An example of visualization of our descriptor for the applied features is given in Fig. 7.5. From a close inspective look at the plots in the figure, one can see that there is a remarkable similarity in feature structure (leading to similar color values in the Figure) among sequences of walk, jog, and run actions, and between sequences of wave and clap actions. Intuitively, this is due to the fact of high closeness of similar types of actions.

7.3.4 Fuzzy feature selection

In this section, we describe our method for feature selection based on temporally adaptive decomposition of action sequences into a finite number of time slices in a fuzzy way, which is targeted at the removal of irrelevance and redundancy in the features set, so that not only does the reduced set of features speed up the action classification process by removing class irrelevant features, but it also provides at

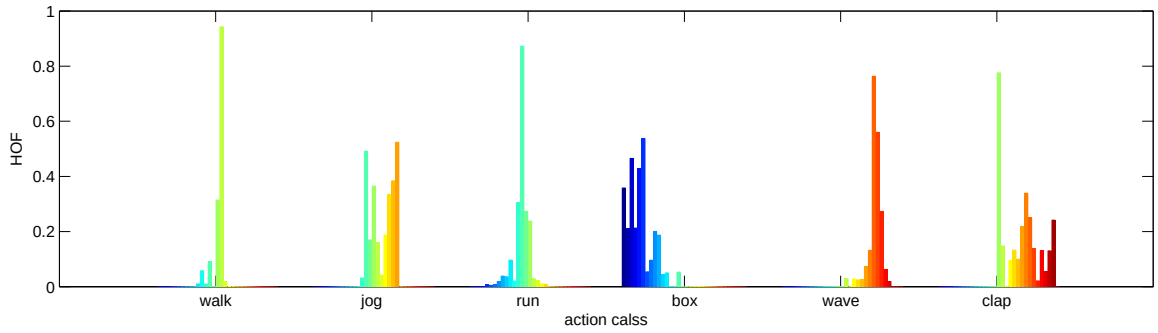


FIG. 7.5. Visualization of the proposed descriptor (with $K = 32$) for HOF features extracted from sample sequences of walk, jog, run, box, wave, and clap actions.

least the same quality of action classification as the original one. Eventually, this enables the proposed approach to achieve better feature reduction ratios without losses in recognition accuracy. As discussed in the previous section, a normalized histogram based on HOG features can be constructed at each instance of time t:

$$\mathbf{h}_t = (h_{t;1}, h_{t;2}, \ldots, h_{t;K})^\top \tag{7.20}$$

where K (the number of histogram bins) is a parameter of choice, which has a direct influence on the eventual performance of the recognition system. Since the flow features in Eq. (7.20) can be computed at each instant time of a given sequence (i.e., action snippet), the action snippet can be represented as a time series of these features: $A = \{\mathbf{h_t}\}_{t=0}^{\tau-1}$ which provides us an attractive and rigorous approach to classify and recognize actions. To obtain the final feature vector for each action snippet, each action snippet is split into several time-slices defined by linguistic intervals [123]. A Gaussian fuzzy membership function is used to describe each of these intervals. The general forms of these membership functions is given by

$$\mathcal{G}_j(t; \alpha, \beta, \gamma) = e^{-\left|\frac{t-\alpha}{\beta}\right|^\gamma} \tag{7.21}$$

where $\alpha, \beta,$ andγ are three scalar parameters of the fuzzy function; i.e., the center, width, and the fuzzification factor which is a weighting exponent on each fuzzy membership, respectively. Therefore, a feature vector for a time-slice can be generated by calculating the weighted average feature vector of all frames within the time-slice. More formally, the directional feature vector for time-slice j is given by,

$$\mathcal{H}_j = \frac{1}{\Delta t} \sum_{t \in slice_j} \mathcal{G}_j(t)\mathbf{h}_t,\ j = 1, 2, \ldots, m \tag{7.22}$$

where $\mathcal{G}_j(t)$ is the Gaussian membership function representing the j-th time slice, Δt is the duration of the time slice in frames, and m is the total number of time slices

into which the action snippet is divided. Accordingly, the full feature vector for an action snippet can be straightforwardly derived by concatenating all m feature vectors of its time slices as follows,

$$\mathcal{A} = \mathcal{H}_1 \oplus \mathcal{H}_2 \oplus \cdots \oplus \mathcal{H}_m \tag{7.23}$$

where $\oplus$ is the concatenation operator. From the above mentioned, it follows that the process of slicing action snippets into a finite number of temporal steps would achieve the primary goal of effective feature dimensionality reduction and de-correlation by removing probable redundancy in the features set, while retaining the information essential for effective recognition of actions. For this purpose, each action sequence is treated as a time series composed of low-dimensional feature vectors corresponding to decomposition of the sequence into several time slices. More specifically, we keep only m multidimensional feature vectors corresponding to the m time slices, instead of taking all the feature vectors of all the frames in the action sequence. These m vectors form the feature space for action representation and classification.

It bears mentioning that m is a parameter of choice, where $m \ll n, n$ is the number of frames in the action sequence. To investigate whether and how the overall recognition results are affected by different values for m, in our experiments, different values of the parameter m were tried, each lies in the range of 1 to 5. The value that generates the highest average recognition accuracy over all runs would be selected. As a final note here it should also to be mentioned that the directional features are efficiently computed using fuzzy histograms that enables real-time implementation of the proposed action recognition method.

7.3.5 SVM based action classification

In this section, our goal is to classify actions according to the fuzzy descriptors mentioned previously. Human action recognition can be modeled as a multi-dimensional classification problem having one class for each action, and the goal is to assign a class label to a given action. For this purpose, we use one-vs.-rest SVMs (Support Vector Machines) with RBF (Radial Basis Function) kernels. For SVMs, the one-vs.-rest approach is widely adopted for handling the multi-class problem by constructing the decision rule based on multiple binary classification tasks.

Generally speaking, there are various supervised learning algorithms by which an action recognizer can be trained to recognize patterns of motion over time. In this work, we propose to employ SVMs in our framework due to their outstanding generalization capability and reputation of a highly accurate paradigm. SVMs [137]

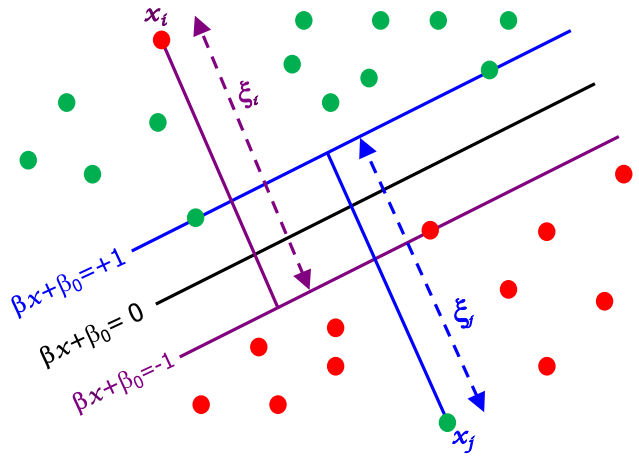


FIG. 7.6. Generalized optimal separating hyperplane.

are based on the Structure Risk Minimization principle from computational theory, and are a solution to data overfitting in neural networks. Originally, SVMs were designed to handle dichotomic classes in a higher dimensional space where a maximal separating hyperplane is created. On each side of this hyperplane, two parallel hyperplanes are conducted. Then SVM attempts to find the separating hyperplane that maximizes the distance between the two parallel hyperplanes (see Fig. 7.6). Intuitively, a good separation is achieved by the hyperplane having the largest distance. Hence the larger the margin the lower the generalization error of the classifier. More formally, let $\mathcal{D} = \{(\mathbf{x}_i, y_i) \,|\, \mathbf{x}_i \in \mathbb{R}^d, y_i \in \{-1, +1\}\}$ be a training set, Coretes and Vapnik [137] have argued that this problem is best approached by allowing some examples to violate the margin constraints . These potential violations can be formulated using some positive slack variables ξ_i and a penalty parameter $C \geq 0$ that penalize the margin violations. Thus the optimal separating hyperplane is determined by solving the following QP problem:

$$\min_{\boldsymbol{\beta},\beta_0} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_i \xi_i \tag{7.24}$$

$$\text{subject to} \quad (y_i(\langle \mathbf{x}_i, \boldsymbol{\beta}\rangle + \beta_0) \geq 1 - \xi_i \quad \forall i) \wedge (\xi_i \geq 0 \quad \forall i).$$

Geometrically, $\boldsymbol{\beta} \in \mathbb{R}^d$ is a vector going through the origin point and perpendicular to the separating hyperplane. The offset parameter β_0 is added to allow the margin to increase, and to not force the hyperplane to pass through the origin that restricts the solution. For computational purposes it is more convenient to solve SVM in its dual formulation. This can be accomplished by forming the Lagrangian and then optimizing over the Lagrange multiplier $\boldsymbol{\alpha}$. The resulting decision function has weight vector $\boldsymbol{\beta} = \sum_i \alpha_i \mathbf{x}_i y_i,\ 0 \leq \alpha_i \leq C$. The instances $\mathbf{x}_i$ with $\alpha_i > 0$ are termed *support vectors*, as they uniquely define the maximum margin hyperplane.

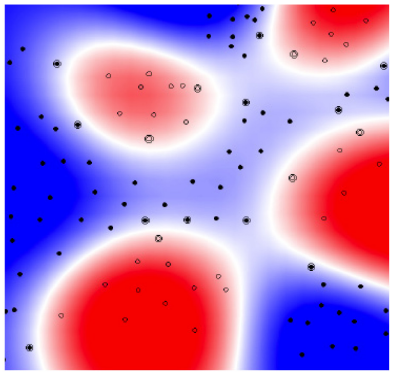


FIG. 7.7. An example for nonlinear RBF kernel[3].

For the proposed approach, several classes of actions are defined and hence several one-vs.-all SVM classifiers are trained on the fuzzy directional features extracted from the action sequences in the training dataset. The feature vectors of the training set are fed into SVM classifiers in order to learn the differences among the features of each action class. In this work, we used one of the most popular and successful kernels, the RBF (or exponential) kernel, defined as

$$\kappa(\mathbf{x}, \mathbf{y}) = \exp(\|\mathbf{x} - \mathbf{y}\|^2/(2\sigma^2)) \tag{7.25}$$

where σ is the kernel width, which can be regarded as a tuning parameter. It is noteworthy to mention here that the SVMs with RBF have evolved as a flexible and powerful tool which is potentially able to create models that handle non-linearly separable data by mapping original features of the training data to a higher dimensional feature space to enable linear separation for classification. In this higher dimensional space, linear functions (or separators) can be constructed, which is potentially able to produce non-linear boundaries (see Fig. 7.7 above) when mapped back to the original input feature space. Another important point to underscore here is that, for RBF kernel, there is a set of parameters (e.g, c and γ) for which several tests were carried out in order to establish their optimum values.

7.3.6 Experiments and discussion

In this section, we present the experimental results and discussion of the proposed framework for human action recognition in real-world video sequences. The reported results here are based on our feature extraction technique described in detail in Section 7.3.3 and 7.3.4 (i.e., fuzzy HOF-based features) and obtained with the IIKT action recognition dataset introduced at the beginning of this chapter that

[3]The plot is generated by Bell SVM applet.

we created for the purpose of recognizing human actions in realistic scenarios. In this study, first of all, the experiments have been conducted to gauge the potential recognition capabilities of the proposed recognition system. The chapter refers also to the results of a series of experiments performed to quantify the effect on recognition performance of altering the feature description parameters (i.e., K and m) in order to establish the optimum recognition rate.

As there was no control over the video capturing process, the action sequences used in our experiments exhibit some degree of variation in the actors, scale, pose, camera views, appearance inside the same action category, coupled with cluttered background and different illumination conditions. Considering that most previous research experiments were conducted in controlled or partially controlled environments (e.g., KTH and Weizmann datasets), we intuitively expect that the experimental results using this dataset will be more realistic. As mentioned previously, this action dataset contains a total of six categories of interest to be recognized, namely walking, jogging, running, boxing, hand waving and hand clapping, performed several times by nine subjects. The test data used in experiments consists of a total of 300 action snippets derived from the video sequences recorded in the dataset. These streams were saved in AVI format with a resolution of 640×480-pixel frame dimensions with 24-bit color depth at 30 fps frame rate. An additional total of 480 action streams are utilized to train the six-action SVM model.

A series of experiments with different feature description parameters (K and m) was run to assess the effectiveness of the proposed technique for action recognition in realistic settings. We extracted about 360 directional features (for the case $K = 18$) from each action video, and then applied our fuzzy approach for feature selection described in Section 7.3.4 to reduce the dimension of the fuzzy feature descriptor to 90. Fig. 7.8 shows an example of visualization of the proposed fuzzy descriptor for the directional features extracted from different action categories. By inspecting the figure, one can observe that the descriptor reflects the actual similarity/dissimilarity between different categories of actions at each temporal step. Thus, to quantify the degree of similarity or dissimilarity between two actions, a measure of similarity can be reliably computed based on a distance (e.g. Euclidean distance) between these descriptors. One more interesting observation is that the descriptor remains constant or slightly changes with time; this suggests that a relatively few number of time slices will suffice to construct such a descriptor. With the eventual goal of developing a high performance action recognition system, we investigate the recognition performance of the proposed recognition framework under the values of the feature description parameters (K and m) varying. Towards this goal, we compute such descriptors a total of 20 times for all samples in training set (i.e., the

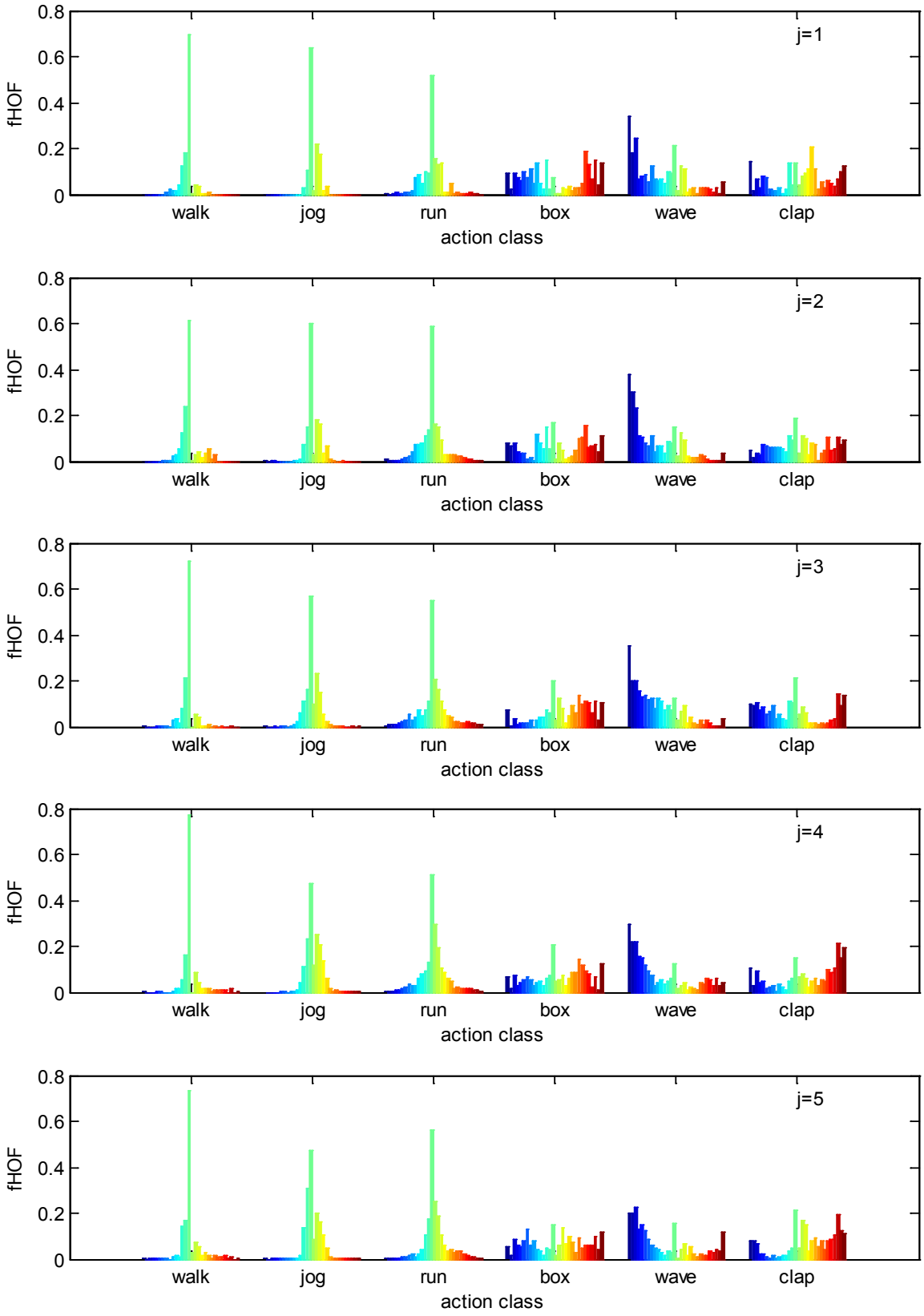


FIG. 7.8. An example of visualization of the proposed descriptor for directional features extracted from different action categories at five temporal steps $m = 5$.

number of all possible combinations of values of the parameters m and K, where $m \in \{1, 2, 3, 4, 5\}$ and $K \in \{4, 8, 12, 18\}$). Therefore, m fuzzy membership functions should be defined to represent different time slices of a given action sequence, as shown in Fig. 7.9. Note that for the sake of visualization, each fuzzy membership function in the Figure is plotted in a unique color.

To evaluate qualitatively and quantitatively the system's performance, the experiments were performed for all possible combinations of values of the feature parameters. To facilitate the visualization of the system's performance, the confusion matrices that tabulate the correct and incorrect classifications are calculated through majority voting. The classification performance of the system for test dataset can be presented directly in the form of confusion tables. Instead, for the sake of clarity, we graphically represent these confusion tables through a series of 3D bar plots (see Fig. 7.10). In this figure, we see a series of 3D plots that visualizes the confusion in recognition results for each action category, each corresponding to a combination of feature representation parameters. By inspecting all plots shown in the figure, it is explicitly observed, as expected, that the feature representation parameters K and m are both significant and directly affect the recognition results.

Furthermore, the overall accuracy (or correct recognition rate) metric is employed to gauge the holistic performance of the proposed recognition scheme. The dependency of the overall recognition rate on the feature parameters has a shape similar to shown in Fig. 7.11. Having a closer look at the figure, one can see that in terms of recognition rate, the larger values of both parameters provide the greatest improvement in performance, and generally are the most important. In other words, the larger the values of feature parameters are, the better the holistic performance is. For the sake of brevity, as a final remark in this section, we only mention that in our computational experiments, all the routines considered in this study were coded in Visual Studio 2008 and executed on a PC equipped with an Intel Core 2 processor operating at 2.8 GHz with 8 MB of cache and 4 GB of SDRAM.

Action Localization:

In this subsection, we describes the results of a final simple experiment conducted with the purpose of localizing the moving objects as motion regions of interest (ROI) identified by motion information. The analysis of the spatial location distribution of the flow features generated by our proposed fuzzy framework can efficiently contribute to a fast and accurate estimation of the 2D position of the centroid of these features based on the average of the coordinates of all feature points in motion ROI. More formally, the centroid of an action, at each time instant, is calculated

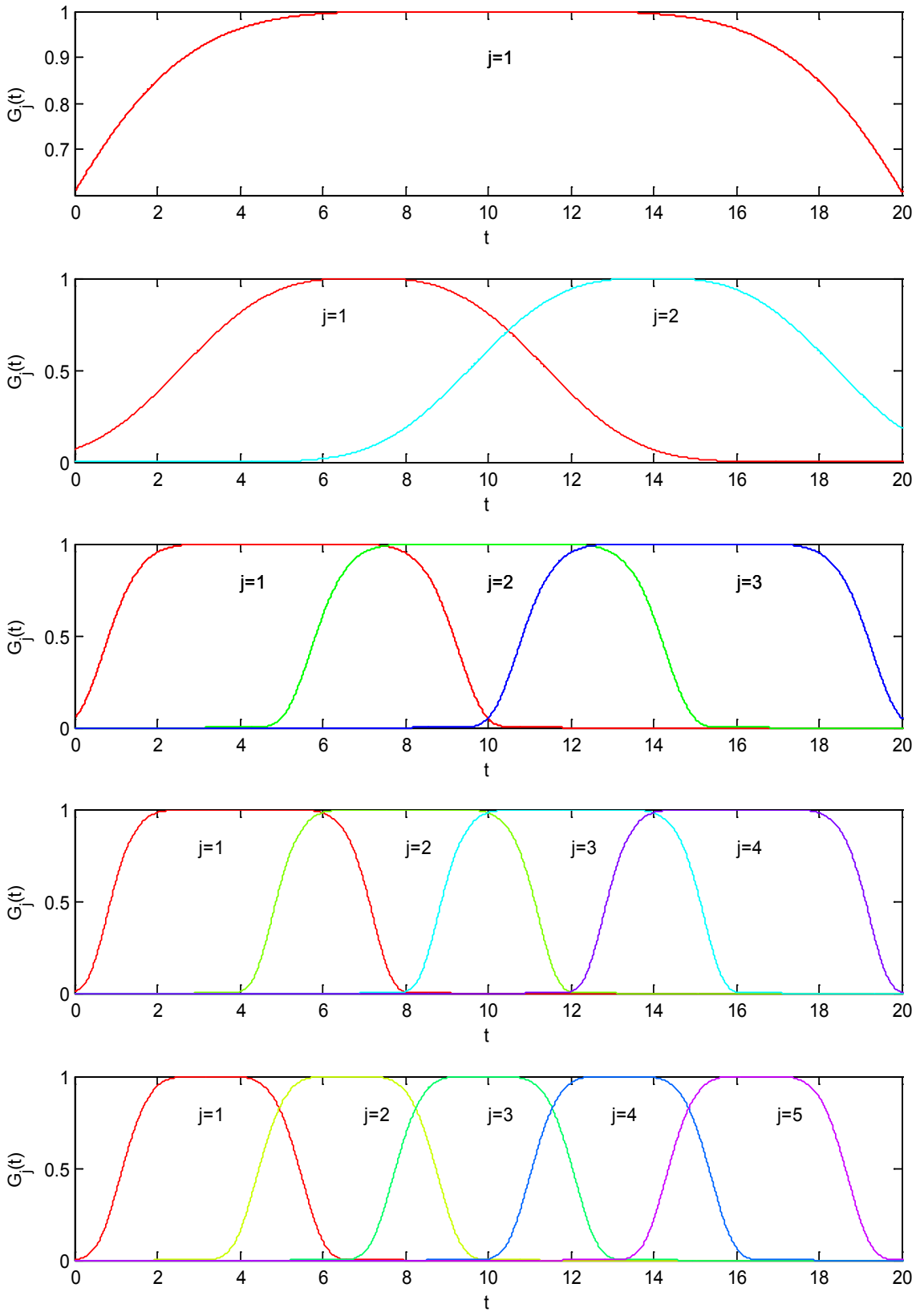
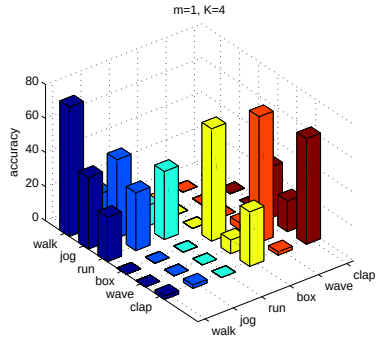
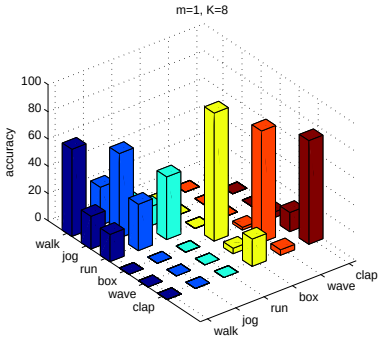
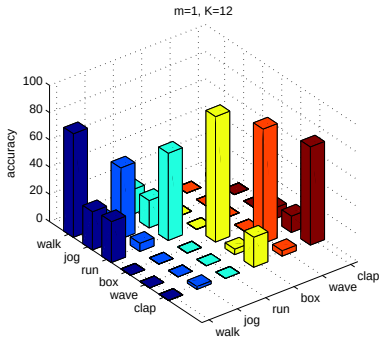
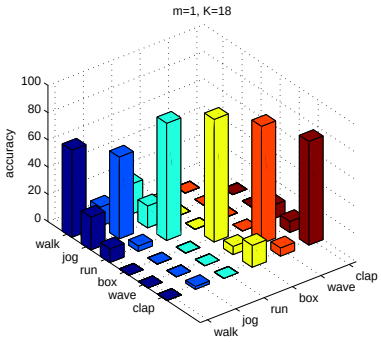


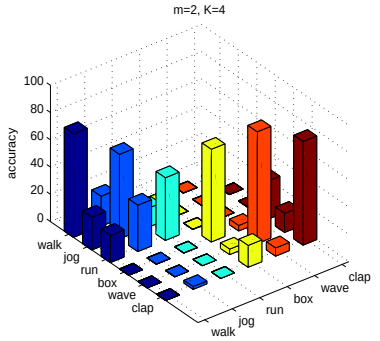
FIG. 7.9. Fuzzy Gaussian membership functions used to represent temporal steps.

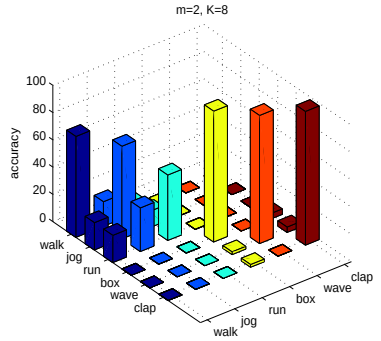

m=1, K=4
accuracy
0
20
40
60
80
walk
jog
run
box
wave
clap
walk
jog
run
box
wave
clap

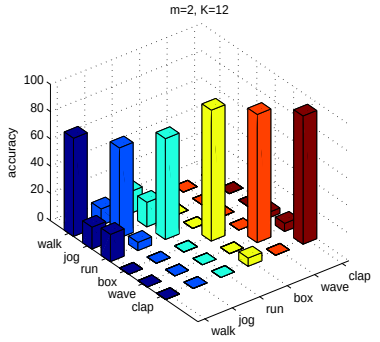

m=1, K=8
accuracy
0
20
40
60
80
100
walk
jog
run
box
wave
clap
walk
jog
run
box
wave
clap

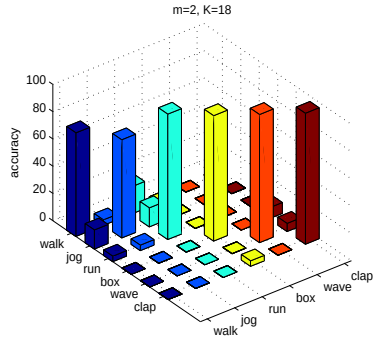

m=1, K=12
accuracy
0
20
40
60
80
100
walk
jog
run
box
wave
clap
walk
jog
run
box
wave
clap

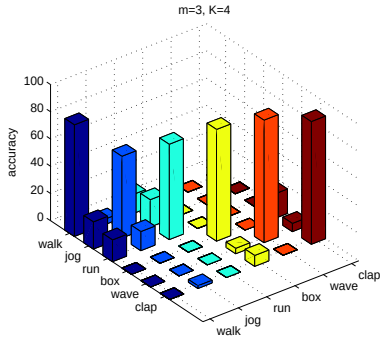

m=1, K=18
accuracy
0
20
40
60
80
100
walk
jog
run
box
wave
clap
walk
jog
run
box
wave
clap

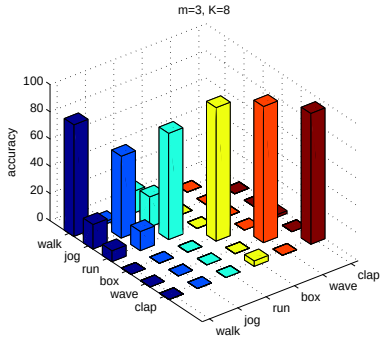

m=2, K=4
accuracy
0
20
40
60
80
100
walk
jog
run
box
wave
clap
walk
jog
run
box
wave
clap

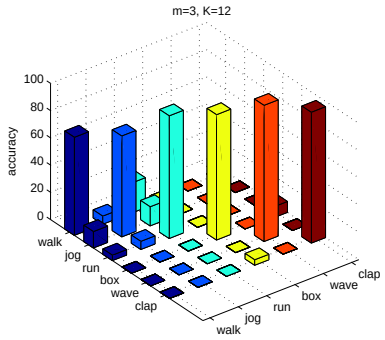

m=2, K=8
accuracy
0
20
40
60
80
100
walk
jog
run
box
wave
clap
walk
jog
run
box
wave
clap

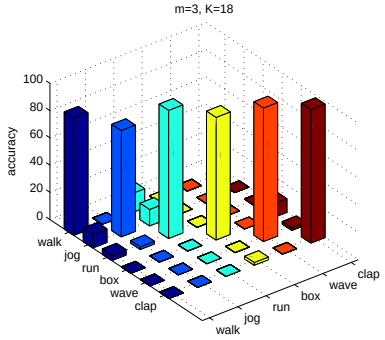

m=2, K=12
accuracy
0
20
40
60
80
100
walk
jog
run
box
wave
clap
walk
jog
run
box
wave
clap

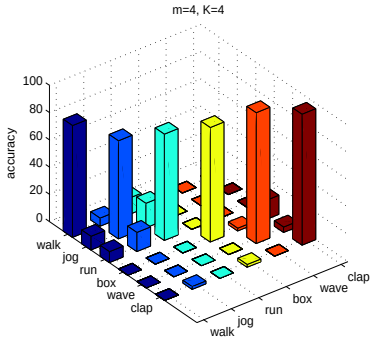

m=2, K=18
accuracy
0
20
40
60
80
100
walk
jog
run
box
wave
clap
walk
jog
run
box
wave
clap

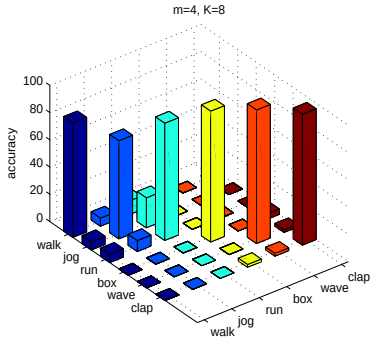

m=3, K=4
accuracy
100
80
60
40
20
0
walk
jog
run
box
wave
clap
walk
jog
run
box
wave
clap

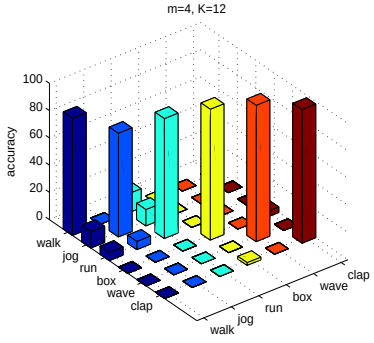

m=3, K=8
accuracy
100
80
60
40
20
0
walk
jog
run
box
wave
clap
walk
jog
run
box
wave
clap

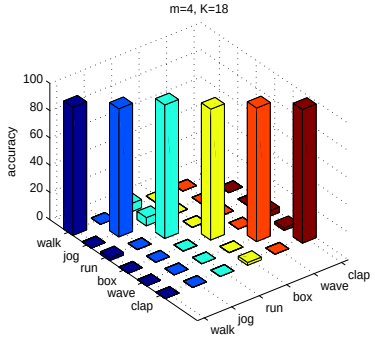

m=3, K=12
accuracy
100
80
60
40
20
0
walk
jog
run
box
wave
clap
walk
jog
run
box
wave
clap


m=3, K=18
accuracy
100
80
60
40
20
0
walk
jog
run
box
wave
clap
walk
jog
run
box
wave
clap


m=4, K=4
accuracy
100
80
60
40
20
0
walk
jog
run
box
wave
clap
walk
jog
run
box
wave
clap


m=4, K=8
accuracy
100
80
60
40
20
0
walk
jog
run
box
wave
clap
walk
jog
run
box
wave
clap


m=4, K=12
accuracy
100
80
60
40
20
0
walk
jog
run
box
wave
clap
walk
jog
run
box
wave
clap


m=4, K=18
accuracy
100
80
60
40
20
0
walk
jog
run
box
wave
clap
walk
jog
run
box
wave
clap

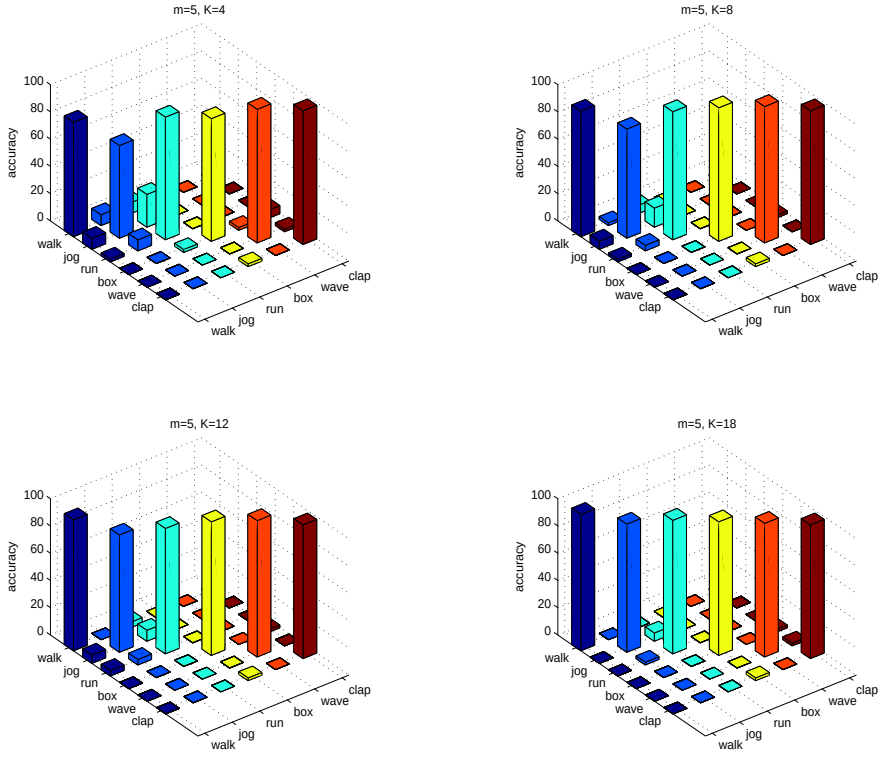


FIG. 7.10. 3D bar plots visualizing the confusion in the action recognition results, each corresponding to different values of the feature parameters K and m.

according to the following expression:

$$\mu_x = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \mu_y = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{7.26}$$

where (μ_x, μ_y) denote 2D coordinates of the centroid of the features. This centroid coincides with the estimated center of mass of the moving ROI (i.e. action actor here). In a similar vein, the dimensions of the moving object are estimated by

$$\sigma_x = 2\sqrt{3\eta_{xx}}, \quad \sigma_y = 2\sqrt{3\eta_{yy}} \tag{7.27}$$

where η_{xx} and η_{yy} are the central moments of the corresponding centroid. In practice, this approach has proved to be significantly more efficient for scenes with a relatively stable background, even with very high levels of noise. In Fig. 7.12, some results of action localization are depicted following this approach.

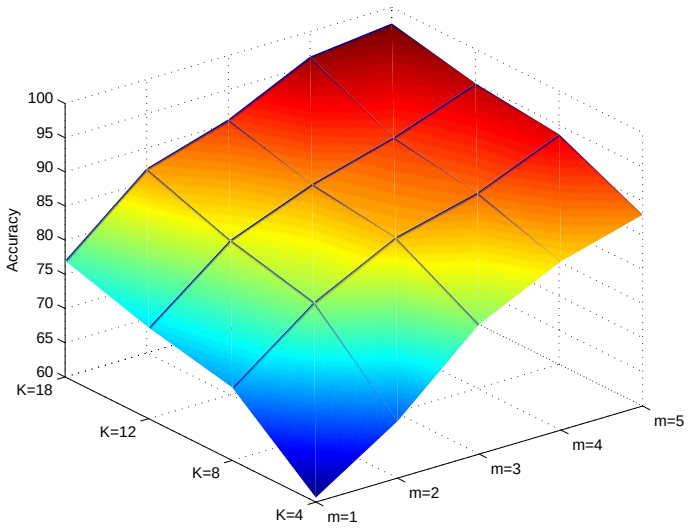
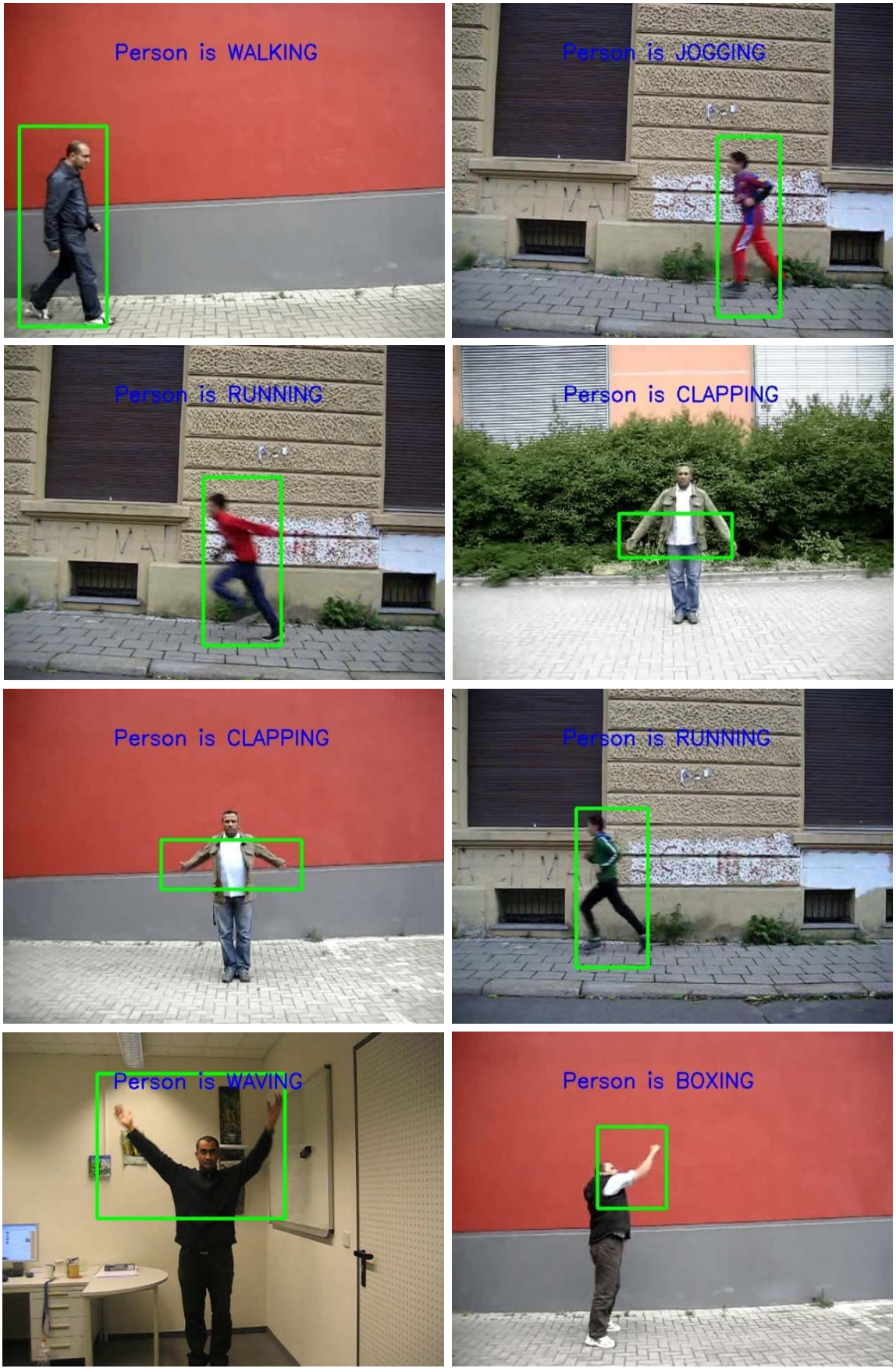


FIG. 7.11. Overall action recognition performance of the proposed framework as a two-dimensional function of the feature parameters K and m.

7.4 Summary and Conclusion

In this chapter, in the beginning, we described the dataset (IIKT action dataset) that we used in this work, and then we presented the details of our proposed approach for action recognition in real-world scenes and the experiments designed to evaluate this approach. This dataset includes realistic video streams of nine different persons, each performing six actions: 'walking', 'jogging', 'running', 'boxing', 'hand waving' and 'hand clapping'. On the basis of the proposed approach towards action recognition in realistic scenarios, a new fuzzy framework for representing and recognizing human actions in real-world video sequences has been presented. In this work, a compact and computationally-efficient descriptor; the fuzzy motion descriptor is constructed based on directional features of optical flow and fuzzy temporal slicing. The one-vs.-rest SVM classifiers have been trained automatically in the feature space for activity classification. The simplicity and computational efficiency of the employed features allow this approach to be more amenable for real-time implementation. It is noteworthy to point out here that the presented experiments conducted so far have demonstrated two points of considerable interest. First, the feature representation parameters K and m are both significant and directly affect the recognition results. Secondly, in terms of holistic performance, the larger values of both parameters provide the greatest improvement in overall recognition rate, and generally are the most important. In other words, the larger the values of the feature parameters are, the better the overall recognition performance is. Finally, for the sake of brevity here, we only affirm that the best overall recognition accuracy


Person is WALKING
Person is JOGGING
Person is RUNNING
Person is CLAPPING
Person is CLAPPING
Person is RUNNING
Person is WAVING
Person is BOXING

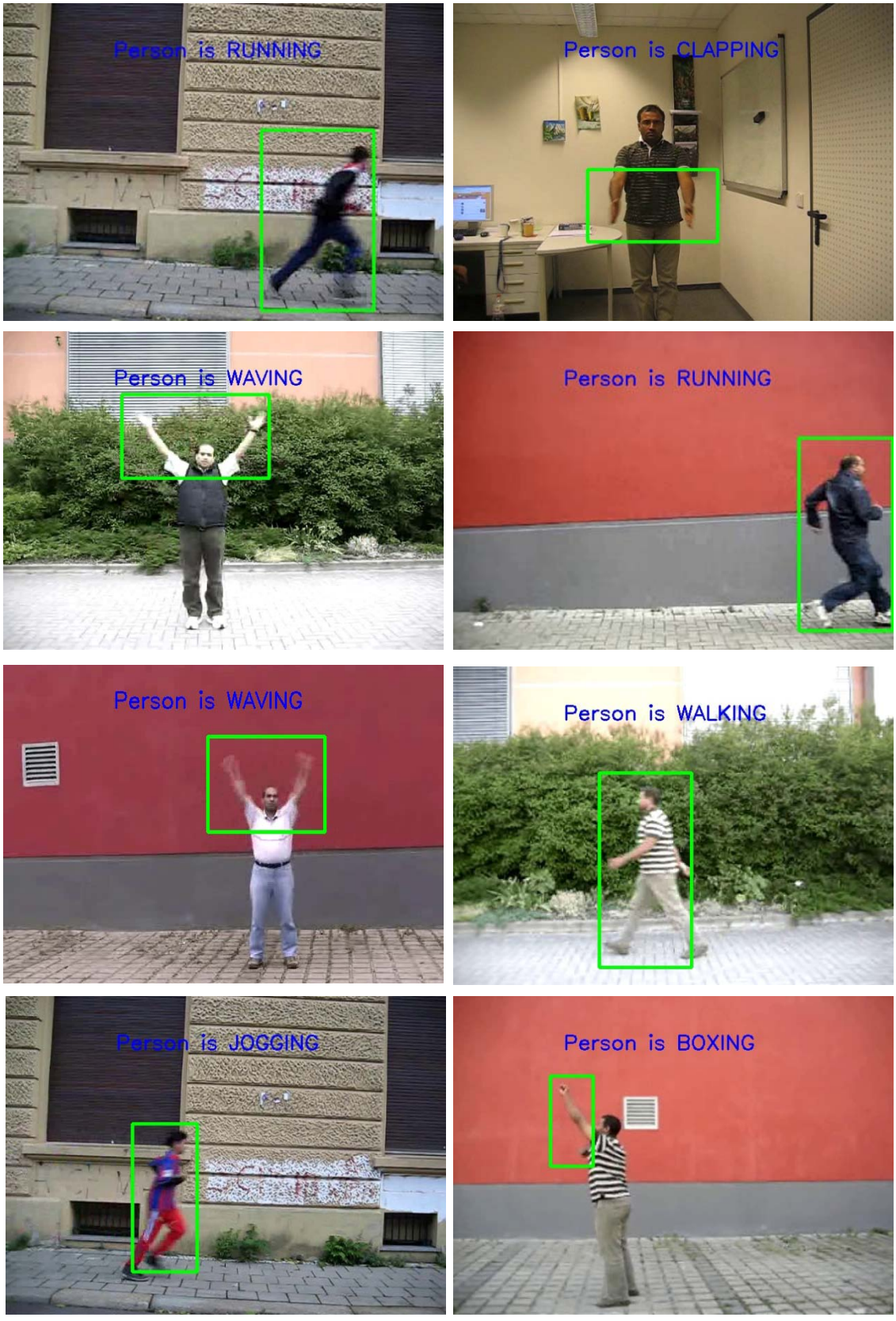


FIG. 7.12. Some results of action localization and recognition in our dataset.

(corresponding to $K = 18$ and $m = 5$) achieved by the proposed approach is 96.3 % which can be regarded as "encouraging", and confirm the basic correctness of the approach, considering the realistic working environments. However, some further investigations on larger realistic datasets may be necessary to enable us to discuss the substantive correctness, robustness, and large-scale feasibility of the approach.

CHAPTER 8

Conclusions & Future Perspectives

In the framework of this doctoral dissertation, the problem of vision-based representation and recognition of human actions in video data has thoroughly been investigated. The research performed for this thesis has led to the several contributions to the area of automatic human action recognition. This short concluding chapter contains two sections. In Section 8.1, the key contributions of the thesis are summed up, and some conclusions from the preceding investigations and experiments are also drawn. In the light of the drawn conclusions, some possible directions for future research within this area, either as an extension of the theory presented in this thesis, or as an alternative are suggested in Section 8.2).

8.1 Summary of the Thesis

Due to its great practical application particularly in human-computer interaction and intelligent systems, the problem of human action recognition in video data has received increasing attention recently from computer vision and pattern recognition community. The overall objective of the work contained in this thesis was to investigate and propose suitable methodologies for developing efficient action models for better visual analysis and interpretation of video data. As discussed earlier, initial chapters have been devoted to tackle the problem of recognizing the observed human actions from video data. In addition, the focus of research has been toward the investigation of a variety of distinctive visual features (i.e., shape features and motion features) for the vision-based representation and recognition of human actions. In Chapter 4, various types of features have been explored

and presented for action recognition. The tentative conclusion drawn from our experimental results is that the careful choice of visual features has a significant impact on an automatic action recognition system, and that several visual feature sets do not seem to serve the recognition performances. This can be regarded as an evaluation of the features with respect to recognition of specific actions.

To fulfill the overall objective of the thesis, several approaches were developed and used to achieve robust action recognition. In the following paragraphs, we briefly outline these approaches. As for the first approach, a new framework for action recognition has been presented, based on log-polar histogram features. The main contribution of this approach is twofold. On the one hand, a reliable neural model as a classifier is used for the task of action recognition. On the other hand, we unfold how the temporal shape variations can be accurately described using a time series of fuzzy log-polar histograms. Preliminary results on KTH and Weizmann action datasets have shown that, with this approach, actions can be recognized with overall recognition rates of 94.3% and 97.8%, respectively. These results compare favorably with those of other investigators published in the literature.

With the second approach, a Bayesian model for human action recognition based on multiple cues has been developed. In a nutshell, this approach proceeds as follows. First, a series of temporal silhouettes of the moving human body parts are extracted from an action clip. Next, each action clip is split into several time-slices represented by fuzzy intervals. As shape features, a variety of shape descriptors both boundary-based (e.g., Fourier descriptors, curvature features, etc.) and region-based (e.g., invariant moments, moment-based features, etc.) are then extracted from the silhouettes. Finally, an NB (Naïve Bayes) classifier is learned in the feature space for action classification. Our preliminary results with KTH action dataset are promising and show and effectiveness and robustness of the approach. Despite their stability and compactness, chord-length shape features have received relatively little attention in the human activity recognition literature. In the third approach, a new methodology for action recognition has been proposed, based on chord-length shape features. In this work, the most interesting contributions can be summarized as follows. We first show how a compact and computationally efficient shape descriptor; the chord-length shape features is constructed using 1-D chord-length functions. Second, we unfold how to use Gaussian membership functions to partition action videos into a number of temporal states to reduce the dimensionality of extracted features. On the KTH action dataset, through this approach, encouraging results have been achieved, which compare favorably with those reported in the literature, while maintaining real-time guarantees.

As for our empirical approach towards action recognition in real-world video

data, an innovative fuzzy framework for representing and recognizing actions in realistic videos has been proposed, based on motion vector distribution characteristics. Within this framework, a compact and computationally-efficient fuzzy descriptor is constructed based on fuzzy directional features. Then, several one-vs.-rest SVMs are trained in the feature space for action classification. The computational complexity of the employed features is relatively low, which guarantees their efficient calculation at real-time. In a set of preliminary experiments on our real-world dataset, we observed that the feature representation parameters directly affect the recognition results. In addition, in terms of the holistic performance, the larger values of the feature parameters provide the greatest improvement in overall recognition rate. The highest overall recognition accuracy achieved using this approach is 96.3% which can be regarded as ”promising“, considering the realistic working environments, and confirm the basic correctness of the approach. However, more comprehensive experimental studies on larger realistic datasets appear to be necessary to to validate the applicability and scalability of the approach.

8.2 Future Perspectives

In the course of this dissertation's work, the focus was mainly directed towards investigating a variety of distinctive visual features for modeling human actions in video data. In order to achieve this goal, several contributions have been made through the research in this study. In the following few paragraphs, we sketch possible future research directions into which the presented work of this thesis can be continued and extended. Strictly speaking, our future work will be organized along three different lines. As a first line of future work, we plan to investigate extensions to the proposed techniques to recognition of unconstrained real-life human actions, since it is a point of great importance to explore the empirical validation of the proposed approaches on large scale realistic and more complex datasets presenting many technical challenges in data handling, such as object articulation, occlusion, and significant background clutter. These key issues merit to be explored more fully in our future work.

Throughout the course of this dissertation, the focus of action recognition has been drawn to the case where there is only a single person in the scene performing a specific action. However, the recognition of more complex activities (i.e. interactions) involving more than one subject (e.g., kissing, hugging, holding-hands, etc.) in the scene performing the action of the interest seems obviously to be of potential application to the design of more efficient human-computer interaction and intelligent systems. Therefore, another significant topic to be investigated

extensively in our future work is the question of how the presented approaches can be adapted or extended in order to deal with such more complex activities.

As is well known, there are tow major types of learning models. For the work described in this thesis, supervised learning has been adopted, whereby the training data (i.e. action features and class labels) are given to train the classifier (e.g., SVM, ANN, and NB). Then, given a set of features extracted from a new observation (i.e. video sequence), the learned classifier can assign the new observation with a proper action. However, unsupervised learning is also feasible and valuable for the analysis of human actions. Therefore, a third line of study that we intend to pursue in our future work is concerned with semi-supervised and unsupervised action recognition. Unsupervised methods for learning human actions are largely based on clustering feature space. Another aspect that would also be of interest to our future work is modeling and analysis of long-term activities (e.g., "Eating", "Going Shopping", "Preparing Meal", etc.) that have tremendous potential to support pervasive applications, especially in the home care and healthcare domains.

Bibliography

[1] A. Bobick and J. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 257–267, 2001.

[2] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 1395–1402, 2005.

[3] S. Ali, A. Basharat, and M. Shah, "Chaotic invariants for human action recognition," in *IEEE International Conference on Computer Vision (ICCV)*, vol. 206, pp. 1–8, January 2007.

[4] I. Laptev and T. Lindeberg, "Space-time interest points," in *IEEE International Conference on Computer Vision (ICCV)*, 2003.

[5] R. Kaundal, A. S. Kapoor, and G. P. S. Raghava, "Machine learning techniques in disease forecasting: A case study on rice blast prediction," *BMC Bioinformatics*, vol. 485, pp. 1471–2105, 2006.

[6] S. Marsland, *Machine Learning: An Algorithmic Perspective*. Chapman & Hall/CRC, 1st ed., 2009.

[7] C. Schüldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proceedings of International Conference on Pattern Recognition (ICPR'04)*, vol. 3, (Cambridge, UK), pp. 32–36, 2004.

[8] S. Sadek, A. Al-Hamadi, B. Michaelis, and U. Sayed, "Towards robust human action retrieval in video," in *Proceedings of the British Machine Vision Conference (BMVC'10)*, (Aberystwyth, UK), September 2010.

[9] G. Garibotto and C. Cibei, "3-d scene analysis by real-time stereovision," in *IEEE International Conference on Image Processing*, vol. 2, pp. 105–108, 2005.

[10] K. Shearer and S. Venkatesh, "Detection of setting and subject information in documentary video," in *International Conference on Multimedia Computing and Systems*, vol. 1, pp. 797–801, 1999.

[11] M. R. Naphade and T. S. Huang, "Detecting semantic concepts using context and audiovisual features," in *IEEE Workshop on Detection and Recognition of Events in Video*, vol. I, pp. 92–98, 2001.

[12] C. C. Fung and N. Jerrat, "A neural network based intelligent intruders detection and tracking system using CCTV images," in *Proceedings of TENCON*, vol. 2, pp. 409–414, 2000.

[13] C. Sacchi, C. Regazzoni, and G. Vernazza, "A neural network-based image processing system for detection of vandal acts in unmanned railway environments," in *11th International Conference on Image Analysis and Processing*, pp. 529–534, 2001.

[14] L. Xie, S.-F. Chang, A. Divakaran, and H. Sun, "Structure analysis of soccer video with hidden Markov models," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 4096–4099, 2002.

[15] P. Peursum, S. Venkatesh, G. A. W. West, and H. H. Bui, "Object labelling from human action recognition," in *First IEEE International Conference on Pervasive Computing and Communications (PerCom'03)*, pp. 399–406, 2003.

[16] M. R. Naphade, T. Kristjansson, B. Frey, and T. S. Huang, "Probabilistic multimedia objects (multijects): A novel approach to video indexing and retrieval in multimedia systems," in *IEEE International Conference on Image Processing (ICIP'98)*, vol. 3, pp. 536–540, 1998.

[17] A. Datta, M. Shah, and N. D. V. Lobo, "Person-on-person violence detection in video data," in *16th International Conference on Pattern Recognition*, vol. 1, pp. 433–438, 2002.

[18] J. Dever, N. Lobo, and M. Shah, "Automatic visual recognition of armed robbery," in *Proceedings of 16th International Conference on Pattern Recognition (ICPR)*, vol. 1, pp. 451–455, 2002.

[19] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 2247–2253, 2007.

[20] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *CVPR*, 2008.

[21] J. K. Aggarwal and Q. Cai, "Human motion analysis: A review," *Computer Vision and Image Understanding (CVIU)*, vol. 73, no. 3, pp. 428–440, 1999.

[22] N. Nguyen, D. Q. Phung, S. Venkatesh, and H. H. Bui, "Learning and detecting activities from movement trajectories using the hierarchical Hidden Markov Models," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 955–960, 2005.

[23] E. Yu and J. K. Aggarwal, "Detection of fence climbing from monocular video," in *IEEE International Conference on Pattern Recognition (ICPR)*, pp. 375–378, 2006.

[24] S. Gong and T. Xiang, "Recognition of group activities using dynamic probabilistic networks," in *9th International Conference on Computer Vision (ICCV'03)*, 2003.

[25] P. Dai, H. Di, L. Dong, L. Tao, and G. Xu, "Group interaction analysis in dynamic context," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 1, no. 38, pp. 275–282, 2008.

[26] D. Damen and D. Hogg, "Recognizing linked events: Searching the space of feasible explanations," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 927–934, 2009.

[27] Y. Shi, Y. Huang, D. Minnen, A. F. Bobick, and I. A. Essa, "Propagation networks for recognition of partially ordered sequential action," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 862–869, 2004.

[28] N. Oliver, E. Horvitz, and A. Garg, "Layered representations for human activity recognition," in *IEEE International Conference on Multimodal Interfaces (ICMI)*, 2002.

[29] D. J. Moore and I. A. Essa, "Recognizing multitasked activities from video using stochastic context-free grammar," in *AAAI/IAAI*, pp. 770–776, 2002.

[30] S.-W. Joo and R. Chellappa, "Attribute grammar-based event recognition and anomaly detection," in *IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2006.

[31] D. Minnen, I. A. Essa, and T. Starner, "Expectation grammars: Leveraging high-level expectations for activity recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 626–632, 2003.

[32] Y. A. Ivanov and A. F. Bobick, "Recognition of visual activities and interactions by stochastic parsing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 852–872, 2000.

[33] K. M. Kitani, Y. Sato, and A. Sugimoto, "Recovering the basic structure of human activities from a video-based symbol string," in *IEEE Workshop on Motion and Video Computing (WMVC)*, 2007.

[34] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis, "Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[35] S. D. Tran and L. S. Davis, "Event modeling and recognition using Markov logic networks," in *Proceedings of European Conference on Computer Vision (ECCV)*, pp. 610–623, 2008.

[36] C. S. Pinhanez and A. F. Bobick, "Human action detection using pnf propagation of temporal constraints," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1998.

[37] J. M. Siskind, "Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic," *Journal of Artificial Intelligence Research (JAIR)*, vol. 15, pp. 31–90, 2001.

[38] R. Nevatia, T. Zhao, and S. Hongeng, "Hierarchical language-based representation of events in video streams," in *IEEE Workshop on Event Mining*, 2003.

[39] V.-T. Vu, F. Brèmond, and M. Thonnat, "Automatic video interpretation: A novel algorithm for temporal scenario recognition," in *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1295–1302, 2003.

[40] M. S. Ryoo and J. K. Aggarwal, "Recognition of composite human activities through context-free grammar based representation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1709–1718, 2006.

[41] E. Aldelson and J. R. Bergen, "Spatiotemporal energy models for the perception of motion," *Journal of the Optical Society of America*, vol. 2, no. 2, pp. 284–299, 1985.

[42] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action MACH: A spatio-temporal maximum average correlation height filter for action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[43] E. Shechtman and M. Irani, "Space-time behavior based correlation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 405–412, 2005.

[44] Y. Ke, R. Sukthankar, and M. Hebert, "Spatio-temporal shape and flow correlation for action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[45] G. Johansson, "Visual motion perception," *Scientific American*, vol. 232, no. 6, pp. 76–88, 1975.

[46] S. Niyogi and E. Adelson, "Analyzing and recognizing walking figures in XYT," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 469–474, 1994.

[47] J. A. Webb and J. K. Aggarwal, "Structure from motion of rigid and jointed objects," *Artificial Intelligence*, vol. 19, pp. 107–130, September 1982.

[48] Y. Sheikh, M. Sheikh, and M. Shah, "Exploring the space of a human action," in *IEEE International Conference on Computer Vision (ICCV)*, vol. 1, pp. 144–149, 2005.

[49] A. Yilmaz and M. Shah, "Recognizing human actions in videos acquired by uncalibrated moving cameras," in *IEEE International Conference on Computer Vision (ICCV)*, 2005.

[50] C. Rao and M. Shah, "View-invariance in action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 316–322, 2001.

[51] P. Dóllar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, pp. 65–72, 2005.

[52] J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," in *British Machine Vision Conference (BMVC)*, 2006.

[53] M. S. Ryoo and J. K. Aggarwal, "Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities," in *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[54] O. Chomat and J. Crowley, "Probabilistic recognition of activity using local appearance," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 1999.

[55] L. Zelnik-Manor and M. Irani, "Event-based analysis of video," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[56] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference*, pp. 147–152, 1988.

[57] D. G. Lowe, "Object recognition from local scale-invariant features," in *IEEE International Conference on Computer Vision ICCV*, pp. 1150–1157, 1999.

[58] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *International Conference on Computer Vision & Pattern Recognition*, vol. 2, pp. 886–893, June 2005.

[59] J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *International Journal of Computer Vision (IJCV)*, vol. 79, pp. 299–318, 2008.

[60] A. Yilmaz and M. Shah, "Actions sketch: a novel action representation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 984–989, 2005.

[61] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *ACM International Conference on Multimedia (ACM MM)*, pp. 357–360, 2007.

[62] M. Bregonzio, S. Gong, and T. Xiang, "Recognising action as clouds of space-time interest points," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[63] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[64] K. Rapantzikos, Y. Avrithis, and S. Kollias, "Dense saliency-based spatiotemporal feature points for action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[65] J. Liu and M. Shah, "Learning human actions via information maximization," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[66] S. Savarese, A. DelPozo, J. Niebles, and L. Fei-Fei, "Spatial-temporal correlations for unsupervised action classification," in *IEEE Workshop on Motion and Video Computing (WMVC)*, pp. 1–8, 2008.

[67] S.-F. Wong, T.-K. Kim, and R. Cipolla, "Learning motion categories using both semantic and structural information," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*, 2007.

[68] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26, pp. 43–49, 1978.

[69] A. Veeraraghavan, R. Chellappa, , and A. Roy-Chowdhury, "The function space of an activity," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 959–968, 2006.

[70] T. Darrell and A. Pentland, "Space-time gestures," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 335–340, 1993.

[71] D. Gavrila and L. Davis, "Towards 3-d model-based tracking and recognition of human movement," in *International Workshop on Face and Gesture Recognition*, pp. 272–277, 1995.

[72] A. Efros, A. Berg, G. Mori, , and J. Malik, "Recognizing action at a distance," in *IEEE International Conference on Computer Vision (ICCV)*, vol. 2, pp. 726–733, 2003.

[73] Y. Yacoob and M. Black, "Parameterized modeling and recognition of activities," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 120–127, 1998.

[74] R. Lublinerman, N. Ozay, and D. Z. O. Camps, "Activity recognition from silhouettes using linear systems and model (in)validation techniques," in *IEEE Int. Conference on Pattern Recognition (ICPR)*, pp. 347–350, 2006.

[75] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 267–296, February 1989.

[76] Z. Ghahramani, "Learning dynamic bayesian networks," in *Adaptive Processing of Sequences and Data Structures*, vol. 1387 of *Lecture Notes in Artificial Intelligence*, pp. 168–187, Springer-Verlag, 1998.

[77] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden Markov model," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 379–385, 1992.

[78] T. Starner and A. Pentland, "Real-time american sign language recognition from video using hidden Markov models," in *International Symposium on Computer Vision*, 1995.

[79] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. IT-13, pp. 260–269, April 1967.

[80] M. Brand, "Coupled hidden Markov models for modeling interacting processes," *Technical Report 405, Massachusetts Institute of Technology Media Lab Perceptual Computing*, 1997.

[81] N. M. Oliver, B. Rosario, and A. P. Pentland, "A bayesian computer vision system for modeling human interactions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 831–843, 2000.

[82] S. Park and J. K. Aggarwal, "A hierarchical bayesian network for event recognition of human actions and interactions," *Multimedia Systems*, vol. 10, no. 2, pp. 164–179, 2004.

[83] P. Natarajan and R. Nevatia, "Coupled hidden semi Markov models for activity recognition," in *IEEE Workshop on Motion and Video Computing (WMVC)*, 2007.

[84] D. Weinland, "Action representation and recognition," *PhD thesis, Institut National Polytechnique de Grenoble*, 2008.

[85] A. F. Bobick and A. D. Wilson, "A state-based approach to the representation and recognition of gesture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 12, pp. 1325–1337, 1997.

[86] F. Lv and R. Nevatia, "Single view human action recognition using key pose matching and viterbi path searching," in *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2007.

[87] S. Sadek, A. Al-Hamadi, A. Wannig, B. Michaelis, and U. Sayed, "A new approach to image segmentation via fuzzification of Rènyi entropy of generalized distributions," in *Proceedings of Int. Conference on Image, Signal and Vision Computing (ICISVC'09)*, (Singapore), pp. 598–603, August 2009.

[88] S. Sadek, A. Al-Hamadi, B. Michaelis, and U. Sayed, "An efficient method for noisy cell image segmentation using generalized α-entropy," in *Proceedings of International Symposium on Signal Processing, Image Processing and Pattern Recognition (SIP'09)*, Lecture Notes in Computer Science, (Jeju Island, Korea), pp. 33–40, Springer-Verlag Berlin/Heidelberg, November 2009.

[89] A. L. Ratan, O. Maron, W. E. L. Grimson, and T. Lozano-Perez, "A framework for learning query concepts in image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 423–431, 1999.

[90] Y. Weiss, "Segmentation using eigenvectors: A unifying view," in *Proceedings of the International Conference on Computer Vision*, vol. 2, pp. 975–982, 1999.

[91] C. Ware, *Information is Visualization*. Morgan Kaufmann, Los Altos, 2000.

[92] W. Metzger, "Gesetze des sehens," *Waldemar Kramer Verlag, Frankfurt am Main, 2nd edition*, 1953.

[93] N. Ouerhani and H. Högli, "MAPS: Multiscale attention-based presegmentation of color images," in *4th International Conference on Scale-Space theories in Computer Vision*, pp. 537–549, 2003.

[94] T. Lehmann, W. Oberschelp, E. Pelikan, and R. Repges, "Bildverarbeitung für die medizin," *Springer-Verlag Berlin*, 1997.

[95] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics (SMC-9)*, vol. 9, no. 1, pp. 62–66, 1979.

[96] C. E. Shannon and W. Weaver, "The mathematical theory of communication," *Urbana, University of Illinois Press*, 1949.

[97] A. Rényi, "On a theorem of P. Erdös and its application in information theory," *Mathematica*, vol. 1, pp. 341–344, 1959.

[98] C. Tsallis, S. Abe, and Y. Okamoto, *Nonextensive Statistical Mechanics and its Applications*. Series Lecture Notes in Physics, Springer, March 2001.

[99] W. Tatsuaki and S. Takeshi, "When nonextensive entropy becomes extensive," *Physica A.*, vol. 301, pp. 284–290, 2001.

[100] R. C. Gonzalez and R. E. Woods, *Digital Image Processing Using Matlab*. Prentice Hall, Inc, Upper Saddle River, NJ, 2nd Edition, 2003.

[101] I. Levner and H. Zhang, "Classification-driven watershed segmentation," *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1437–1445, 2007.

[102] A. C. Bovik, *Handbook of Image and Video Processing*. Academic Press, 2000.

[103] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: A systematic survey," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 294–307, 2005.

[104] H. Tao, H. Sawhney, and R. Kumar, "Dynamic layer representation and its applications to tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 134–141, 2000.

[105] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia, "Event detection and analysis from video streams," *IEEE Transactions on Pattern Analysis and Machine Vision*, vol. 23, no. 8, pp. 873–889, 2001.

[106] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, and L. Wixson, "A system for video surveillance and monitoring," Technical Report CMU-RI-TR-00-12, CMU, Robotics Institute, Carnegie Mellon University, May 2000.

[107] M. Irani, B. Rousso, and S. Peleg, "Computing occluding and transparent motions," *Int. J. Computer Vision*, vol. 12, no. 1, pp. 5–16, 1994.

[108] J. J. Gibson, "On the analysis of change in the optic array," *Scandinavian I. Psychol*, vol. 18, pp. 161–163, 1977.

[109] E. P. Simoncelli, "Design of multi-dimensional derivative filters," in *IEEE Int. Conf. Image Processing*, vol. 1, pp. 790–793, 1994.

[110] B. Lucas and T. Kanade, "Performance of optical flow techniques," in *Proceedings of DARPA IU Workshop*, pp. 121–130, 1981.

[111] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[112] A. Tekalp, *Digital Video Processing*. Prentice Hall, 1995.

[113] J. Fan, G. Zeng, M. Body, and M.-S. Hacid, "Seeded region growing: An extensive and comparative study," *Pattern Recognition Letter*, vol. 8, no. 26, pp. 1139–1156, 2005.

[114] S. Sadek, A. Al-Hamadi, B. Michaelis, and U. Sayed, "Real-time automatic traffic accident recognition using HOF," in *Proceedings of the 20th International conference on Pattern Recognition (ICPR'10)*, (Istanbul, Turkey), pp. 3348–3351, August 2010.

[115] Y. Bar-Shalom and X. R. Li, *Estimation and Tracking: Principles, Techniques, and Software*. Boston, MA: Artech House, 1993.

[116] S. J. McKenna, Y. Raja, and S. Gong., "Object tracking using adaptive color mixture models," in *Proceedings of Asian Conf. Computer Vision*, pp. 615–622, 1998.

[117] S. J. McKenna, Y. Raja, and S. Gong, "Tracking color objects using adaptive mixture models," *Image and Vision Computing*, vol. 17, pp. 223–229, 1999.

[118] Y. Raja, S. J. McKenna, and S. Gong, "Tracking and segmenting people in varying lighting conditions using color," in *Proceedings of IEEE Int. Conf. Automatic Face Gesture Recognition*, pp. 228–233, 1998.

[119] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pp. 246–252, 1999.

[120] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2, pp. 107–123, 2005.

[121] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *IJCV*, vol. 37, no. 2, pp. 151–172, 2000.

[122] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

[123] S. Sadek, A. Al-Hamadi, B. Michaelis, and U. Sayed, "Human activity recognition: A scheme using multiple cues," in *Proceedings of the International Symposium on Visual Computing (ISVC'10)*, vol. 1, (Las Vegas, Nevada, USA), pp. 574–583, November 2010.

[124] S. Sadek, A. Al-Hamadi, M. Elmezain, and B. Michaelis, "Human activity recognition using temporal shape moments," in *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT2010)*, (Luxor, Egypt), pp. 79–84, December 2010.

[125] D. Zhang and G. Lu, "A comparative study of fourier descriptors for shape representation and retrieval," in *Proceedings of 5th Asian Conference on Computer Vision*, 2002.

[126] H. Kauppinen, T. Seppanen, and M. Pietikainen, "An experimental comparison of auto-regressive and fourier-based descriptors in 2-d shape classification," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 2, pp. 201–207, 1995.

[127] R. B. Yadava, N. K. Nishchala, A. K. Gupta, and V. K. Rastogi, "Retrieval and classification of shape-based objects using fourier, generic fourier, and wavelet-fourier descriptors technique: A comparative study," *Optics and Lasers in Engineering*, vol. 45, no. 6, pp. 695–708, 2007.

[128] D. Zhang and G. Lu, "A comparative study of curvature scale space and fourier descriptors for shape-based image retrieval," *Visual Communication and Image Representation*, vol. 14, no. 1, pp. 39–57, 2003.

[129] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[130] S. X. Liao and M. Pawlak, "On image analysis by moments," *Pattern Anal. Machine Intell.*, vol. 18, no. 3, pp. 254–266, 1996.

[131] R. R. Bailey and M. Srinath, "Orthogonal moment features for use with parametric and non-parametric classifiers," *IEEE Trans. Patterns Analysis Machine Intelligence*, vol. 18, no. 4, pp. 389–399, 1996.

[132] M.-K. Hu, "Pattern recognition by moment invariants," *IEEE Transactions on Information Theory*, vol. 49, pp. 14–28, 1961.

[133] B. P. Bogert, M. J. R. Healy, and J. W. Tukey, *The Quefrency Analysis of Time Series for Echoes: Cepstrum, Pseudo Auto covariance, Cross-Cepstrum and Saphe Cracking*, ch. 15, pp. 209–243. Proceedings of the Symposium on Time Series Analysis, New York: John Wiley and Sons, 1963.

[134] N. Alajlan, M. S. Kamel, and G. Freeman, "Multi-object image retrieval based on shape and topology," *Signal Processing: Image Communication*, vol. 21, no. 10, pp. 904–918, 2006.

[135] S. Haykin, *Neural Networks: A Comprehensive Foundation, 2nd ed*. New York: Macmillan College Publishing Company, Inc., 1994.

[136] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Computation*, vol. 7, pp. 1–58, 1992.

[137] V. Vapnik, *The nature of statistical learning theory*. New York: Springer, 1995.

[138] C. J. C. Burges and B. Schoelkopf, "Improving the accuracy and speed of support vector learning machines," *Advances in neural information processing systems*, vol. 9, pp. 375–381, 1997.

[139] P. Langley, W. Iba, and K. Thompson, "An analysis of bayesian classifier," in *Proceedings of the 10th National Conference on Artificial Intelligence*, pp. 223–228, 1992.

[140] S. Bhattacharyya, P. Dutta, and U. Maulik, "Object extraction using self organizing neural network with a multi-level sigmoidal transfer function," in *Proceedings of 5th International Conference on Advances in Pattern Recognition*, pp. 435–438, 2003.

[141] S. Sadek, A. Al-Hamadi, B. Michaelis, and U. Sayed, "A robust neural system for objectionable image recognition," in *Proceedings of Second International Conference on Machine Vision (ICMV2009)*, (Dubai, UAE), pp. 32–36, 2009.

[142] C. Cortes and V. Vapnik, "Support-vector networks," *Journal of Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[143] N. I. M. Gould and P. L. Toint, "A quadratic programming bibliography," `http://www.optimization-online.org/DBHTML/2001/02/285.html`, 2001.

[144] N. I. M. Gould and P. L. Toint, "A quadratic programming page," `http://www.numerical.rl.ac.uk/qp/qp.html`.

[145] N. Cristianini and J. Shawe-Taylor, "An introduction to support vector machines," *Cambridge University Press*, 2000.

[146] D. D. Lewis, "Naïve (bayes) at forty: The independence assumption in information retrieval," in *Proceedings of the 10th European Conference on Machine Learning (ECML-98)*, vol. 1398, pp. 4–15, 1998.

[147] W. M. Bolstad, "Introduction to bayesian statistics," *Wiley & Sons*, pp. 55–105, 2004.

[148] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, "A biologically inspired system for action recognition," in *ICCV*, pp. 1–8, 2007.

[149] A. Kläser, M. Marszaek, and C. Schmid, "A spatiotemporal descriptor based on 3d-gradients," in *BMVC*, pp. 995–1004, 2008.

[150] G. Willems, T. Tuytelaars, and L. Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *ECCV (2)*, vol. 5303 of *Lecture Notes in Computer Science*, pp. 650–663, Springer, 2008.

[151] S. Sadek, A. Al-Hamadi, B. Michaelis, and U. Sayed, "An action recognition scheme using fuzzy log-polar histogram and temporal self-similarity," *EURASIP Journal on Advances in Signal Processing*, January 2011.

[152] S. Sadek, A. Al-Hamadi, B. Michaelis, and U. Sayed, "An efficient method for real-time activity recognition," in *Proceedings of the International Conference on Soft Computing and Pattern Recognition (SoCPaR 2010)*, (Paris, France), pp. 7–10, December 2010.

[153] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 264–271, 2003.

[154] H. Maurase and S. Nayar, "Visual learning and recognition of 3d objects from appearance," *International Journal of Computer Vision (IJCV)*, vol. 14, pp. 5–24, 1995.

[155] J. R. Movellan, "Visual speech recognition with stochastic networks," *Advances in Neural Information Processing Systems*, vol. 7, pp. 851–858, 1995.

[156] The Georgia-Tech Gait Recognition Database. `http://www.cc.gatech.edu/cpl/projects/hid/images.html`, April 2004.

[157] R. Gross and J. Shi, "The CMU Motion of Body (MoBo) Database," *Technical Report CMU-RI-TR-01-18, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA*, 2001.

[158] C. Schüldt, B. Caputo, and I. Laptev, "Action recognition based on local space-time features," Master Thesis, School of Electrical Engineering, Royal Institute of Technology, 2004.

[159] Weizmann Institute of Science. `http://www.weizmann.ac.il/acadaff/Scientific_Activities/current/weizmann.html`.

[160] S. Sadek, A. Al-Hamadi, B. Michaelis, and U. Sayed, "A new method for image classification based on multi-level neural networks," in *Proceedings of International Conference on Signal and Image Processing (ICSIP2009)*, (Amsterdam, Netherlands), pp. 197–200, September 2009.

[161] Y. Ke, R. Sukthankar, and M. Hebert, "Efficient visual event detection using volumetric features," in *IEEE Conference on Computer Vision (ICCV)*, vol. 1, pp. 166–173, 2005.

[162] Y. Wang and G. Mori, "Max-margin hidden conditional random fields for human action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[163] T. K. Kim, S. F. Wong, and R. Cipolla, "Tensor canonical correlation analysis for action classification," in *CVPR*, 2007.

[164] Z. Zhang, Y. Hu, S. Chan, and L.-T. Chia, "Motion context: A new representation for human action recognition," *ECCV*, vol. 4, pp. 817–829, 2008.

[165] A. Fathi and G. Mori, "Action recognition by learning midlevel motion features," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[166] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.

[167] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Machine Learning*, vol. 29, pp. 103–137, 1997.

[168] S. Suzuki and K. Abe, "Topological structural analysis of digitized binary images by border following," *CVGIP*, vol. 30, no. 1, pp. 32–46, 1985.

[169] S. Sadek, A. Al-Hamadi, B. Michaelis, and U. Sayed, "An SVM approach for activity recognition based on chord-length-function shape features," in *IEEE International Conference on Image Processing (ICIP'12)*, (Florida, U.S.A.), pp. 767–770, October 2012.

[170] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," *Lecture Notes in Computer Science*, vol. 2749, pp. 363–370, 2003.

Concise Curriculum Vitae

Name:	Samy S. M. Bakheet
Citizenship:	Egyptian
Marital Status:	Married with two kids
Date of Birth:	12th Dec, 1973, in Sohag, Egypt
Mail:	P.O. Box 4120, 39016 Magdeburg, Germany
E-mail:	samy.bakheet@gmail.com

Qualifications:

2008 – Present	Pursuing a Ph.D. degree in Technical Computer Science, Otto-von-Guericke University Magdeburg (FEIT), Germany.
2002 – 2005	**M.Sc.** in Computer Science, Sohag University, Egypt.
2001 – 2002	**PG Diploma** in Computer Science, Sohag University, Egypt.
1996 – 2000	**B.Sc.** in Computer Science, Sohag University, Egypt.
1992 – 1996	**B.Sc.** in Mathematics, South Valley University, Sohag, Egypt.

Professional Experience:

2008 – Present	Ph.D. Scholar, Dept. of Technical Computer Science, Otto-von-Guericke University Magdeburg (FEIT), Germany.
2005 – 2008	Research associate, Sohag University, Egypt.
2001 – 2005	Research assistant, South Valley University, Sohag, Egypt.

Magdeburg, 28.06.2013

Samy Bakheet

Related Publications

Most of the material contained in this doctorial dissertation is partly based on the following collection of refereed papers published in a variety of peer-reviewed journals and/or proceedings of well reputed international conferences/symposia.

1) **S. Sadek**, A. Al-Hamadi, B. Michaelis, and U. Sayed, "Chord-length shape features for human activity recognition," *Journal ISRN Machine Vision*, vol. 1, pp. 1–9, November 2012.

2) **S. Sadek**, A. Al-Hamadi, B. Michaelis, and U. Sayed, "Human action recognition via affine moment invariants," in *21st International Conference on Pattern Recognition (ICPR'12)*, Tsukuba Science City, Japan, November 2012, pp. 218–221.

3) **S. Sadek**, A. Al-Hamadi, B. Michaelis, and U. Sayed, "An SVM approach for activity recognition based on chord-length-function shape features," in *IEEE International Conference on Image Processing (ICIP'12)*, Florida, U.S.A., October 2012, pp. 767–770.

4) **S. Sadek**, A. Al-Hamadi, B. Michaelis, and U. Sayed, "A fast statistical approach for human activity recognition," *International Journal of Intelligence Science (IJIS)*, vol. 2, no. 1, pp. 9–15, January 2012.

5) **S. Sadek**, A. Al-Hamadi, B. Michaelis, and U. Sayed, "Face detection and localization in color images: An efficient neural approach," *Journal of Software Engineering and Applications (JSEA)*, vol. 4, pp. 682–687, December 2011.

6) **S. Sadek**, A. Al-Hamadi, B. Michaelis, and U. Sayed, "Human action recognition: A novel scheme using fuzzy log-polar histogram and temporal self-similarity," *EURASIP Journal on Advances in Signal Processing*, January 2011.

7) **S. Sadek**, A. Al-Hamadi, M. Elmezain, B. Michaelis, and U. Sayed, "Human activity recognition using temporal shape moments," in *IEEE International*

Symposium on Signal Processing and Information Technology (ISSPIT'10), Luxor, Egypt, December 2010, pp. 79–84.

8) **S. Sadek**, A. Al-Hamadi, B. Michaelis, and U. Sayed, "An efficient method for real-time activity recognition," in *Proceedings of the International Conference on Soft Computing and Pattern Recognition (SoCPaR'10)*, Paris, France, December 2010, pp. 7–10.

9) **S. Sadek**, A. Al-Hamadi, B. Michaelis, and U. Sayed, "Human activity recognition: A scheme using multiple cues," in *Proceedings of the International Symposium on Visual Computing (ISVC'10)*, Las Vegas, Nevada, USA, November 2010, vol. 1, pp. 574–583.

10) **S. Sadek**, A. Al-Hamadi, B. Michaelis, and U. Sayed, "Towards robust human action retrieval in video," in *Proceedings of the British Machine Vision Conference (BMVC'10)*, Aberystwyth, UK, September 2010.

11) **S. Sadek**, A. Al-Hamadi, B. Michaelis, and U. Sayed, "A statistical framework for real-time traffic accident recognition," *Journal of Signal and Information Processing (JSIP)*, vol. 1, pp. 77–81, 2010.

12) **S. Sadek**, A. Al-Hamadi, B. Michaelis, and U. Sayed, "Real-time automatic traffic accident recognition using HOF," in *Proceedings of the 20th International conference on Pattern Recognition (ICPR'10)*, Istanbul, Turkey, August 2010, pp. 3348–3351.

13) **S. Sadek**, A. Al-Hamadi, B. Michaelis, and U. Sayed, "Efficient region-based image querying," *Int. Journal of Computing*, vol. 2, no. 10, pp. 1–6, June 2010.

14) M. Elmezain, A. Al-Hamadi, **S. Sadek**, and B. Michaelis, "Robust methods for hand gesture spotting and recognition using hidden markov models and conditional random fields," in *IEEE Int. Symposium on Signal Processing and Information Technology (ISSPIT'10)*, Luxor, Egypt, December 2010, pp. 131–136.

15) **S. Sadek**, A. Al-Hamadi, A. Wannig, B. Michaelis, and U. Sayed, "A new approach to image segmentation via fuzzification of Rènyi entropy of generalized distributions," in *Proceedings of International Conference on Image, Signal and Vision Computing (ICISVC'09)*, Singapore, August 2009, pp. 598–603.

16) **S. Sadek**, A. Al-Hamadi, B. Michaelis, and U. Sayed, "A new method for image classification based on multi-level neural networks," in *Proceedings of International Conference on Signal and Image Processing (ICSIP'09)*, Amsterdam, Netherlands, September 2009, pp. 197–200.

17) **S. Sadek**, A. Al-Hamadi, B. Michaelis, and U. Sayed, "Cubic-spline neural network-based system for image retrieval," in *Proceedings of Sixth International IEEE Conference on Image Processing (ICIP'09)*, Cairo, Egypt, December 2009, pp. 273–276.

18) **S. Sadek**, A. Al-Hamadi, B. Michaelis, and U. Sayed, "An image classification approach using multilevel neural networks," in *Proceedings of IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS'09)*, Shanghai, China, 2009, pp. 180–183.

19) **S. Sadek**, A. Al-Hamadi, B. Michaelis, and U. Sayed, "An efficient method for noisy cell image segmentation using generalized α-entropy," in *Proceedings of International Symposium on Signal Processing, Image Processing and Pattern Recognition (SIP'09)*, Jeju Island, Korea, November 2009, Lecture Notes in Computer Science, pp. 33–40, Springer-Verlag Berlin/Heidelberg.

20) **S. Sadek**, A. Al-Hamadi, B. Michaelis, and U. Sayed, "An efficient approach for region-based image classification and retrieval," in *Proceedings of International Symposium on Signal Processing, Image Processing and Pattern Recognition (SIP'09)*, Jeju Island, Korea, November 2009, Lecture Notes in Computer Science, pp. 56–64, Springer-Verlag Berlin/Heidelberg.

21) **S. Sadek**, A. Al-Hamadi, B. Michaelis, and U. Sayed, "A robust neural system for objectionable image recognition," in *Proceedings of Second International Conference on Machine Vision (ICMV'09)*, Dubai, UAE, 2009, pp. 32–36.

22) **S. Sadek**, A. Al-Hamadi, B. Michaelis, and U. Sayed, "Image retrieval using cubic spline neural networks," *International Journal of Video & Image Processing and Network Security (IJIPNS)*, vol. 9, no. 10, pp. 17–22, 2009.

23) U. Sayed, **S. Sadek**, and B. Michaelis, "Two phases neural network-based system for pornographic image classification," in *5th International Conference: Sciences Of Electronic,Technologies Of Information and Telecommunications (SETIT'09)*, TUNISIA, March 2009, pp. 1–6.

24) **S. Sadek**, U. Sayed, and H. H. Amin, "An neural network for face detection and localization in color images," in *Proceedings of First Engineering Conference for Young Researchers EM3AC-07-Engineering between Theory and Practice*, Assiut, Egypt, May 2007, pp. 168–171.