

STRUCTURED AND SEQUENTIAL REPRESENTATIONS FOR HUMAN
ACTION RECOGNITION

by

Oya Çeliktutan

B.S., Electronics Engineering, Uludağ University, 2005

M.S., Electrical and Electronics Engineering, Boğaziçi University, 2008

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Electrical and Electronics Engineering Department

Boğaziçi University

2013

STRUCTURED AND SEQUENTIAL REPRESENTATIONS FOR HUMAN
ACTION RECOGNITION

APPROVED BY:

Prof. Bülent Sankur
(Thesis Supervisor)

Assoc. Prof. Burak Acar

Prof. Lale Akarun

Assoc. Prof. Murat Saraçlar

Assoc. Prof. Christian Wolf

DATE OF APPROVAL: 06.09.2013

ACKNOWLEDGEMENTS

To my family...

First of all, I would like to express my deepest sense of gratitude to my supervisor, Prof. Bülent Sankur, for his invaluable guidance and his infinite support not only in my academic studies but also in all aspects of my life. He has always inspired me by his never-ending energy and great enthusiasm for life, research and teaching, and impressed me by his huge knowledge on nearly everything.

I am deeply grateful to my co-supervisor, Assoc. Prof. Christian Wolf. It was a great chance for me to visit LIRIS, INSA de Lyon, France during my doctoral studies and to work with him. He has always found a trade-off between letting his students stand on their own feet and leading them. I have learnt a lot from him in a short time.

I am thankful to members of my thesis committee, Prof. Lale Akarun, Assoc. Prof. Burak Acar and Assoc. Prof. Murat Saraçlar. They provided me a critical reading and valuable suggestions which have been very important for the improvement of this dissertation. I would also like to thank Assist. Prof. Ali Albert Salah, Assoc. Prof. A. Taylan Cemgil and Ceyhun Burak Akgül for providing guidance and encouragement.

I feel so lucky to have such wonderful friends, İpek Şen, Erinç Dikici, Doğaç Başaran and Çisel Aras. Swimming times, Sirtaki lessons, weekly classical music concerts. These are only a few of my beautiful memories that has enlivened my life between never-ending deadlines. I am thankful for all the delightful moments shared together. Especially, I would like to thank İpek for being always there when I need her. I would also like to thank Kültür Dostları team, Gönenc Tarakçioğlu and Neslihan Gerek, for their motivation, and Neşe Alyüz for weekly coffee break sessions. Many thanks to our excellent secretary Leyla Çeken for her infinite support and to Yasin Çitkaya for his kind help in tedious assistantship duties.

It was a great pleasure to be a member of BUSIM. I would like to thank all former and current BUSIM members. However, there are some special people, no one could take their place: Hatice Çınar Akakın, Helin Dutağacı, Ebru Arısoy and Bilgin Esmé. I am thankful to them for their invaluable friendship. I greatly enjoyed collaborating with Cem Sübakan and Sezer Ulukaya, especially, Cem who has carried on the Bayesian spirit in the laboratory.

I would also like to thank all LIRIS, INSA de Lyon members for their warm hospitality during my stay in France. Especially, Peng Wang and Youyao Zhang, I believe that we laid the foundation for a lifelong friendship. I thank Eric Lombardi for his collaboration as well as for his effort to teach me GPU implementation. Also thanks to Mingyuan Jiu for brainstorming sessions on action recognition.

I have no words to express my gratitude to my parents İpek and Turgay Çeliktutan. None of this would be possible without their endless love and encouragement. Millions of thanks to my dear brother Onur, my grandmother Gönül Erol and my parents-in-law Teslime and İbrahim Dikici for their warm support. Finally, my special thanks to mon époux Çağatay Dikici for all the wonderful moments, especially for his invaluable guidance and endless encouragement to finalize this dissertation and to pursue my dreams.

This thesis was supported by Boğaziçi University Research Fund under Project No. 12A02D2.

ABSTRACT

STRUCTURED AND SEQUENTIAL REPRESENTATIONS FOR HUMAN ACTION RECOGNITION

Human action recognition problem is one of the most challenging problems in the computer vision domain, and plays an emerging role in various fields of study. In this thesis, we investigate structured and sequential representations of spatio-temporal data for recognizing human actions and for measuring action performance quality. In video sequences, we characterize each action with a graphical structure of its spatio-temporal interest points and each such interest point is qualified by its cuboid descriptors. In the case of depth data, an action is represented by the sequence of skeleton joints. Given such descriptors, we solve the human action recognition problem through a hyper-graph matching formulation. As is known, hyper-graph matching problem is NP-complete. We simplify the problem in two stages to enable a fast solution: In the first stage, we take into consideration the physical constraints such as time sequentiality and time irreversibility for the actions; in the second stage we approximate the problem using a sparse subset of spatio-temporal interest points. The reduced problem is then elegantly solved with the dynamic programming technique. Our approach results in competitive performance figures vis-à-vis the state-of-the-art action recognition algorithms. The proposed hyper-graph matching formulation has also been applied to the problem of the quality of action rendition. Finally, we present an alternative formulation of the action recognition problem via Hidden Markov Models (HMMs). To learn HMM parameters, contrary to the conventional approach, Expectation-Maximization algorithm, we demonstrate the practical employment of a spectral algorithm. Given the large variations in action sequences, we resort to a clustering scheme for exploring the subgroups in the training data and for learning multiple HMMs per action category.

ÖZET

YAPISAL VE ARDIŞIK GÖSTERİMLER İLE İNSAN EDİMLERİNİ TANILAMA

İnsan edimlerini tanılama en zorlu bilgisayarla görü problemlerinden biridir ve çok geniş uygulama alanlarına sahip olması bakımından oldukça önemli bir rol oynamaktadır. Video verisinden insan edimlerini analiz etmek amacıyla, bu çalışmada yapısal ve ardışık gösterimleri göz önüne aldık. Her bir edimi, imge dizilerinde, uzam-zamansal ilgin noktaları ve görünüşe dayalı öznitelikler ile betimlerken, derinlik verisinde ise bir iskelet dizisi şeklinde ifade ettik. Bu bağlamda, insan edimlerini tanılama problemini hiper-çizge eşleme ile formüle ettik. Bilindiği üzere, hiper-çizge eşleme NP-tam bir problemdir. Verimli bir şekilde çözüme ulaşmak amacıyla, bu çalışmada problemi iki aşamada indirgedik. İlk aşamada, insan edimlerinin zamanda ardışıklık özelliklerini göz önüne aldık. İndirgenmiş problemin dinamik programlama tekniği ile verimli bir şekilde çözülebileceğini gösterdik. İkinci aşamada ise çizge modelini seyrek bir ilgin nokta kümesinden oluşturarak yaklaştık. Yaklaşımımız literatürdeki yöntemler ile başa baş bir sonuç vermektedir. Geliştirilen algoritmayı aynı zamanda derinlik verisine ve edim tanılama literatüründe farklı bir probleme uyguladık. Buradaki amaç, insanların bir eğitmen önderliğinde verilen edimleri hangi ölçüde doğru yaptığını öznel olarak nicelemektir. Hiper-çizgeye dayalı önerilen yöntem doğru ve yanlış hareketlerin ayırt edilmesi ve yürütülen bir edime öznel bir puan atanması problemlerine uygulanmıştır. Ayrıca, edim tanılama probleminin Saklı Markov Modelleri ile alternatif bir formülasyonunu önerdik. Sık kullanılan parametre kestirimi yöntemi beklenti-enbüyütme algoritmasının yerine, bu çalışmada yeni bir parametre kestirimi yönteminin pratik uygulanması vurgulanmaktadır. Bu parametre kestirimi yöntemi, insan edimlerinin sınıf-içi çeşitliliğini etkin bir biçimde ele almak amacıyla, edim sınıfı başına çoklu Saklı Markov Modellerinin sonsuz katışımının öğrenilmesinde kullanılmıştır.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
ABSTRACT	v
ÖZET	vi
LIST OF FIGURES	x
LIST OF TABLES	xii
LIST OF SYMBOLS	xiv
LIST OF ACRONYMS/ABBREVIATIONS	xv
1. INTRODUCTION	1
2. STATE-OF-THE-ART	8
2.1. Literature on Action Recognition	8
2.1.1. Actions: Data and Descriptor Types	8
2.1.1.1. Low-level descriptors	9
2.1.1.2. Mid-level descriptors	12
2.1.1.3. High-level descriptors	13
2.1.2. Review of Action Recognition Methods	14
2.1.2.1. BoW and its extensions	14
2.1.2.2. The use of Graphs	17
2.1.2.3. Sequential methods	19
2.1.2.4. The use of Trees	21
2.1.2.5. Parts-based structured methods or SVMs	22
2.2. Benchmark Datasets	23
3. SPATIO-TEMPORAL HYPER-GRAPH MATCHING	29
3.1. Related Literature on Graph Matching	31
3.2. General Problem Formulation	35
3.3. Spatio-Temporal Matching	40
3.3.1. Properties of Spatio-Temporal Data	40
3.3.2. Matching	41
3.3.3. Computational Complexity	46
3.4. Graphical Structure	48

3.4.1.	Single-chain-single-point Model	50
3.4.2.	Single-chain-multiple-points Model	52
3.4.3.	Multiple-chains Model	53
3.5.	Application to Video-based Action Recognition	54
3.5.1.	STIP Detection and Description	55
3.5.1.1.	3D Harris detector	55
3.5.1.2.	2D Gabor filters	56
3.5.1.3.	Cuboid descriptors	57
3.5.2.	Learning Discriminative Graph Prototypes	57
3.5.3.	Experimental Setup	58
3.5.4.	Choice of Graphical Structure	59
3.5.5.	Prototype Selection	61
3.5.6.	Comparison with State-of-the-art	64
3.5.7.	A Real-time GPU Implementation	65
3.6.	Summary	65
4.	HYPER-GRAPH BASED ANALYSIS OF SKELETON SEQUENCES	71
4.1.	Related Literature on Quality Assessment	72
4.2.	Pose Descriptor Extraction	73
4.2.1.	Angle-based Pose Descriptor	73
4.2.2.	Distance-based Pose Descriptor	75
4.2.3.	Pose Quantization	76
4.3.	Graph-based Sequence Alignment	76
4.4.	Learning Model Graphs	78
4.5.	Action Quality Assessment	79
4.5.1.	Classification: Correct vs. Wrong	79
4.5.2.	Quality Assessment	80
4.6.	Experiments and Results	81
4.6.1.	Datasets and Experimental Setup	81
4.6.2.	Action Recognition Results	83
4.6.3.	Comparison with State-of-the-art Methods	83
4.6.4.	Quality Assessment Result	86
4.6.4.1.	Classification results	86

4.6.4.2.	Regression analysis	86
4.7.	Summary	88
5.	MIXTURE OF HIDDEN MARKOV MODELS	90
5.1.	Hidden Markov Models	91
5.1.1.	Learning Model Parameters of HMM	92
5.1.1.1.	Expectation-Maximization algorithm	92
5.1.1.2.	Spectral algorithm	94
5.2.	Mixtures of Hidden Markov Models	97
5.2.1.	Spectral Learning of Mixture of HMMs	98
5.3.	Application to Human Action Recognition	99
5.3.1.	Spatio-temporal Interest Point Detection and Action Description	99
5.3.2.	Experimental Results	101
5.4.	Summary	101
6.	CONCLUSION	104
6.1.	Fast Hyper-Graph Matching for Spatio-temporal Data	104
6.2.	Skeleton-based Human Action Analysis	107
6.3.	Mixture of Hidden Markov Models	108
6.4.	Conclusions	109
	REFERENCES	112

LIST OF FIGURES

Figure 1.1.	Illustration of the action <i>answer phone</i> variations over the shape and size of the object, the style of the subject, pose factor and camera view angle.	3
Figure 2.1.	<i>KTH</i> dataset is a typical example of “one actor, one action, uniform background” concept [1].	24
Figure 2.2.	Recent research trends stress the recognition of the actions in the wild, some examples are illustrated from <i>Hollywood-2</i> dataset [2]. .	25
Figure 2.3.	<i>LIRIS Human Activities Dataset</i> [3] is one pioneering multi-modal dataset in that it contains complex human activities in a realistic setting.	26
Figure 3.1.	Classification of graph matching algorithms pertinent to our perspective.	32
Figure 3.2.	Illustration of graph matching problem in our formulation.	37
Figure 3.3.	Illustration of calculating the geometric deformation, $D_g(\cdot)$	39
Figure 3.4.	Separation of node assignments.	42
Figure 3.5.	An example hyper-graph.	47
Figure 3.6.	Proposed graphical structure.	49
Figure 3.7.	Computation of HoG and HoF [4]. Courtesy of Ivan Laptev.	57

Figure 3.8.	Examples for matched sequences.	67
Figure 4.1.	Angles computed for skeleton representation.	74
Figure 4.2.	(a) Illustration of torso basis; (b) Illustration of the spherical coordinates, i.e., radius R , inclination angle α_j and azimuth angle β_j , defined based on the torso frame.	75
Figure 4.3.	Each triangle models the spatial relationship between three consecutive skeletons in the triangular structure graph.	77
Figure 4.4.	Example illustrations.	84
Figure 4.5.	Calculated distance, QM , vs. simulated mismatch (perturbation variance).	88
Figure 5.1.	Directed Acyclic Graph (DAG) (a) of HMM and (b) of MHMM.	93
Figure 5.2.	Expectation-Maximization algorithm for HMM.	94
Figure 5.3.	HMM spectral learning algorithm.	98
Figure 5.4.	Each box (frame) is described by the number of detected STIPs within the blocks. An action can be regarded as a sequence of boxes.100	

LIST OF TABLES

Table 1.1.	Overview of the organization of the thesis.	7
Table 2.1.	How can an action be represented?	10
Table 2.2.	Taxonomy of the reviewed approaches.	15
Table 2.3.	Overview of multi-modal datasets for human activity recognition. .	28
Table 3.1.	Definitions and notations used throughout this chapter.	36
Table 3.2.	Confusion matrices before prototype selection.	62
Table 3.3.	Confusion matrices after prototype selection.	63
Table 3.4.	Summary of the experimental results. Run time (ms/frame) is computed for matching one model graph on a CPU with 2.8GHz and 8GB RAM.	64
Table 3.5.	Comparison with the state-of-the-art methods on the KTH database.	66
Table 3.6.	Run times in milliseconds for a CPU and two different GPUs and for 4 different scene block sizes.	68
Table 3.7.	Summary of the computational complexities and the recognition performances of the proposed approaches.	69
Table 4.1.	The action types used in the experiments.	82

Table 4.2.	Recognition performances (%) on MSR: MSR Action 3D Dataset and WSU: WorkoutSU Gesture datasets. L is the number of model graphs used in the experiments.	85
Table 4.3.	Recognition performances (%) under additive Gaussian noise. We set σ value to 0.1, 0.5 and 1 for mild, medium and severe noise cases, respectively.	85
Table 4.4.	Recognition performances (%) for MSR Action 3D Dataset [5] in cross-subject test setting. The respective standard deviations are 5, 7.5, and 13.2 for the last column.	86
Table 4.5.	Recognition performances (%) for Workout SU-10 Gesture dataset [6] in cross-subject test setting.	87
Table 4.6.	Classification performance of the proposed framework: correctly performed sequences vs. wrongly performed sequences. Overall classification performance is 86.6%.	87
Table 5.1.	Confusion matrices for (a) HMM with EM learning; (b) HMM with spectral learning; (c) MHMM with spectral learning.	103

LIST OF SYMBOLS

$D(\cdot)$	Space-time geometric distortion between two triangles
$D_g(\cdot)$	Euclidean distance between sets of angles
$D_t(\cdot)$	Truncated time differences
\mathcal{E}	Set of edges
f_i	Local appearance feature vector of a point p_i
\mathcal{G}	Generic hyper-graph
O	Observation probability matrix
p_i	Space-time position of a point
T	State transition probability matrix
T^t	Maximum allowable time difference
$U(\cdot)$	Euclidean distance between feature vectors
\mathcal{V}	Set of nodes
W^d	Penalty for dummy assignment
W^t	Maximum penalty for temporal distortion
$x_i = j$	Variable : i^{th} model node is assigned to j^{th} scene node
x	The whole set of assignments: $x = \{x_i\}, i = 1..M$
$\Delta(\cdot)$	Time difference between two pairs of nodes
ϵ	Dummy value (node is assigned)
$\lambda_1, \lambda_2, \lambda_3$	Weighting parameters
π_i	Initial state probability
$\phi^{(\cdot)}(i, j, k)$	Angle with respect to the point j

LIST OF ACRONYMS/ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
BoW	Bag of Words
DAG	Directed Acyclic Graph
DPM	Deformable Part-based Model
DTW	Dynamic Time Warping
EM	Expectation-Maximization
HAR	Human Action Recognition
HMM	Hidden Markov Models
HoF	Histogram of Flow
HoG	Histogram of Gradient
LSVM	Latent Support Vector Machine
MHMM	Mixture of Hidden Markov Models
PCA	Principle Component Analysis
RDF	Random Decision Forest
RGB-D	Red Green Blue-Depth
STIP	Spatio-Temporal Interest Point
SVM	Support Vector Machine

1. INTRODUCTION

The problem of Human Action Recognition, in simple words, can be defined as “design of an automated system that can recognize what action is being performed, given an image or a sequence of images of one or more people performing an action”. The terms “action” and “activity” are inconsistently used in the literature. Although there is no clear distinction between these terms, we refer to an action as one of the semantic segments of an activity, or simply a delineated part of a series of motion or movements that composes an activity. Sample actions are *walk*, *hand wave*, *answer phone*, *tennis serve* etc., corresponding to body displacements, gesturing, office activity, sports activity, respectively. By extension, an activity denotes a combination of action instances having a successive or concurrent relationship in the temporal domain. While actions are generally executed by an individual and last for a relatively short duration of time, activities may include actions of several people, usually in interaction, and are characterized by longer time scales. “*A person unlocks a room, and then enters*”, “*two people meet and handshake*” and “*a person gives an object to one another*” can be given as example activities.

In the last two decades, human action recognition has gained a lot of importance as a research area, and has become one of the fastest growing fields in the computer vision community. Its connections to various fields of study, critical application areas and potential marketing impact all render this topic very attractive for the scientists. In the following, its relevant applications are presented under three major disciplines: (i) sociology; (ii) medicine; (iii) computer science or robotics.

Monitoring of the behavior and activities of individuals or of crowds, namely surveillance, is useful for such applications as maintaining social control and security, preventing or investigating crimes, and detecting threats. For example, an algorithm with an acceptable false alarm rate can be used to attract the attention of the security personnel to anomalous activities and hence to improve the security level. Design of such systems has been studied by many researchers [7] with the focus of collective

activity recognition [8–10], crowd analysis [11, 12], abnormal behavior recognition [13] and behavioral biometrics. Especially, in behavioral biometrics, identifying people based on their distinctive gait pattern [14] has gained interest as a complementary part in multi-biometrics, or in the absence of other modalities such as face or fingerprint.

Assistive technology (e.g. smart homes) and robotics have become instrumental for promoting the health and wellbeing of the elderly living at home, and improving the quality of life of disabled people. In the same vein, gaming and virtual reality systems have emerged in rehabilitation of individuals with physical and cognitive disabilities resulting from neurological conditions such as stroke, acquired brain injury and in the diagnosis and treatment of developmental disorders such as autism and Asperger syndromes. One can cite smart homes, fall detection [15], movement quality assessment of impaired people [16], analysis of parent-child social interactions for early autism detection [17] as currently prominent research topics in this field.

Gestures and actions of people form a rich source of information for creating ergonomic and effective human-machine interfaces. They also offer insight into understanding personality traits, emotions and mood which are important clues to develop emotionally intelligent systems that adapt and respond better to the user’s need. Such human-centric interfaces are now becoming widespread in ambient intelligence, cinema, entertainment and gaming industry where some concrete examples can be given as animation [18], customizable hand gesture recognition interfaces [19, 20] for interacting with the laptops, smartphones and cars, interactive smart rooms [21, 22], smart televisions, game consoles such as Xbox 360 [23], and interactive self learning tools [24, 25].

Finally, action/activity recognition techniques can be utilized for processing the ever-increasing amount of video data available. If we consider daily uploaded images, videos to social networking websites such as Facebook, YouTube (100 hours of video are uploaded to YouTube every minute [26]), videos recorded by TV-channels etc., we can have an idea on the incredible amount of data available and the growing need to datamine this information by effective techniques enabling content-based video analysis, video annotation and retrieval [27–29], and video summarization [30].

The application of human action understanding to a variety of aspects of life is enormous. However, despite the available data and the advances in motion capturing technology, solution to this problem has proven to be extremely difficult (arguably one of the most challenging problems ever faced by the computer vision community) due to inherent variations such as the randomness and variability in the rate, style, posture of the subject, as well as the multitude of confounding factors such as the presence of multiple subjects, viewing angle, camera motion, background changes, resolution, illumination, occlusions and clothing. Consider a simple action of *answer phone*. As illustrated in Figure 1.1, the object can vary in appearance, shape and size; the manner of holding it, with left hand or right hand, the movements of the person, the body posture, camera view angle etc. can all differ to cause significant intra-class variations exceeding even inter-class variations. An action recognition system that works well under and across all these variations, and that infers the target action in a time efficient manner has not been feasible yet. The techniques developed so far is far from the intended goal and there is a large room for improvement in this area.



Figure 1.1. Illustration of the action *answer phone* variations over the shape and size of the object, the style of the subject, pose factor and camera view angle.

Since the problem at hand is really challenging, the researchers make full use of all types of machine learning methods ranging from Support Vector Machines, Random Decision Forests to probabilistic graphs such as Bayesian networks, Markov Random Fields etc. There can be several paradigms to the action recognition problem: non-structured, structured, sequential and holistic approaches etc. While non-structured methods like Bag of Words approach does not take into account the configuration of the descriptors in the video, structured approaches like graphical structures model the configuration of the descriptors both in space and time or their parent-child relationships. In the same vein, sequential approaches consider the order of the observed descriptors. Finally, the holistic approaches handle an action as a 3-dimensional object in the spatio-temporal domain and rest upon volumetric representations.

In this thesis, we investigate the use of structured and sequential representations in order to recognize human actions. We proceed from simple to complex actions in two different settings. We can summarize our achievements and novelties under two items in connection to these specific settings.

- *Action recognition from video sequences.* In this setting, we aim to recognize actions in video sequences, and focus on “one actor, one action, uniform background” case. Our proposed methodology relies on the configuration of the interest points both in the space and time domain. A spatio-temporal interest point (STIP) can be defined as a point exhibiting saliency in the space and time domains, i.e., high gradient or local maxima of the spatio-temporal filter responses. Early approaches exploit these interest points in a Bag-of-Words (BOW) [31] formalism which converts the local descriptors encoding STIPs into a numerical representation by disregarding most of the structural information. On the other hand, graph-structured representations offer a natural description for this type of information. In this context, our main contributions can be given as follows.
 - (i) *Hyper-graph Matching.* We present a novel unified representation of the local descriptors and their geometrical configuration by the use of hyper-graphs where the nodes of the graph correspond to the spatio-temporal interest points and neighborhood relationship is derived from proximity information.

So a hyper-graph or point-set matching algorithm quantifies the deformation between two graphs, i.e., a model graph and a scene graph, and a distance measure is deduced which can be further used for action recognition in a classification scheme.


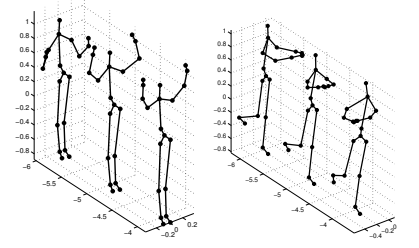
- (ii) Graph matching is a commonly used technique in object recognition. Temporal domain extensions of graphs have only relatively recently started being applied due to its extra computational complexity. As our main methodological contribution we show that, in the case of spatio-temporal data, typical of video applications, the exact solution to the point set matching problem with hyper-graphs can be calculated in polynomial complexity when the hyper-graph is constructed by respecting certain properties of the time domain.
- (iii) We introduce a new graphical structure which is specifically designed for the spatio-temporal data. The proposed algorithm when applied to graphs designed with this special structure, allows calculation of matches with computational complexity that grows linearly in both the number of model nodes and the number of scene nodes.
- (iv) *Hidden Markov Models*. Hidden Markov Models (HMMs) and their variants are widely employed for sequence alignment and classification in the action recognition domain. Likewise, we learn a Mixture of HMMs (MHMMs) for recognition. As is known, the exact inference requires the evaluation of an intractable integral over the HMM parameters. The classical methods use Expectation-Maximization (EM) algorithm for parameter estimation. One disadvantage of EM-based algorithms is that they need a good initialization; yet there is no guarantee to reach the global optimal. Therefore, we present a spectral method to estimate the parameters of HMM; the latter method is computationally practical and deterministic, but requires larger number of samples. Given the large variations in action sequences, we resort to a clustering scheme for exploring the subgroups in the training data and we use the spectral method to efficiently learn multiple HMMs per action category in the spirit of MHMMs. In this context, we more aim to demonstrate the practical usage of spectral learning of MHMMs rather than to improve the

action recognition performance.

- *Analysis of actions in depth sequences.* In the second setting, we utilize depth sequences recorded with a consumer depth camera and the tracked skeleton joints. The main contribution is that we adapt our hyper-graph-based method to align two dynamic skeleton sequences, and apply it both to action recognition task and to the objective quantification of the goodness of the action performance. The automated measurement of “action quality” has the potential to monitor and gauge action imitations, for example, during a physical therapy or dance lessons. Action correspondences are again established through the graph matching formalism that jointly measures spatio-temporal domain deformations. In addition to recognizing actions, the deformation measure has also been used for separating acceptable and unacceptable action performances and for a continuous quantification of the action performance quality.

The organization of the thesis is tabulated in Table 1.1. Chapter 2 gives a brief survey on the action recognition literature. Chapter 3 introduces the efficient hyper-graph matching approach for spatio-temporal data and Chapter 4 presents the application of the hyper-graph based method to the depth modality. Chapter 5 introduces the spectral learning approach for Mixture of Hidden Markov Models. Finally, Chapter 6 concludes.

Table 1.1. Overview of the organization of the thesis.

Chapter	Methodology	Problem	Scenario
Chapter 3	Hyper-Graph Matching	Action recognition from video sequences	
Chapter 5	Hidden Markov Models		
Chapter 4	Hyper-Graph Matching	Analysis of actions in skeleton (depth) sequences	

2. STATE-OF-THE-ART

In this chapter, we present a survey of non-exhaustive action recognition methods in the last decade and available datasets for evaluation.

2.1. Literature on Action Recognition

There is a very large scale of human action and activity recognition approaches available in the literature and these approaches have been extensively reviewed in recent papers [32–35]. These approaches can be categorized in various ways, for example, based on the criteria of the type or modality of the observed data (still image, video sequence, Motion Capture or depth data), on the information source underlying the methodology (low-level, mid-level or high-level descriptors), on the methodological assumptions (e.g., unstructured or structured in the spatio-temporal domain, sequential, volumetric) and on the intended goal (e.g., multi-view action recognition, collective activity recognition, activity recognition for interaction etc.).

Despite the difficulty of finding clear-cut distinctions since algorithms often share techniques common to more than one category, we have found it useful to classify them based on the specific machine learning technique. However, before delving into the state-of-the-art approaches, we briefly introduce the data types and common descriptors for action or activity recognition. The following section will make clear which method needs to be used and how for each data type, feature or specific task.

2.1.1. Actions: Data and Descriptor Types

An action or activity is a function of time, it may be perceived as extending over a whole period of time or as a series of repeated occurrences or as a series of states, e.g., begin, apex and end. This interpretation is the essential first step for recognizing actions and for realizing the subsequent tasks, e.g., segmentation, semantic reasoning from these primitives. Notice that, there also exist several studies [36–41]

that have a remarkable ability to recognize actions from a single image. However, inherent limitations of still image or video data restrict open-ended improvements in human action recognition, e.g., in the presence of severe illumination artifacts, camera view-point change and dynamic complex backgrounds.

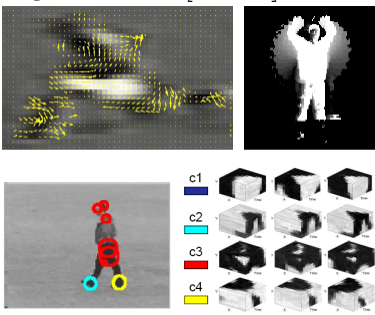
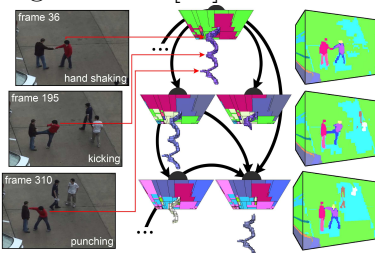
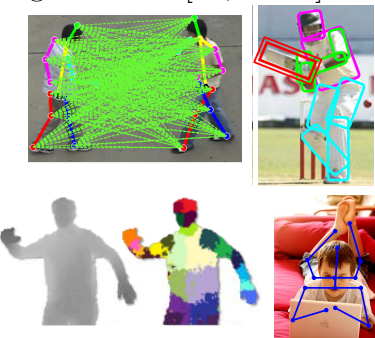
Recently, many researchers have shifted their focus onto gesture and/or action analysis based on the depth sequences with the development of low-cost consumer cameras. One advantage of depth data modality is that it can potentially mitigate the problems encountered in the case of still image or video data. Secondly, more informative representations of human body movements can be obtained based on depth maps and point clouds. For this reason, in the last three years, seminal works for action recognition has been revamped and quickly adapted to the depth data modality.

Another interesting data type could be Motion Capture (MoCap) data. In this case, body-worn sensors are used to measure the motion of each part. However, this modality is substantially different than the video-based action recognition methods. In this work, we limited ourselves to vision-based methods only.

Given the above data types, the descriptors (features) can be subsumed under the following three categories: (i) Low-level descriptors; (ii) Mid-level descriptors; (iii) High-level descriptors. An overview of these descriptors is given in Table 2.1.

2.1.1.1. Low-level descriptors. The early methods for low-level feature extraction benefit from two main information sources, shape and motion. Shape-based methods consider an action as a 3-dimensional (3D) object in the space-time domain. They characterize the spatial appearance by background subtracted silhouettes, blobs, contours or gradients, and deduce the volume from stacking together the tracked contours [49] or blobs [50]. Features can be moments, skeletal curve, medial axis, chain codes, geometric features (e.g., peaks, pits, valleys) and 3D shape descriptors that are intended to describe global appearance, boundary or volumetric properties of the delineated volume. A pioneering approach, proposed by Bobick and Davis [42], accumulated the

Table 2.1. How can an action be represented?

Category	Approaches	Illustration
Low-level descriptors	<ul style="list-style-type: none"> -Silhouette volumes and contour stack descriptors; -Moments, 3D shape descriptors; -Optical flow, trajectories, motion field descriptors; -Spatio-temporal interest points detection and description etc. 	Figures from [42–44]. 
Mid-level descriptors	<ul style="list-style-type: none"> -Learning the spatio-temporal relationship of the interest points; -Grouping interest points based on the proximity information or grouping trajectories; -Spatio-temporal tubes etc. 	Figure from [45]. 
High-level descriptors	<ul style="list-style-type: none"> -Pose, poselets, skeletons; -Object detection; -Scene understanding; -Interaction context, crowd context; -Semantic attributes etc. 	Figures from [41, 46–48]. 

blobs over the frames into a static image, called as “Motion History Images”. Two other approaches, proposed in [49] and [50], best exemplify the holistic or volumetric approaches. Although these methods are simple and fast, they can properly work in just controlled settings. Shape and appearance are prone to errors of background subtraction, light variations, shadows, and clothing. Moreover, silhouettes are not adequate enough for describing the action as the motion and shape information inside the contour is ignored.

Second category of methods, motion, can be relatively descriptive, and, at the same time, quite robust against shape and appearance variations. A popular approach is to estimate the motion field by optical flow and to directly extract features from the resulting flow pattern [43, 51]. A prominent method in this case, proposed by Ke *et al.* [52], extended the Haar type features [53] to 3D volume and applied to the dense optical flow pattern. Another way to describe the motion is trajectory of the interest points. The fact remains that the optical flow is sensitive to noise and illumination changes, and suffers from high computational complexity. Likewise, motion trajectories are affected by rotations, camera position, scale changes, and occlusions.

Currently, holistic or volumetric methods solely based on shape or motion have become out-of-date, however, a significant improvement has been noted with the local methods combining both sources of information. These methods focus more on the local neighborhoods in spatio-temporal domain, and thus become more robust against the aforementioned impediments and also bypass the tedious segmentation task. One outstanding method is the spatio-temporal interest point (STIP) detector. A STIP detector can be considered as a spatio-temporal filter that gives a high response at the points exhibiting significance, e.g., high gradient or local maxima, in the spatio-temporal domain. The most popular interest point detection methods are periodic (1D Gabor filters) [54], 3D Harris corner detector [44], extension of SIFT to time domain [55], 3D Hessian [56], 3D Gabor filters [57], 2D Gabor filters applied on the region of interest [58]. It is typical to describe the local neighborhood of an interest point by concatenation of gradient values [54], of optical flow values [54], or of spatio-temporal jets [1], histogram of gradient and optical flow values (HoG and HoF) [44], and

3D Scale-invariant Feature Transform (SIFT) [55]. Comparison of the existing STIP detection and description methods with sparse or dense sampling (using a regular grid) strategies have been well studied in the papers [59, 60]. The effort still continues to incorporate other channels, e.g., color SIFTs [61], or for the depth data [62, 63]. For example, Hadfield and Bowden [63] recently extended the benchmark interest-point detectors [44, 54, 56] into the depth modality.

2.1.1.2. Mid-level descriptors. Low-level descriptors either capture only the local motion information within a small spatio-temporal patch, namely cuboid, or describe the holistic nature. For this reason, mid-level descriptors have been proposed to extract and to encode the information within a cuboid that is large enough to capture more informative motion. Mid-level descriptors are generally built on the low-level descriptors and possess no semantic meaning as them.

Most of the studies combine the spatio-temporal interest points to encode their relationship. Fathi and Mori [64] first extracted low-level optical flow information, and then used a method in the spirit of AdaBoost to build mid-level descriptors from them. In their scheme, low-level features served as weak classifiers, and a mid-level feature was a weighted combination of them where the weights were assigned by the AdaBoost algorithm proportional to the discrimination capability between pairs of action classes. For the same purpose, [65] and [66] used Random Decision Forests (RDF), differently, considered histogram of gradient and optical flow values as low-level features. For example, Hu *et al.* [66] proposed a two-level RDF framework. In the first RDF pass, the surviving features were used to compose mid-level descriptors, and the second pass was served for the classification from the mid-level features. They treated the space and the time domains separately, in other words, two types of mid-level features have been obtained covering the spatial structure and the temporal pairwise relationships.

Wang *et al.* [67] divided the video into a number of volumes of fixed size, and extracted spatio-temporal orientation energy as low-level features. Mid-level features are constructed by finding the connected volumes that have similar motion character-

istics. Similarly, Brendel and Todorovic [45] also took into account the homogeneity in terms of pixel intensity and motion at multiple scales, and oversegmented video to obtain $2D + t$ tubes.

In the context of BoW formalism, mid-level descriptors taking into account only pairwise spatio-temporal proximity [68] and a collection of the neighboring points [69] or modeling the spatio-temporal relationship of these neighboring points [70] can be considered as examples.

Since STIPs are more relevant to our focus area, we mentioned STIP-based mid-level descriptors in detail. Note that it is possible to see the instances that combine other types of low-level features in the literature, e.g., clustering trajectories into groups [71].

2.1.1.3. High-level descriptors. A high-level descriptor can be a human pose, an object, a semantic attribute, scene context or even a primitive human action. They also benefit from low-level or mid-level like features, but there is not a direct hierarchical relationship.

Pose estimation significantly assists the recognition of actions. An approach could be to represent each action by a set of key poses and search similar poses within a test sequence. For this reason, there is a considerable amount of work in the literature for pose estimation. For example, pictorial structures [72], deformable part-based models [73] and their variants [74, 75] are popularized specially for this task. However, estimating pose from still images or videos is very challenging per se, and occlusions have not yet been properly handled. In the cases where the body is occluded or in the close-up recordings where only a portion of the body is visible, poselets offer a good solution [76, 77]. A poselet captures a salient pattern corresponding to a particular part of the human pose under a given viewpoint, and it is obtained by tightly clustering parts both in appearance and configuration. On the other side, depth data is also likely to be helpful for pose estimation. As a case in point, it is really encouraging to witness

the real-time skeleton tracking algorithm developed by Shotton *et al.* [47] from depth sequences. This scheme results in 20 body (skeleton) joints per frame and proves to be adequate for tracking in the presence of appearance changes and varying background. Actually, it is much more favorable to combine these two modalities (intensity and depth) for complex activity recognition [78–80].

In addition to the pose information, object recognition, scene understanding or interaction context all give an insight into human action recognition problem. For this reason, integrating object-human context [48, 78, 79], scene context [78], person-person context [46, 78] or crowd context [81] have stirred great interest in the literature. Note that there exist remarkably good off-the-shelf codes that are publicly available for object detection and pose estimation. Many approaches [79, 82, 83] first run a bunch of object detectors and pose estimators, and then directly integrate these outcomes to their methodology.

A final type of high-level descriptors could be the semantic attributes. A semantic attribute is generally characterized by a text that describe a primitive human action (“talking on phone”, “riding”, “stretching arm”) [83, 84], a spatio-temporal movement of a body part (“single leg motion”, “torso twist”, “torso up-down motion”) [85], a physical property of the subject (“female”, “have backpack”, “wearing suit”) [86, 87] etc.

2.1.2. Review of Action Recognition Methods

Rapidly increasing interest on human action recognition has spawned a large number of different methodologies proposed in the literature. In this section, we only present a categorization of the methods based on their prominent characteristics, and specifically limited to relevance of our work. The taxonomy of the reviewed methods is given in Table 2.2.

2.1.2.1. BoW and its extensions. Local spatio-temporal features extracted from a few thousand cuboids provide a compact yet rich representation of the visual data. From

Table 2.2. Taxonomy of the reviewed approaches.

Human Action Recognition Methods	
Bag-of-Words and its extensions	Dollar <i>et al.</i> [54], Schuldt <i>et al.</i> [1], Niebles <i>et al.</i> [88,89], Laptev <i>et al.</i> [4], Ryoo and Aggarwal [90], Ta <i>et al.</i> [68], Bilinski and Br�mond [69], Li <i>et al.</i> [5], Ofli <i>et al.</i> [91], Lin <i>et al.</i> [92], Bettadapura <i>et al.</i> [93].
The use of Graphs	Ta <i>et al.</i> [94], Gaur <i>et al.</i> [95], Borzeshi <i>et al.</i> [96], Liu <i>et al.</i> [97], Yuan <i>et al.</i> [98], Chen and Grauman [82], Raja <i>et al.</i> [40], Yao and Fei-Fei [41], Yi and Pavlovic [99]. Brendel and Todorovic [45], Todorovic [100].
Sequential methods	Chaaroui <i>et al.</i> [101], Dyana and Das [102], Savarese <i>et al.</i> [103], Cuntoor <i>et al.</i> [104], Li <i>et al.</i> [105], Lv and Nevatia [106], Zhang <i>et al.</i> [107], Xia <i>et al.</i> [108], Alexiadis <i>et al.</i> [24], Raptis <i>et al.</i> [109], Bianco <i>et al.</i> [25], Wang <i>et al.</i> [110], �elebi <i>et al.</i> [111], Negin <i>et al.</i> [6], Ellis <i>et al.</i> [112].
The use of Trees	Mikolajczyk and Uemura [113], Jiang <i>et al.</i> [114], Choi <i>et al.</i> [81], Oshin <i>et al.</i> [115], Hu <i>et al.</i> [66], Miranda <i>et al.</i> [116].
Part-based structured models (Support Vector Machines)	Delaitre <i>et al.</i> [117], Tian <i>et al.</i> [118], Filipovych and Ribeiro [119], Sharma <i>et al.</i> [87], Liu <i>et al.</i> [85], Raptis and Sigal [120].

these features, one pioneering approach for recognition is Bag-of-Words (BoW) formalism that was originally developed for image classification [31]. BoW basically measures the occurrence of the local features with taking into account their membership to different action classes. More explicitly, it performs k-means clustering to find the representative cluster centers, and then convert the congregated local features around these centers into a histogram, and finally, compute the distance between these histograms [54] or feed them into a classifier, e.g., Support Vector Machines (SVMs) [1], for recognition.

BoW is very fast, simple and successful, if our only goal is to uncover the action. For more sophisticated goals –such as modeling of action co-occurrences or high level interpretation of the scene–, it is patently not appropriate. Its limitation is to neglect the structural information encoded in the video, e.g., the spatio-temporal relationship of the descriptors. To remedy its shortcomings, structural information has been imposed by many studies [4, 68, 69, 88–90]. For example, Laptev *et al.* [4] divided the space-time volume into several grids and constructed spatio-temporal histograms where each histogram is referenced with its grid position. The latter methods [88–90] focus more on the detection and localization of actions in the video rather than classification.

Some methods [89] extended the BoW approach with the topic models. In traditional BoW, we have a number of videos that contain visual words from a vocabulary and simply count the number of occurrences of each word within the video. The topic models are also in the same spirit, additionally, we introduce a latent topic variable associated with each occurrence of a word. BoW formalism assigns each word to a topic independently. This assumption can be constrained given that the topics generate words condition on the previous word or on some spatial-temporal consistency between the words. Note that, in the computer vision literature, words correspond to local spatio-temporal descriptors and topics to different action categories. A case in point, Niebles *et al.* [89] learned the distribution of the local descriptors and recognize multiple actions by using probabilistic Latent Semantic Analysis (pLSA) and Latent Dirichlet Allocation (LDA). In another work [88], they built a probabilistic model on the top of the BoW representations. Once the interest points are clustered, they model

the relative spatial arrangement of clusters as well the distribution of the appearance and position of interest points within each cluster in a probabilistic framework. Instead of building histograms from individual descriptors, Ryoo and Aggarwal [90] described the pairwise spatio-temporal relations between the local features based on a set of rules and transformed these relations into a 3D histogram. The similarity between two histograms was measured by kernel matching.

The concept of the BoW was also adapted to the depth data modality by Li *et al.* [5]. Bag of 3D points was used to characterize the salient poses, which were probabilistically modeled by so called “action graphs”. In a similar way, Ofli *et al.* [91] divided each sequence into small temporal segments, namely M-frame length subsequences. They took into account a number of joints having maximum variance within each subsequence and then used these most informative joints to construct histograms for action recognition.

The use of BoW formalism is still on-trend, however, in contrast to the previous efforts, we observe that the recent methods [92,93] are more characterized by enriched descriptors and intended for activity recognition in complex scenarios.

2.1.2.2. The use of Graphs. Graphs by nature offer a model for describing the spatio-temporal relationships. As we propose in Section 3.5, several methods [94,95] structure the interest points into a graph for matching. Ta *et al.* [94] have built graphs from proximity information by thresholding distances between interest points [54] in space and time. Similarly, Gaur *et al.* [95] modeled the relationship of spatio-temporal interest points [44] in a local neighborhood, i.e., they have built local feature graphs from small temporal segments instead of the whole video. These temporally ordered local feature graphs composed the so-called String of Feature Graphs (SFGs). Dynamic Time Warping (DTW) was utilized to measure the similarity between a model and a scene SFG. In [96], graphs were not directly matched in the spatio-temporal domain. Instead, a number of interest points is structured into a graph per frame, which is called as “frame graph”. Scores generated by matching a scene frame graph across a set of

model frame graphs were fed to Hidden Markov Models (HMMs) for classification.

Graph embedding has been used in several studies [97, 98]. For example, Liu *et al.* [97] used graphs to model the relationship between different components. While the nodes can be spatio-temporal descriptors, spin images or action classes, edges represent the strength of the relationship between these components, i.e. feature-feature, action-feature or action-action similarities. The graph was then embedded into a k -dimensional Euclidean space; hence correspondence was solved by spectral technique.

The approach proposed by Chen and Grauman [82] can be considered in this category as well. They first divided the test video into several subvolumes where each subvolume was scored by Support Vector Machines (SVMs) with respect to its relevance to a specific action class, in terms of the appearance and motion descriptors. Then, branch-and-cut algorithm searched the best scoring connected subvolumes, referred to as a “max-subgraph”, for each action class.

Probabilistic models have also been widely used in the literature [45, 100, 105–107]. Brendel and Todorovic [45] built weighted directed graphs from adjacency and hierarchical relationships of spatio-temporal tubes produced from oversegmenting the video. Given a set of training graphs, a graph can be generated by mixing these relationships through a permutation matrix which encodes the correspondences between the nodes. This generative probabilistic model was used to weight the representative edges and nodes as well as to eliminate the outliers. Once the weights were learned, a test graph was matched to the available model graphs by estimating the permutation matrix and assigned to the class of the closest model in the least squares sense.

Graphs are also used to model different types of data: still images [40, 41] and MoCap data [99]. Inspired from pictorial structures [72], Raja *et al.* [40] jointly estimate the pose and action using a pose-action graph where the nodes correspond to the five body parts, e.g., head, hands, feet, and the energy function is formulated based on both detected body parts and possible relative positioning of parts given the action

class. In [41], Yao and Fei-Fei structure salient points on the human body, namely skeletal joints, into a graph. Body joints are detected again by pictorial structures [72]; however, the main difference is that they use the method in [121] to recover depth information and, then attribute the 3D position information of the joints as well as appearance features to the nodes and 3D pose features to the edges. Similarly, in [99], graphs model the relationship of skeletal joints from MoCap Data.

2.1.2.3. Sequential methods. The linear and causal nature of the time dimension is frequently used to devise methods based on sequence alignment. The widely used Dynamic Time Warping (DTW) is an appropriate technique for matching two sequences in different lengths [101, 114]. Other example methods based on sequence alignment are chain graph models [105–107], trajectory matching with Gabor filters [102], correlation [103], learning salient state transitions by HMMs [96, 104] and modeling the evolution of silhouettes over time [122].

Chaaroui *et al.* [101] proposed a multi-view action recognition method based on learning a set of key poses in the spirit of BoW. They characterized each sequence frame by frame with the key poses, and aligned two different sequences by DTW. In [103], Savarese *et al.* measured the correlation of spatio-temporal (ST) patterns and constructed ST correlograms. Then, Probabilistic Latent Semantic Analysis (pLSA) was used to learn different human actions from these correlograms.

Li *et al.* [105] used a weighted directed graph where they assigned the nodes to frames with salient postures or silhouettes that were shared among different action categories. Here, edges encoded the transition probabilities between these postures. Given the training data, each action was learned and represented by several paths in the graphs. In the same vein, Lv and Nevatia [106] used salient postures for single-view action recognition in [106]. They modeled each action as well as linking different actions by chain graphs, called as Action Nets. Then, recognizing actions was boiled down to searching the most likely path by the Viterbi algorithm. Finally, to recognize actions performed by two interacting people, e.g. boxing, hand shaking etc., Zhang *et*

al. [107] also proposed a time-expanded chain graphical model.

Again in this category, depth-based action recognition methods can be coarsely examined along two main aspects. First, a skeleton is transformed into a feature representation robust to intrinsic properties such as body size, variance within the action class etc., and extrinsic conditions like camera position. Secondly, model skeleton sequences and a scene skeleton sequence are temporally aligned to each other and compared with respect to their kinematic details, e.g., speed, amplitude, position of skeleton joints, trajectory etc. The available approaches to temporal alignment typically utilize three methods or variations of them: Dynamic Time Warping (DTW) [25,111], Hidden Markov Models (HMM) [108] and correlation [6,24].

Xia *et al.* [108] represented each skeleton as a histogram of joint positions that was obtained by converting Cartesian coordinates of the joint positions to spherical coordinates and dividing the sphere enclosing the skeleton into several grids. Each action video was described by a sequence of poses, which were actually characterized by histograms, and Hidden Markov Models (HMMs) were used for classification.

One prominent robust skeleton representation was proposed by Raptis *et al.* [109]. Briefly, Principal Component Analysis (PCA) was applied on the torso joint positions. The resulting basis, called as “torso frame”, was used to estimate the orientation of the human body and accordingly to extract a set of features, e.g., limb joint angles with respect to the torso frame. These features were used to train cascaded correlation-based classifiers and the reliability of matching two sequences was evaluated by a distance metric based on DTW. Following this, Negin *et al.* [6] proposed a correlation-based method based on the torso frame features [109] and employed Random Decision Forests to learn the discriminative features per action class.

Introducing a skeleton joint weighting or a feature selection mechanism has been shown to improve the action recognition performance in previous studies [6,91,110,111]. Wang *et al.* [110] proposed a discriminative joint mining approach in which the goal was to select the best subset of joints that increases a confidence measure and at

the same time decreases an ambiguity measure. Their learning scheme resulted both in most informative joints and training sequences per each action class. They used Multiclass-Multiple Kernel Learning for classification where each kernel corresponds to a learned discriminative prototype. In another work, Çelebi *et al.* [111] used weighted DTW where they learned the weights of each joint during a training phase. Finally, Ellis *et al.* [112] used GentleBoost for selecting a set of best features corresponding to the informative joints and logistic regression for classification.

2.1.2.4. The use of Trees. Trees offer a representation for handling aforementioned structural properties by nature. However, we observe that they are often utilized for fast searching a query feature within a large vocabulary. For example, Mikolajczyk and Uemura [113] built a vocabulary from appearance-motion features and exploited randomized kd-trees to match a query feature to the vocabulary. A sequence was classified by accumulating the weights of the matched features over time. Similarly, Jiang *et al.* [114] used trees for assigning each frame to a shape-motion prototype, and aligned the two sequences of prototypes by DTW.

The most recent methods have been used Random Decision Forests (RDFs) or Random Ferns [115] for learning discriminative and representative features [6, 66, 81]. Choi *et al.* [81] proposed a method to analyze crowd behavior, e.g., “queuing in a line”, “talking”. Given the location, pose of each individual per frame and the velocity, they learned the most discriminating spatio-temporal regions by RDFs, and then modeled their spatio-temporal relationships by Markov Random Fields (MRF) for a specific activity class. Oshin *et al.* [115] extracted relative-motion-based descriptors, inspired from the Random Ferns concept. To capture the relative motion, each fern applied a series of binary tests to the input cuboids within a portion of the video, and these outputs were then aggregated and concatenated to form a histogram representation. They used Support Vector Machines (SVMs) for classification.

In the context of depth modality, Miranda *et al.* [116] used RDFs for action recognition. They first learned a set of key poses with multi-class Support Vector

Machines (SVMs), and then fed these poses to the RDFs. They characterized each pose (skeleton) by a modified version of the torso frame features [109].

2.1.2.5. Parts-based structured methods or SVMs. Although Deformable Part-based Models (DPMs) [73] are frequently used for pose estimation, we have encountered only two methods [117, 118] for action recognition in the literature. A few explanatory notes for DPMs are in order. DPM assumes that each object is composed of a root filter and several parts, as is common, each characterized by HoG features. Its key aspect is that, for a certain object category and viewpoint, it learns the discriminative parts as well as their best configuration by using Latent Support Vector Machines (LSVM) where the spatial position of the parts are regarded as the latent variables. To detect an object in an image, searching over all possible configurations of the parts is efficiently performed by dynamic programming technique. For example, in the case of pose estimation, parts can be considered as the limbs.

Delaitre *et al.* [117] used DPMs to recognize actions in still images and also showed that the DPMs perform better when combined with the Bag-of-Words approach. Up to our knowledge, the first method that extends DPMs to spatio-temporal domain was proposed by Tian *et al.* [118]. Their method retained exactly the same formulation as in [73] while generalizing the parts to capture spatio-temporal information. More explicitly, each part is represented by a volume, and the model learns the displacement of parts both in space and time.

Similarly, Filipovych and Ribeiro [119] integrated spatio-temporal configuration, appearance and human-object interactions into a parts-based model. Sharma *et al.* [87] recognized both attributes and actions from still images by parts-based loosely structured models. In other words, their model allows learning all possible variations, articulations from a large number of candidate parts while the highly structured models assume some priori information or initialization heuristics taking into account a few spatially constrained templates. Their formulation was basically based on linear SVM with hinge loss.

Besides part-based models, Latent SVMs have also been used to discriminatively learn different types of components. Liu *et al.* [85] treated the semantic attributes that describe the movements of the body parts as the latent variables. Raptis and Sigal [120] considered the key-frames featuring in a particular action sequence, each encoded with spatial features by the notion of poselets [76]. Their SVM formulation allows both learning the discriminative key-poses and their temporal relationships.

2.2. Benchmark Datasets

Since standard datasets are compulsory for experimental assessment and performance comparisons of different algorithms, for selection of proper solutions to practical applications, we briefly review the most relevant works that have been dedicated to generate standard testbeds for action detection and recognition systems.

Large spectrum of human action recognition applications creates need for different types of data collections, consequently, there are a plethora of datasets publicly available in the literature. These datasets have been extensively studied in a recent survey [123]. Here, we only recall the most prominent ones, and, additionally, multi-modal (RGB-D) datasets which were not addressed in [123], and introduce our own multi-modal dataset, so called *LIRIS Human Activities Dataset* [3].

The earliest datasets focus on simple periodic actions, e.g., running, walking, boxing, hand-clapping etc., with usually uniform background and static camera. Each video sequence includes a single person performing only one action (please refer to Figure 2.1). Typical examples are the *KTH* dataset [1] and the *Weizmann* dataset [124]. However, the performance on these datasets have been saturated. More complex actions and cluttered and dynamic backgrounds are covered by *CAVIAR* [125], *ETISEO* [126], *UIUC* [127] and *MSR* [128] action datasets, in which the recordings took place in shopping centers, hallways, metro stations or on the street.

More realistic datasets include videos of a series of actions or co-occurring actions performed by one or more people, namely, activities or events, from real-world scenes



Figure 2.1. *KTH* dataset is a typical example of “one actor, one action, uniform background” concept [1].

and generally collected for surveillance purposes. In this context, sample datasets that focus person-person interaction are *CAVIAR* [125], *BEHAVE* [129], *CASIA* [130], *i3DPost* [131], *TV Human Interactions* [132], *UT-Interaction* [133], *VideoWeb* [134] datasets. Several datasets feature crowd behavior, for instance *PETS 2009* [135], *ETISEO* [126], or group activities, for instance *BEHAVE* [129] and *Collective Activity* [136]. Person-object interactions were addressed by *CASIA* [130], where the object can be a car, door, telephone, baggage etc. Finally, daily activities in a natural kitchen environment are dealt by *University of Rochester Activities of Daily Living Dataset* [137] and the relatively challenging *TUM Kitchen* [138] dataset.

Different camera views render activity recognition problem much harder than the recognition with fixed viewing direction. An algorithm that is trained on a single view and works well across all view variations has not been feasible yet. Therefore, multi-view datasets include several simultaneous views for each scene: *BEHAVE* [129], *CASIA* [130], *CAVIAR* [125], *ETISEO*, [126], *IXMAS* [139], *i3DPost* [131], *MuHaVi* [140], *UCF-ARG* [141], *VideoWeb* [134] and *Multiple Cameras Fall* [15]. Aerial views are handled by *UCF Aerial* [142] and *UCF-ARG* [141].

Many datasets can be defined as “controlled” in that they are collected within the framework of a defined experimental setup. Uncontrolled databases, on the other hand, are collected without any constraints, and they are appropriately called sometimes “ac-

tions in the wild”. Recently, datasets collected from Youtube, dailymotion and broadcast television channels, movies, have aroused a lot of interest due to the huge amount of web sources in contrast to the laborious process of building controlled databases. Moreover, they provide more realistic and challenging videos. These datasets exhibit a large variability in background, camera view angle, camera motion, resolution, environmental conditions, number of subjects and the style of the acting. Prominent examples of wild datasets are *BEHAVE* [129], *HMDB51* [143], *Hollywood* [144], *Hollywood-2* [2], *Olympic Sports* [145], *TV Human Interaction* [132], *UCF Youtube* [146], *UCF Sports* and *UCF 50* [147]. Some instances are illustrated in Figure 2.2.



Figure 2.2. Recent research trends stress the recognition of the actions in the wild, some examples are illustrated from *Hollywood-2* dataset [2].

The recent introduction of low-cost depth cameras, e.g., Microsoft Kinect, Asus Xtion, Primesense Carmine and Capri etc., has created wide spread interest in activity recognition from depth sequences. On the one hand, depth data offers a solution to coping with complex colored background, camera view variations and camera motion etc., as well as alternative representations based on depth maps, point clouds or human skeletons. On the other hand, the downside is that the current technology only allows detecting objects within a short distance to the depth sensor, i.e., reliable results are obtained within 3-4 meters. As a result, there is a considerable amount of publicly available depth datasets, so-called “RGB-D” or “multi-modal” datasets, in the literature. Among these, *MSR Gesture 3D* [148], *MSRC-12 Kinect Gesture Dataset* [149] and the *ChaLearn Gesture Dataset* [150] focus on gesture recognition and body sign language understanding. Actions in the context of interacting with game consoles are handled in *MSR Action 3D Dataset* [148], basic actions such as jumping, hand clapping, stand up etc. in *Berkeley Multimodal Human Action Database (MHAD)* [151], 10 physical therapy exercises in *WokoutSU-10 Exercise Dataset* [6], and 12 different



Figure 2.3. *LIRIS Human Activities Dataset* [3] is one pioneering multi-modal dataset in that it contains complex human activities in a realistic setting. From left to right, gray scale images captured by Kinect, their corresponding depth images and color images from the Sony camcorder are illustrated respectively.

types of tennis shots in *THree Dimensional Tennis Shots (THETIS)* [152] dataset. Finally, the recognition of daily activities is addressed in the *Cornell Human Activities dataset (CAD)* [153], the *RGBD-HuDaAct dataset* [154] and the *MSR Daily Activity 3D dataset* [148]. These activities are usually individual activities, such as got to bed, get up, drink water, eat meal, answer phone, enter a room, lie down on a sofa, play guitar, write on a whiteboard etc., in a natural home environment.

In the scope of multi-modal datasets, we collected *LIRIS Human Activities Dataset* [3] that is particularized for recognizing complex activities in a realistic office environment. These activities are ranging from individual activities including human-object interactions, e.g., pick up/put down an object, unlock a door and then enter, leave a baggage, to human-human interactions or group activities, e.g., hand shake, discussion, give an object to another. As shown in Figure 2.3, the dataset was shot in two settings: (i) a dynamic Kinect camera installed on a mobile robot; (ii) a static Sony consumer

camera. Its comparatively difficult aspects stem from the camera movements (a large variety of camera view angle), activities running in the background and the same type of activities occurring in different contexts, for example, discussion of several people on the white board or around the table.

In Table 2.3, we summarize the highlights of the RGB-D or multi-modal datasets.

Table 2.3. Overview of multi-modal datasets for human activity recognition.

Dataset	Contents	Highlights
MSR Action 3D [148]	10 subjects, 567 samples, 20 actions for interacting with game consoles: arm wave, forward kick, tennis serve etc.	Annotation: tracked human skeleton joints.
MRSC-12 [149]	30 subjects, 594 samples, 12 gestures: (i) iconic gestures: shoot a pistol, throw an object, kick etc.; (ii) metaphoric gestures: start music, raise volume etc.	Annotation: tracked human skeleton joints.
WSU-10 [6]	15 subjects, 1500 samples, 10 physical therapy exercises: lateral stepping, freestanding squats, oblique stretch etc.	Annotation: tracked human skeleton joints.
MHAD [151]	12 subjects, 660 samples, 11 basic actions: jump in place, punch, sit down then stand up etc.	Source: 5 different modalities (motion capture system, multi-view stereo vision camera arrays, Kinect cameras, wireless accelerometers and microphones).
MSR Daily Activity 3D [148]	10 subjects, 320 samples, 16 activities: read book, write on a paper, use vacuum cleaner etc.	Annotation: tracked human skeleton joints (RGB channel and depth channel are not strictly synchronized).
CAD-60 [153]	4 subjects, 60 samples, 12 daily activities: brush teeth, chop, relax on couch etc.	Annotation: tracked human skeleton joints.
CAD-120 [153]	4 subjects, 120 samples, 10 complex activities: take medicine, microwave food, clean objects etc.	Annotation: sub-activity labels such as opening, placing, object affordance labels such as drinkable, closeable, tracked human skeleton joints and tracked object bounding boxes.
RGB-D Human Daily Activities [154]	30 subjects, 1189 samples, 12 daily activities: going to bed, eating meal, taking off the jacket etc.	Annotation: activity label.
LIRIS Human Activities [3]	18 subjects, 461 samples, 10 complex activities: discussion, give an item to another, unlock a room and enter etc.	Source: 2 different modalities (Kinect camera mounted on a mobile robot and a consumer camcorder). Annotation: video activity tag and tracked human bounding boxes with the activity label.

3. SPATIO-TEMPORAL HYPER-GRAPH MATCHING

Many computer vision problems have been formulated in terms of graphs and their associated algorithms, since graphs provide a structured, flexible and powerful representation for visual data. This representation has been successfully used in problems such as tracking [155, 156], object recognition [157–160], object categorization [161], and shape matching [162, 163].

Consider a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ in a typical computer vision problem. It consists of a set of nodes \mathcal{V} with associated geometric and appearance features, and of edges \mathcal{E} that represent structural relationships between nodes. Graph matching problem is basically searching the best correspondence between two graphs, i.e., the one that represents the model - a model graph \mathcal{G}^M - and the other one that represents the scene - a scene graph \mathcal{G}^S . In this thesis, we formulate the graph matching problem as the following generic energy-minimization problem [159]:

$$E(x) = \sum_i U(x_i) + \sum_{(i,j) \in \mathcal{E}} D(x_i, x_j) \quad (3.1)$$

where $x = \{x_i\}$ is a set of discrete variables representing an assignment between two sets of nodes, a model set and a scene set to which the model set is tentatively assigned. In this formulation, the energy $E(\cdot)$ is a non-negative scalar value which is lower for likely assignments respecting certain invariances and geometric transformations, whereas it results in higher values for unrealistic and unlikely assignments. Other alternative notations use matrices, but they are mathematically equivalent (e.g. [164, 165]).

Here, the $U(\cdot)$ term denotes the distance between intrinsic properties of nodes. These properties represent most frequently appearance information such as SIFT [166], histograms of gradient and optical flow [44], shape contexts [167] etc. Therefore, $U(\cdot)$ refers to a distance function on these properties between the model and scene. $D(\cdot)$, on the other hand, addresses the similarity of node pairs. It models the geometric

deformation of the assignment with respect to some invariance.

There are two drawbacks of considering only pairwise edges. It restricts geometrical coherence constraints to distance measures, and edge pairs are not invariant to scale changes. A more general case involving n -tuples of nodes, namely, hyper-graph matching, was proposed in the context of object recognition by Zass and Shashua [168]. A hyper-graph is a generalization of graphs allowing for edges (*hyper-edges*, strictly speaking) to connect any number of vertices, typically more than two. In this thesis, we consider 3-tuples of nodes. The energy function becomes an extension of the classical formulation 3.1, but where inter-nodal distortion is handled through a ternary term that enables scale-invariant matching, in the spirit of [165, 168]:

$$E(x) = \sum_i U(x_i) + \sum_{(i,j,k) \in \mathcal{E}} D(x_i, x_j, x_k). \quad (3.2)$$

Despite its popularity and effectiveness, the computation of graph matching still remains a challenging task. In fact, its computational complexity have made its use on data having a large number of nodes, e.g., video, intractable in practice. Useful formulations as in Equations 3.1 and 3.2 are known to be NP-hard combinatorial problems [159]. While the graph isomorphism problem is conjectured to be solvable in polynomial time, it is known that subgraph isomorphism - exact subgraph matching - is NP-complete [169]. Solutions for practical graphs therefore rely on approximations or heuristics.

In this thesis, we propose a novel method for matching hyper-graphs embedded in space-time focused on action recognition applications in videos; in particular, on localizing and recognizing human actions in video or depth sequences. In this context, we show that the complexity of the matching problem can be reduced to a tractable level and the global minimum can be calculated by taking into account special properties of the time domain, i.e., causality, linear order of time and one-to-one mapping of time instants. Furthermore, we propose an approximate graphical structure specifically

designed for the spatio-temporal data and derive an exact minimization algorithm for this special structure. In other words, we show that a better solution can be obtained by matching approximate data models instead of using an approximate matching algorithm applied to the complete data model.

3.1. Related Literature on Graph Matching

Since the general problem is NP-hard [159], a great effort has gone to optimization of graph matching algorithms in the machine learning community. The extensive research on practical solutions to graph matching can be analyzed under different perspectives. One of the most common classifications is “exact matching vis-à-vis inexact matching” [170]. Exact matching tries to find a strictly structure-preserving correspondence between the two graphs (e.g., graph isomorphism) or at least between their subparts (e.g., subgraph isomorphism), while inexact matching allows compromises in the correspondence principle by admitting structural deformations up to some extent. This work will be more to analyze the available methods in terms of their solution guarantees, namely “approximate solution vis-à-vis exact solution”, whether they perform exact or inexact matching. Below, we briefly review the methods relevant to our work as illustrated in Figure 3.1.

Approximate solution. Approximate matching algorithms do not guarantee global optimum, as the energy can get trapped in some local minimum. Many different types of approaches have been proposed to approximate the solution of the graph matching problem. Among these, spectral methods [162, 165] study the similarities between the eigenstructures of the adjacency or Laplacian matrices of the graphs. In particular, Duchenne *et al.* [165] generalized the spectral matching method from the pairwise graphs presented in [162] to hyper-graphs by using a tensor-based algorithm to represent affinity between feature tuples, which is then solved as an eigen-problem on the assignment matrix. More explicitly, they solved the relaxed problem by using a multi-dimensional power iteration method, and obtained a sparse output by taking into account l_1 -norm constraints instead of the classical l_2 -norm. Leordeanu *et al.* [171] made an improvement on the solution to the integer quadratic programming problem

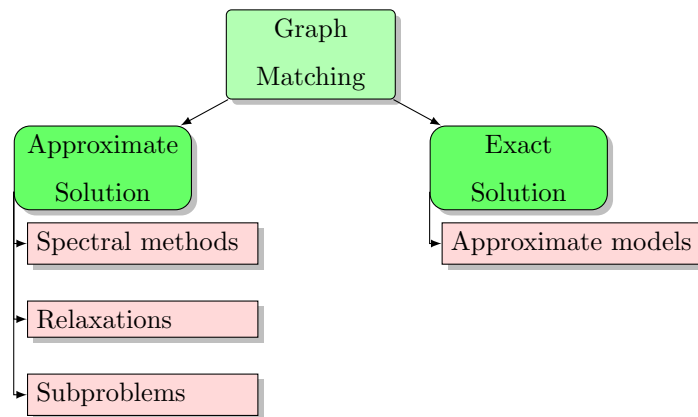


Figure 3.1. Classification of graph matching algorithms pertinent to our perspective.

Our proposed method falls into the class of exact solution-approximate models.

in [165] by introducing a semi-supervised learning approach. In the same vein, Lee *et al.* [172] approached this problem via the random walk concept.

Some methods solve a relaxation of the original combinatorial problem. Zass and Shashua [168] presented a soft hyper-graph matching method between sets of features that proceeds through an iterative successive projection algorithm in a probabilistic setting. They extended the Sinkhorn algorithm [173], which is used for soft assignment in combinatorial problems, to obtain a global optimum in the special case when the two graphs have the same number of vertices and an exact matching is desired. They also presented a sampling scheme to handle the combinatorial explosion due to the degree of hyper-graphs. Zaslavskiy *et al.* [174] employed a convex-concave programming approach to solve the least-squares problem over the permutation matrices. Recall that a permutation matrix encodes one possible correspondence between the vertices of two graphs. More explicitly, they proposed two relaxations to the quadratic assignment problem over the set of permutation matrices which results in one quadratic convex and one quadratic concave optimization problem. They obtained an approximate solution of the matching problem through a path following algorithm which tracks a path of local minimum by linearly interpolating convex and concave formulations.

Another approach is to decompose the original matching problem into subprob-

lems, which is then solved with different optimization tools [159, 175]. Lin *et al.* [176] first determined a number of subproblems where each subproblem is characterized by local assignment candidates, i.e., by plausible matches for model and scene local structures. For an example in action recognition domain, these local structures can correspond to the human body parts. Then, they built a candidacy graph representation by taking into account these possible candidates on a layered (hierarchical) structure and formulated the matching problem as a multiple coloring problem. Finally, Duchenne *et al.* [161] extended one dimensional multi-label graph cuts minimization algorithm to images for optimizing the Markov Random Fields (MRFs).

The methods described above more focus on object recognition problem in still images or in 3D as practical application. We have encountered only several previous works that literally applied graph matching to human action recognition problem in video sequences. As mentioned in Chapter 2, [94] and [95] structured the spatio-temporal interest points into a graph. Ta *et al.* [94] have built hyper-graphs from proximity information by thresholding distances between interest points both in space and time, and match a scene graph to a model graph by using the algorithm of Duchenne *et al.* in [165]. Similarly, Gaur *et al.* [95] modeled the relationship of spatio-temporal interest points within small temporal segments and obtained temporally ordered graphs. Then, a model graph sequence and a scene graph sequence were aligned by Dynamic Time Warping where the distance between two graphs was formalized by the method of Leordeanu and Hebert in [162]. Differently, Brendel and Todorovic [45] formulated the graph matching problem as a weighted least squares problem on the set permutation matrices. Recall that they built directed graphs from adjacency and hierarchical relationships of spatio-temporal regions and learned the representative nodes and edges in the weighted least squares sense from a set of training graphs.

Exact solution. Truly exact minimization algorithms always find a solution that is the global minimum of the energy function. On the other hand, they require high computational time in general. In order to achieve matching in polynomial time, one approach, *introduced in this chapter*, is to approximate the data model (graphical structure) as opposed to applying an approximate matching algorithm to the complete

data model. This can be achieved by filtering out the unfruitful portion of the data before matching. As a case in point, a method for object recognition has been proposed by Caetano *et al.* [177] which approximates the model graph by building a k-tree randomly from the spatial interest points of the object. Then, matching was calculated using the classical junction tree algorithm [178] known for solving the inference problem in Bayesian Networks.

Regarding the previous works on graph matching and action recognition, we can highlight the commonalities and the novelties of our proposed approach as follows.

- The nodes of our graphs correspond to spatio-temporal interest points in video sequences (Section 3.5) or skeleton joints in depth sequences (Chapter 4), and the neighborhood relationships are derived from proximity information as in [94, 95].
- Our formulation also tackles the time warping problem and aligns two sequences.
- The proposed method enables both localization and recognition of human actions in a query video as [45, 94, 95]. Localization corresponds to matching an action model point set within a usually much larger scene point set, and to generate a matching score for classification.
- Unlike the previous works [45, 94, 95], we do not search approximate solutions, matching is done with an exact minimization algorithm.
- A tractable exact minimization algorithm is derived by assuming realistic time-domain constraints. We show that, under these constraints, the hyper-graph matching problem –formulated in Equation 3.2– can be efficiently solved in the spirit of dynamic programming technique. In this case, the complexity of the correspondence problem can be bounded by a small exponent.
- We introduce a graphical structure which spreads over the spatio-temporal domain. To further reduce the complexity, we approximate it by proposing three different graph building strategies, each assumes very few interest points per frame.

Notice that we are using the terms “approximate matching - approximate solution” and “exact solution - exact minimization” interchangeably within this chapter.

The rest of the chapter is organized as follows: Section 3.2 formulates the graph matching problem for spatio-temporal data. Section 3.3 discusses the temporal properties of the action data and proposes an optimal space-time matching algorithm taking advantage of these properties. In Section 3.4, we introduce the structuring of model sequences into a graph and derive an algorithm which further reduces the computational complexity of the matching algorithm. Subsequent Section 3.5 and Chapter 4, make clear which components need be adapted and how for two specific tasks, namely, action recognition from video-sequences and analysis of depth sequences.

3.2. General Problem Formulation

In our work, we formulate the problem as a particular case of the general correspondence problem between two point sets. The objective is to assign points from the model set to points in the scene set, such that geometrical invariance is satisfied. We solve this problem through a global energy minimization procedure, which operates on a hyper-graph constructed from the model point set. The M points of the model are organized as a hyper-graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where \mathcal{V} is the set of nodes (corresponding to the points) and \mathcal{E} is the set of edges. From now on we will call (in loose language) hyper-graphs “graphs” and hyper-edges “edges”. The edges \mathcal{E} in our graph connect sets of three nodes, thus triangles. The notation used in this chapter is summarized in Table 3.1.

While our method requires the points in the model video to be structured into a graph, this is not necessarily so for the points in the scene video. In other words, the scene data consists of arbitrary number of space-time interest points per frame, without any superimposed organization. However if desired, any prior structural information available on the scene data can be integrated easily in the minimization framework. In this case, the problem reduces to a classical graph-matching problem. Our formulation is therefore more general but can also deal with graph matching.

We illustrate the graph matching problem in Figure 3.2. Notice that the mapping is neither subjective as not all scene nodes need to be assigned, nor injective since more

Table 3.1. Definitions and notations used throughout this chapter.

Symbol	Definition
p_i	Space-time position of a point : [$p_i^{\langle x \rangle}$ $p_i^{\langle y \rangle}$ $p_i^{\langle t \rangle}$]
f_i	Local appearance feature vector of a point p_i
$\langle m \rangle, \langle s \rangle$	Super-script indicators for “model” and “scene”
M, S	Number of model and of scene nodes; $S \gg M$;
$\overline{M}, \overline{S}$	Number of model and of scene frames;
\overline{M}_i	Number of model points per frame i ,
ϵ	Dummy value (node is assigned)
$\lambda_1, \lambda_2, \lambda_3$	Weighting parameters
W^d	Penalty for dummy assignment
W^t	Maximum penalty for temporal distortion
T^t	Maximum allowable time difference
$x_i = j$	Variable : i th model node is assigned to j th scene node
x	The whole set of assignments: $x = \{x_i\}, i = 1..M$
Terms	
$U(\cdot)$	Euclidean distance between a model and a scene appearance feature vector
$D(\cdot, \cdot, \cdot)$	Space-time geometric distortion between two triangles
$D_g(\cdot, \cdot, \cdot)$	Euclidean distance between sets of angles
$D_t(\cdot, \cdot, \cdot)$	Truncated time differences
$\Delta(i, j)$	Time difference between two pairs of nodes
$\phi^{\langle \cdot \rangle}(i, j, k)$	Angle with respect to the point j
When variables x_i are split into z_i and $y_{i,l}$ (Section 3.3)	
$z_i = j$	Variable : i th model frame is assigned to j th scene frame
$y_{i,l} = k$	Variable : l th node of i th model frame is assigned to k th node of z_i th scene frame
Graphical structure	
\mathcal{G}	Generic hyper-graph
\mathcal{V}	Set of nodes
\mathcal{E}	Set of edges
\mathcal{E}^i	Set of edges with latest time instant i
\mathcal{R}^i	<i>Reach</i> of frame i : set of edges which reach into the past of frame i and which are also part of i or of its future
\mathcal{X}^i	Set of all variables z_p or $y_{p,l}$ involved in the edges of of the reach \mathcal{R}^i

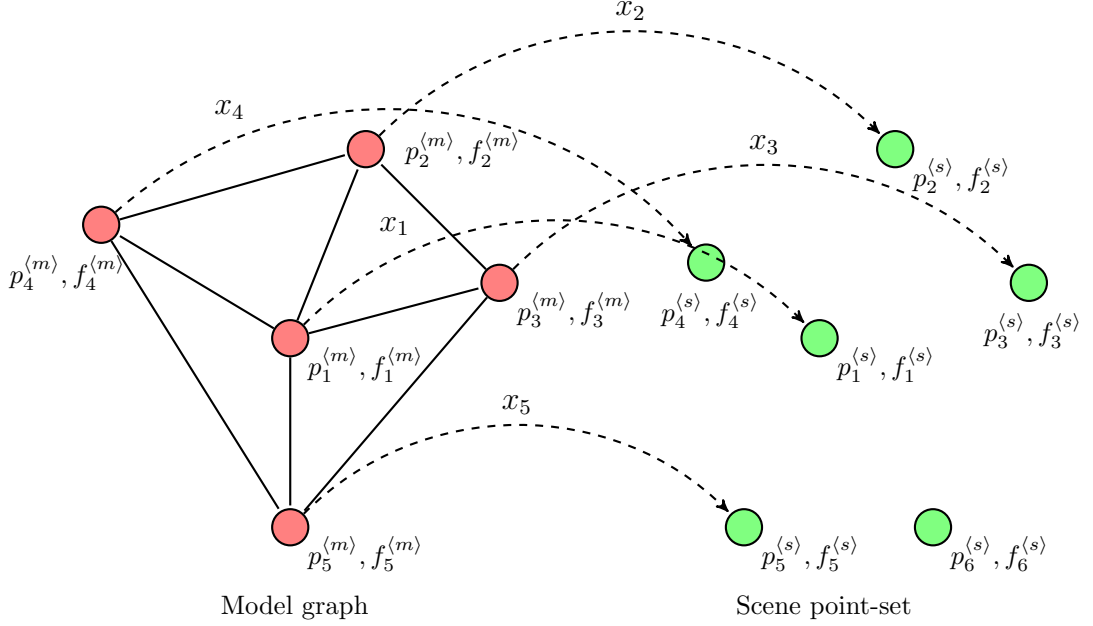


Figure 3.2. Illustration of graph matching problem in our formulation.

than one model point can be assigned to a scene node.

Each point i in the two sets (model and scene) is assigned a space-time position $p_i = [p_i^{(x)} \ p_i^{(y)} \ p_i^{(t)}]^T$, and a feature vector f_i that describes the appearance in a local space-time neighborhood around this point. When necessary, we will distinguish between model and scene variables by the superscripts $\langle m \rangle$ and $\langle s \rangle$: $p_i^{(m)}$, $f_i^{(m)}$, $p_i^{(s)}$, $f_i^{(s)}$ etc. Note that symbols in superscripts enclosed in angle brackets $\langle \cdot \rangle$ are not numerical indices, they are mere symbols indicating a category.

Each node i of the model graph is assigned a discrete variable x_i , $i = 1..M$, which represents the mapping from the i th model node to some scene node, and can take values from $\{1 \dots S\}$, where S is the number of scene nodes. We use the shorthand notation x to denote the whole set of map variables x_i . A solution of the matching problem is given through the values of the x_i , that is, through the optimization of the set x , where $x_i = j$, $i = 1..M$, is interpreted as model node i being assigned to scene node $j = 1..S$. To handle cases where no reliable match can be found, e.g., due to occlusions, an additional dummy value ϵ is admitted, which semantically means that

no assignment has been found for the given variable ($x_i = \epsilon$). Note that the use of dummy value can be regarded as an additional node for the scene graph.

Each combination of assignments x evaluates to a figure of merit in terms of an energy function $E(x)$, which will be given below. In principle, the energy should be lower for assignments that correspond to a realistic transformation from the model image to the scene image, and it should be high otherwise. We search for the assignments that minimize this energy.

In higher order matching, typically, hyper-edges connect 3 nodes, which allows to formulate geometrical constraints between pairs of triangles. In addition to feature similarity, geometrical similarity of point triples can be measured in terms of angles of the respective triangles; notice that this measure is scale invariant. Our proposed energy function is as follows:

$$E(x) = \lambda_1 \sum_i U(x_i) + \lambda_2 \sum_{(i,j,k) \in \mathcal{E}} D(x_i, x_j, x_k) \quad (3.3)$$

where U is a data-attached term taking into account feature distances, D is the geometric deformation between two space-time triangles, namely the cost for assigning a scene node triple to a model node triple, and λ_1 and λ_2 are the weighting parameters. For convenience, dependencies on all values over which we do not optimize have been omitted. U is defined as the Euclidean distance between the appearance features of assigned points in the case of a candidate match, and it takes a penalty value W^d for the dummy assignments:

$$U(x_i) = \begin{cases} W^d & \text{if } x_i = \epsilon, \\ \|f_i^{\langle m \rangle} - f_{x_i}^{\langle s \rangle}\| & \text{else.} \end{cases} \quad (3.4)$$

The D term separately handles the internal angles of node triples and time location differences between node pairs. Since our data is embedded in space-time, angles

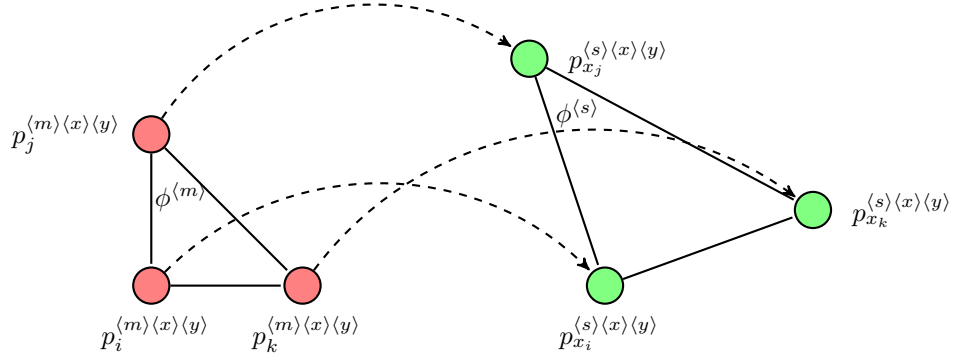


Figure 3.3. Illustration of calculating the geometric deformation, $D_g(\cdot)$.

include a temporal component that is not related to scale changes induced by zooming.

We therefore split the geometry term D into a temporal deformation term D^t and a spatial geometric deformation term D^g :

$$D(x_i, x_j, x_k) = D^t(x_i, x_j, x_k) + \lambda_3 D^g(x_i, x_j, x_k) \quad (3.5)$$

where the temporal distortion D^t is defined as truncated time differences in terms of the number of frames over two pairs of nodes of the triangle:

$$D^t(x_i, x_j, x_k) = \begin{cases} W^t & \text{if } \Delta(i, j) > T^t \vee \Delta(j, k) > T^t, \\ \Delta(i, j) + \Delta(j, k) & \text{else} \end{cases} \quad (3.6)$$

$$\Delta(i, j) = |(p_i^{\langle m \rangle \langle t \rangle} - p_j^{\langle m \rangle \langle t \rangle}) - (p_{x_i}^{\langle s \rangle \langle t \rangle} - p_{x_j}^{\langle s \rangle \langle t \rangle})|. \quad (3.7)$$

First, $\Delta(i, j)$ is the time interval differences in the assignment of model node pair (i, j) to scene node pair (x_i, x_j) , and T^t bounds the node pair differences in time. The time warping term penalizes the discrepancy in the extent of time between model node pairs and the corresponding scene node pairs. The node pairs in the model should not

be too close or too far from each other, and likewise for the scene node pairs.

Secondly, D^g is defined over differences of angles:

$$D^g(x_i, x_j, x_k) = \left\| \begin{array}{l} \phi^{\langle m \rangle}(i, j, k) - \phi^{\langle s \rangle}(x_i, x_j, x_k) \\ \phi^{\langle m \rangle}(j, i, k) - \phi^{\langle s \rangle}(x_j, x_i, x_k) \end{array} \right\|. \quad (3.8)$$

Here, $\phi^{\langle m \rangle}(i, j, k)$ and $\phi^{\langle s \rangle}(x_i, x_j, x_k)$ denote the angles subtended at point j and x_j for, respectively, model and scene triangles indexed by (i, j, k) and (x_i, x_j, x_k) as illustrated in Figure 3.3. It should be noted that the nodes of the space-time triangles are themselves spatial triangles within frames. In other words, our geometric deformation is measured in terms of the geometrical distance between triangles only in spatial domain which ensures robustness against camera zooming in contrast to Ta *et al.* [94].

3.3. Spatio-Temporal Matching

3.3.1. Properties of Spatio-Temporal Data

Our data structure is embedded in space-time. We assume the following commonly accepted properties of space-time to derive an efficient algorithm:

Assumption 3.1. *Causality.* Each point in the two sets (i.e., model and scene) lies in a 3-dimensional space: $[p_i^{\langle x \rangle} p_i^{\langle y \rangle} p_i^{\langle t \rangle}]^T$. However, the spatial and temporal dimensions should not be treated in the same way. While objects (and humans) can undergo arbitrary geometrical transformations like translation and rotation, which is subsumed by geometrical matching invariance in our formulation, time series data (human movements) can normally not be reversed. In a correct match, the temporal order of the points should be retained, which can be formalized as follows

$$\forall i, j : p_i^{\langle m \rangle \langle t \rangle} \leq p_j^{\langle m \rangle \langle t \rangle} \Leftrightarrow p_{x_i}^{\langle s \rangle \langle t \rangle} \leq p_{x_j}^{\langle s \rangle \langle t \rangle}. \quad (3.9)$$

Let us recall that the superscript $\langle t \rangle$ stands for the time dimension, and it is not an index.

Assumption 3.2. *Temporal closeness.* Another reasonable assumption is that the extent of time warping between model and scene time axes must be limited. In other words, two points which are close in time must be close in both the model set and the scene set. This property can be used to further decrease the search space during subgraph matching. Since our graph is created from proximity information (time distances have been thresholded to construct the hyper-edges), this can be formalized as follows:

$$\forall i, j, k \in \mathcal{E} : |p_{x_i}^{\langle s \rangle \langle t \rangle} - p_{x_j}^{\langle s \rangle \langle t \rangle}| < T^t \quad \wedge \quad |p_{x_j}^{\langle s \rangle \langle t \rangle} - p_{x_k}^{\langle s \rangle \langle t \rangle}| < T^t. \quad (3.10)$$

This property is also addressed in Equations 3.6 and 3.7.

Assumption 3.3. *One-to-one mapping of time instants.* We assume that time instants cannot be split or merged. In other words, all points of a model frame (time instant) should be matched to points of one and only one scene frame, and vice versa.

$$\forall i, j : (p_i^{\langle m \rangle \langle t \rangle} = p_j^{\langle m \rangle \langle t \rangle}) \Leftrightarrow (p_{x_i}^{\langle s \rangle \langle t \rangle} = p_{x_j}^{\langle s \rangle \langle t \rangle}). \quad (3.11)$$

3.3.2. Matching

In this section, we derive an exact minimization algorithm. Assumption 3.3 implies that a correct sequence match is an injective map, that is, it consists of a collection of single model-frame-to-scene-frame matches. We therefore first reformulate the energy function in Equation 3.3 by splitting each variable x_i into two subsumed variables z_i and $y_{i,l}$, which are interpreted as follows: z_i , a frame variable, denotes the index of the scene frame that is matched to i th model frame. The number of model frames and scene frames are denoted as \overline{M} and \overline{S} , respectively. Each model frame i also possesses a number \overline{M}_i of node variables $y_{i,1}, \dots, y_{i,\overline{M}_i}$, where $y_{i,l}$ denotes the index of scene node that is assigned to l th node at frame i in the model graph. Note that

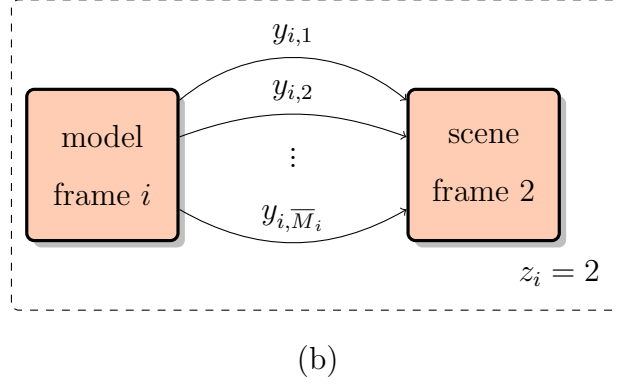
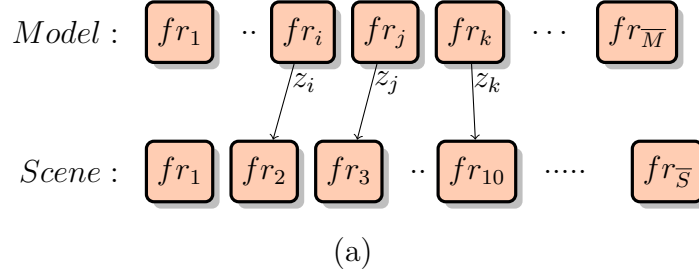


Figure 3.4. Separation of node assignments. (a) Frame variable set z_c denotes a triple assignment of model frames i, j, k to the scene frames; (b) Each frame variable z_i can assume \overline{M}_i node variables.

the number of possible values for variable $y_{i,l}$ depends on the value of z_i , since different frames may contain different number of nodes. The separation of node assignments is exemplified in Figure 3.4.

The objective remains to calculate the globally optimal assignment of all nodes of the graph, i.e., the optimal values for all variables z_i and $y_{i,l}, \forall i \forall l$. In other terms, a node of a given frame i is not necessarily matched to the (locally only) best fitting node in the frame-to-frame sense. This will be detailed in the rest of this section.

For convenience, we will also simplify the notation by representing a hyper-edge (the corresponding frame indices and node indices) as c and the corresponding variables as (z_c, y_c) ; in other words, (z_c) implies a triple assignment of model frames (say, i, j, k), to scene frames; and (y_c) implies the connection of the three model nodes to the three frames (z_c) . For ease of notation we also drop the parameters λ_1 and λ_2 which can be

absorbed into the potentials U and D .

The reformulated energy function is now given as:

$$E(z, y) = \sum_{(i,l) \in \overline{M} \times \overline{M}_i} U(z_i, y_{i,l}) + \sum_{c \in \mathcal{E}} D(z_c, y_c). \quad (3.12)$$

To summarize, we continue with the following variables and definitions:

- Each model frame is injectively assigned via $z_i = j$ to a scene frame from among $j = \{1, \dots, \overline{S}\}$.
- Nodes within each such possibly matched model frame to a scene frame are assigned to the nodes in the scene frame via $y_{i,l}$.
- Assumptions 3.1-3.3 are observed in the temporal assignments.

We now introduce a decomposition of the set of hyper-edges \mathcal{E} into disjoint subsets \mathcal{E}^i , where \mathcal{E}^i is the set of all hyper-edges which contain at least one node with temporal coordinate equal to i and no node has a higher (later) temporal coordinate. It is clear that the set of all possible sets \mathcal{E}^i forms a complete partition of \mathcal{E} , i.e. $\mathcal{E} = \bigcup_i \mathcal{E}^i$. We can now exchange sums and minima according to this partitioning:

$$\begin{aligned} \min_{z,y} E(z, y) = & \\ & \min_{z_1; y_{1,1}, \dots, y_{1, \overline{M}_1}} \left[\sum_{l=1}^{\overline{M}_1} U(z_1, y_{1,l}) + \sum_{c \in \mathcal{E}^1} D(z_c, y_c) + \right. \\ & \min_{z_2; y_{2,1}, \dots, y_{2, \overline{M}_2}} \left[\sum_{l=1}^{\overline{M}_2} U(z_2, y_{2,l}) + \sum_{c \in \mathcal{E}^2} D(z_c, y_c) + \right. \\ & \vdots \\ & \left. \left. \min_{z_{\overline{M}}; y_{\overline{M},1}, \dots, y_{\overline{M}, \overline{M}_{\overline{M}}}} \left[\sum_{l=1}^{\overline{M}_{\overline{M}}} U(z_{\overline{M}}, y_{\overline{M},l}) + \sum_{c \in \mathcal{E}^{\overline{M}}} D(z_c, y_c) \right] \dots \right] \right]. \end{aligned} \quad (3.13)$$

Hyper-edges have variable temporal spans, which makes it impossible to define a

recursion scheme with regular structure. We therefore define the concept of the *reach* \mathcal{R}^i of frame i , which consists of the set of edges which reach into the past of frame i and which are also part of i or of its future:

$$\mathcal{R}^i = \{c \in \mathcal{E} : [\min^{(t)}(c) < i] \wedge [\max^{(t)}(c) \geq i]\} \quad (3.14)$$

where $\min^{(t)}(c)$ and $\max^{(t)}(c)$ are, respectively, the minimum and the maximum temporal coordinate of the nodes of edge c . Note that, by definition $\mathcal{E}^i \subseteq \mathcal{R}^i$.

We also introduce the expression \mathcal{X}^i for the set of all variables z_p or $y_{p,l}$ involved in the reach \mathcal{R}^i :

$$\mathcal{X}^i = \{(z_p, y_{p,l}) : [\exists q, r : (p, q, r) \in c] \wedge [c \in \mathcal{R}^i]\}. \quad (3.15)$$

Finally, the set of variables \mathcal{X}^i that belongs to the frames before i is denoted as \mathcal{X}_-^i

$$\mathcal{X}_-^i = \{(z_p, y_{p,l}) \in \mathcal{X}^i : p < i\}. \quad (3.16)$$

Now, a recursive calculation scheme for Equation 3.13 can be devised by defining a recursive variable α_i which minimizes the variables of a given frame as a function of the reach variables before it as follows. For the inner most bracket we can set

$$\alpha_{\bar{M}}(\mathcal{X}_-^{\bar{M}}) = \min_{z_{\bar{M}}; y_{\bar{M},1}, \dots, y_{\bar{M}, \bar{M}}} \left[\sum_{l=1}^{\bar{M}} U(z_{\bar{M}}, y_{\bar{M},l}) + \sum_{c \in \mathcal{E}^{\bar{M}}} D(z_c, y_c) \right]. \quad (3.17)$$

For the next inner bracket, a brute force calculation would give

$$\alpha_{\bar{M}-1}(\mathcal{X}_-^{(\bar{M}-1)}) = \min_{z_{\bar{M}-1}; y_{\bar{M}-1,1}, \dots, y_{\bar{M}-1, \bar{M}-1}} \left[\sum_{l=1}^{\bar{M}-1} U(z_{\bar{M}-1}, y_{\bar{M}-1,l}) + \sum_{l=1}^{\bar{M}} U(z_{\bar{M}}, y_{\bar{M},l}) + \sum_{c \in \mathcal{E}^{\bar{M}-1} \cup \mathcal{E}^{\bar{M}}} D(z_c, y_c) \right].$$

The backward recursive expression can be obtained by using $\alpha_{\overline{M}}(\mathcal{X}_{-}^{\overline{M}})$:

$$\alpha_{\overline{M}-1}(\mathcal{X}_{-}^{(\overline{M}-1)}) = \min_{z_{\overline{M}-1}; y_{\overline{M}-1,1}, \dots, y_{\overline{M}-1, \overline{M}_{\overline{M}-1}}} \left[\sum_{l=1}^{\overline{M}_{\overline{M}-1}} U(z_{\overline{M}-1}, y_{\overline{M}-1,l}) + \sum_{c \in \mathcal{E}^{\overline{M}-1}} D(z_c, y_c) + \alpha_{\overline{M}}(\mathcal{X}_{-}^{\overline{M}}) \right]. \quad (3.18)$$

Finally, the general recursive formula can be given as

$$\alpha_i(\mathcal{X}_{-}^i) = \min_{z_i; y_{i,1}, \dots, y_{i, \overline{M}_i}} \left[\sum_{l=1}^{\overline{M}_i} U(z_i, y_{i,l}) + \sum_{c \in \mathcal{E}^i} D(z_c, y_c) + \alpha_{i+1}(\mathcal{X}_{-}^{(i+1)}) \right]. \quad (3.19)$$

It should be noted that Equation 3.19 is valid, if the following relation holds:

$$\mathcal{X}_{-}^{(i+1)} \subseteq \left(\mathcal{X}_{-}^i \cup \{z_i; y_{i,1}, \dots, y_{i, \overline{M}_i}\} \right). \quad (3.20)$$

Illustrative example of the recursion scheme. Figure 3.5 is a simple example illustrating how the recursion scheme works. The vertical blue bars correspond to the frame instances. Our model graph assumes five frame variables z_i , each associated with one or more node variables $y_{i,l}$. The recursion starts from the last frame $i = 5$ where we define the respective reach variable and the assignment variable sets as $\mathcal{R}^5 = \{(z_3, z_4, z_5)\}$, $\mathcal{X}^5 = \{(z_3, y_{3,1}), (z_4, y_{4,1}), (z_5, y_{5,1})\}$ and $\mathcal{X}_{-}^5 = \{(z_3, y_{3,1}), (z_4, y_{4,1})\}$. Thus, we obtain

$$\alpha_5(\mathcal{X}_{-}^5) = \min_{z_5, y_{5,1}} \left[U(z_5, y_{5,1}) + D(z_3, y_{3,1}, z_4, y_{4,1}, z_5, y_{5,1}) \right].$$

And the recursion continues as follows. For $i = 4$, we can set $\mathcal{R}^4 = \{(z_2, z_3, z_4), (z_3, z_4, z_5)\}$, $\mathcal{X}^4 = \{(z_2, y_{2,1}), (z_3, y_{3,1}), (z_4, y_{4,1}), (z_5, y_{5,1})\}$ and $\mathcal{X}_{-}^4 = \{(z_2, y_{2,1}), (z_3, y_{3,1})\}$, the recursion iterates by calculating α_4 from α_5 as below

$$\alpha_4(\mathcal{X}_-^4) = \min_{z_4, y_{4,1}} \left[U(z_4, y_{4,1}) + D(z_2, y_{2,1}, z_3, y_{3,1}, z_4, y_{4,1}) + \alpha_5(\mathcal{X}_-^5) \right].$$

Finally, for $i = 3$, we obtain

$$\alpha_3(\mathcal{X}_-^3) = \min_{z_3, y_{3,1}} \left[U(z_3, y_{3,1}) + D(z_1, y_{1,1}, z_2, y_{2,1}, z_3, y_{3,1}) \right. \\ \left. + D(z_2, y_{2,2}, z_2, y_{2,3}, z_3, y_{3,1}) + \alpha_4(\mathcal{X}_-^4) \right],$$

where $\mathcal{R}^3 = \{(z_1, z_2, z_3), (z_2, z_3, z_4), (z_3, z_4, z_5)\}$, $\mathcal{X}^3 = \{(z_1, y_{1,1}), (z_2, y_{2,1:3}), \dots, (z_5, y_{5,1})\}$ and $\mathcal{X}_-^3 = \{(z_1, y_{1,1}), (z_2, y_{2,1:3})\}$. Note that we can write these relations because Equation 3.20 is always satisfied.

3.3.3. Computational Complexity

The recursion starts at the last frame $i = \overline{M}$ and iterates by calculating α_i from α_{i+1} . At each step, a minimum is calculated over all variables of frame i for all possible values of the variables in \mathcal{X}_-^i . The computational complexity can thus be bounded by the *maximum* number of variables z and y in $(\mathcal{X}_-^i \cup \{z_i; y_{i,1}, \dots, y_{i, \overline{M}_i}\})$, which are denoted by $|\mathcal{X}_z^{i*}|, |\mathcal{X}_y^{i*}|$ in the following expression:

$$O\left(\overline{M} \cdot \overline{S}^{|\mathcal{X}_z^{i*}|} \cdot R^{|\mathcal{X}_y^{i*}|}\right) \quad (3.21)$$

where \overline{M} and \overline{S} are the number of model and scene frames, R is the maximum number of nodes per frame in the scene sequence. The complexity is thus very much lower than the complexity of the brute force approach, which is given by $O(MS^M|\mathcal{E}|)$. Let us recall that S is the total number of nodes in the scene and M is the total number of nodes in the model, i.e. $S \gg \overline{S}$ and $S \gg R$. Furthermore, both $|\mathcal{X}_z^i|$ and $|\mathcal{X}_y^i|$ are bounded and in fact quite small when the graph is constructed from proximity constraints. However, for general graphs, it is still too high for practical usage. The next section will introduce a special structure which further decreases complexity.

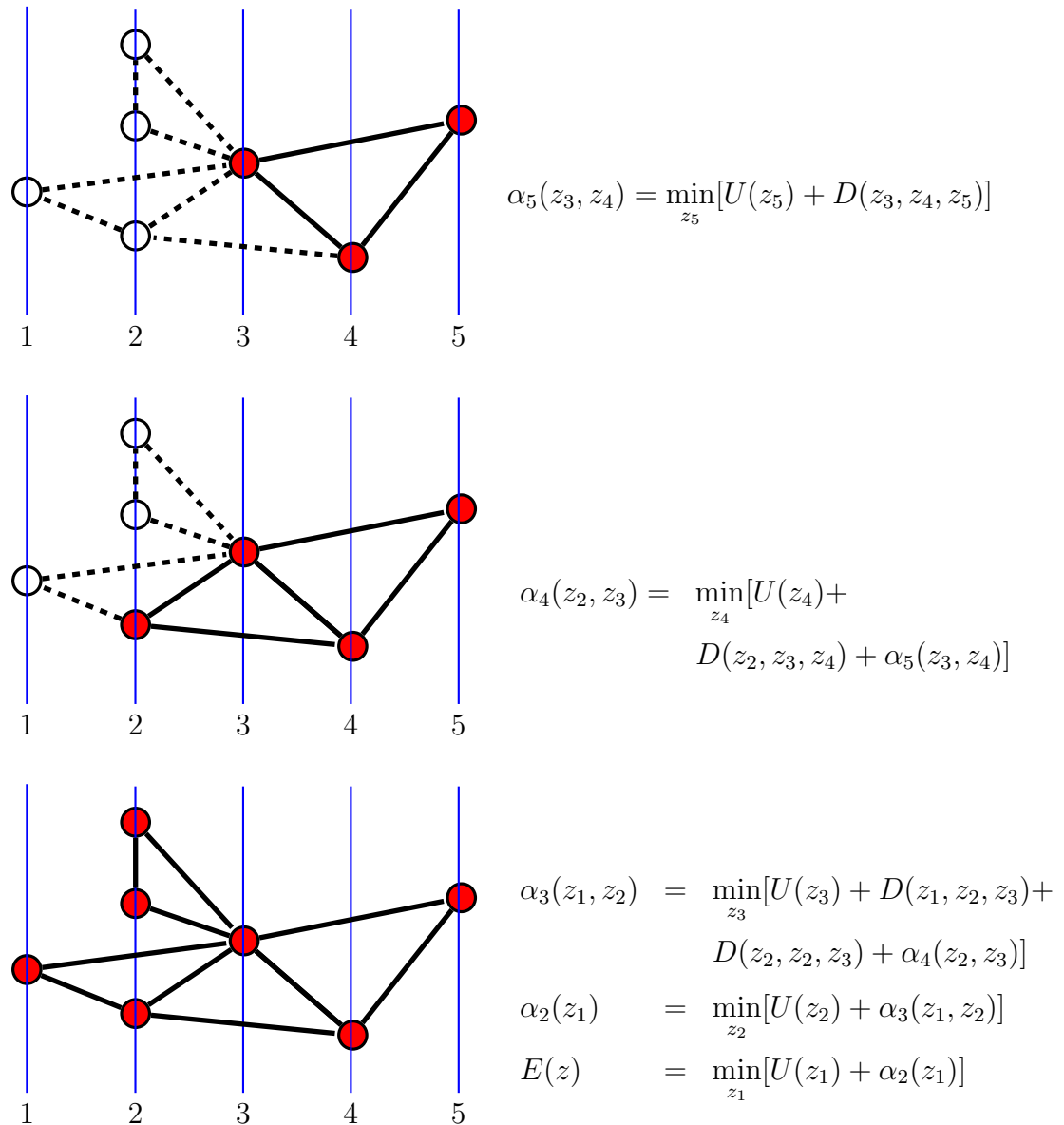


Figure 3.5. An example hyper-graph. Vertical blue bars correspond to the frames.

The recursion starts from the last frame $i = 5$ and iterates by calculating α_i from α_{i+1} where the evolution of the iteration is illustrated with solid lines. For simplicity, node variables $y_{i,l}$ are absorbed into the frame variables z_i . At each step, a minimum is calculated over all variables of frame i for all possible values of the variables in \mathcal{X}_- .

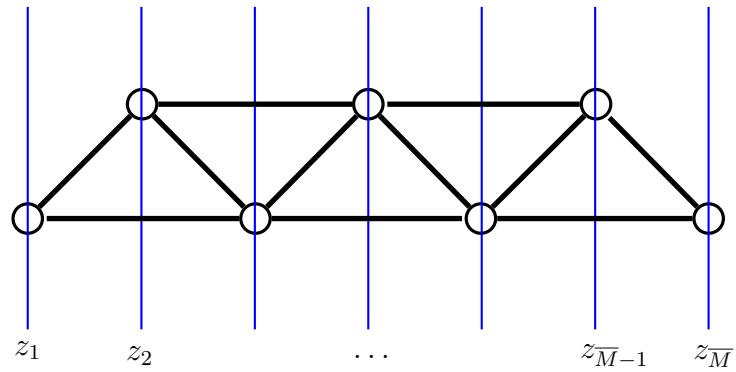
3.4. Graphical Structure

As discussed before, classical methods such as spectral approaches provide approximate solutions to the NP-complete graph matching problem since the exact minimization is infeasible. In this work, we advocate an alternative idea, which is to approximate the problem – the graphical structure in this case – and then to solve the newly formulated problem exactly. This is particularly appealing in point matching problems where the structure of the graph is less related to the description of the object per se, but rather more to the constraints in the matching process. We recall that the graphical structure is obtained from adjacency or proximity information, the description of the object can be approximated by respecting the properties of the spatio-temporal data (Section 3.3.1).

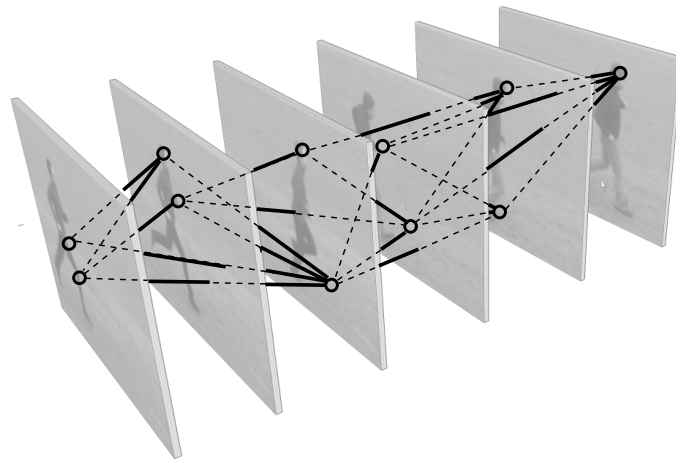
The approximation to the graphical structure is as follows. The graphical structure (the set \mathcal{E} of edges) is restricted by constraints on the combinations of temporal coordinates. The triplets (i, j, k) of temporal coordinates allowed in a hyper-edge are restricted to triplets of consecutive frames: $(i, j, k) = (i, i + 1, i + 2)$. Depending on the visual content of a video, there may be frames which do not contain any space time interest points, and therefore no nodes in the model graph. These empty frames are not taken into account when triplets of consecutive frame numbers are considered. One can encounter frames without STIPs more often in one video vis-à-vis another if the action is enacted more slowly in the former. Thus this skipping of frames is in fact instrumental for the implicit time warping of sequences.

This structure can be visualized by a meta graph, which contains a single node for each (non empty) frame and a hyper-edge connecting the triplets of consecutive frames, as seen in Figure 3.6a. The meta graph is planar and with triangular structure. Figure 3.6b shows a model graph which satisfies the restrictions described above. Note that each triangle in the meta graph (Figure 3.6a) corresponds to a set of triangles in the model graph (Figure 3.6b).

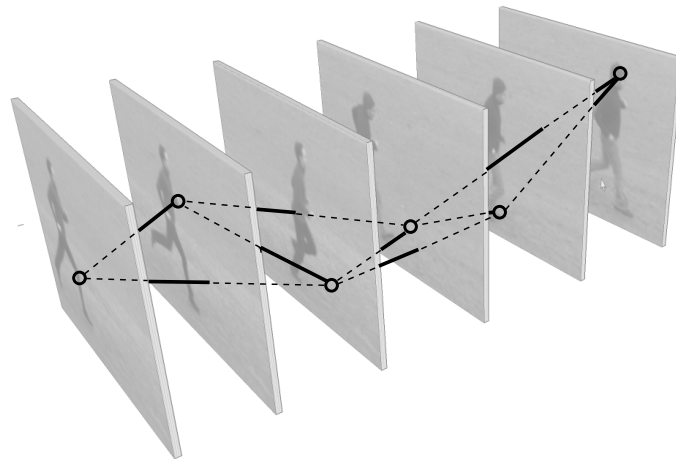
Given these restrictions, we propose three different model graphs and three dif-



(a)



(b)



(c)

Figure 3.6. Proposed graphical structure. (a) A special graphical structure designed for very low computational complexity: a second order chain. This meta-graph describes the restrictions on the temporal coordinates of model graphs; (b) a sample model graph satisfying the restrictions in (a); (c) a sample model graph limited to a single point per frame (the single-point-single-chain model).

ferent associated matching algorithms. They differ in the number of model nodes per frame and the way they are linked.

3.4.1. Single-chain-single-point Model

In this model, we keep only a single node per model frame by choosing the most salient one, e.g., the one with the highest confidence. However, no restrictions are applied to the scene frames which may contain an arbitrary number of points. In this case, the graphical structure of the meta graph is identical to the graphical structure of the model graph. An example of this model graph is given in Figure 3.6c. In particular, each model point i is connected to its two immediate frame-wise predecessors $i - 1$ and $i - 2$ as well as to its two immediate successors $i + 1$ and $i + 2$.

Given this special graphical structure, the matching problem can be solved as follows. The energy function can be simplified, since the couple of discrete variables $(z_i, y_{i,l}), l = 1$ can be simplified by dropping all variables $y_{i,l}$. The frame variable z_i is indeed sufficient to describe the assignment of the frame as well as the assignment of the point, which are one-to-one related. The neighborhood system of this simplified graph can be described in a very simple way using the index of the nodes of graph, similar to the dependency graph of a second order Markov chain:

$$E(z) = \sum_{i=1}^{\bar{M}} U(z_i) + \sum_{i=3}^{\bar{M}} D(z_i, z_{i-1}, z_{i-2}). \quad (3.22)$$

The reach of this structure is constant and consists of two edges only, as $\mathcal{R}^i = \{(z_{i-2}, z_{i-1}, z_i), (z_{i-1}, z_i, z_{i+1})\}$. The general recursive formula of the inference algorithm can be derived as

$$\alpha_i(z_{i-1}, z_{i-2}) = \min_{z_i} \left[U(z_i) + D(z_i, z_{i-1}, z_{i-2}) + \alpha_{i+1}(z_i, z_{i-1}) \right] \quad (3.23)$$

with the initialization

$$\alpha_{\overline{M}}(z_{\overline{M}-1}, z_{\overline{M}-2}) = \min_{z_{\overline{M}}} [U(z_{\overline{M}}) + D(z_{\overline{M}}, z_{\overline{M}-1}, z_{\overline{M}-2})]. \quad (3.24)$$

During the calculation of the trellis, the arguments of the minima in Equation 3.23 are stored in a table $\beta_i(z_{i-1}, z_{i-2})$. Once the trellis is completed, the optimal assignment can be calculated through classical backtracking:

$$\widehat{z}_i = \beta_i(z_{i-1}, z_{i-2}), \quad (3.25)$$

starting from an initial search for z_1 and z_2 :

$$(\widehat{z}_1, \widehat{z}_2) = \arg \min_{z_1, z_2} [U(z_1) + U(z_2) + \alpha_3(z_1, z_2)]. \quad (3.26)$$

The algorithm as given above is of complexity $O(\overline{M} \cdot \overline{S}^3 \cdot R)$. Recall that the total number of model and scene nodes are denoted by M and S , a trellis is calculated in an $M \times S \times S$ matrix, where each cell corresponds to a possible value of a given variable. The calculation of each cell requires to iterate over all S^2 possible combinations of its two successors.

Exploiting the assumptions on the spatio-temporal data introduced in Section 3.3.1, the complexity can be decreased further:

- *Assumption 3.1.* Taking causality constraints into account, we can prune many combinations from the trellis of the optimization algorithm. In particular, if we calculate possibilities in the trellis given a certain assignment for a given variable z_i , all values for its predecessors z_{i-1} and z_{i-2} must be necessarily *before* z_i , i.e., lower.
- *Assumption 3.2:* Similar as above, given a certain assignment for a given variable z_i , we will allow a maximum number of T^t possibilities for the values of the successors z_{i+1} , z_{i+2} , which are required to be *close*.

Thus, the expression in Equation 3.23 is only calculated for values (z_{i-1}, z_{i-2})

satisfying the following constraints:

$$\begin{aligned} |z_i - z_{i-1}| < T^t \quad \wedge \quad |z_{i-1} - z_{i-2}| < T^t \quad \wedge \\ z_i > z_{i-1} \quad \wedge \quad z_{i-1} > z_{i-2}. \end{aligned} \tag{3.27}$$

These pruning measures decrease the complexity to $O(\overline{M} \cdot \overline{S} \cdot T^{t^2} \cdot R)$, where T^t is a small constant measured in the number of frames and R is generally very small, so the complexity becomes linear on the number of points in the scene: $O(\overline{M} \cdot \overline{S})$.

Note that our formulation does not require or assume any probabilistic modeling. We used dynamic programming algorithm extended to include second-order dependencies. At a first glimpse it could be suspected that the single-point-per-frame approach could be too limited to adequately capture the essence of an action sequence. Experiments have shown, however, that the single chain performs surprisingly well. It should be noted again, that no restrictions have been imposed on the scene, in other words, none of the scene points have been eliminated.

3.4.2. Single-chain-multiple-points Model

Keeping multiple points per model frame and solving the exact minimization problem is of polynomial complexity, as has been proven in Section 3.3. However, the complexity is still too high even with the restrictions imposed by the meta-graph structure given in Figure 3.6a. We propose the following (partially) greedy algorithm, which separates frame assignments and node assignments into two consecutive steps:

- For each possible frame assignment z_i , we solve the node assignment variables $y_{i,l}$ with local geometric and appearance information extracted from the given frames.
- We calculate the solution for the frame assignment variables z_i by minimizing the energy function. In this step, the node assignment variables $y_{i,l}$ are considered constant.

The node assignments in the first step can be done by minimizing appearance information alone, i.e., for each model node we select a scene node having minimal feature difference. To make this assignment more robust, we do not take this decision individually per node, but we take the decision jointly for all nodes of each model frame. Let's assume that we select the most salient three points per each model frame. In this case, the criteria for the assignment is a weighted combination of appearance feature distances and geometrical deformation of each in-frame triangle formed by the model points.

The second step, which optimally aligns the frames given the pre-calculated node assignments, is equivalent to the matching algorithm of the single point model described in Section 3.4.1. However, the graphical structure on which the algorithm operates is now the meta graph (given in Figure 3.6a) and not the model graph (given in Figure 3.6b). In the multiple point case these two graphs are not identical. For this reason, the terms $U(\cdot)$ and $D(\cdot)$, originally defined on edges in the model graph, i.e., on triangles in space-time, are now defined on edges in the meta graph, i.e. on sets of triangles in space-time. They can be set as sums over the respective triangles in the set:

$$\begin{aligned} U(z_i) &= \sum_{l=1}^R U_l(z_i, y_{i,l}) \\ D(z_i, z_{i-1}, z_{i-2}) &= \sum_{c \in (i, i-1, i-2)} D_c(z_c, y_c). \end{aligned} \tag{3.28}$$

Here the expressions $U_l(\cdot)$ and $D_c(\cdot)$ are defined, respectively, as equivalent to $U(\cdot)$ in Equation 3.4, and as equivalent to $D(\cdot)$ in Equation 3.5.

The computational complexity of the global frame alignment step is identical to the single point model, apart from a small loop over the nodes of each frame. For example, in our case ($R = 3$), the complexity becomes $O(\overline{M} \cdot \overline{S} \cdot T^{t^2} \cdot R^3)$.

3.4.3. Multiple-chains Model

The model described in Section 3.4.2 has allowed taking into account multiple points per frame. In this case, we approximate the matching algorithm by separating

the set of discrete variables into two subsets and by solving for each subset independently. An alternative way is to approximate the model by separating the full graph into a set of independent model graphs, each one featuring a single point per frame, i.e., each one being of the type shown in Figure 3.6c. As in Section 3.4.1, the exact global solution of each individual chain is computed.

In this case, the definitions for $U(z_i)$ and $D(z_i, z_{i-1}, z_{i-2})$ are identical to the single-point model. The energy function to be minimized is a sum over the energy functions corresponding to the individual graphs (chains) as follows:

$$E^*(z) = \frac{1}{N} \sum_{k=1}^N E_k(z) \quad (3.29)$$

where N is the number of single chain models and $E_k(\cdot)$ is the k th single-chain-single-point energy function which is formulated in Equation 3.22. In other words, we match each single-chain-single-point model to a scene sequence and average the resulting N energies to reach a final decision. Therefore, the computational complexity increases linearly with N as compared to the single-chain model. Note that N is typically a small number.

Subsequent Section 3.5 and Chapter 4 present the experimental results for two specific problems, namely, action recognition from video-sequences and analysis of depth sequences.

3.5. Application to Video-based Action Recognition

In this section, we discuss the application of the proposed hyper-graph matching algorithm to the human action recognition problem in video sequences. Given a training set of model action graphs, our strategy is to calculate the matching energy of a given scene sequence with each model graph, and infer the action class by the nearest-neighbor classification rule. In the sequel, we first present the preprocessing steps, namely, we describe the detection of the spatio-temporal interest points, description of

their shape and motion, and pruning unfruitful model graphs. We then continue with the experimental setup for action recognition and its results.

3.5.1. STIP Detection and Description

We used two different methods to extract spatio-temporal interest points: 3D Harris detector [4] and 2D Gabor filters [58]. The detected interest points constitute the nodes of the graphs.

3.5.1.1. 3D Harris detector. The well known Harris interest point detector was developed by Harris and Stephens [179] and was extended into the spatio-temporal domain by Laptev *et al.* [4]. The detector is based on the spatio-temporal second moment matrix ψ of the Gaussian smoothed video volume I [4].

The spatio-temporal second moment matrix ψ is composed of first order spatial and temporal gradients averaged using the spatio-temporal separable Gaussian kernel $g(\cdot)$:

$$\psi(x, y, t; \sigma, \tau) = g(x, y, t; \sigma, \tau) * \begin{pmatrix} L_x^2 & L_x L_y & L_x L_t \\ L_x L_y & L_y^2 & L_y L_t \\ L_x L_t & L_y L_t & L_t^2 \end{pmatrix}, \quad (3.30)$$

where σ , τ are defined as the spatial and temporal scales, respectively, and the first order gradients can be calculated as follows:

$$L_x = \partial_x (g(x, y, t; \sigma, \tau) * I(x, y, t)) \quad (3.31)$$

$$L_y = \partial_y (g(x, y, t; \sigma, \tau) * I(x, y, t)) \quad (3.32)$$

$$L_t = \partial_t (g(x, y, t; \sigma, \tau) * I(x, y, t)). \quad (3.33)$$

Note that the scale parameters for spatial and temporal domain are treated separately as the spatial and temporal extents of the events are independent.

To detect interest points, we search for the locations where ψ contains three large eigenvalues, a_1, a_2, a_3 , namely, strong intensity variation along three orthogonal spatio-temporal directions. Instead of calculating eigenvalues at every video location (x, y, t) , we find the local maxima of the following function

$$H = \det(\psi) - k \cdot (\text{trace}(\psi))^3 = a_1 a_2 a_3 - k(a_1 + a_2 + a_3)^3. \quad (3.34)$$

In other words, the spatio-temporal interest points, i.e., the points showing high gradient both in space and time, are at the locations of the local maxima of H . Laptev *et al.* [4] also proposed to extract points at multiple levels by using a set of scale parameters. In our experiments, we used the off-the-shelf code in [4]. The parameter set is defined as follows: $\sigma^2 = 4, 8, 16, 32, 64, 28$ and $\tau^2 = 2, 4$ and $k = 0.0005$.

3.5.1.2. 2D Gabor filters. Gabor filters have been successfully used in many application such as iris recognition, face recognition etc. In the action recognition domain, several researchers [54, 57, 58] have also used Gabor filters to detect spatio-temporal interest points. In our work, we use 2D Gabor filters as proposed by Bregonzio *et al.* in [58] where the real component of a Gabor filter is composed of a sinusoidal carrier and a Gaussian envelope:

$$G(x, y) = \cos(2\pi \cdot (\mu_0 x + \nu_0 y) + \theta_i) \cdot \exp\left(-\frac{x^2 + y^2}{2\rho^2}\right), \quad (3.35)$$

where ρ is the width of the Gaussian envelope, θ_i is the orientation of the filter and μ_0, ν_0 are the spatial frequencies that controls the scale of the filter.

To detect interest points, Bregonzio *et al.* [58] proposed to take the difference between the consecutive frames and to convolve the resulting image with a bank of 2D Gabor filters having different orientation and scale parameters. The different responses are calculated for different parameter pair and then summed up. Spatio-temporal interest points are located at local maxima of the resulting function. In our experiments,

we used the publicly available code from [58]. We used the same parameter setting as in [58] where we considered five different θ_i values, $\theta_i = \{0^\circ, 22^\circ, 45^\circ, 67^\circ, 90^\circ\}$, and one scale parameter by setting $\mu_0 = \nu_0 = \frac{1}{2\rho}$ and $\rho = 11$.

3.5.1.3. Cuboid descriptors. For appearance features f_i , we have used the well known histogram of gradient and optical flows (HoG+HoF) extracted with the publicly available code from [4]. Extension of HoG and HoF to temporal domain results in a position dependent histogram. The histogram computation is illustrated in Figure 3.7. The local neighborhood of a detected point is divided into a grid with $M \times M \times N$ (i.e., $3 \times 3 \times 2$) spatio-temporal blocks. For each block, 4-bin gradient and 5-bin optical flow histograms are computed and concatenated into a 162-length feature vector.

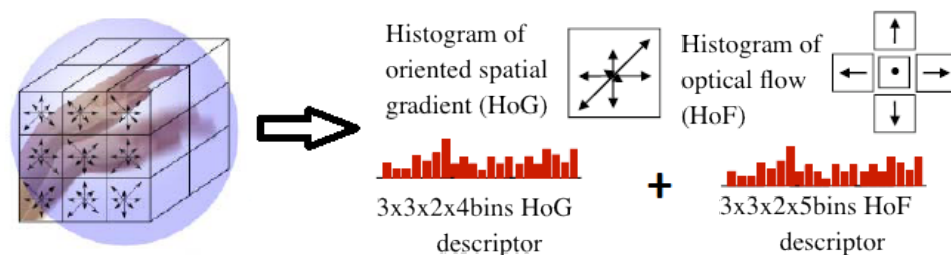


Figure 3.7. Computation of HoG and HoF [4]. Courtesy of Ivan Laptev.

3.5.2. Learning Discriminative Graph Prototypes

In prototype-based approaches, prototype selection plays a key role in recognition performance. In the context of action recognition, intra-class variability can be large enough to eclipse the discrimination among different classes. For this reason, we learned a reduced set of representative model graphs called prototypes and used Nearest Prototype Classifier (NPC) for recognition. The problem can be formulated as follows.

Let $\{G_i^{(m)}\}_{i=1}^{|L|}$ be a set of model graph prototypes obtained from the training set L . Our goal is to find a subset of discriminative graph prototypes, SDG , which increases the measure $ACC(V, \{G_i^{(m)}\}_{i=1}^N)$ defined as recognition accuracy on a validation set V

using the model graph prototypes $\{G_i^{(m)}\}_{i=1}^N \in L$, $N < |L|$:

$$SDG = \arg \max_{\{G_i^{(m)}\}_{i=1}^N \in L} ACC(V, \{G_i^{(m)}\}_{i=1}^N). \quad (3.36)$$

However, searching exhaustively for the optimal SDG is a combinatorial problem. Instead, we used a greedy search algorithm called as Sequential Floating Forward Search (SFFS), which provides discriminative prototypes in ensembles at a much lower computational cost. SFFS has been successfully used as a supervised feature selection method in many previous studies [180]. The main steps of SFFS can be briefly described as follows:

- (i) Start with one prototype per class that yields the best performance and proceed to add, one at a time, conditionally the prototype that enables the biggest improvement in performance.
- (ii) After a number of addition steps, remove one or more of the prototypes if their deletion improves the performance.
- (iii) Repeat Step 2 and 3, until no more performance improvement is observed.

3.5.3. Experimental Setup

The proposed matching algorithms have been evaluated on the widely used public KTH database [1]. This database contains 25 subjects performing 6 actions (*walking*, *jogging*, *running*, *handwaving*, *handclapping* and *boxing*) recorded in four different scenarios including indoor/outdoor scenes and different camera viewpoints, totally 599 video sequences (one is corrupted). Each video sequence is also composed of three or four action subsequences, resulting in 2391 subsequences in total. The subdivision of the sequences we used is the same as the one provided on the official dataset website [181]. In our experiments, we have used these subsequences to construct the model graphs where each one consists of 20 to 30 frames, and such that each frame contains at least one or more salient interest points.

Parameters have been estimated or fixed as follows. The penalty parameter W^d

should theoretically be higher than the average energy of correctly matched triangles and lower than the average energy of incompatible triangles. We define W^d as the mean value of all, compatible or incompatible, triangle matching energies. The weighting parameters are set so that each distortion measure has the same range of values: $\lambda_1 = 0.6$, $\lambda_2 = 0.2$, $\lambda_3 = 5$, $T^t = 10$, and $W^t = 20$.

All experiments use the leave-one-subject-out (LOSO) strategy. Action classes on the unseen subjects are recognized with a nearest prototype classifier (NPC). The distance between scene and model prototypes is based on the matching energy given in Equation 3.3. However, experiments have shown that the best performance is obtained if only the appearance terms $U(\cdot)$ are used for distance calculation instead of the full energy in Equation 3.3. This results in the following two step process:

- (i) Node correspondences: The correspondence problem is solved using the total energy in Equation 3.3, that is the sum of $U(\cdot)$ and $D(\cdot)$ terms. This step provides a solution for the hidden variables x .
- (ii) Decision for action class: The distance between model and scene graphs is calculated using only the $U(\cdot)$ term evaluated on the assignments calculated in the first step, i.e., on solutions for variables x .

Results have been analyzed in terms of three criteria: (i) performance of the various graphical structures with different number of interest points per frame; (ii) impact of prototype selection; and (iii) computational efficiency. These will be detailed in the following section.

3.5.4. Choice of Graphical Structure

The three graphical structures introduced in Section 3.4 have been evaluated on the dataset:

- Single-chain-single-point model: The interest points are detected by the 3D Harris

detector [4]. Frame of a model sequence is represented by a single most salient point. There are no such restrictions on the scene sequences.

- Single-chain-multiple-points model: The 3D Harris detector results in very few interest points, i.e., the number of points varies between 1 and 4 per frame on average. It is a good choice, if our only goal is to select a single point per frame. However, in the case of multiple-points model, it is not appropriate. We therefore used 2D Gabor filters [58] and extracted at least 20 points per frame. To eliminate outliers, we fitted a bounding box on the human body and applied canny edge detector to choose the points that are on or closer to the human silhouette. $N = 3$ points per frame are used in the model sequence. To capture the global property of the human body, these three points are selected so as to constitute the biggest triangle. There are no restrictions on the scene sequences, only the points outside the bounding box are eliminated.
- Multiple-chains model: We again used 2D Gabor filters [58] to extract interest points. $N = 2$ single-chain-single-point models are matched separately. A final decision is obtained by averaging the matching energies over chains.

In Table 3.2, the results for different graphical structures are given. The best recognition performance of the proposed scheme is found to be 89.2% with the single-chain model. Looking at the confusion matrices in Table 3.2, it can be observed that major part of the errors are due to confusion between the *jogging* and *running* classes, which are indeed quite similar.

At first sight it might come as a surprise that the single-point model performs slightly better than the two alternative models using multiple points per frame. However, the single-chain-multiple-point model does not fully exploit the rich space-time geometry of the problem. In fact, we force this model to choose the triple of interest points per frame (that will constitute one of the nodes of its hyper-graph) all from the same model frame and the same scene frame. Even though the subsequent hyper-graph matching uses the space-time geometry, it cannot make-up for the lost flexibility in the in-frame triple point correspondence stage.

The multiple-chain model, on the other hand, is similar in nature to the single-chain model as it consists of chains successively established with mutually exclusive model nodes (scene nodes can be re-used). Individual performance of chains are 85.6% and 84.6%, respectively. The additional information content gathered by chains seems to translate into better performance, which increases to 86.6% when we do score fusion (average the energies). A plausible explanation is that two models, single-chain and multiple-chain, basically differs in the interest point detection method used, 3D Harris detector [4] and 2D Gabor filters [58], respectively, which is noisy for multiple-chains model. However, this gap is compensated by the prototype selection algorithm as shown in the following section.

3.5.5. Prototype Selection

We conjectured that the discrimination power can be improved by judicious selection of prototypes, in effect removing the noisy prototypes. We learned a set of discriminative prototypes by the method introduced in Section 3.5.2. We used the same data partition protocol (8/8/9) given in [1]: Model graph prototypes are created from the training subjects and the prototype selection is optimized over the validation set. We have determined the optimal number of prototypes by a grid search where the increments were in groups of 5 graphs. In Table 3.3, we have presented our results after prototype selection. For example, for single-chain-single-point model, SFFS yielded 50 models out of the initial 750 ones as the best subset of model graph prototypes. Learning prototypes increased the test performance by 2 percentage points, up to 91%. As expected, *handwaving* and *jogging* sequences benefit the most from dictionary learning (see Table 3.3). Similar observations can be made for single-chain-multiple-points and multiple-chains model. The number of model graph prototypes are reduced to 250 and 50 from 750, respectively. To sum up, Table 3.4 summarizes our experimental results and compares run times of each graphical structures. Note that the run times given do not include interest point detection and feature extraction, but these are negligible compared to the matching requirements.

Sample matched model and scene sequences are illustrated in Figure 3.8 where

Table 3.2. Confusion matrices before prototype selection (a) single-chain-single-point model, (b) single chain-multiple-points model and (c) multiple-chains model. Respective average accuracies are 89.2%, 87.6% and 86.6%. (B: Box, HC: Handclap, HW: Handwave, J: Jog, R: Run, W: Walk).

	B	HC	HW	J	R	W
B	86	13	0	0	1	0
HC	1	99	0	0	0	0
HW	1	10	89	0	0	0
J	0	0	0	74	26	0
R	0	0	0	10	90	0
W	0	0	0	3	0	97

	B	HC	HW	J	R	W
B	96	2	0	0	2	0
HC	1	99	0	0	0	0
HW	2	7	91	0	0	0
J	0	0	0	69	31	0
R	0	0	0	14	84	2
W	0	0	0	8	5	87

	B	HC	HW	J	R	W
B	94	6	0	0	0	0
HC	3	97	0	0	0	0
HW	3	12	85	0	0	0
J	0	0	0	93	6	1
R	0	0	0	40	59	1
W	0	0	0	8	0	92

Table 3.3. Confusion matrices after prototype selection: (a) single-chain-single-point model, (b) single chain-multiple-points model and (c) multiple-chains model. Respective average accuracies are 91%, 90.2% and 90.8%. (B: Box, HC: Handclap, HW: Handwave, J: Jog, R: Run, W: Walk).

	B	HC	HW	J	R	W
B	92	3	3	2	0	0
HC	0	97	3	0	0	0
HW	0	3	97	0	0	0
J	0	0	0	88	9	3
R	0	0	0	19	75	6
W	0	0	0	0	3	97

	B	HC	HW	J	R	W
B	94	0	6	0	0	0
HC	0	97	3	0	0	0
HW	3	5	92	0	0	0
J	0	0	0	92	3	5
R	0	0	0	8	89	3
W	0	0	0	17	5	78

	B	HC	HW	J	R	W
B	94	6	0	0	0	0
HC	0	100	0	0	0	0
HW	0	6	94	0	0	0
J	0	0	0	91	3	6
R	0	0	0	23	77	0
W	0	0	0	11	0	89

Table 3.4. Summary of the experimental results. Run time (ms/frame) is computed for matching one model graph on a CPU with 2.8GHz and 8GB RAM.

Method	Before prototype selection	After prototype selection	Run time (ms/fr)
Single-chain-single-point model	89.2%	91%	4.1ms
Single-chain-multiple-points model	87.6%	90.2%	15.6ms
Multiple-chains model	86.6%	90.8%	12.6ms

in each sub-figure the left image is from the model sequence and the corresponding matched scene frame is given on its right. One can observe that the proposed method is also successful at localizing a model action sequence in a much longer scene sequence (recall $\bar{S} \gg \bar{M}$). We illustrate in Figure 3.8a-c successfully recognized actions *handwaving*, *handclapping*, *walking*, and in Figure 3.8d a misclassification case, where *running* was recognized as *jogging*.

3.5.6. Comparison with State-of-the-art

We would like to point out that although many research results have been published on the KTH database, most of these results cannot be directly compared due to their differing evaluation protocols, as has been indicated in the detailed report on the KTH database [182]. Nevertheless, for completeness we give our performance results along with those of some of the state-of-the-art methods as reported in the respective original papers. Details of these methods were discussed in Chapter 2. In Table 3.5, we report average action recognition performance and computation time of the compared methods. Although methods to calculate run times and protocols employed differ between the papers, this table is intended to give an overall idea. We claim that our method provides a good compromise between performance and complexity. For ex-

ample, the method proposed by Ta *et al.* [94] is the closest approach to our method: they also used graph matching but based on spectral methods, so that they do approximate matching of the exact problem. Recall that ours was the exact minimization of the approximated graph problem. As can be seen, our approach shows a competitive performance but with a very low computational time.

3.5.7. A Real-time GPU Implementation

A GPU implementation enables real-time performance on standard medium end GPUs, e.g., a Nvidia GeForce GTS450¹. Table 3.6 compares single-chain/single-point model run times of the CPU implementation in Matlab/C and the GPU implementation running on different GPUs with different characteristics, especially the number of calculation units. The run times are given for matching a single model graph with 30 nodes against scene blocks of different lengths. If the scene video is segmented into smaller blocks of 60 frames, which is necessary for continuous video processing, real time performance can be achieved even on the low end GPU model. With these smaller chunks of scene data, matching all 50 graph models to a block of 60 frames (roughly 2 seconds of video) takes about 3 ms regardless of the GPU model.

The processing time of 3 ms per frame is very much lower than the requirement for real time processing, which is 40 ms for video acquired at 25 frames per second. Additional processing will be required in order to treat overlapping blocks, which increases running time to 6 ms per frame.

3.6. Summary

In this chapter, we showed that the exact solution to the point set matching problem with hyper-graphs can be calculated with bounded complexity in the case of spatio-temporal data. This is enabled when the correspondence problem is constrained by taking into account the sequential nature of time and causality of human actions. We show that, under these assumptions, the energy function can be minimized in the spirit

¹GPU implementation of the proposed algorithm was done by Eric Lombardi.

Table 3.5. Comparison with the state-of-the-art methods on the KTH database (E: Performance Evaluation, R: Run time, LOOCV: Leave-one-out-cross validation, NA: Not available).

Work	Average performance	Run time	Remarks
Ta <i>et al.</i> [94]	91.2%	46.7s	E: LOOCV R: Matching 1s video with 98 model graphs
Borzeshi <i>et al.</i> [96]	70.2%	NA	E: 8/8/9 R: NA
Brendel & Todorovic [45]	NA	10s	E: NA R: Matching 1000 nodes model-graph with 2000+ nodes scene-graph
Lv & Nevatia [106]	NA	5.1s	E: NA R: s/frame
Savarese <i>et al.</i> [103]	86.8%	NA	E: LOOCV R: NA
Ryoo & Aggarwal [90]	93.8%	NA	E: LOOCV R: NA
Mikolajczyk & Uemura [113]	95.3%	0.5s to 3s (5.5s to 8s with SVM)	E: LOOCV R: s/frame
Jiang <i>et al.</i> [114]	93.4%	NA	E: LOOCV, all scenarios in one R: NA
<i>Our method</i> (single-chain-single-point model)	91%	0.2s	E: 8/8/9 R: s/frame, matching with 50 model graphs

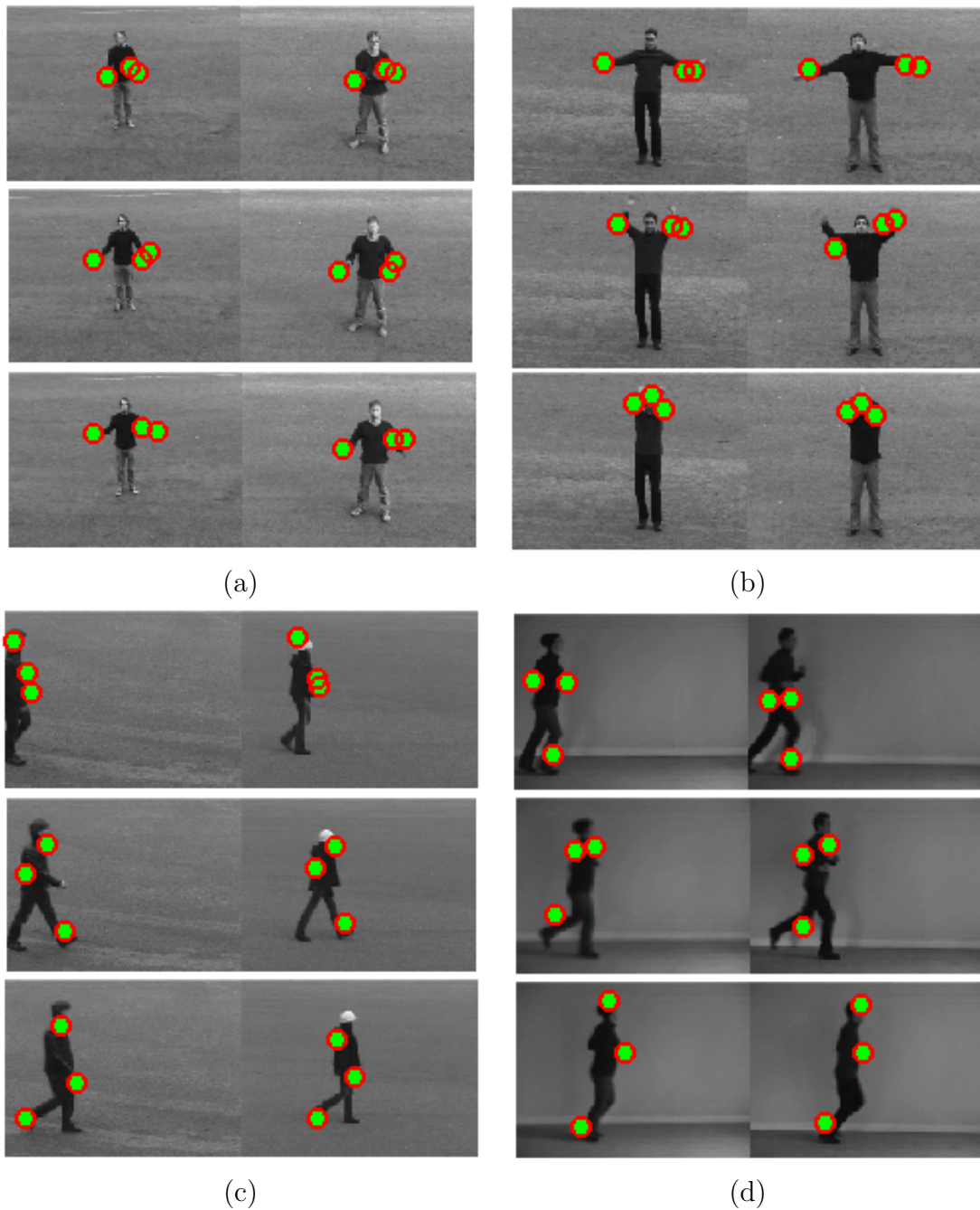


Figure 3.8. Examples for matched sequences. (a) Model: Handclap (first column), Scene: Handclap (second column); (b) Model: Handwave, Scene: Handwave; (c) Model: Walk, Scene: Walk; (d) Model: Jog, Scene: Run. Here, (a), (b) and (c) show a case of correct recognition, while (b) is a case of misclassification.

of dynamic programming technique which indeed leads to the global minimum. These simplifications reduce the exponent of the complexity, but it still remains exponential

Table 3.6. Run times in milliseconds for a CPU and two different GPUs and for 4 different scene block sizes. The last column on the right gives times per frame for matching the whole set of 50 model graphs (*Real time).

Implementation	Scene	Scene	A single model		All 50 models
	Nodes	Frames	Time (ms)	Time/fr (ms)	Time/fr (ms)
CPU: Intel Core 2 Duo, E8600 @ 3.33Ghz, Matlab/C(mex)	754	723	2900	4.01	200.5
Nvidia GeForce GTS450, 192 cuda cores, 128 bit memory interface	754	723	748	1.03	51.5
	60	55	4	0.07	3.5*
Nvidia GeForce GTX560, 336 cuda cores, 256 bit memory interface	754	723	405	0.56	28
	60	55	4	0.07	3.5*

in nature. We have therefore introduced an approximation to the graphical structure with three different graph building strategies: (i) single-chain-single-point model; (ii) single-chain-multiple-points model; and (iii) multiple-chains model.

As a proof of concept, our algorithm has been tested on the KTH database where we compared these three graphical structures in terms of action recognition performance and computational efficiency. We summarized our results in Table 3.7.

To remind, we quantify the computational complexity by the following variables.

- (i) M and S are the number of model nodes and the number of scene nodes, respectively. Note that $S \gg M$. $|\mathcal{E}|$ is the number of edges.
- (ii) \bar{M} and \bar{S} are the number of model frames and the number of scene frames, respectively. Note that $S \gg \bar{S}$ and $\bar{S} \gg \bar{M}$.
- (iii) In the approximated graphical structure, the distance between the nodes in the time domain is thresholded by T^t .
- (iv) R is the maximum number of interest points per frame in the *scene* sequence.

Table 3.7. Summary of the computational complexities and the recognition performances of the proposed approaches.

Method	Complexity	Recognition Performance
Brute-force approach	$O(M \cdot S^M \cdot \mathcal{E})$	-
Single-chain-single-point model	$O(\bar{M} \cdot \bar{S} \cdot T^{t^2} \cdot R)$	91%
Single-chain-multiple-points model	$O(\bar{M} \cdot \bar{S} \cdot T^{t^2} \cdot R^3)$	90.2%
Multiple-chains model	$O(N \cdot \bar{M} \cdot \bar{S} \cdot T^{t^2} \cdot R)$	90.8%

(v) N is the number of matched single chains-single point model.

Each model allows performing exact minimization with complexity that grows only linearly in the number of model frames and the number of scene frames, and grows exponentially in the temporal search range and the number of interest-points per scene frame. As shown, the complexity of the proposed approaches is very much lower than the complexity of the brute force approach.

We notice that single-chain-single-point model surpasses the other two models, multiple-points and multiple-chains models, not only in performance, albeit slightly, but also in computational time. One reason could be that these two approaches (single-point versus multiple-points) differ in the interest point detection method. We conjecture that the interest points detected by 3D Harris detector [4] are more informative and discriminative than the interest points by 2D Gabor filters [58]. We tested and validated this conjecture by comparing “single-chain-single-point model+3D Harris detector” performance with “single-chain-single-point model+2D Gabor filter” performance where we obtained 89.2% and 85.6% recognition performances, respectively, before the prototype selection takes place. However, this gap in performance was compensated by eliminating noisy prototypes from the model graph set.

The proposed scheme shows a modest performance (91%) as compared to the best performance obtained on the KTH database (95.3%). However, it has the advantage of very low runtime. Our algorithm is faster by an order of magnitude with respect to its nearest competitor in speed (Table 3.5).

4. HYPER-GRAPH BASED ANALYSIS OF SKELETON SEQUENCES

In this chapter, we present the application of the proposed graph matching algorithm to action recognition from depth data. In this context, we use depth sequences recorded with a Kinect camera [23], which yields the coordinates of the tracked skeleton joints [47].

We address a second specific problem in the action analysis domain, that of automatic action quality assessment. The goal is to evaluate the performance of a person performing an action as guided by instructions or by exemplars. The exemplar can be in the form of a video of an agent performing the action, the instructions in the form of a text describing the kinematic details of an action or in the form of by an instructor/person in interaction. Instances of this problem in the literature are evaluation of karate performance [25], of dancer’s performance [183], quality assessment of reach and grasp movements of stroke survivors [16], and the gaming/entertainment industry. In this work, we particularly aim to assess the quality of physical exercises for therapeutic purposes.

We model the spatio-temporal relationship between the skeleton joints by the single-chain-multiple-points model (Chapter 3, Section 3.4.2). Since the in-frame point correspondence problem is already solved by skeleton-tracking algorithm, we do not need to perform any greedy algorithm for assigning node variables (see Section 3.4.2). Therefore, the matching problem is reduced to ensuring the temporal consistency between joints, namely the temporal alignment. The reduced problem seems to be somewhat easier as compared to the unconstrained graph matching problem in Chapter 3, but it is still very challenging due to spiky, noisy joint data delivered by the Kinect system or data missing due to severe occlusions. Once the skeleton sequences are aligned, the action quality is measured based on the spatial and temporal deformation measures with respect to a reference subject, e.g., an instructor.

We begin with the related literature on quality assessment. Then, we describe the three important aspects of the skeleton-based algorithm: (i) skeleton representation (normalization and pose descriptor extraction); (ii) sequence alignment by graph matching; (iii) performance assessment. Comparative experimental results on two action datasets are also presented.

4.1. Related Literature on Quality Assessment

In the literature, we have encountered only few studies for automatic action quality assessment. These prior works have used Motion Capture (MoCap) data [184, 185] or RGB video sequences [186]. A few approaches have been recently proposed based on skeleton tracking. For example, in [25], Bianco and Tisato recognize Karate moves based on skeletal joints. They select triplets of joints manually for hand and foot techniques, and represent each move by the angles of joint triplets. A set of key poses is obtained via K-means clustering and Dynamic Time Warping (DTW) is used for aligning two sequences of poses. The resulting normalized DTW distance is used for performance evaluation and a regression analysis is done in order to validate the relationship between the DTW distance and the subjective scoring. Essid *et al.* [183] use three different quality measures and then combine them for salsa dance performance evaluation. These measures are computed based on positions, velocities and 3D motion (flow) of the joints. This method uses quaternionic correlation to estimate the time-shift between two dancing sequences. Venkataraman *et al.* [16] propose a correlation-based approach to quantify the movements of stroke survivors. They model the movements of a human differently, from a dynamical system perspective and reconstruct a phase space to infer geometrical and topological information from observations by means of attractor. Their features rely on the shape features from the reconstructed phase space. However, the main drawback of these methods is the use of correlation, as it does not incorporate time warping.

In our work, we use the proposed hyper-graph matching algorithm to align two skeleton sequences. Compared to the scheme in Chapter 3, the spatio-temporal interest points in the graph matching algorithm are replaced with skeleton joints, and

then characterize each skeleton by a pose descriptor. We enrich the angle-based pose descriptor proposed in [25] and concatenate it with a distance-based pose descriptor. We also propose a novel method for elimination irrelevant model graphs. Experimental results are reported on both action recognition and quality assessment of physical exercises.

4.2. Pose Descriptor Extraction

Prior to any feature extraction, we applied a preprocessing stage to normalize skeletons. Each joint n of a skeleton is represented at a time instant i by its 3 coordinates $p_{i,n} = [p_{i,n}^{\langle x \rangle} p_{i,n}^{\langle y \rangle} p_{i,n}^{\langle z \rangle}]$. First, the skeletons are normalized in order to render each skeleton independent from position and body size. For each time instant (frame), we scaled the Euclidean distance between connected skeleton joints so that the inner distance between the hip and the center of shoulders is set to unit length, and then we translated joint positions so that the hip center coincides with the origin of the coordinate system. Secondly, we applied vector median filtering on the time-trajectories of each joint, where for the n th joint the \overline{M} long joint coordinate sequence is given by

$$P_{i,n} = \begin{bmatrix} p_{1,n}^{\langle x \rangle} & p_{2,n}^{\langle x \rangle} & \cdots & p_{\overline{M},n}^{\langle x \rangle} \\ p_{1,n}^{\langle y \rangle} & p_{2,n}^{\langle y \rangle} & \cdots & p_{\overline{M},n}^{\langle y \rangle} \\ p_{1,n}^{\langle z \rangle} & p_{2,n}^{\langle z \rangle} & \cdots & p_{\overline{M},n}^{\langle z \rangle} \end{bmatrix}. \quad (4.1)$$

Vector median filtering aims at removing possible coordinate spikes and reducing noise. Following the preprocessing stage, we extract two types of pose descriptors for action recognition and quality assessment.

4.2.1. Angle-based Pose Descriptor

In the literature, it has been shown that scale and orientation-invariant features can be obtained by using angles between consecutive joints. Each skeleton is represented by 14 angles as illustrated in Figure 4.1. Bianco *et al.* [25] characterized each joint with only the subtended angle. We believe that to fully describe the orientation

in 3D, one should calculate both the subtended angle and the orientation angle of the plane defined by the three points (joints).

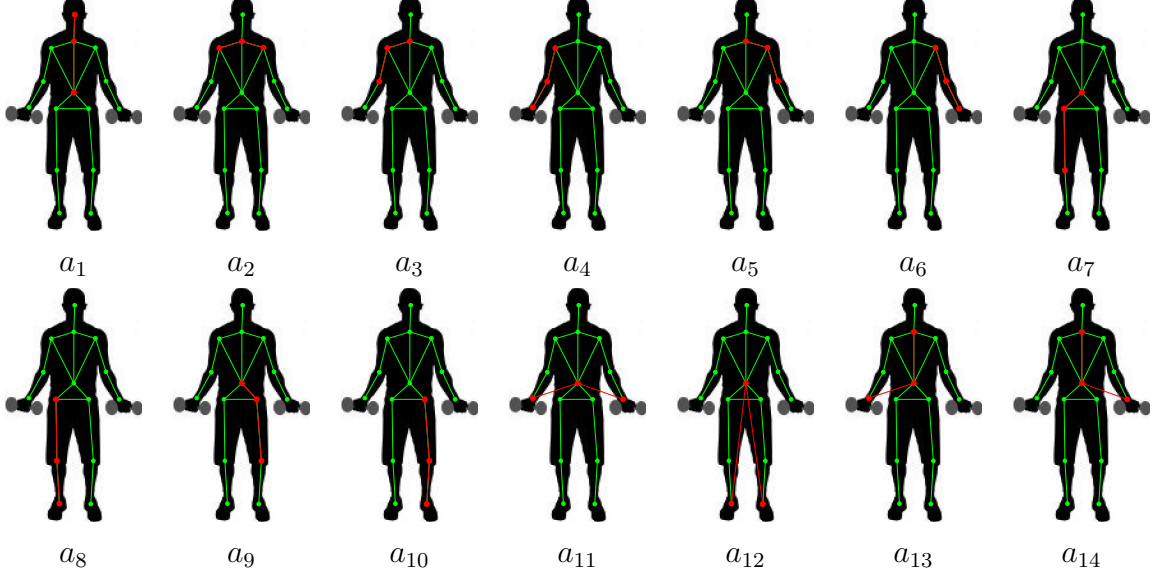


Figure 4.1. Angles computed for skeleton representation.

To describe the orientation angles of the plane defined by joint triples, we calculate the inclination and azimuth angles with respect to the torso basis, as in Raptis *et al.* in [109]. We apply PCA to the six torso joint positions (shoulder left/right/center and hip left/right/center) and find a torso basis $\{u_x, u_y, u_z\}$ in which u_x aligns with the line that connects the shoulders, u_y coincides with the line along the spine, and finally, u_z corresponds to depth directional vector. The torso joints and the calculated angles are illustrated in Figure 4.2. Each angle is defined by three joints p_s , p_c and p_e at any time instant. For simplicity, we ignore the time index i for the moment. As illustrated in Figure 4.2b, the vector b , extension of the vector $\overrightarrow{p_s p_c}$, is the normal of the plane \mathcal{S} centered at the p_c . We calculate the angles as follows:

- The inclination angle $\alpha(n)$ is computed between $\overrightarrow{p_c p_e}$ and b .
- The azimuth angle $\beta(n)$ is defined between \hat{u}_x and \hat{p}_e which are the projections of u_x and p_e on the plane \mathcal{S} , respectively.
- We also compute the angle η between the directional vector z from the depth sensor and the body orientation vector u_z from the torso basis, to measure the

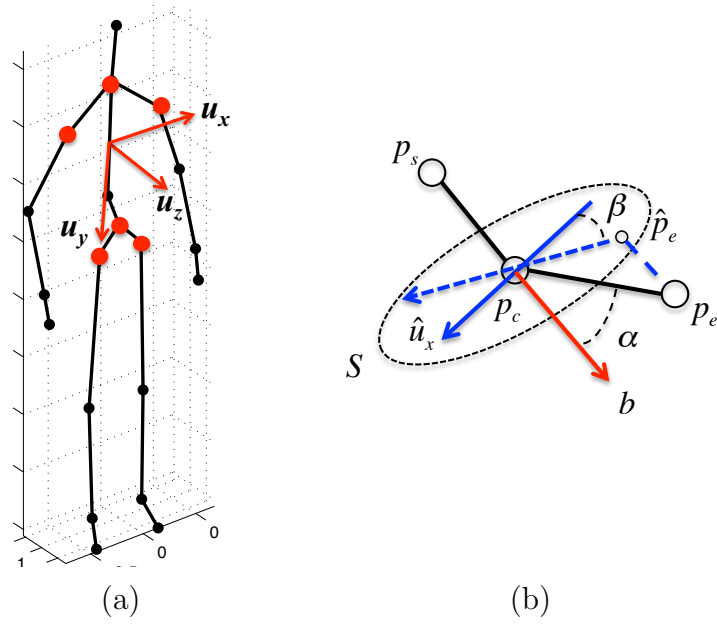


Figure 4.2. (a) Illustration of torso basis; (b) Illustration of the spherical coordinates, i.e., radius R , inclination angle α_j and azimuth angle β_j , defined based on the torso frame.

bending of the body.

Thus, we can define the angle-based pose descriptor vector as $\mathbf{a} = \{a(1), \dots, a(14), \eta\} \in \mathbb{R}^{29}$ where $a(n) = \{\alpha(n), \beta(n)\}$.

4.2.2. Distance-based Pose Descriptor

Complementarily, we calculate the Euclidean distance between joint-position pairs, $p_{i,n}$ and $p_{i,k}$. Let N be the number of joints. In our experiments, we ignore the joints related to hand and foot, since these are more prone to errors, and set $N = 15$. We define the distance-based pose descriptor vector as $\mathbf{d} = \{d(n) | n = 1, 2, \dots, 15\} \in \mathbb{R}^{210}$ where $d(n) = \{d(n, k) | k \neq n\} \in \mathbb{R}^{14}$ and $d(n, k) = \|p_{i,n} - p_{i,k}\|_2$.

Finally, we simply obtain the joint pose descriptor vector f_i as the concatenation of the two pose, angle-based and distance-based, descriptor vectors, $f_i = \{\mathbf{a}, \mathbf{d}\}$.

4.2.3. Pose Quantization

We quantize the skeleton pose space in order to decrease redundancy in the temporal domain. Since K-means clustering is widely used for extracting the key poses [25, 108], we adapted it to cluster skeleton poses represented by the joint pose descriptor f_i and obtained a set of key poses. Given a sequence of key poses, we sample and assign each skeleton at a time instant to its closest cluster center, i.e., its closest key pose, and obtain an abstract representation for each skeleton sequence.

4.3. Graph-based Sequence Alignment

In this section, we customize the single-chain-multiple-points-model (please refer to Chapter 3, Section 3.4.2) for aligning the tracked skeleton sequences. In this context, each action sequence is structured into a single chain graph where the frame variables z_i coincide with the frames (skeletons) and each z_i assumes N node variables $y_{i,n}$, namely skeleton joints. Since the Kinect system provides the positions of the skeleton joints, the problem of assigning the node variables has already been solved. However, the problem of which model frame to match with which scene frame remains still very challenging in nature due to noisy, missing or occluded joints. In this context also, graph matching offers a robust and flexible solution to the sequence alignment problem against these problems.

Below, we invoke our proposed graph energy function as defined in Section 3.4.2:

$$E(z) = \lambda_1 \sum_{i=1}^{\bar{M}} U(z_i) + \lambda_2 \sum_{i=3}^{\bar{M}} D(z_i, z_{i-1}, z_{i-2}). \quad (4.2)$$

$U(\cdot)$ is now defined as the Euclidean distance between the pose descriptors of skeletons, and $D(\cdot)$ again measures the spatio-temporal deformation in terms of two terms, the time warping penalty term $D^t(\cdot)$ and the geometrical deformation term $D^g(\cdot)$. For convenience, we recall $U(\cdot)$

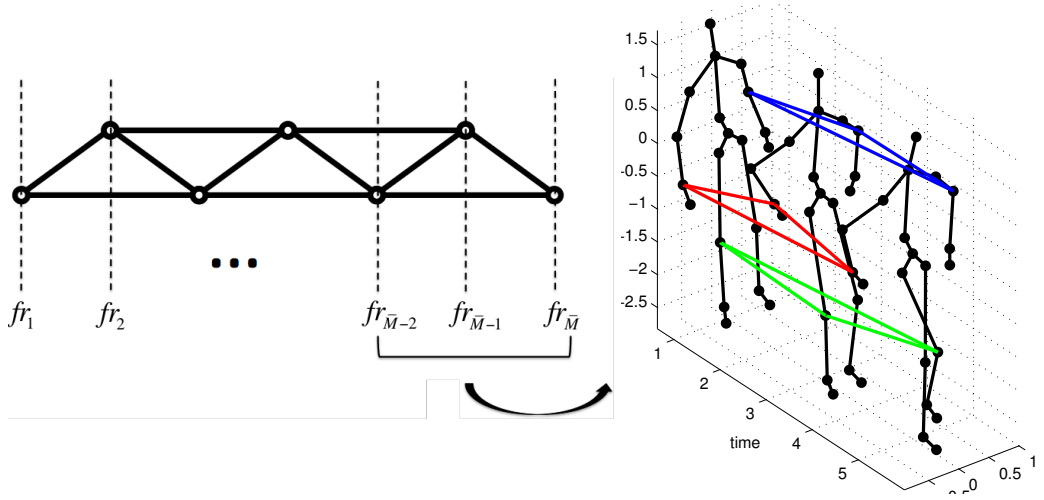


Figure 4.3. Each triangle models the spatial relationship between three consecutive skeletons in the triangular structure graph.

$$U(z_i) = \begin{cases} W^d & \text{if } z_i = \epsilon, \\ \|f_i^{(m)} - f_{z_i}^{(s)}\| & \text{else,} \end{cases} \quad (4.3)$$

where ϵ is the dummy assignment and W^d its penalty, and $D(\cdot)$

$$D(z_i, z_{i-1}, z_{i-2}) = \sum_{c \in (i, i-1, i-2)} D_c(z_c, y_c). \quad (4.4)$$

Recall that $D^t(\cdot)$ is the time differences between the model nodes and their corresponding scene nodes truncated by the threshold T^t . Slightly different from the single-chain-multiple-points-model in Section 3.4.2, $D^g(\cdot)$ is measured by summing the deformation terms over the triplets of consecutive joints. This can be best explained by Figure 4.3 where each skeleton joint is connected to its two preceding and its two succeeding neighbors. In other words, each triangle models the temporal relationship of three consecutive skeleton joints, all of the same type. If we put into formulation, let $p_{c,n}$ denotes the triangle vertices associated with the 3 dimensional Cartesian coordinates of n th joint in three consecutive frames, $\{i, i-1, i-2\}$. $D^g(\cdot)$ is defined as the summation of the differences between the angles of triangle formed by the model triplet $p_{c,n}^{(m)}$ and

the corresponding angles of the triangle formed by its corresponding scene triplet $p_{z_c,n}^{(s)}$:

$$D^g(z_i, z_{i-1}, z_{i-2}) = \sum_{n=1}^N \|\psi(p_{z_c,n}^{(m)}) - \psi(p_{z_c,n}^{(s)})\| \quad (4.5)$$

where $\psi(\cdot)$ denotes the two triangle angles in Equation 3.8 and N is the number of skeleton joints per frame.

Note that the computational complexity, in this case, is even lower than the complexity of the single-chain-single-point-model, i.e. $O(\overline{M} \cdot \overline{S} \cdot T^2)$. Recall that \overline{M} and \overline{S} are respective number of scene and model frames. In the sequel, we show how these measures can be used for action recognition and performance assessment as well as for learning a set of discriminative and representative model graphs.

4.4. Learning Model Graphs

In this chapter, inspired from the algorithm in [67], we follow an approach for learning and mining prototypes different from what we had used in Section 3.5.2. Given a set of L model graph prototypes $G_j^{(m)}$ obtained from a training set, our goal is to find a subset of representative and discriminative graph prototypes. Let E_i^j be the energy of a model graph $G_j^{(m)}$ matched to a scene graph $G_i^{(s)}$ which is explicitly formalized in Equation 4.2. We define two measures: discriminability measure and representability measure.

Discriminability measure for $G_j^{(m)}$ is formulated as the ratio of between-class variance to within-class variance:

$$DIS_j = \frac{\sum_{k=1}^K N_k (\bar{\mu}_k^j - \bar{\mu}^j)^2}{\sum_{k=1}^K \sum_{i \in C_k} (\mu_i^j - \bar{\mu}_k^j)^2} \quad (4.6)$$

where K is the number of action classes. Let N_k be number of samples in each action

class C_k , $\bar{\mu}_k^j$ and $\bar{\mu}^j$ can be defined as follows:

$$\bar{\mu}_k^j = \frac{1}{N_k} \sum_{i \in C_k} E_i^j; \quad \bar{\mu}^j = \frac{1}{\sum_{k=1}^K N_k} \sum_{k=1}^K \bar{\mu}_k^j. \quad (4.7)$$

Representability measure, REP_j , evaluates the ratio of the scene graphs that are covered by the model graph $G_j^{(m)}$. Given a model graph and a scene graph pertaining to the same action class, the model graph covers the scene graph, if it is in the l -nearest neighborhood of the scene graph. REP_j is therefore defined as the ratio of the number of covered scene graphs to all scene graphs.

We choose the model graphs satisfying these two requirements. In our experiments, we normalize these two measures and calculate the mean value. We choose the model graphs that have higher mean and set the number of selected prototypes that gives the best performance on the training set. Indeed, experimental results show that the choice of representative and discriminative model graphs to construct a learned dictionary improves the performances as compared to when all the models were used.

4.5. Action Quality Assessment

We test and assess the utility of our scheme in two separate frameworks: (i) To classify a variety of action sequences into respective “correctly performed” and “wrongly performed” classes; (ii) To automatically grade the similitude of an action as compared with a model sequence, as a measure of the actor’s performance level. In the context of quality assessment, the correct sequences (the model set) will typically be the actions of an instructor or a software agent, and the test sequences, which can prove correct or incorrect, will be those of a novice performer.

4.5.1. Classification: Correct vs. Wrong

Given a set of selected model graphs and a set of scene graphs labeled as correct or incorrect, the goal is to discriminate action sequences that are deemed to

be wrongly performed from correctly performed ones. For this purpose, we define a feature vector for each scene graph $G_i^{(s)}$ in terms of the matching energy to the model graphs $\{G_j^{(m)}, j = 1 \dots L\}$ in the library. The feature vector has the form $F_i^{(s)} = \{E_i^j, j = 1 \dots L\}$, where E_i^j is the matching energy in Equation 4.2 between the model graph $G_j^{(m)}$ and the scene graph $G_i^{(s)}$, and L is the number of model graphs. Then using the feature vector, $F_i^{(s)}$, we train a linear Support Vector Machine (SVM) separately for each action type to classify sequences as correctly and wrongly performed. Notice that due to the intra-variation of action sequences, we cannot use a single prototype sequence, but we aggregate via SVM the scores of the distances of the test sequence to all realizations of the model actions. In other words, there are multiple ways of doing the action wrongly, and there are variations among the correct ones.

4.5.2. Quality Assessment

The second task is to gauge the goodness of the performance given that the action has been performed correctly. To this effect we can use the matching energy per sequence, or per frame, per limb, or even per joint. Consider two aligned sequences, where MT is some model sequence and ST some scene sequence aligned to the model sequence. These sequences are characterized by their respective sequences of pose descriptors, namely $MT = \{f_i^{(m)}; i = 1 \dots \bar{M}\}$ and $ST = \{f_i^{(s)}; i = 1 \dots \bar{M}\}$. Our goal is to infer the offsets between the joints in MT and ST

The offset of a component (joint) n of an descriptor between the pair of $f_{i,n}^{(m)}$ and $f_{i,n}^{(s)}$ at a time instant i is computed as

$$\delta_n = |f_{i,n}^{(m)} - f_{i,n}^{(s)}|. \quad (4.8)$$

We define $\Delta = \sum_{n=1}^N \delta_n + \delta_t$ for each skeleton of n joints, δ_t as the time difference, N is the number of joints, and $d = (\Delta_1, \dots, \Delta_{\bar{M}})$ for the entire sequence of \bar{M} frames. A quality measure (QM) is defined in terms of the overall distance, i.e., $QM = \sum_{m=1}^{\bar{M}} \Delta_m$. We have observed, as expected that QM (distance) is proportional to the deterioration

in the action quality.

This distance measure can be also specialized for different subsets of body joints, for example upper body or lower body joints or to only one joint in order pinpoint the deficiencies or differences between the performance of the test subject and that of the instructor.

4.6. Experiments and Results

The proposed framework was tested on MSR Action 3D dataset [5] and WorkoutSU-10 Gesture dataset [6]. Experimental evaluations have demonstrated the utility of the proposed scheme for the following two tasks: action recognition and action quality assessment.

4.6.1. Datasets and Experimental Setup

The experiments are conducted under three settings as follows.

- MSR Action 3D Dataset [5]. MSR Action 3D dataset is one of the early collections recorded with a depth sensor. There are 20 actions performed by 10 subjects. Each subject performs each action 2 or 3 times, resulting in 567 recordings in total. However, we have used 557 recordings in our experiments as in [110]. Each recording contains a depth map sequence with a resolution of 640×480 and the corresponding coordinates of the 20 skeleton joints. Actions are selected in the context of interacting with game consoles, for example, arm wave, forward kick, tennis serve etc.
- Workout SU-10 Gesture dataset [6]. This dataset has the same recording format as MSR Action 3D dataset. However, the context is physical exercises performed for therapeutic purposes. Lateral stepping, hip adductor stretch, freestanding squats, oblique stretch etc. can be given as examples of exercises (actions). There are 15 subjects and 10 different exercises. Each subject repeats an exercise 10 times, resulting in 1500 action sequences in total. In our experiments, we used

600 sequences for training and tested our algorithm on the unseen part of the dataset (we ignored one subject, and tested on 800 sequences).

The action types occurring in MSR Action 3D Dataset [5] and Workout SU-10 Gesture dataset [6] are listed in Table 4.1.

Table 4.1. The action types used in the experiments.

Dataset	Acronym	Action type
MSR Action 3D Dataset [5]	AS1	horizontal arm wave, hammer, forward punch, high throw, hand clap, bend, tennis swing, pick & throw
	AS2	high arm wave, hand catch, draw x, draw tick, draw circle, two hand wave, side-boxing, forward kick
	AS3	high throw, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pick & throw
Workout SU-10 Gesture Dataset [6]	A1	single leg balance with hip flexion
	A2	single leg balance trunk rotation
	A3	lateral stepping
	B1	thoracic rotation bar on shoulder
	B2	hip adductor stretch 1
	B3	hip adductor stretch 2
	C1	dumbbell curl-to-press
	C2	freestanding squats
	C3	transverse horizontal dumbbell punch
	C4	lateral trunk/oblique stretch

- Perturbed dataset. We used Workout SU-10 Gesture dataset [6] for performance assessment. Since the subjective quality labels on this dataset is absent, we created the metadata artificially. We perturbed the positions of selected joints in these sequences with Gaussian noise, and the perturbation level (noise variance) was assumed as the quality metadata. We considered mild to severe perturbations

applied to various subsets of joints, i.e., left/right arm and left/right leg. Ten levels of perturbation strength were used, as represented by standard deviations, $\sigma = 0.1 : 0.15 : 1.5$. Recalling that the spine length of the skeleton was set to 1, can give us an idea of the extent of perturbation. Sample perturbed skeleton sequences are illustrated in Figure 4.4. These perturbations are not constrained in that the kinematics of the arm or leg movements are not taken into account. However for mild perturbations, movements of the skeleton and Figure 4.4 show that the results are plausible and convenient for the performance evaluation under limb joint uncertainties.

Finally, the parameters, λ , W^d , are set on the training set. It should be noted that we applied pose quantization only on the model graphs where we find setting $K = 512$ adequate in pose quantization.

4.6.2. Action Recognition Results

We used nearest neighbor classifier where the distance measure was the matching energy in Equation 4.2. The average performance is found to be 72.9% and 99.5% on MSR Action 3D Dataset and WorkoutSU Gesture dataset, respectively. These results are given in Table 4.2.

The performance when joint subsets were perturbed with additive Gaussian noise are presented in Table 4.3. We observe that the performance degrades gracefully with increasing limb noise.

4.6.3. Comparison with State-of-the-art Methods

For completeness, we compared our algorithm with the state-of-the-art methods [16], [5], [6] which were described in Chapter 2.

In Table 4.4, we tabulated our results on MSR Action 3D Dataset [5]. The performance is computed using a cross-subject test setting where half of the subjects

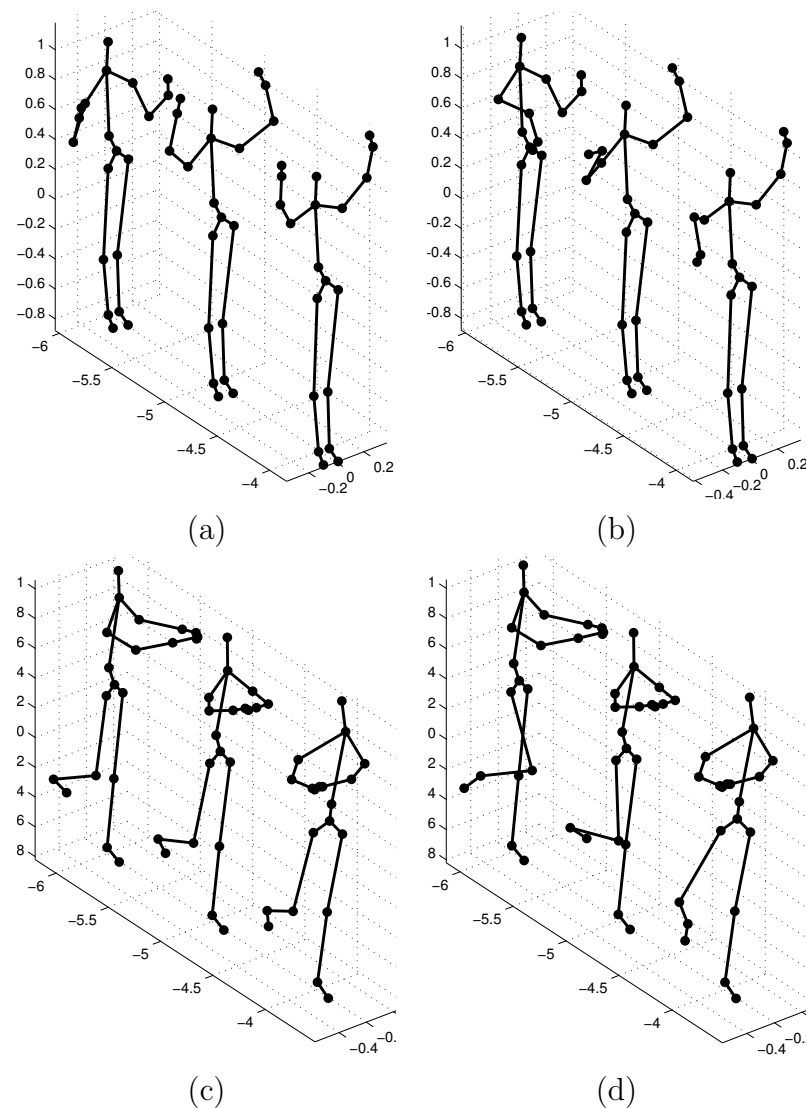


Figure 4.4. Example illustrations. (a)-(b) Respective original sequence and perturbed sequence of action type C1; (c)-(d) Respective original sequence and perturbed sequence of action type A2. Note that, while (b) is a severe noise case, (d) is an example of mild noise case.

were used for training and testing was conducted on the unseen portion of the subjects. Li *et al.* [5] proposed to divide the dataset into three subsets in order to reduce the computational complexity during training. Each action set (AS) consists of eight action classes similar in context. We repeated the same experimental setup 100 times where, each time, we randomly selected five subjects for training and used the rest for testing. Finally, we reported the average recognition rate over all repetitions as well as the

Table 4.2. Recognition performances (%) on MSR: MSR Action 3D Dataset and WSU: WorkoutSU Gesture datasets. L is the number of model graphs used in the experiments.

Dataset	Number of action classes	Performance w/o Graph Mining	Performance w/ Graph Mining
MSR	20	71.8 ($L = 291$)	72.9 ($L = 271$)
WSU	10	94.9 ($L = 600$)	99.5 ($L = 507$)

Table 4.3. Recognition performances (%) under additive Gaussian noise. We set σ value to 0.1, 0.5 and 1 for mild, medium and severe noise cases, respectively.

Perturbation Type	Performance w/o Graph Mining	Performance w/ Graph Mining
Mild noise	93.4	98.8
Medium noise	86.4	96.2
Severe noise	75.7	87.8
No perturbation	94.9	99.5

corresponding standard deviations in the last column of Table 4.4. As seen Table 4.4, the proposed method performs better in AS1 and AS2, and has a competitive performance in AS3. Our proposed method is more successful in overall performance, especially in discriminating actions with similar movements.

In Table 4.5, we compared our results on Workout SU-10 Gesture dataset [6] with the method proposed in [6]. In Table 4.2, we have used 14 subjects as opposed to 12 subjects by Negin *et al.* [6]. For a fair comparison, we used the same experimental setup, namely, cross-subject test setting where we used the same six subjects for training and the same remaining six subjects for testing as in [6]. As seen, our proposed method with graph mining scheme performs better as compared with Negin *et al.* [6].

Table 4.4. Recognition performances (%) for MSR Action 3D Dataset [5] in cross-subject test setting. The respective standard deviations are 5, 7.5, and 13.2 for the last column.

Action Set	Venkataraman <i>et al.</i> [16]	Li <i>et al.</i> [5]	Proposed Method w/o Graph Mining
AS1	77.5	72.9	84.5
AS2	63.1	71.9	85.0
AS3	87.0	79.2	72.2
Overall	75.9	74.7	80.5

4.6.4. Quality Assessment Result

4.6.4.1. Classification results. In Table 4.6, we present the performance of our framework for discriminating correctly performed and wrongly performed sequences. We consider a set of 400 sequences. We obtained a set of wrongly performed sequences by distorting each sequence with $\sigma = 0.5, 1$, which results in 800 sequences in total. We trained a separate SVM classifier for each action class on the features as described in Section 4.5 and used a leave-one-subject-out test setting. Our average classification performance is found to be 86.6%.

4.6.4.2. Regression analysis. The rendition of the action obviously deteriorates with the addition of the joint noise and we expect our action quality measure, that is, the calculated total distance between two matched sequences, to be proportional. We used a regression analysis between the noise variance and the matching energies to the model sequences. We considered 10 different σ values and four different body parts (left/right arm and left/right leg) and generated 3600 different sequences for action classes A1, A2 and A3. We matched each perturbed sequence with its original sequence and calculated the quality measure, QM , as in Section 4.5. As discussed before, in the absence of human evaluations, in order to find the relationship with the calculated distance and the perturbation variances, we applied Support Vector Regression (SVR) with

Table 4.5. Recognition performances (%) for Workout SU-10 Gesture dataset [6] in cross-subject test setting.

Action	Negin <i>et al.</i> [6]	Proposed Method w/o Graph Mining	Proposed Method w/ Graph Mining
A1	100	96.7	100
A2	93.3	100	100
A3	98.3	100	100
B1	98.3	100	100
B2	96.6	100	100
B3	100	73.3	98.3
C1	100	100	100
C2	98.3	100	100
C3	100	100	100
C4	95.0	91.5	98.3
Overall	98.0	96.1	99.6

Table 4.6. Classification performance of the proposed framework: correctly performed sequences vs. wrongly performed sequences. Overall classification performance is 86.6%.

	Correct. Perform.	Wrong. Perform.
Correct. Perform.	85.1	14.9
Wrong. Perform.	11.8	88.2

polynomial kernel variety. Figure 4.5 verifies the relationship between the calculated quality score (distance) and the simulated mismatch (perturbation variance).

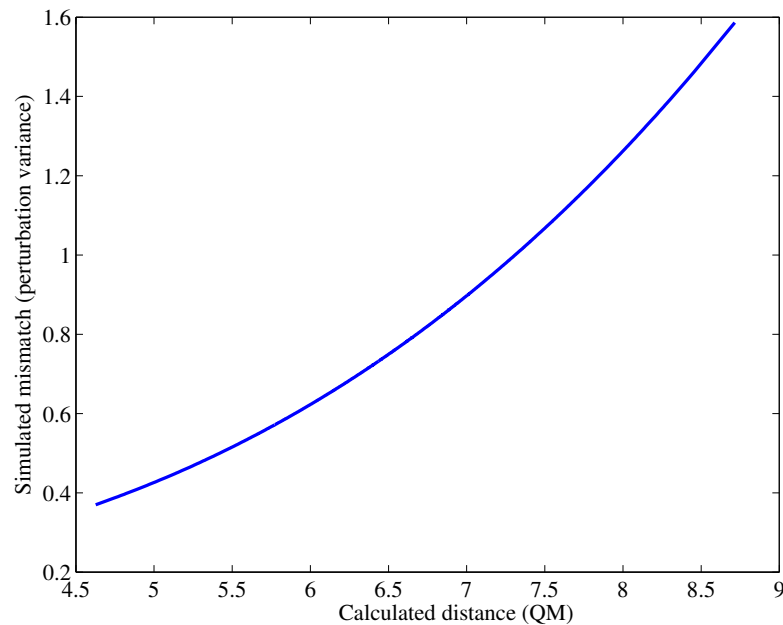


Figure 4.5. Calculated distance, QM , vs. simulated mismatch (perturbation variance).

4.7. Summary

In this chapter, we extended the graph-based action recognition scheme to skeleton pose sequences, and in addition we developed an action quality assessment method. We also introduced an effective algorithm to select the most discriminative prototypes in the training set. The experimental results show that the graph matching scheme has a competitive action recognition capability, it proves to be useful as an action quality monitor on a continuous scale, and that the prototype selection improves the overall performance.

We demonstrated the viability of the proposed algorithm on MSR Action 3D dataset [5] and Workout SU-10 Gesture dataset [6]. In our experiments, we obtained 72.9% performance on the more challenging and noisier MSR Action 3D dataset and

99.5% performance on Workout on the SU-10 Gesture datasets. In either case, our method surpasses the performance of its counterparts, which are based on correlation [6, 16] and BoW [5], respectively.

For action quality assessment experiments, in the absence of scored and labeled data, we created a perturbed dataset where the noise variance was interpreted as the meta-data. Our regression results showed that the action perturbation (noise variance) and the distance measure between a scene sequence and its aligned model sequence has a plausible relationship. This distance measure has been used for separating acceptable (original) and unacceptable (severely noise perturbed) action performances.

Finally, since the Kinect system delivers the skeleton joint positions in real-time, in fact our scheme can be used to give instantaneous feedback on the quality of action, whether over a portion of the action or a subset of joints.

5. MIXTURE OF HIDDEN MARKOV MODELS

Hidden Markov Models (HMM) is a fundamental statistical method for dealing with temporal patterns, as is expected, it has been widely used in the action recognition domain. We had already cited several instances of its use in modeling human movements [96, 104, 108, 187] (please refer to Chapter 2). In this chapter, we apply Hidden Markov Models (HMMs) to the action recognition problem, where the contribution consists of an alternative and novel parameter learning algorithm. We propose a spectral learning method for efficiently estimating the parameters of the Hidden Markov Models (HMM) as well as those of the Mixtures of Hidden Markov Models (MHMM). We demonstrate the viability of the proposed algorithm on video sequences like those in the KTH dataset [1].²

The parameter learning task in HMM can be formulated as finding a set of parameters that maximizes the likelihood of the training sequences. This formulation, however, requires the evaluation of an intractable integral over the parameters. There is no algorithm in the open literature for solving this problem exactly. As a consequence, popular learning algorithms derive a local maximum likelihood by using search heuristics, for example, Expectation-Maximization algorithm. Although this algorithm is efficient, it does not guarantee the global optimum as it suffers from vagaries of initializations and inadequate number of iterations. To this effect, spectral learning algorithm has been proposed as a viable and computationally efficient alternative to EM approach for maximum likelihood estimation. Given a sufficient amount of data, Hsu *et al.* [188] have recently shown that hidden variables such as the model parameters can be estimated by only low-order moments of the observed sequence. In this chapter, we investigate the practical deployment of this spectral method [188] against its counterpart EM algorithm.

Previous works [96, 187] represented each action class by a single Hidden Markov Model. However, a single representative model may not suffice in that, in reality,

²The content of this chapter was joint work with Yusuf Cem Sübakan.

human action instantiations show great variability depending upon the subject, the environment, viewing angle etc. In this context, there is a good reason to use mixture models which is a good approach to handle data containing multiple subgroups. In Section 5.2.1, we show that the extended version of the spectral method Hsu *et al.* [188] can be used for learning *infinite* Mixtures of Hidden Markov Models (MHMM) where the number of subgroups is unknown. We experimentally validate that learning infinite mixture models of action classes improves the classification accuracy.

The rest of this chapter is organized as follows. First, we define Hidden Markov Models and present the parameter learning by the EM approach and the spectral method. Then, we introduce the mixture of HMMs in conjunction with the spectral learning algorithm. Finally, the experimental results highlight the potential impact of the proposed algorithm on the KTH action dataset [1].

5.1. Hidden Markov Models

The Hidden Markov Model defines a probability distribution over sequences of hidden states (h_t) and observation (x_t). An HMM can be formally defined with the following elements [189]:

- Hidden states, h_t , that can take values from $\{1, \dots, M\}$ at a time instant t , where M is the number of states.
- Observations, x_t , that can take values from $\{1, \dots, N\}$ at a time instant t , where N is the number of observations and $M \leq N$.
- State transition probability matrix

$$T_{ij} = p(h_t = i | h_{t-1} = j), \quad T \in \mathbb{R}^{M \times M} \quad (5.1)$$

where $T_{ij} \geq 0$ and $\sum_j T_{ij} = 1$ for $j \in \{1, \dots, M\}$.

- Observation probability matrix

$$O_{ij} = p(x_t = i | h_t = j), \quad O \in \mathbb{R}^{N \times M}. \quad (5.2)$$

- Initial state probabilities

$$\boldsymbol{\pi} = [\pi_i] \quad \text{where } \pi_i = p(h_1 = i). \quad (5.3)$$

The parameter set of an HMM is denoted as $\lambda = \{O, T, \boldsymbol{\pi}\}$, since M and N are implicitly defined in the other parameters. Directed graphical model representing an HMM is given in Figure 5.1a. Given a training sequence, $\mathbf{x} = \{x_1, \dots, x_S\}$, and a hidden state sequence, $\mathbf{h} = \{h_1, \dots, h_S\}$, the joint probability can be calculated by

$$p(\mathbf{x}, \mathbf{h}|\lambda) = p(h_1|\boldsymbol{\pi}) \cdot \left(\prod_{t=2}^S p(h_t|h_{t-1}, T) \right) \cdot \left(\prod_{t=1}^S p(x_t|h_t, O) \right). \quad (5.4)$$

We are interested in estimating the model parameters that maximizes the probability generating \mathbf{x} , namely, to find λ^* that maximizes the likelihood $p(\mathbf{x}|\lambda)$. The likelihood can be obtained by marginalizing the joint probability, namely, summing up over all possible \mathbf{h} 's:

$$p(\mathbf{x}|\lambda) = \sum_{\text{all } \mathbf{h}} p(\mathbf{x}, \mathbf{h}|\lambda). \quad (5.5)$$

5.1.1. Learning Model Parameters of HMM

In this section, we present two methods for learning model parameters: (i) classical maximum likelihood based learning by Expectation-Maximization procedure and (ii) spectral learning based on the method of moments.

5.1.1.1. Expectation-Maximization algorithm. To learn the optimal model parameters λ^* , one has to maximize the logarithm of the likelihood:

$$\begin{aligned} \lambda^* &= \arg \max_{\lambda} \log p(\mathbf{x}|\lambda) \\ &= \arg \max_{\lambda} \log \sum_{\mathbf{h}} p(\mathbf{x}, \mathbf{h}|\lambda). \end{aligned} \quad (5.6)$$

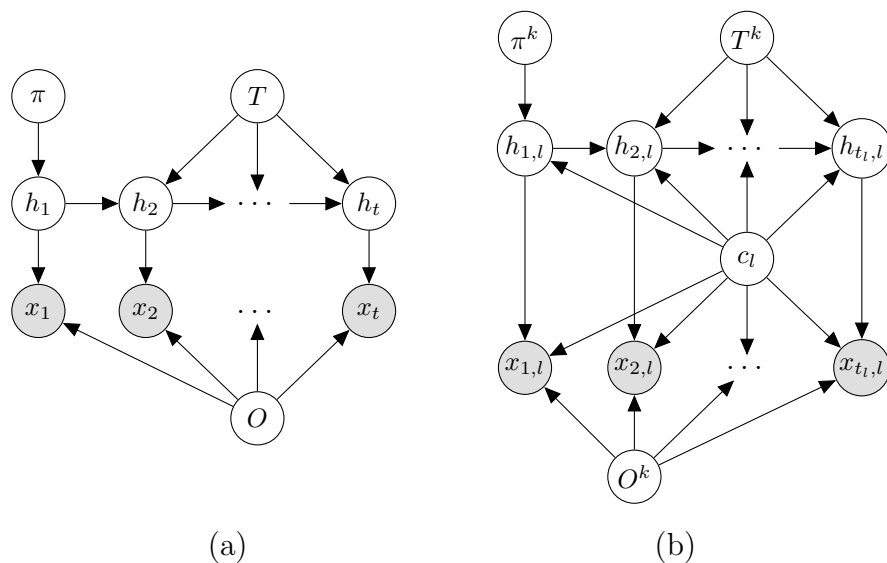


Figure 5.1. Directed Acyclic Graph (DAG) (a) of HMM and (b) of MHMM.

However, there is no closed form solution for this maximization problem since the likelihood $p(\mathbf{x}|\lambda)$ is defined with a summation over the hidden variables \mathbf{h} that prevents the logarithm to act directly on the distribution. Instead, given λ , we first find a lower bound for Equation 5.5, and then maximize this lower bound. For a valid probability distribution $q(\mathbf{h})$, we can define the lower bound of Equation 5.5 by the following function:

$$\sum_{\mathbf{h}} q(\mathbf{h}) \log \frac{p(\mathbf{x}, \mathbf{h}|\lambda)}{q(\mathbf{h})} \quad (5.7)$$

due to Jensen's inequality [190].

Given some initial values for model parameters denoted by λ^{old} , Expectation-Maximization algorithm is a two-step iterative procedure. In Expectation (E) step, we choose $q(\mathbf{h})$ such that Equation 5.5 and Equation 5.7 are equal for λ^{old} . This is achieved by setting $q(\mathbf{h}) = p(\mathbf{h}|\mathbf{x}, \lambda^{old})$ for every possible state sequence \mathbf{h} . In Maximization (M) step, we look for the new parameter values, λ^{new} , that maximizes Equation 5.7 such that conditions on λ^{new} are satisfied. EM algorithm is summarized in Figure 5.2.

It should be noted that we give a naive approach to EM where there are M^S

possible \mathbf{h} in the E step, which is not practical. This can be efficiently done by exploiting the conditional independence properties of the HMM using forward-backward algorithm. The details can be found in [191].

EM is an efficient algorithm, however, learning parameters of an HMM using EM is a non-convex problem with many local maxima. EM will converge to any local maximum based on the initial values, therefore one has to run the algorithm multiple times with different initial values.

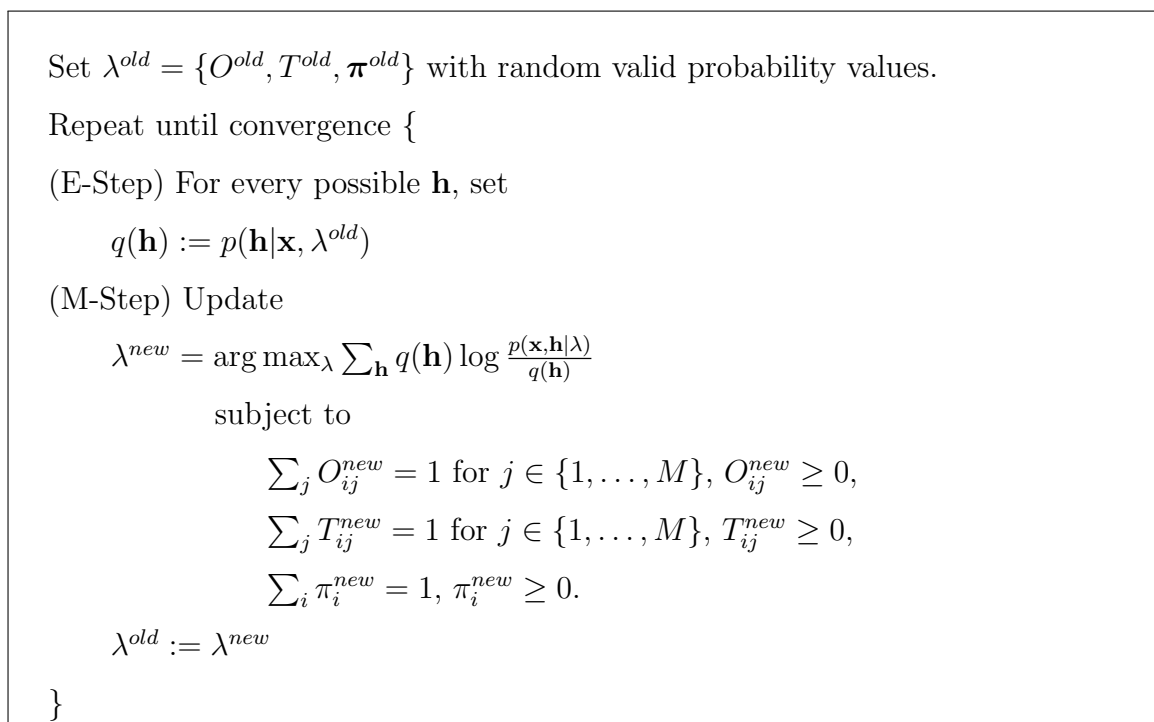


Figure 5.2. Expectation-Maximization algorithm for HMM.

5.1.1.2. Spectral algorithm. Spectral methods have been proposed as an alternative to overcome the shortcomings of EM and to bypass the tedious estimation of the maximum likelihood. They basically operate on the eigen-structure of the empirical (observable) moments to elicit the underlying model parameters and assure a unique accurate estimation of them given a sufficient number of training samples. We first present how such a connection between the low-order observable moments and the model parameters can be established in the case of an HMM based on the work proposed by Hsu *et al.* [188, 192], so called “method of moments”.

Method of moments [188,192] learns a representation that is based on the following observable operator view:

Lemma 5.1. For $x = 1, \dots, N$ define

$$A_x = T \text{diag}(O(x, :)) \quad (5.8)$$

where $O(x, :)$ denotes the x th row of the matrix O . Given an observed sequence \mathbf{x} and the parameters λ :

$$p(\mathbf{x}|\lambda) = \mathbf{1}_M^T A_{x_S} \dots A_{x_1} \boldsymbol{\pi} \quad (5.9)$$

where $\mathbf{1}_M$ is a vector of all ones of length M .

Proof.

$$p(\mathbf{x}|\lambda) = \sum_{h_{S+1}} \sum_{h_S} p(h_{S+1}|h_S) p(x_S|h_S) \dots \sum_{h_2} p(h_3|h_2) p(x_2|h_2) \underbrace{\sum_{h_1} p(h_2|h_1) p(x_1|h_1) p(h_1)}_{A_{x_1} \boldsymbol{\pi}}.$$

$$\underbrace{\hspace{15em}}_{A_{x_2} A_{x_1} \boldsymbol{\pi}}$$

$$\underbrace{\hspace{25em}}_{A_{x_S} \dots A_{x_2} A_{x_1} \boldsymbol{\pi}}$$

$$\underbrace{\hspace{30em}}_{\mathbf{1}_M^T A_{x_S} \dots A_{x_2} A_{x_1} \boldsymbol{\pi}}$$

□

We assume that the HMM obey the following condition.

Assumption 5.2. $\boldsymbol{\pi} > 0$ element wise, and O is rank M .

The conditions on $\boldsymbol{\pi}$ requires that each state has non-zero probability. This is satisfied if the Markov chain specified by T is ergodic and $\boldsymbol{\pi}$ is its stationary distribution. The condition on O prevents the ambiguity in which a state i has an output distribution equal to a convex combination of other states' output distributions, and thus

guarantees an explicit expression of the model parameters in terms of the observable quantities.

Following this, the HMM representation is defined by the observable quantities, namely the marginal probabilities of observation singletons, pairs and triples:

$$(P_1)_i = p(x_1 = i) \quad (5.10)$$

$$(P_{3,1})_{ij} = p(x_3 = i, x_1 = j) \quad (5.11)$$

$$(P_{3,x,1})_{ij} = p(x_3 = i, x_2 = x, x_1 = j), \quad (5.12)$$

where $P_1 \in \mathbb{R}^N$ is a vector, and $P_{3,1}, P_{3,x,1} \in \mathbb{R}^{N \times N}$ are the matrices. Moreover, it is straightforward to show the connection between these observable quantities and the model parameters:

$$P_1 = O\boldsymbol{\pi} \quad (5.13)$$

$$P_{3,1} = OTT \text{diag}(\boldsymbol{\pi}) O^T \quad (5.14)$$

$$P_{3,x,1} = OA_x T \text{diag}(\boldsymbol{\pi}) O^T. \quad (5.15)$$

The representations further depends on the matrices $U \in \mathbb{R}^{N \times M}$ and $V \in \mathbb{R}^{N \times M}$ that obeys the following condition and lemma.

Assumption 5.3. $U^T O$ and $V^T O$ are invertible.

Lemma 5.4. Assume $\boldsymbol{\pi} > 0$, and O and T are rank M . Let $U \in \mathbb{R}^{N \times M}$ and $V \in \mathbb{R}^{N \times M}$ be matrices such that both $U^T O$ and $V^T O$ are invertible. Then $U^T P_{3,1} V$ is invertible, and for all x , the observable operator, $B_x \in \mathbb{R}^{M \times M}$, is given by

$$B_x := (U^T O T) \text{diag}(O(x, :)) (U^T O T)^{-1}. \quad (5.16)$$

Then, B_x can be written in terms of the observable quantities

$$B_x = (U^T P_{3,x,1} V) (U^T P_{3,1} V)^{-1}. \quad (5.17)$$

Proof. Since $P_{3,1} = OTTdiag(\boldsymbol{\pi})O^T$, we can express $U^T P_{3,x,1}V$ as

$$U^T P_{3,x,1}V = U^T O A_x T diag(\boldsymbol{\pi}) O^T V \quad (5.18)$$

$$= U^T OT diag(O(x, :)) T diag(\boldsymbol{\pi}) O^T V \quad (5.19)$$

$$= U^T OT diag(O(x, :)) (U^T OT)^{-1} (U^T OT) T diag(\boldsymbol{\pi}) O^T V \quad (5.20)$$

$$= (U^T OT) diag(O(x, :)) (U^T OT)^{-1} U^T P_{3,1} V. \quad (5.21)$$

Thus, $B_x = (U^T P_{3,x,1}V)(U^T P_{3,1}V)^{-1}$ is satisfied. \square

The algorithm is motivated by Lemma 5.4 in that we compute the Singular Value Decomposition (SVD) of an empirical estimate of $P_{3,1}$ to discover U and V that satisfy Assumption 5.3. More explicitly, B_x reveals T and O as follows. First, we independently sample a number of independent observation pairs and triples, $p(x_3, x_1)$ and $p(x_3, x_2, x_1)$, to form empirical estimates of $P_{3,1} \in \mathbb{R}^{N \times N}$ and $P_{3,2,1} \in \mathbb{R}^{N \times N \times N}$. We then compute the SVD of $P_{3,1}$ to discover U and V that are the left and right singular vectors that corresponds to the M largest singular values and recover B_x from $(U^T P_{3,x,1}V)(U^T P_{3,1}V)^{-1}$. Note that, if we compute the eigen-decomposition of B_x , eigenvalues are the x th row of O and T can be estimated from the eigenvectors, $U^T OT$. Since we admit that $\boldsymbol{\pi}$ is the steady state distribution in Assumption 5.2, we can estimate it as the eigenvector of T associated to the eigenvalue 1.

We summarize the spectral algorithm for estimating parameters in Figure 5.3.

5.2. Mixtures of Hidden Markov Models

Mixtures of Hidden Markov Models (MHMMs) can be interpreted as the combination of K independent “standard” HMMs. A MHMM is illustrated in Figure 5.1b. The key point is to introduce a hidden cluster indicator, c_l , where $c_l = k$ indicates that the observation sequence \mathbf{x}_l is generated from the k th HMM with a transition matrix T^k and an observation matrix O^k . Given L observation sequences, to estimate

1. Independently sample a number of observation pairs and triples to form empirical estimates $\hat{P}_{3,1}$ and $\hat{P}_{3,2,1}$;
 2. Compute the SVD of $\hat{P}_{3,1}$, and let \hat{U} and \hat{V} be the matrix of left and right singular values corresponding to the M largest singular values;
 3. Estimate the first row of \hat{O} matrix and $R = \hat{U}^T \hat{O} \hat{T}$ by computing the eigen-decomposition of $B_1 = R \text{diag}(O(1, :)) R^{-1}$, namely, $\hat{O}(1, :) = R^{-1} B_1 R$;
- for** $n = 2$ to N **do**
- $$\hat{O}(n, :) = R^{-1} B_n R;$$
- end for**
- $$temp = (\hat{U} \hat{O})^{-1} R;$$
4. Normalize $temp$ to have columns sum up to one, and set $\hat{T} = temp$.

Figure 5.3. HMM spectral learning algorithm.

the model parameters, one has to maximize the following likelihood over all cluster parameter sets:

$$p(\mathbf{x}_1, \dots, \mathbf{x}_L | \lambda^1, \dots, \lambda^K) = \sum_{c_l} \sum_{\mathbf{h}_l} \prod_{l=1}^L p(\mathbf{x}_l, \mathbf{h}_l, c_l | \lambda^1, \dots, \lambda^K). \quad (5.22)$$

For this maximization problem, an EM algorithm can be derived similar to Section 5.1.1.1. The only difference is the additional summation over c_l terms. However, for each iteration, the evaluation of the E step increases linearly with the number of clusters, K . Below, we propose an alternative method that combines k-means clustering method and spectral learning approach.

5.2.1. Spectral Learning of Mixture of HMMs

The extension of the method of moments to the Mixtures of HMMs is not straightforward. We therefore use method of moments as a subroutine in a K-means type algorithm. The basic steps of the algorithm can be summarized as follows:

- (i) For each action class, assume N training sequences \mathbf{x}_n and K independent standard HMMs. Randomly initialize the cluster indicator variables $\mathbf{c} = \{c_1, \dots, c_N\}$.
- (ii) Given \mathbf{c} , compute empirically $P_{3,1}^k$ and $P_{3,2,1}^k$, $k = 1, \dots, K$, and estimate the model parameters λ^k of each HMM using the spectral algorithm given in Figure 5.3.
- (iii) Update the cluster-training sequence assignments, in other words, assign each training sequence, x_n , to the cluster that has the best fitting model parameters:

$$c_n = \arg \max_k p(\mathbf{x}_n | \lambda^k). \quad (5.23)$$

- (iv) Repeat steps 2 and 3 until convergence.

It should be noted that, in a finite mixture model, number of components has to be specified apriori. However, in reality, we do not have such an information about the number of subgroups available in the data. The algorithm can be extended to infinite mixture models by defining a term that creates a new, non-pre-existing cluster, if the available cluster model parameters can not model the training sequence well enough. This term can be simply characterized by $p(\mathbf{x}_n | \lambda^{all})$ where λ^{all} can also be empirically learned from all training samples, $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, regardless of the clusters. More explicitly, if $p(\mathbf{x}_n | \lambda^{all}) > p(\mathbf{x}_n | \lambda^k)$, $\forall k$, we create a new cluster $K + 1$. By extension, if one of the clusters becomes empty, we decrease the number of clusters by one, and relabel the remaining ones.

5.3. Application to Human Action Recognition

In this section, we demonstrate the practical usage of the proposed approaches on the human action recognition problem from video sequences. First, we briefly mention action description and observation model selection, and then continue with the experimental results.

5.3.1. Spatio-temporal Interest Point Detection and Action Description

For each action frame, we first place a bounding box on the human body, and then, divide the box into 9 blocks. We characterize a box by the number of detected

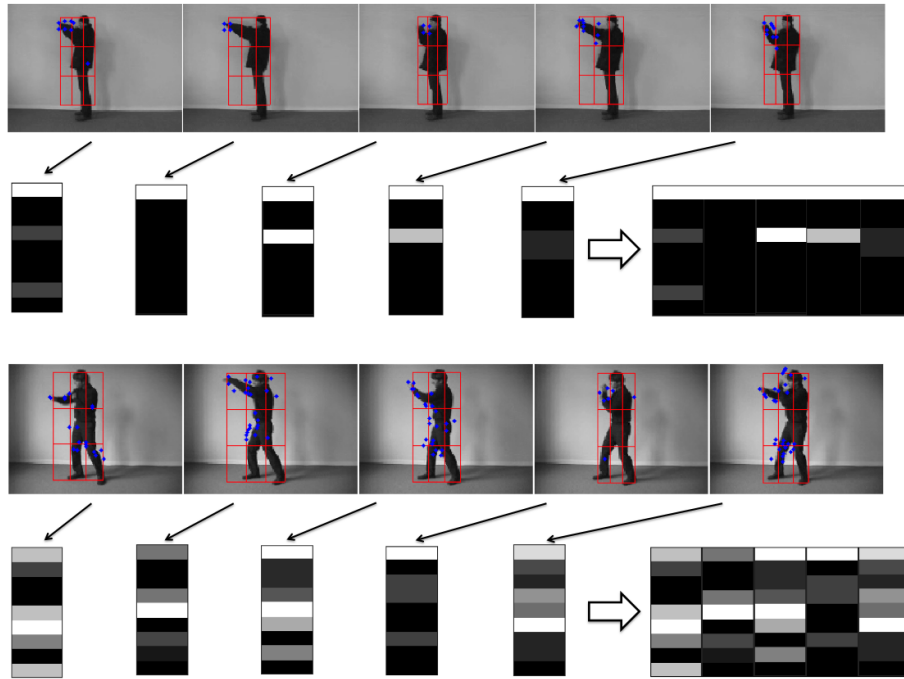


Figure 5.4. Each box (frame) is described by the number of detected STIPs within the blocks. An action can be regarded as a sequence of boxes.

spatio-temporal interest points (STIPs) within each block. Yamato *et al.* [187] also used a similar descriptor in conjunction with HMMs. They first converted each frame to a binary image which is then divided into blocks, and count the occurring 0/1 pixels within each block. Differently, we treat the number of STIPs within each box as the observations x_t at a time instant t . This procedure is illustrated in Figure 5.4. As shown, consider two “boxing” action instantiations. In the first sequence, there is no motion in the legs, whereas in the second we have substantial leg movement. Training a Mixture of HMMs for each class can effectively capture the within-class-variance. The box and interest points are automatically extracted by using the off-the-shelf code in [58]. Finally, we opt to use Poisson distribution to model the observation since each observation or box is represented by the STIPs count.

The rationale to use a rather simple set of features to describe human actions is to be able to run the HMM model as a proof of concept.

5.3.2. Experimental Results

We tested the proposed approaches on KTH dataset [1]. For each action class, we use 64 sequences for training and 36 sequences for testing. We present our results in three settings:

- (i) HMM with EM learning: We train a single HMM per action class and learn the model parameters by EM. We assign a test sequence to the highest likelihood action class.
- (ii) HMM with spectral learning: We train a single HMM per action class and learn the model parameters by spectral method. We assign a test sequence as in (i).
- (iii) MHMM with spectral learning: We train more than one HMM per action class and learn the model parameters by spectral method. We assign a test sequence based on the likelihoods averaged over the clusters pertaining to a certain action class.

The corresponding results are tabulated in Table 5.1. In both of the learning approaches, it is obvious that learning Mixture of HMMs significantly increases the recognition performance. In the case of a single HMM, spectral learning is a lagging runner up behind EM. However, this gap is compensated in learning MHMMs. More importantly, given this accuracy, the run-time of the spectral learning is competitive. For example, for learning a MHMM, if we use EM, each iteration takes 7.5 seconds. On the other hand, 3.1 seconds is needed per iteration, if we use spectral learning for parameter estimation.

5.4. Summary

In this chapter, we examined the practical deployment of HMM in handling action recognition problem from a parameter learning perspective. For this purpose, we compared two parameter learning approaches, EM and spectral learning. First, we show that spectral learning can be competitive as EM, moreover, reduced computational time and being insensitive to initialization are an added bonus. Secondly, we demon-

strated that training Mixtures of HMMs per class effectively captures the within-class variations, and consequently increases the recognition performance significantly.

In this chapter, the best performance, 74.0%, was obtained by MHMMs on KTH action dataset [1]. This performance, which is well below the state of the art (Chapter 3) should be interpreted with the notion that it is only a proof of concept result. In fact, we have used a rather a rather simple set of features to only demonstrate the competitiveness of the spectral algorithm with respect to the EM method and the capability of MHMMs in handling the inherent variations of human actions. More sophisticated features such as appearance features should be used with the spectral methods to prove its true performance.

Table 5.1. Confusion matrices for (a) HMM with EM learning; (b) HMM with spectral learning; (c) MHMM with spectral learning. Respective recognition accuracies: 70.6%, 66.5%, and 74.0%. (B: Box, HC: Handclap, HW: Handwave, J: Jog, R: Run, W: Walk)

	B	HC	HW	J	R	W
B	89	3	3	0	0	5
HC	11	86	3	0	0	0
HW	3	17	80	0	0	0
J	0	0	0	47	20	33
R	2	2	0	55	30	11
W	0	0	0	8	0	92

	B	HC	HW	J	R	W
B	89	0	3	0	0	8
HC	20	55	25	0	0	0
HW	0	0	100	0	0	0
J	0	0	0	20	30	50
R	3	3	0	20	47	27
W	0	0	0	0	3	97

	B	HC	HW	J	R	W
B	89	3	0	0	0	8
HC	14	69	17	0	0	0
HW	3	0	97	0	0	0
J	0	0	0	36	22	42
R	5	0	0	28	56	11
W	0	0	0	3	0	97

6. CONCLUSION

In this thesis, we have investigated structured and sequential representations of video data for recognizing selected sets of human actions and for measuring their enacting quality. In video sequences, we have characterized each action with a graphical structure of its spatio-temporal interest points and cuboid descriptors, whilst in the case of depth data an action has been represented by the sequence of its skeleton joints. Given these descriptors, we formulated the human action recognition problem through two essential machine learning techniques, Hyper-graph Matching and Hidden Markov Models.

The main achievements, novelties of the thesis research and the potential future directions are summarized in the subsequent Sections 6.1, 6.2 and 6.3. Finally, we draw our conclusions from the state-of-the art literature in Section 6.4.

6.1. Fast Hyper-Graph Matching for Spatio-temporal Data

The widely used graph matching technique provides a powerful solution to point set correspondence problem by combining both sources of information, the configurational and the appearance sources. In the general case, its major drawback is that the useful formulations result in an NP-hard combinatorial problem. In the computer vision literature, as also elaborated in Chapter 3, we have observed that there are two common strategies to achieve efficient graph matching: (i) to approximate the solution and (ii) to approximate the graphical structure.

A significant body of work has been devoted to approximate the solution, e.g., spectral methods, relaxation of the optimization problem, local search heuristics etc. There are several applications of the approximate matching algorithms in the action recognition domain [45, 94, 95]. This strategy usually results in non-optimal solutions and one does not have much of a control in the nature of approximation. However, the strategy of approximating the solution is found to be more appropriate in the case of

large graphs.

The alternative strategy, that is, the methods based on approximating the graphical structures in order to enable an exact optimal solution, results in matching of smaller graphs. We have encountered only few works in the literature that employ the strategy of approximating a graphical structure. An example is the graph structured into a k-tree in [177] for object recognition.

In this context, the main theoretical contributions and novelties of the thesis can be summarized as follows.

- We developed an efficient hyper-graph matching method for spatio-temporal data. A hyper-graph was built using the spatio-temporal relationships between the interest points and the nodes of the graph were attributed cuboid descriptors. The matching problem formulation represents the conversion of the exact minimization problem into a simpler but feasible problem without compromising the optimality of the solution. In this simplification, three realistic assumptions were made: (i) causality of human movements; (ii) sequential nature of human movements; and (iii) one-to-one mapping of time instants. We showed that, under these assumption, the correspondence problem can be decomposed into a set of subproblems such that each subproblem can be solved recursively in terms of the others, and hence derived an exact minimization algorithm. Finally, the minimization problem was efficiently solved by using the dynamic programming technique.
- We further reduced the computational complexity significantly by introducing the notion of sparse graphs. In this case, the reduced graphical structures represent an approximation, but otherwise the graph matching is solved exactly. We proposed three different graphical approximation structures, all characterized by a strong reduction in the number of interest points, in fact, each one assumes very few interest points, i.e., from 1 to 3 points per frame. We show that our algorithm can successfully recognize actions with an order of magnitude reduction in the computation time.

- We found out that the performance improves when the training set is judiciously selected. A lean training set selected with a search algorithm performs quite better as compared to employing the entire set. To this effect, we customized the feature selection algorithm called Sequential Floating Forward Search, for learning a set of representative and discriminative model graph prototypes over a validation set.

As a proof of concept, we first applied the proposed algorithm to action recognition in video sequences. Spatio-temporal interest points were detected by means of 3D Harris corner detector and 2D Gabor filters. As shape and motion descriptors, we utilized histogram of gradients (HoG) and histogram of optical flow (HoF). The hyper-graph matching energy between two point sets, i.e., a model and a scene, was further used for recognition and localization of actions.

We can draw the following conclusions. Hyper-graph-based representation of spatio-temporal interest points adds an important information to the problem formulation and significantly improves the performance as compared to a scrambled collection of spatio-temporal patches. When one is interested in localizing actions or segmenting movements, our matching formulation offers a natural solution. However, the complexity is still high in the case of large graphs. While our method is a good and efficient solution for “one action, one actor and uniform background” paradigm, sparse sampling of the interest points leaves many uncovered volumes in the video which will be detrimental to deducing scene context and to recognizing complex activities.

The proposed algorithm can be improved along several avenues.

- *Learning the parameters.* Recall that our energy formulation in Equation 3.3 is composed of three terms: feature dissimilarity, geometric deformation and time warping penalty. Since each action has a different nature both in space and time, we believe that learning the parameters, λ 's, T^t and W^d , per action category would boost the performance. For example, the parameters can be learned using Support Vector Machines in a similar way as in [46]. An alternative

way to formalize problem could be a model-based approach in a probabilistic setting where the matching of triple of nodes can be modeled by a probabilistic distribution.

- *Smart selection of interest points or triangles.* In our experiments, the single-chain-single-point model works surprisingly well and fast for recognizing periodic actions, e.g., boxing, walking. However, the single-chain-single-point model is limited in capturing local information that could not be discriminative for action classes different despite containing similar movements, for example, drinking coffee and answering phone. On the other hand, the single-chain-multiple-points model has the potential to better capture the global motion of the human body. However, the interest points detected by Gabor filters are found to be less informative. We conjecture that the performance of the single-chain-multiple-points model or multiple chains model can be further increased by learning informative interest points and spatio-temporal triangles for each action category in the spirit of Sequential Floating Forward Search.
- *Descriptors to be explored.* We have used spatio-temporal interest points for matching two action sequences. However, our graphical structure enables to replace the interest points by mid-level or high-level features. For example, in Chapter 4, we used tracked skeleton joints [47] from depth sequences. Another potential features could be the use of pairwise spatio-temporal features as in [68, 69], spatio-temporal regions [45] or human body parts [73, 193].

6.2. Skeleton-based Human Action Analysis

Our proposed hyper-graph matching algorithm assumes a structured representation both in space and time domains. It is therefore closely related to sequence alignment methods in that it exploits the sequential nature of the time in a similar way. In our work, we demonstrated the usefulness of our method for aligning tracked skeleton sequences from depth data. We obtained the skeletons by using the off-the-shelf skeleton tracking system in [47]. Each action sequence is structured into a chain graph where the nodes coincide with the skeleton joints in frames, and the edges model their temporal relationship. More explicitly, each node is associated to exactly one

frame characterized by a skeleton; an edge models the temporal relationship between the joints of the skeleton in neighboring frames. We encoded each node with angle-based and distance-based pose descriptors. To eliminate the confounding model graph prototypes, we proposed a novel prototype selection method that intends to minimize the within-class variance, at the same time, to maximize between-class variance. This approach enabled us to achieve reliable performance both in recognizing actions and in quantifying the action quality.

We conclude that when the spatio-temporal interest points have a semantic meaning as in the case of the joints of the human skeleton, our matching algorithm offers a good solution both for action recognition and sequence alignment problems, hence automatic action quality assessment, and it is also capable of handling noisy joint estimations.

As a future direction, we believe that learning informative joints would play an important role. For example, a weighted alignment formulation can be utilized in the spirit of the weighted Dynamic Time Warping in [111] in which the weights of each joint per class were learned during a training phase. Another potential application could be recognizing the interaction between humans based on their skeletons as in [46].

6.3. Mixture of Hidden Markov Models

Hidden Markov Model (HMM) has been widely used tool for the sequential pattern recognition in the literature. Although Hidden Markov Models for action recognition in video sequences is not a new concept. To the best of our knowledge, we are the first ones to investigate alternative parameter learning approaches for HMMs in the action recognition domain. In the context of our work, our contributions can be summarized as follows.

- We demonstrated that spectral parameter learning algorithm [188] can be a viable algorithm against the conventional Expectation-Maximization (EM) algorithm. The performance of spectral learning algorithm appears modest as compared

to the EM. The most important advantage of the spectral estimation methods is that it is not sensitive to different initializations, i.e., it always guarantee a unique solution. To obtain better performance, spectral method can be used to initialize EM.

- To mitigate the large within-class variance of human actions, we learned a Mixture of Hidden Markov Models (MHMMs) for each action category. Our experimental results verified that MHMMs per action are superior to learning a single HMM. A similar idea was proposed by Felzenszwalb *et al.* [73] where mixture of deformable part-based models assumes multiple components for handling front and side views. As a novelty for the MHMM applications, we automatically learned the number of mixture components from the training samples. To this effect, we proposed a method for learning infinite mixtures of HMMs in the spirit of K-means algorithm. Our experimental results showed that spectral learning method for MHMMs is adequate to infer human actions with the advantage of lower run time.

In our proof of concept approach, we used extremely simple features, that is, the count of interest points within the blocks of the bounding box of the action subject. We believe that integrating rich STIP descriptors such as HoG and HoF [4] in the MHMM scheme should improve the recognition performance results. This is in fact part of the future work.

In conclusion, action recognition has come a long way from its meager beginning with Motion History Images [42] and Bag-of-Words formalism [31] etc. The current trend is to combine spatial configuration, temporal consecutiveness of the descriptors or hierarchical relationships between the descriptors. We believe that we have integrated this aspect carried in our graph matching and sequence alignment formulations.

6.4. Conclusions

Our conclusions drawn from the state-of-the art literature are as follows:

- *Sparse sampling vs. dense sampling.* While the sparse sampling of interest points is easier to handle within the framework of graph models, and it has proven to be adequate for action recognition in the case of single actor and of non-cluttered background, dense sampling seems more powerful to recognize complex activities [82]. Its main advantage is that dense sampling with a regular grid on the spatio-temporal domain guarantees coverage of the entire object and exploitation of the scene context. However, dealing with dense features brings significantly higher computational load, and it is not obvious how one can handle the cloud of spatio-temporal interest points. One possible way to proceed would be to find similarities of spatio-temporal interest point clouds using thin plate spline based robust point matching [194]; alternatively, one can consider the heat kernel signature approach [195].
- *Higher-level descriptors.* Until recently, the majority of action recognition works have been using low-level descriptors, e.g., most of them in the form of spatio-temporal center-surround contrast points, that is, spatio-temporal saliencies. In recent years, however, pose estimation techniques have grown exponentially in number, both in the RGB and the depth modalities. Integration high level descriptors, such as poselets, human pose estimation, detection of scene objects, has been shown to improve the understanding of the scene and the recognition of complex activities. Though extraction of high level descriptors is a challenging task per se. In this context, mid-level features [45, 67] seems a promising research direction. We believe it is worthwhile exploring alternative approaches to integrate them into the graphical structure approach.
- *Multi-modal data.* With the development of low-cost depth cameras, RGBD datasets that combine intensity and depth sequences have become an emerging field. Depth data remedies the limitations of the intensity sequences and is promising for handling multiple views, variability in clothes, cluttered backgrounds etc. Works that jointly exploit spatio-temporal depth and appearance data have started to appear. Recently built realistic RGBD datasets will boost the research in this direction.
- *Complexity level of the problems.* The action recognition problem can be considered to be solved for single action, single actor and uniform background case, at

least in the view of current databases such as Weizman [124], KTH [1]. The solution of this restricted problem has the potential to benefit certain applications in human-machine interaction, for example, the gaming industry. On the other hand, for more complex and realistic scenes involving arbitrary views, multiple actors, complex activities and cluttered background, the current technology is very far from a general robust solution stage.

REFERENCES

1. Schuldt, C., I. Laptev and B. Caputo, “Recognizing Human Actions: A Local SVM Approach”, *International Conference on Pattern Recognition*, pp. 32–36, 2004.
2. Laptev, I., “Hollywood2: Human Actions and Scenes Dataset”, <http://www.irisa.fr/vista/actions/hollywood2/>, 2013, accessed at July 2013.
3. Wolf, C., J. Mille, E. Lombardi, O. Çeliktutan, M. Jiu, M. Baccouche, E. Delandrea, C.-E. Bichot, C. Garcia and B. Sankur, *The LIRIS Human Activities Dataset and the ICPR 2012 Human Activities Recognition and Localization Competition*, Tech. rep., LIRIS Laboratory, 2012.
4. Laptev, I., M. Marszalek, C. Schmid and B. Rozenfeld, “Learning Realistic Human Actions from Movies”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.
5. Li, W. Q., Z. Y. Zhang and Z. C. Liu, “Action Recognition Based on a Bag of 3D Points”, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 9–14, 2010.
6. Negin, F., F. Özdemir, C. B. Akgül, K. A. Yüksel and A. Erçil, “A Decision Forest Based Feature Selection Framework for Action Recognition from RGB-Depth Cameras”, *Image Analysis and Recognition*, Vol. 7950 of *Lecture Notes in Computer Science*, pp. 648–657, 2013.
7. Cristani, M., R. Raghavendra, A. D. Bue and V. Murino, “Human Behavior Analysis in Video Surveillance: A Social Signal Processing Perspective”, *Neurocomputing*, Vol. 100, No. 7, pp. 86–97, 2013.
8. Choi, W. and S. Savarese, “A Unified Framework for Multi-target Tracking and

- Collective Activity Recognition”, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato and C. Schmid (Editors), *European Conference on Computer Vision*, Vol. 7575 of *Lecture Notes in Computer Science*, pp. 215–230, 2012.
9. Lan, T., W. Yang, Y. Wang and G. Mori, “Beyond Actions: Discriminative Models for Contextual Group Activities”, *Proceedings of Neural Information Processing Systems*, 2010.
 10. Kaneko, T., M. Shimosaka, S. Odashima, R. Fukui and T. Sato, “Consistent Collective Activity Recognition with Fully Connected CRFs”, *21st International Conference on Pattern Recognition*, pp. 2792–2795, 2012.
 11. Jacques Junior, J. C. S., S. Raupp Musse and C. R. Jung, “Crowd Analysis Using Computer Vision Techniques”, *Signal Processing Magazine, IEEE*, Vol. 27, No. 5, pp. 66–77, 2010.
 12. Rodriguez, M., S. Ali and T. Kanade, “Tracking in Unstructured Crowded Scenes”, *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 1389–1396, 2009.
 13. Popoola, O. P. and K. Wang, “Video-Based Abnormal Human Behavior Recognition—A Review”, *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, Vol. 42, No. 6, pp. 865–878, 2012.
 14. Sarkar, S., P. J. Phillips, Z. Liu, I. R. Vega, P. Grother and K. W. Bowyer, “The humanID Gait challenge problem: data sets, performance, and analysis”, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 27, No. 2, pp. 162–177, 2005.
 15. Auvinet, E., F. Multon, A. Saint-Arnaud, J. Rousseau and J. Meunier, “Fall Detection With Multiple Cameras: An Occlusion-Resistant Method Based on 3-D Silhouette Vertical Distribution”, *Information Technology in Biomedicine, IEEE Transactions on*, Vol. 15, No. 2, pp. 290–300, 2011.

16. Venkataraman, V., P. Turaga, N. Lehrer, M. Baran, T. Rikakis and S. L. Wolf, “Attractor-Shape for Dynamical Analysis of Human Movement: Applications in Stroke Rehabilitation and Action Recognition”, *International Workshop on Human Activity Understanding from 3D data*, 2013.
17. Rehg, J. M., G. D. Abowd, A. Rozga, M. Romero, M. A. Clements, S. Sclaroff, I. Essa, O. Y. Ousley, Y. Li, C. Kim, H. Rao, J. C. Kim, L. L. Presti, J. Zhang, D. Lantsman, J. Bidwell and Z. Ye, “Decoding Children’s Social Behavior”, *IEEE International Conference on Computer Vision and Pattern Recognition*, 2013.
18. Forsyth, D. A., O. Arikan, L. Ikemoto, J. O’Brien and D. Ramanan, “Computational Studies of Human Motion: Part 1, Tracking and Motion Synthesis”, *Foundations and Trends in Computer Graphics and Vision*, Vol. 1, No. 2-3, pp. 77–254, 2005.
19. Keskin, C., E. Berger and L. Akarun, “A Unified Framework for Concurrent Usage of Hand Gesture, Shape and Pose”, *IEEE Conference on Computer Vision and Pattern Recognition, Workshop on Gesture Recognition*, 2012.
20. EyeSight, “EyeSight Solutions”, <http://eyesight-tech.com>, accessed at July 2013.
21. Pentland, A., “Smart Rooms, Smart Clothes”, *International Conference on Pattern Recognition*, Vol. 2, pp. 949–953, 1998.
22. Charif, H. and S. J. McKenna, “Tracking the Activity of Participants in a Meeting”, *Machine Vision and Applications*, Vol. 17, No. 2, pp. 83–93, 2006.
23. Microsoft, “Introducing Kinect for Xbox 360”, <http://www.xbox.com/en-US/kinect>, accessed at July 2013.
24. Alexiadis, D. S., P. Kelly, P. Daras, N. E. O’Connor, T. Boubekeur and M. B. Moussa, “Evaluating a Dancer’s Performance using Kinect-based Skeleton Track-

- ing”, *Proceedings of the 19th ACM international conference on Multimedia*, pp. 659–662, 2011.
25. Bianco, S. and F. Tisato, “Karate Moves Recognition from Skeletal Motion”, *Proceedings of 3D Image Processing and Applications*, 2013.
 26. YouTube, “YouTube Statistics”, <http://www.youtube.com/yt/press/statistics.html>, accessed at July 2013.
 27. Lan, T., Y. Wang, G. Mori and S. N. Robinovitch, “Retrieving Actions in Group Contexts”, K. Kutulakos (Editor), *Trends and Topics in Computer Vision*, Vol. 6553 of *Lecture Notes in Computer Science*, pp. 181–194, 2012.
 28. Jones, S., L. Shao, J. Zhang and Y. Liu, “Relevance Feedback for Real-world Human Action Retrieval”, *Pattern Recognition Letters*, Vol. 33, No. 4, pp. 446–452, 2012.
 29. Tang, J., L. Shao and X. Zhen, “Human Action Retrieval via Efficient Feature Matching”, *IEEE International Conference on Advanced Video and Signal-based Surveillance*, 2013.
 30. Aydemir, M. S., U. Ergul, A. Guclu and M. E. Karsligil, “Video Summarization using Simple Action Patterns”, *21st International Conference on Pattern Recognition*, pp. 2047–2050, 2012.
 31. Sivic, J. and A. Zisserman, “Video Google: A Text Retrieval Approach to Object Matching in Videos”, *IEEE International Conference on Computer Vision*, pp. 1470–1477, 2003.
 32. Turaga, P., R. Chellappa, V. S. Subrahmanian and O. Udrea, “Machine Recognition of Human Activities: A Survey”, *Circuits and Systems for Video Technology, IEEE Transactions on*, Vol. 18, No. 11, pp. 1473–1488, 2008.
 33. Poppe, R., “A Survey on Vision-based Human Action Recognition”, *Image and*

- Vision Computing*, Vol. 28, No. 6, pp. 976–990, 2010.
34. Weinland, D., R. Ronfard and E. Boyer, “A Survey of Vision-based Methods for Action Representation, Segmentation and Recognition”, *Computer Vision and Image Understanding*, Vol. 115, No. 2, pp. 224–241, 2011.
 35. Aggarwal, J. and M. Ryoo, “Human Activity Analysis: A Review”, *ACM Computing Surveys*, Vol. 43, No. 3, pp. 1–43, 2011.
 36. Ikizler, N., R. G. Cinbis, S. Pehlivan and P. Duygulu, “Recognizing Actions in Still Images”, *International Conference on Pattern Recognition*, 2008.
 37. Thurau, C. and V. Hlavac, “Pose Primitive Based Human Action Recognition in Videos or Still Images”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
 38. Yang, W., Y. Yang and G. Mori, “Recognizing Human Actions from Still Images with Latent Poses”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.
 39. Maji, S., L. Bourdev and J. Malik, “Action Recognition from a Distributed Representation of Pose and Appearance”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
 40. Raja, K., I. Laptev, P. Perez and L. Oisel, “Joint Pose Estimation and Action Recognition in Image Graphs”, *IEEE International Conference on Image Processing*, pp. 25–28, 2011.
 41. Yao, B. and L. Fei-Fei, “Action Recognition with Exemplar Based 2.5D Graph Matching”, *Proceedings of the European Conference on Computer Vision*, 2012.
 42. Bobick, A. F. and J. W. 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.

43. Efros, A. A., A. C. Berg, G. Mori and J. Malik, “Recognizing Action at a Distance”, *Proceedings of the Ninth IEEE International Conference on Computer Vision*, pp. 726–733, 2003.
44. Laptev, I., “On Space-Time Interest Points”, *International Journal of Computer Vision*, Vol. 64, No. 2-3, pp. 107–123, 2005.
45. Brendel, W. and S. Todorovic, “Learning Spatiotemporal Graphs of Human Activities”, *IEEE International Conference on Computer Vision*, 2011.
46. Meng, L., L. Qing, P. Yang, J. Miao, X. Chen and D. N. Metaxas, “Activity Recognition Based on Semantic Spatial Relation”, *International Conference on Pattern Recognition*, pp. 609–612, 2012.
47. Shotton, J., A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore and A. Blake., “Real-time Human Pose Recognition in Parts from Single Depth Images”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1297–1304, 2011.
48. Yao, B. and L. Fei-Fei, “Modeling Mutual Context of Object and Human Pose in Human-object Interaction Activities”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 17–24, 2010.
49. Yilmaz, A. and M. Shah, “Actions Sketch: A Novel Action Representation”, *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 984–989, 2005.
50. Gorelick, L., M. Blank, E. Shechtman, M. Irani and R. Basri, “Actions as Space-time Shapes”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 12, pp. 2247–2253, 2007.
51. Black, M. J., Y. Yacoob, A. D. Jepson and D. J. Fleet, “Learning Parameterized Models of Image Motion”, *IEEE Conference on Computer Vision and Pattern*

- Recognition*, pp. 561–567, 1997.
52. Ke, Y., R. Sukthankar and M. Hebert, “Efficient Visual Event Detection Using Volumetric Features”, *Tenth IEEE International Conference on Computer Vision*, Vol. 1, pp. 166–173, 2005.
 53. Viola, P. and M. Jones, “Robust Real-time Object Detection”, *International Journal of Computer Vision*, Vol. 57, No. 2, pp. 137–154, 2001.
 54. Dollár, P., V. Rabaud, G. Cottrell and S. Belongie, “Behavior Recognition via Sparse Spatio-Temporal Features”, *VS-PETS Workshop at the IEEE International Conference on Computer Vision*, 2005.
 55. Scovanner, P., S. Ali and M. Shah, “A 3D SIFT Descriptor and Its Application to Action Recognition”, *International Conference on ACM Multimedia*, pp. 357–360, 2007.
 56. Willems, G., T. Tuytelaars and L. V. Gool, “An Efficient Dense and Scale-invariant Spatio-temporal Interest Point Detector”, *Proceedings of the European Conference of Computer Vision*, 2008.
 57. Ning, H., Y. Hu and T. S. Huang, “Searching Human Behaviors using Spatial-temporal Words”, *IEEE International Conference on Image Processing*, Vol. 6, pp. 337–340, 2007.
 58. Bregonzio, M., S. Gong and T. Xiang, “Recognising Action as Clouds of Space-time Interest Points”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1948–1955, 2009.
 59. Wang, H., M. M. Ullah, A. Kläser, I. Laptev and C. Schmid, “Evaluation of Local Spatio-temporal Features for Action Recognition”, *British Machine Vision Conference*, 2009.
 60. Tamrakar, A., S. Ali, Q. Yu, J. Liu, O. Javed, A. Divakaran, H. Cheng and

- H. Sawhney, “Evaluation of Low-level Features and Their Combinations for Complex Event Detection in Open Source Videos”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3681–3688, 2012.
61. Everts, I., J. C. V. Gemert and T. Gevers, “Evaluation of Color STIPs for Human Action Recognition”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
 62. Xia, L. and J. K. Aggarwal, “Spatio-Temporal Depth Cuboid Similarity Feature for Activity Recognition Using Depth Camera”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
 63. Hadfield, S. and R. Bowden, “Hollywood 3D: Recognizing Actions in 3D Natural Scenes”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
 64. Fathi, A. and G. Mori, “Action Recognition by Learning Mid-level Motion Features”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.
 65. Liu, C., Y. Kong, X. Wu and Y. Jia, “Action Recognition with Discriminative Mid-level Features”, *IEEE Conference on Pattern Recognition*, pp. 3366–3369, 2012.
 66. Hu, J., Y. Kong and Y. Fu, “Activity Recognition by Learning Structural and Pairwise Mid-level Features Using Random Forest”, *IEEE Conference on Automatic Face and Gesture Recognition*, 2013.
 67. Wang, L., Y. Qiao and X. Tang, “Motionlets: Mid-Level 3D Parts for Human Motion Recognition”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
 68. Ta, A. P., C. Wolf, G. Lavoue, A. Başkurt and J. M. Jolion, “Pairwise Features for Human Action Recognition”, *International Conference on Pattern Recognition*,

- pp. 3224–3227, 2010.
69. Bilinski, P. and F. Bremond, “Statistics of Pairwise Co-occurring Local Spatio-temporal Features for Human Action Recognition”, *Proceedings of the 4th International Workshop on Video Event Categorization, Tagging and Retrieval, in conjunction with 12th European Conference on Computer Vision*, Vol. 7583, pp. 311–320, 2012.
 70. Zhang, Y., X. Liu, M.-C. Chang, W. Ge and T. Chen, “Spatio-Temporal Phrases for Activity Recognition”, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato and C. Schmid (Editors), *European Conference on Computer Vision*, Vol. 7574 of *Lecture Notes in Computer Science*, pp. 707–721, 2012.
 71. Cheng, G., Y. Wan, W. Santiteerakul, S. Tang and B. P. Buckles, “Action Recognition with Temporal Relationships”, *Computer Vision and Pattern Recognition Workshops*, 2013.
 72. Felzenszwalb, P. F. and D. P. Huttenlocher, “Pictorial Structures for Object Recognition”, *International Journal of Computer Vision*, Vol. 61, No. 1, pp. 55–79, 2005.
 73. Felzenszwalb, P. F., R. Girshick, D. McAllester and D. Ramanan, “Object Detection with Discriminatively Trained Part-Based Models”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 9, pp. 1627–1645, 2010.
 74. Johnson, S. and M. Everingham, “Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation”, *British Machine Vision Conference*, pp. 1–11, 2010.
 75. Yang, Y. and D. Ramanan, “Articulated Pose Estimation with Flexible Mixtures-of-Parts”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1385–1392, 2011.

76. Bourdev, L. and J. Malik, “Poselets: Body Part Detectors Trained using 3D Human Pose Annotations”, *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 1365–1372, 2009.
77. Maji, S., L. Bourdev and J. Malik, “Action Recognition from A Distributed Representation of Pose and Appearance”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3177–3184, 2011.
78. Ni, B., Y. Pei, Z. Liang, L. Lin and P. Moulin, “Integrating Multi-Stage Depth-Induced Contextual Information For Human Action Recognition and Localization”, *IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, 2013.
79. Koppula, H. S., R. Gupta and A. Saxena, “Learning Human Activities and Object Affordances from RGB-D Videos”, *International Journal of Robotics Research*, Vol. 32, No. 8, pp. 951–970, 2013.
80. Zhu, Y., W. Chen and G. Guo, “Fusing Spatio-temporal Features and Joints for 3D Action Recognition”, *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013.
81. Choi, W., K. Shahid and S. Savarese, “Learning Context for Collective Activity Recognition”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3273–3280, 2011.
82. Chen, C. and K. Grauman, “Efficient Activity Detection with Max-Subgraph Search”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1274–1281, 2012.
83. Yao, B., X. Jiang, A. Khosla, A. L. Lin, L. Guibas and L. Fei-Fei, “Human Action Recognition by Learning Bases of Action Attributes and Parts”, *IEEE International Conference on Computer Vision*, pp. 1331–1338, 2011.

84. Ryoo, M. S. and J. K. Aggarwal, “Semantic Representation and Recognition of Continued and Recursive Human Activities”, *International Journal of Computer Vision*, Vol. 82, No. 1, pp. 1–24, 2009.
85. Liu, J., B. Kuipers and S. Savarese, “Recognizing Human Actions by Attributes”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3337–3344, 2011.
86. Wang, H. J., Y. L. Lin, C.-Y. Huang, Y.-L. Hou and W. Hsu, “Full Body Human Attribute Detection in Indoor Surveillance Environment Using Color-Depth Information”, *International Conference on Advanced Video and Signal-Based Surveillance*, 2013.
87. Sharma, G., F. Jurie and C. Schmid, “Expanded Parts Model for Human Attribute and Action Recognition in Still Images”, *International Conference on Computer Vision and Pattern Recognition*, 2013.
88. Niebles, J. C. and L. Fei-Fei, “A Hierarchical Model of Shape and Appearance for Human Action Classification”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.
89. Niebles, J. C., H. Wang and L. Fei-Fei, “Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words”, *International Journal of Computer Vision*, Vol. 79, No. 3, pp. 299–318, 2008.
90. Ryoo, M. S. and J. K. Aggarwal, “Spatio-temporal Relationship Match: Video Structure Comparison for Recognition of Complex Human Activities”, *IEEE 12th International Conference on Computer Vision*, pp. 1593–1600, 2009.
91. Offi, F., R. Chaudhry, G. Kurillo, R. Vidal and R. Bajcsy, “Sequence of the Most Informative Joints (SMIJ): A New Representation for Human Skeletal Action Recognition”, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 8–13, 2012.

92. Lin, W., Q. Yu, H. Sawhney and N. Vasconcelos, “Recognizing Activities via Bag of Words for Attribute Dynamics”, *International Conference on Computer Vision and Pattern Recognition*, 2013.
93. Bettadapura, V., G. Schindler, T. Plötz and I. Essa, “Augmenting Bag-of-Words: Data-Driven Discovery of Temporal and Structural Information for Activity Recognition”, *International Conference on Computer Vision and Pattern Recognition*, 2013.
94. Ta, A. P., C. Wolf, G. Lavoue and A. Başkurt, “Recognizing and Localizing Individual Activities Through Graph Matching”, *Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 196–203, 2010.
95. Gaur, U., Y. Zhu, B. Song and A. Roy-Chowdhury, “A String of Feature Graphs Model for Recognition of Complex Activities in Natural Videos”, *IEEE International Conference on Computer Vision*, pp. 2595–2602, 2011.
96. Borzeshi, E. Z., M. Piccardi and R. Y. D. Xu, “A Discriminative Prototype Selection Approach for Graph Embedding in Human Action Recognition”, *IEEE International Conference on, Computer Vision Workshops*, pp. 1295–1301, 2011.
97. Liu, J., S. Ali and M. Shah, “Recognizing Human Actions Using Multiple Features”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.
98. Yuan, Y., H. Zheng, Z. Li and D. Zhang, “Video Action Recognition with Spatio-temporal Graph Embedding and Spline Modeling”, *IEEE International Conference on Acoustics Speech and Signal Processing*, pp. 2422–2425, 2010.
99. Yi, S. and V. Pavlovic, “Sparse Granger Causality Graphs for Human Action Classification”, *International Conference on Pattern Recognition*, pp. 3374–3377, 2012.

100. Todorovic, S., “Human Activities as Stochastic Kronecker Graphs”, *Proceedings of the European Conference on Computer Vision*, 2012.
101. Chaaraoui, A. A., P. Climent-Perez and F. Florez-Revuelta, “An Efficient Approach for Multi-view Human Action Recognition Based on Bag-of-Key-Poses”, *Human Behavior Understanding*, Vol. 7559 of *Lecture Notes in Computer Science*, pp. 29–40, 2012.
102. Dyana, A. and S. Das, “Trajectory Representation using Gabor Features for Motion-based Video Retrieval”, *Pattern Recognition Letters*, Vol. 30, No. 10, pp. 877–892, 2009.
103. Savarese, S., A. Delpozio, J. Niebles and L. Fei-Fei, “Spatial-temporal Correlators for Unsupervised Action Classification”, *In IEEE Workshop on Motion and Video Computing*, 2008.
104. Cuntoor, N. P., B. Yegnanarayana and R. Chellappa, “Activity Modeling Using Event Probability Sequences”, *IEEE Transactions on Image Processing*, Vol. 17, No. 4, pp. 594–607, 2008.
105. Li, W., Z. Zhang and Z. Liu, “Expandable Data-Driven Graphical Modeling of Human Actions Based on Salient Postures”, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 18, No. 11, pp. 1499–1510, 2008.
106. Lv, F. and R. Nevatia, “Single view Human Action Recognition using Key Pose Matching and Viterbi Path Searching”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.
107. Zhang, L., Z. Zeng and Q. Ji, “Probabilistic Image Modeling With an Extended Chain Graph for Human Activity Recognition and Image Segmentation”, *IEEE Transactions on Image Processing*, Vol. 20, No. 9, pp. 2401–2413, 2011.
108. Xia, L., C. C. Chen and J. K. Aggarwal, “View Invariant Human Action Recogni-

- tion Using Histograms of 3D Joints”, *International Workshop on Human Activity Understanding from 3D Data (HAU3D) in conjunction with IEEE Conference on Computer Vision and Pattern Recognition*, pp. 20–27, 2012.
109. Raptis, M., D. Kirovski and H. Hoppe, “Real-time Classification of Dance Gestures from Skeleton Animation”, *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 147–156, 2011.
 110. Wang, J., Z. Liu, Y. Wu and J. Yuan, “Mining Actionlet Ensemble for Action Recognition with Depth Cameras”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1290–1297, 2012.
 111. Celebi, S., A. S. Aydin, T. T. Temiz and T. Arici, “Gesture Recognition using Skeleton Data with Weighted Dynamic Time Warping”, *8th International Conference on Computer Vision Theory and Applications*, 2013.
 112. Ellis, C., S. Z. Masood, M. F. Tappen, J. J. L. Jr. and R. Sukthankar, “Exploring the Trade-off Between Accuracy and Observational Latency in Action Recognition”, *International Journal of Computer Vision*, Vol. 101, No. 3, pp. 420–436, 2013.
 113. Mikolajczyk, K. and H. Uemura, “Action Recognition with Appearance Motion Features and Fast Search Trees”, *Computer Vision and Image Understanding*, Vol. 115, No. 3, pp. 426–438, 2011.
 114. Jiang, Z., Z. Lin and L. Davis, “Recognizing Human Actions by Learning and Matching Shape-Motion Prototype Trees”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 34, No. 3, pp. 533–547, 2012.
 115. Oshin, O., A. Gilbert and R. Bowden, “Capturing the Relative Distribution of Features for Action Recognition”, *IEEE International Conference on Automatic Face and Gesture Recognition*, 2011.

116. Miranda, L., T. Vieira, D. Martinez, T. Lewiner, A. W. Vieira and M. F. M. Campo, “Real-Time Gesture Recognition from Depth Data through Key Poses Learning and Decision Forests”, *Brazilian Symposium on Computer Graphics and Image Processing*, pp. 268–275, 2012.
117. Delaitre, V., I. Laptev and J. Sivic, “Recognizing Human Actions in Still Images: A Study of Bag-of-features and Part-based Representations”, *British Machine Vision Conference*, 2010.
118. Tian, Y., R. Sukthankar and M. Shah, “Spatiotemporal Deformable Part Models for Action Detection”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
119. Filipovych, R. and E. Ribeiro, “Robust Sequence Alignment for Actor-object Interaction Recognition: Discovering Actor-object States”, *Computer Vision and Image Understanding*, Vol. 115, No. 2, pp. 177–193, 2011.
120. Raptis, M. and L. Sigal, “Poselet Key-framing: A Model for Human Activity Recognition”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
121. Taylor, C. J., “Reconstruction of Articulated Objects from Point Correspondences in a Single Uncalibrated Image”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 677–684, 2000.
122. Abdelkader, M. F., W. Abd-Almageed, A. Srivastava and R. Chellappa, “Silhouette-based Gesture and Action Recognition via Modeling Trajectories on Riemannian Shape Manifolds”, *Computer Vision and Image Understanding*, Vol. 115, No. 3, pp. 439–455, 2011.
123. Chaquet, J. M., E. J. Carmona and A. Fernandez-Caballero, “A Survey of Video Datasets for Human Action and Activity Recognition”, *Computer Vision and Image Understanding*, Vol. 117, No. 6, pp. 633–659, 2013.

124. Zelnik-Manor, L. and M. Irani, “Weizmann Event-based Analysis of Video”, <http://www.wisdom.weizmann.ac.il/~vision/VideoAnalysis/Demos/EventDetection/EventDetection.html>, 2013, accessed at July 2013.
125. CAVIAR, “Caviar: Context Aware Vision Using Image-based Active Recognition”, <http://homepages.inf.ed.ac.uk/rbf/CAVIAR/caviar.htm>, 2013, accessed at July 2013.
126. INRIA, “Etiseo Video Understanding Evaluation”, <http://www-sop.inria.fr/orion/ETISEO/index.htm>, 2013, accessed at July 2013.
127. Tran, D., A. Sorokin and D. Forsyth, “Human Activity Recognition with Metric Learning”, <http://vision.cs.uiuc.edu/projects/activity/>, 2013, accessed at July 2013.
128. Yuan, J., Z. Liu and Y. Wu, “Discriminative Video Pattern Search for Efficient Action Detection”, http://users.eecs.northwestern.edu/~jyu410/index_files/actiondetection.html, 2013, accessed at July 2013.
129. Fisher, R., “Behave: Computer-assisted Prescreening of Video Streams for Unusual Activities”, <http://homepages.inf.ed.ac.uk/rbf/BEHAVE/>, 2013, accessed at July 2013.
130. Center for Biometrics and Security Research, “Casia Action Database for Recognition”, <http://www.cbsr.ia.ac.cn/english/Action%20Databases%20EN.asp>, 2013, accessed at July 2013.
131. University of Surrey and CERTH-ITI, “I3dpost Multi-view Human Action Datasets”, http://kahlan.eps.surrey.ac.uk/i3dpost_action/, 2013, accessed at July 2013.
132. Visual Geometry Group, “TV Human Interactions Dataset”, http://www.robots.ox.ac.uk/~vgg/data/tv_human_interactions/index.html, 2013, accessed at July 2013.

cessed at July 2013.

133. Ryoo, M. S. and J. K. Aggarwal, “UT-Interaction Dataset, ICPR Contest on Semantic Description of Human Activities”, http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html, 2013, accessed at July 2013.
134. Video Computing Group, “Videoweb Dataset”, <http://www.ee.ucr.edu/~amitrc/vwdata.php>, 2013, accessed at July 2013.
135. Reading University Computational Vision Group, “Pets 2009 Benchmark Data”, <http://www.cvg.rdg.ac.uk/PETS2009/a.html>, 2013, accessed at July 2013.
136. Choi, W., K. Shahid and S. Savarese, “What Are They Doing? : Collective Activity Classification Using Spatio-Temporal Relationship Among People”, *International Workshop on Visual Surveillance*, 2009.
137. Messing, R., C. Pal and H. Kautz, “Activity Recognition Using the Velocity Histories of Tracked Keypoints”, *International Conference on Computer Vision*, 2009.
138. Tenorth, M., J. Bandouch and M. Beetz, “The TUM Kitchen Data Set of Everyday Manipulation Activities for Motion Tracking and Action Recognition”, *International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences*, 2009.
139. INRIA, “Inria Xmas Motion Acquisition Sequences”, <http://4drepository.inrialpes.fr/public/viewgroup/6>, 2013, accessed at July 2013.
140. Kingston University, “Muhavi: Multicamera Human Action Video Data”, <http://dipersec.king.ac.uk/MuHAVi-MAS/>, 2013, accessed at July 2013.
141. University of Central Florida, “UCF Aerial Camera, Rooftop Camera and Ground Camera Dataset”, <http://vision.eecs.ucf.edu/data/UCF-ARG.html>, 2013, accessed at July 2013.

142. University of Central Florida, “UCF Aerial Action Dataset”, <http://server.cs.ucf.edu/~vision/aerial/index.html>, 2013, accessed at July 2013.
143. Serre Lab, “Hmdb: A Large Video Database for Human Motion Recognition”, <http://serre-lab.clps.brown.edu/resources/HMDB/>, 2013, accessed at July 2013.
144. Laptev, I., “Irisa Download Data/software”, <http://www.irisa.fr/vista/Equipe/People/Laptev/download.html>, 2013, accessed at July 2013.
145. Stanford University, “Olympic Sports Dataset”, <http://vision.stanford.edu/Datasets/OlympicSports/>, 2013, accessed at July 2013.
146. University of Central Florida, “UCF Youtube Action Dataset”, http://www.cs.ucf.edu/~liujg/YouTube_Action_dataset.html, 2013, accessed at July 2013.
147. University of Central Florida, “UCF Sports Action Dataset”, <http://vision.eecs.ucf.edu/datasetsActions.html>, 2013, accessed at July 2013.
148. Microsoft, “MSR Action Recognition Datasets and Codes”, <http://research.microsoft.com/en-us/um/people/zliu/actionrecorsrc/default.htm>, 2013, accessed at July 2013.
149. Fothergill, S., H. M. Mentis, P. Kohli and S. Nowozin, “Instructing People for Training Gestural Interactive Systems”, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1737–1746, 2012.
150. ChaLearn California, “ChaLearn Gesture Dataset (CGD2011)”, <http://gesture.chalearn.org/data>, 2011, accessed at July 2013.
151. Ofli, F., R. Chaudhry, G. Kurillo, R. Vidal and R. Bajcsy, “Berkeley MHAD: A Comprehensive Multimodal Human Action Database”, *IEEE Workshop on Applications on Computer Vision*, 2013.

152. Gourgari, S., G. Goudelis and K. Karpouzis, “THETIS: THree Dimensional Tennis Shots - A Human Action Dataset”, *IEEE International Conference on Computer Vision Workshops*, 2013.
153. Sung, J., H. Koppula, B. Selman and A. Saxena, “Cornell Activity Datasets: CAD-60 & CAD-120”, <http://pr.cs.cornell.edu/humanactivities/data.php>, 2013, accessed at July 2013.
154. Ni, B., G. Wang and P. Moulin, “RGBD-HuDaAct: A Color-Depth Video Database for Human Daily Activity Recognition”, *IEEE Workshop on Consumer Depth Cameras for Computer Vision*, 2011, accessed at July 2013.
155. Thome, N., D. Merad and S. Miguet, “Human Body Labeling and Tracking Using Graph Matching Theory”, *IEEE International Conference on Video and Signal Based Surveillance*, 2006.
156. Zheng, D., H. Xiong and Y. F. Zheng, “A Structured Learning-based Graph Matching For Dynamic Multiple Object Tracking”, *IEEE International Conference on Image Processing*, pp. 2333–2336, 2011.
157. Lee, J. K., J. Oh and S. Hwang, “Clustering of Video Objects by Graph Matching”, *IEEE International Conference on Multimedia and Expo*, pp. 394–397, 2005.
158. Berg, A. C., T. L. Berg and J. Malik, “Shape Matching and Object Recognition using Low Distortion Correspondences”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 26–33, 2005.
159. Torresani, L., V. Kolmogorov and C. Rother, “Feature Correspondence Via Graph Matching: Models and Global Optimization”, *Proceedings of the European Conference of Computer Vision*, pp. 596–609, 2008.
160. Albarelli, A., F. Bergamasco, L. Rossi, S. Vascon and A. Torsello, “A Stable Graph-Based Representation for Object Recognition through High-Order Match-

- ing”, *International Conference on Pattern Recognition*, pp. 3341–3344, 2012.
161. Duchenne, O., A. Joulin and J. Ponce, “A Graph-matching Kernel for Object Categorization”, *IEEE International Conference on Computer Vision*, 2011.
 162. Leordeanu, M. and M. Hebert, “A Spectral Technique for Correspondence Problems Using Pairwise Constraints”, *IEEE International Conference on Computer Vision*, pp. 1482–1489, 2005.
 163. Sharma, J. C. A., R. Horaud and E. Boyer, “Topologically Robust 3D Shape Matching based on Diffusion Geometry and Seed Growing”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2481–2488, 2011.
 164. Leordeanu, M. and M. Hebert, “A Spectral Technique for Correspondence Problems using Pairwise Constraints”, *IEEE International Conference on Computer Vision*, 2005.
 165. Duchenne, O., F. R. Bach, I.-S. Kweon and J. Ponce, “A Tensor-based Algorithm for High-order Graph Matching”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1980–1987, 2009.
 166. Lowe, D., “Distinctive Image Features from Scale-invariant Features”, *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91–110, 2004.
 167. Belongie, S., J. Malik and J. Puzicha, “Matching Shapes”, *Proceedings of the Eighth IEEE International Conference on Computer Vision*, pp. 454–461, 2001.
 168. Zass, R. and A. Shashua, “Probabilistic Graph and Hypergraph Matching”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
 169. Garey, M. R. and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, 1979.
 170. Conte, D., P. Foggia, C. Sansone and M. Vento, “Thirty Years of Graph Match-

- ing in Pattern Recognition”, *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 18, No. 3, pp. 265–298, 2004.
171. Leordeanu, M., A. Zanfir and C. Sminchisescu, “Semi-supervised Learning and Optimization for Hypergraph Matching”, *IEEE International Conference on Computer Vision*, 2011.
 172. Lee, J., M. Cho and K. M. Lee, “Hyper-graph Matching via Reweighted Random Walks”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
 173. Knight, P. A., *The Sinkhorn-Knopp Algorithm: Convergence and Applications*, Tech. rep., 2006.
 174. Zaslavskiy, M., F. Bach and J. P. Vert, “A Path Following Algorithm for the Graph Matching Problem”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, No. 12, pp. 2227–2242, 2009.
 175. Zeng, Y., C. Wang, Y. Wang, X. Gu, D. Samaras and N. Paragios, “Dense Non-rigid Surface Registration Using High-Order Graph Matching”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.
 176. Lin, L., K. Zeng, X. Liu and S. C. Zhu, “Layered Graph Matching by Composite Cluster Sampling with Collaborative and Competitive Interactions”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1351–1358, 2009.
 177. Caetano, T. S., T. Caelli, D. Schuurmans and D. A. C. Barone, “Graphical Models and Point Pattern Matching”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 10, pp. 1646–1663, 2006.
 178. Lauritzen, S. L. and D. J. Spiegelhalter, “Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems”, *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 50, No. 2, pp. 157–224, 1988.

179. Harris, C. and M. Stephens, “A Combined Corner and Edge Detector”, *Alvey Vision Conference*, 1988.
180. Pudil, P., F. J. Ferri, J. Novovicov and J. Kittler, “Floating Search Methods for Feature Selection with Nonmonotonic Criterion Functions”, *International Conference on Pattern Recognition*, Vol. 2, pp. 279–283, 1994.
181. KTH, “Recognition of Human Actions”, <http://www.nada.kth.se/cvap/actions/>, accessed at July 2013.
182. Gao, Z., M. Y. Chen, A. Hauptmann and A. Cai, “Comparing Evaluation Protocols on the KTH Dataset”, *Human Behavior Understanding*, Vol. LNCS 6219, pp. 88–100, 2010.
183. Essid, S., D. Alexiadis, R. Tournemenne, M. Gowing, P. Kelly, D. Monaghan, P. Daras, A. Dremeau and N. E. O’Connor, “An Advanced Virtual Dance Performance Evaluator”, *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2269–2272, 2012.
184. Chua, P., D. Ventura, R. Crivella, T. Camill, B. Daly, N. Hu, J. Hodgins, R. Schaaf and R. Pausch, “Training for Physical Tasks in Virtual Environments: Tai Chi”, *Proceedings of the IEEE Virtual Reality*, pp. 87–94, 2003.
185. Ilg, W., J. Mezger and M. Giese, “Estimation of Skill Levels in Sports Based on Hierarchical Spatio-Temporal Correspondences”, *Pattern Recognition*, Vol. 2781 of *Lecture Notes in Computer Science*, pp. 523–531, 2003.
186. Michelet, S., K. Karp, E. Delaherche, C. Achard and M. Chetouani, “Automatic Imitation Assessment in Interaction”, *Human Behavior Understanding*, Vol. 7559 of *Lecture Notes in Computer Science*, pp. 161–173, 2012.
187. Yamato, J., J. Ohya and K. Ishii, “Recognizing Human Action in Time-Sequential Images using Hidden Markov Model”, *IEEE Conference on Computer Vision and*

- Pattern Recognition*, pp. 379–385, 1992.
188. Hsu, D., S. M. Kakade and T. Zhang, “A Spectral Algorithm for Learning Hidden Markov Models”, *Journal of Computer and System Sciences*, Vol. 78, No. 5, pp. 1460–1480, 2012.
 189. Alpaydin, E., *Introduction to Machine Learning*, The MIT Press, 2010.
 190. Cover, T. M. and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*, Wiley-Interscience, 2006.
 191. Bishop, C. M., *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., 2006.
 192. Anandkumar, A., D. Hsu and S. Kakade, “A Method of Moments for Mixture Models and Hidden Markov Models”, *Conference on Learning Theory*, 2012.
 193. Bourdev, L. and J. Malik, “Poselets: Body Part Detectors Trained Using 3D Human Pose Annotations”, *IEEE 12th International Conference on Computer Vision*, pp. 1365–1372, 2009.
 194. Chui, H. and A. Rangarajan, “A New Point Matching Algorithm for Non-rigid Registration”, *Computer Vision and Image Understanding*, Vol. 89, No. 2-3, pp. 114–141, 2003.
 195. Ovsjanikov, M., Q. Mérigot, F. Mémoli and L. J. Guibas, “One Point Isometric Matching with the Heat Kernel”, *Computer Graphics Forum*, Vol. 29, No. 5, pp. 1555–1564, 2010.