Année 2015

Num d'ordre: 41861

Université Lille 1
Ecole Doctorale Science Pour
l'Ingénieur
Laboratoire CRIStAL (umr cnrs 9189)
Équipe 3D SAM

Università degli Studi di Firenze
Corso di Dottorato in Informatica,
Sistemi e Telecomunicazioni
Media Integration and Communication
Center (MICC)

THÈSE EN COTUTELLE
pour obtenir le grade de
Docteur,
spécialité Informatique

présentée et soutenue publiquement par

Maxime Devanne

le 01/12/2015

3D Human Behavior Understanding by Shape Analysis of Human Motion and Pose

dirigée conjointement par
Mohamed DAOUDI
Pietro PALA

COMPOSITION DU JURY

Mme. Jenny Benois-Pineau	Professeur, Université Bordeaux 1	Rapporteur
M. Nicu Sebe	Professeur, Università di Trento	Rapporteur
M. Mohamed Daoudi	Professeur, Institut Mines-Télécom	Directeur
M. Pietro Pala	Professeur, Università di Firenze	Directeur
M. Alberto Del Bimbo	Professeur, Università di Firenze	Invité
M. Hazem Wannous	Maître de conférences, Université de Lille 1	Invité (encadrant)

Abstract

The emergence of RGB-D sensors providing the 3D structure of both the scene and the human body offers new opportunities for studying human motion and understanding human behaviors. However, the design and development of models for behavior recognition that are both accurate and efficient is a challenging task due to the variability of the human pose, the complexity of human motion and possible interactions with the environment. In this thesis, we address this issue in two main phases by differentiating behaviors according to their complexity. We first focus on the action recognition problem by representing human action as the trajectory of 3D coordinates of human body joints over the time, thus capturing simultaneously the body shape and the dynamics of the motion. The action recognition problem is then formulated as the problem of computing the similarity between shape of trajectories in a Riemannian framework. Experiments carried out on four representative benchmarks demonstrate the potential of the proposed solution in terms of accuracy/latency for a low-latency action recognition. Second, we extend the study to activities by analyzing the evolution of the human pose shape to decompose the motion stream into short motion units. Each motion unit is then characterized by the motion trajectory and depth appearance around hand joints, so as to describe the human motion and interaction with objects. Finally, the sequence of temporal segments is modeled through a Dynamic Naive Bayesian Classifier. Experiments on four representative datasets evaluate the potential of the proposed approach in different contexts, including gesture or activity recognition and online activity detection.

Key words: RGB-D data, 3D human behavior recognition, temporal modeling, shape analysis, Riemannian manifold, online detection.

Résumé

Compréhension de comportements humains 3D par l'analyse de forme de la posture et du mouvement

L'émergence de capteurs de profondeur capturant la structure 3D de la scène et du corps humain offre de nouvelles possibilités pour l'étude du mouvement et la compréhension des comportements humains. Cependant, la conception et le développement de modules de reconnaissance de comportements à la fois précis et efficaces est une tâche difficile en raison de la variabilité de la posture humaine, la complexité du mouvement et les interactions avec l'environnement. Dans cette thèse, nous abordons cette question en deux étapes principales en différenciant les comportements en fonction de leur complexité. Nous nous concentrons d'abord sur le problème de la reconnaissance d'actions en représentant la trajectoire du corps humain au cours du temps, capturant ainsi simultanément la forme du corps et la dynamique du mouvement. Le problème de la reconnaissance d'actions est alors formulé comme le calcul de similitude entre la forme des trajectoires dans un cadre Riemannien. Les expériences menées sur quatre bases de données démontrent le potentiel de la solution en termes de précision/temps de latence de la reconnaissance d'actions. Deuxièmement, nous étendons l'étude aux activités en analysant l'évolution de la forme de la posture pour décomposer la séquence en unités de mouvement. Chaque unité de mouvement est alors caractérisée par la trajectoire de mouvement et l'apparence autour des mains, de manière à décrire le mouvement humain et l'interaction avec les objets. Enfin, la séquence de segments temporels est modélisée par un classifieur Bayésien naif dynamique. Les expériences menées sur quatre bases de données évaluent le potentiel de l'approche dans différents contextes comme la reconnaissance de gestes ou d'activités et la détection en ligne d'activités.

Mots-clés: Données de profondeur, reconnaissance de comportement humain 3D, modélisation temporelle, analyse de forme, variété Riemannienne, détection en ligne.

Sommario

Comprensione del comportamento umano 3D attraverso l'analisi di forma del movimento e della posa

La diffusione di sensori RGB-D capaci di fornire la struttura 3D sia della scena che del corpo umano offre nuove opportunità per studiare i movimenti dell'uomo e capire i suoi comportamenti. Tuttavia, la progettazione e lo sviluppo di modelli per il riconoscimento dei comportamenti che siano tanto accurati quanto efficienti è un problema competitivo a causa della variabilità delle pose, della complessità del moto e delle possibili interazioni con l'ambiente. In questa tesi si affronta il problema in due passi principali, differenziando i comportamenti in base alla loro complessità. Si pone l'attenzione inizialmente sul problema di riconoscere azioni rappresentandole come traiettorie di coordinate 3D dei giunti del corpo nel tempo, catturando al tempo stesso la forma e le dinamiche di moto. Il problema del riconoscimento delle azioni è poi riformulato come il problema di calcolare le similarità tra la forma delle traiettorie in un manifold Riemanniano. Gli esperimenti effettuati su quattro benchmark dimostrano il potenziale della soluzione proposta in termini di accuratezza/latenza del riconoscimento di azioni. Lo studio è poi esteso al riconoscimento di attività analizzando l'evoluzione della forma delle pose per decomporre il flusso di moto in unità di moto. Ogni unità di moto è quindi caratterizzata dalla traiettoria di moto e da una descrizione della profondità nell'intorno dei giunti delle mani, in modo da descrivere il moto e le interazioni con oggetti. Infine, la sequenza di segmenti temporali è modellata attraverso un classificatore Dynamic Naive Bayesian. Il potenziale dell'approccio proposto è valutato su esperimenti con quattro dataset in contesti diversi, inclusi il riconoscimento di gesti e attività e rilevamento di azioni online.

Parole chiave: dati RGB-D, riconoscimento di comportamenti umani 3D, modellazione temporale, analisi della forma, Riemannian manifold, rilevamento online.

Acknowledgement

I would like to express my gratitude to my advisors, Prof. Mohamed Daoudi and Prof. Pietro Pala for guiding me through my research with professionalism, understanding, and patience. Their high experience strongly helped me to make this experience productive and stimulating.

I also thank my co-advisors Dr. Stefano Berretti and Dr. Hazem Wannous for their time and advice in beneficial scientific discussions. Their friendly and encouraging guidance make me more confident in my research.

I am also particularly grateful to Prof. Alberto Del Bimbo for his welcome in his research group as well as his large contribution to make feasible this joint PhD between the University of Lille and the University of Florence.

Special thanks to my PhD committee members for taking the time to participate in this process, and especially the reviewers of the manuscript for having accepted this significant task: Prof. Jenny Benois-Pineau and Prof. Nicu Sebe. All these people made me the honor to be present for this special day despite their highly charged agendas.

The members of both research group have contributed immensely to my personal and professional time in Lille and Florence. That is why I would like to thank all researchers from both teams: Andrew Bagdanov, Lamberto Ballan, Lahoucine Ballihi, Giuseppe Becchi, Boulbaba Ben Amor, Marco Bertini, Paul Audain Desrosiers, Hassen Drira, Andrea Ferracani, Svebor Karaman, Lea Landucci, Giuseppe Lisanti, Iacopo Masi, Federico Pernici, Daniele Pezzatini, Lorenzo Seidenari and Jean-Philippe Vandeborre. A special thanks to my friends and PhD student colleagues with whom I shared this experience: Taleb Alashkar, Claudio Baecchi, Federico Bartoli, Federico Becattini, Enrico Bondi, Quentin De Smedt, Dario Di Fina, Rachid El Khoury, Simone Ercoli, Claudio Ferrari,

Leonardo Galteri, Vincent Léon, Meng Meng, Rim Slama, Francesco Turchini, Tiberio Uricchio and Baiqiang Xia.

I also gratefully acknowledge both the engineering school Telecom Lille and the University of Florence for my co-tutorial thesis funding.

Finally, I thank my family, my friends and roommates for their support and encouragements. Particularly, I would like to thank my fiancee Marine, who also lives this experience against her will, for her continuous support making me very positive along my thesis.

Florence, September 30, 2015.

Publications

International journals

- **Maxime Devanne**, Hazem Wannous, Stefano Berretti, Pietro Pala, MohamedDaoudi and Alberto Del Bimbo. 3D Human Action Recognition by Shape Analysis of Motion Trajectories on Riemannian Manifold. *IEEE Transactions on Cybernetics*, volume 45(7), pages 1340-1352, 2015.

- **Maxime Devanne**, Stefano Berretti, Pietro Pala, Hazem Wannous, Mohamed Daoudi and Alberto Del Bimbo. Motion Units Decomposition of RGB-D Sequences for Human Behavior Understanding. *IEEE Transactions on Cybernetics* (Under review).

International conferences

- **Maxime Devanne**, Hazem Wannous, Stefano Berretti, Pietro Pala, MohamedDaoudi and Alberto Del Bimbo. Space-Time Pose Representation for 3D Human Action Recognition. *SBA Workshop of International Conference on Image Analysis and Processing*, 2013.

- **Maxime Devanne**, Hazem Wannous, Stefano Berretti, Pietro Pala, MohamedDaoudi and Alberto Del Bimbo. Combined Shape Analysis of Human Poses and Motion Units for Action Segmentation and Recognition. *11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 7, pages 1-6, 2015

National conferences

- **Maxime Devanne**, Hazem Wannous, Stefano Berretti, Pietro Pala, MohamedDaoudi and Alberto Del Bimbo. Reconnaissance d'actions humaines 3D par l'analyse de forme des trajectoires de mouvement. *Compression et Représentation des Signaux Audiovisuels (CORESA)*, 2014

Contents

List of Figures

1 Introduction

Sommaire

So as to communicate or interact with real world environments, motion is one of the main manner employed by human people. Hence, analyzing and understanding such human motion is appreciated in many domains of application, like entertainment, medicine, sport, video surveillance, human-machine interfaces, kinesiology and ambient assisted living.

This wide spectrum of potential applications encouraged computer vision society to address the issue of human behavior understanding trough vision-based analysis of the human motion. Human motion analysis in computer vision can be divided into three categories: human motion tracking, analysis of body or body parts movement and human behavior recognition.

Human motion tracking is mainly employed in video surveillance. It relates to the human motion detection in video sequences as well as its tracking over the sequence. In most of the cases, the global motion of the entire body is considered. The analysis of the resulted motions can be used to identify areas of interest in a scene or to determine global attitude of people in the crowd.

Differently, analysis of body movement focus on how the motion is performed. Such analysis are particularly useful in medical domain for detection of abnormal movements or rehabilitation, but also in sport for quality evaluation or statistical assessment of performances. It often considers local motion of certain body parts of interest, so as to achieve more accurate measures of the movement.

Human behavior recognition relates to the local or global analysis of the human motion in order to recognize it and understand its meaning. Such recognition methods rely on similarity computation between a studied behavior and a set of known or learned behaviors, so as to identify to which class of behaviors the query sequence is closer. Human behavior recognition offers a larger panel of application domains than the two previous categories. During this thesis, we focus our study on human behavior recognition and understanding.

Recently, new effective and low-cost depth cameras have been released. These range sensors offer, in addition to RGB images in standard cameras,

a depth map providing for each pixel its corresponding distances with respect to the sensor. The infra-red technology behind these sensors allows them to work in complete darkness and to be robust to light and illumination variation, a common issue in 2D videos analysis. Moreover, the depth information allows the reconstruction of the 3D structural information of the scene, thus facilitating the discernment of objects in the environment as well as the background subtraction. These advantages motivated researchers to investigate depth and resulted 3D data for the task of human behavior recognition by considering both the spatial configuration of the human pose and the dynamic of its motion characterizing the human behavior. While some methods employ extended features from 2D images literature on depth maps, other approaches consider the depth sequence as a 4D spatio-temporal space.

In addition, recent research makes available the real-time estimation of 3D humanoid skeletons from depth images. Such skeletons including a set of 3D connected joints representing various human body parts facilitate the analysis of the human pose and its motion over the time. The effectiveness of skeleton data has been proven for the analysis and recognition of relatively simple behaviors, like human gestures or actions. However, more complex behaviors like activities involve manipulation of objects. So as to characterize such human-object interactions, hybrid approaches combining description of both human motion and objects are appreciated. Such activities also involve more complex human motions. Hence, a temporally local analysis of the motion is often required.

Nevertheless, even if these new depth data provided by RGB-D sensors considerably facilitate the issue of human behavior understanding over 2D videos, important challenges, related to the nature of human motion and its analysis, remain. Indeed, so as to guarantee an effective and robust human behavior recognition system, the analysis should be invariant to geometric transformations of the subject as well as the execution speed of the behavior. Additional challenges appear when a fast recognition is needed or when several behaviors are performed successively in a long

sequence. In these contexts, an online analysis of the motion stream is necessary.

Besides euclidean methods, non-Euclidean like Riemannian methods have also been investigated by modeling human poses and motions on analytic manifolds. Their effectiveness over Euclidean methods have been demonstrated for the task of human behavior recognition from 2D videos.

1.1 Thesis contributions

All considerations stated before motivate us to investigate the issue of human behavior understanding from depth data. So as to face the main challenges, we first propose to recast the problem of human action recognition as a problem of shape analysis of motion trajectories through a Riemannian shape space. We then extend the study to activities by segmenting the motion into short motion units and considering both human movement and depth appearance to characterize human-object interactions. The main contribution of this PhD thesis can be summarized as follows:

- *Human action recognition:* We propose an original translation and rotation invariant representation of an action sequence as a trajectory in a high dimensional space. By concatenating the 3D coordinates of skeleton joints, data representation encodes the shape of the human posture at each frame. By modeling the sequence of frame features along the action as a trajectory, we capture the dynamics of human motion. The analysis and comparison of such trajectories is achieved through the shape analysis framework. Shapes of trajectories are interpreted within a Riemannian manifold and an elastic metric is employed for computing shape similarity, thus improving robustness to the execution speed of actions;

- *Motion unit decomposition of RGB-D sequences:* We extend the study to more complex behaviors, like activities. In so doing, we propose a segmentation method based on the analysis of human pose variation along the sequence. First, the skeleton of each pose is represented

as a 3D curve, which is mapped as a point on a Riemannian manifold. Then, using statistical tools on the manifold permits the decomposition of the sequence into elementary motion units. Then, we combine elastic shape analysis of motion trajectories on a Riemannian manifold, and description of depth appearance around subject hands to temporally describe each motion unit. Finally, using a codebook of motion units to represent a sequence permits capturing the dynamics and recognize the human behavior with a Dynamic Naive Bayesian classifier.

1.2 Thesis organization

The thesis is organized as follows: In Chapter 2, we lay out the issue of human behavior understanding from RGB-D data as well as existing solutions proposed in the state-of-the-art. Chapter 3 introduces the shape analysis framework that we employ to analyze and compare both shape of human pose and motion. Chapter 4, presents the method that we employ for the task of human action recognition and its evaluation in comparison with state-of-the-art on several benchmark action datasets. In Chapter 5, we propose a different method capable of handling identified failure cases and investigate more complex behaviors, like activities. Finally, we conclude this manuscript in Chapter 6 by summarizing contributions of the thesis and proposing directions of future research.

2 State-of-the-art

Sommaire

2.1 Introduction

In computer vision, the problems of human motion analysis and behavior understanding exist since many years and has attracted many researchers notably because of its large panel of potential applications.

In this Chapter, we first define the problem of human behavior understanding with its terminology and its potential applications. Then, we introduce the depth sensor technology as well as RGB-D data provided by such cameras. Benchmark datasets of such data collected for the task of human motion analysis are then presented. Finally, we review the main existing approaches in the state-of-the-art, which provide methodology to face the problem of human behavior understanding.

2.1.1 Human behavior understanding problem

The problem of human behavior understanding can be initially defined as follows: given a set of known sequences of different human behaviors, which of them is performed during an observed test sequence? The problem can then be extended to the analysis of a long unknown motion sequence, where different behaviors are performed successively and should be recognized and localized in the time by the system. Figure 2.1 illustrates this problem.

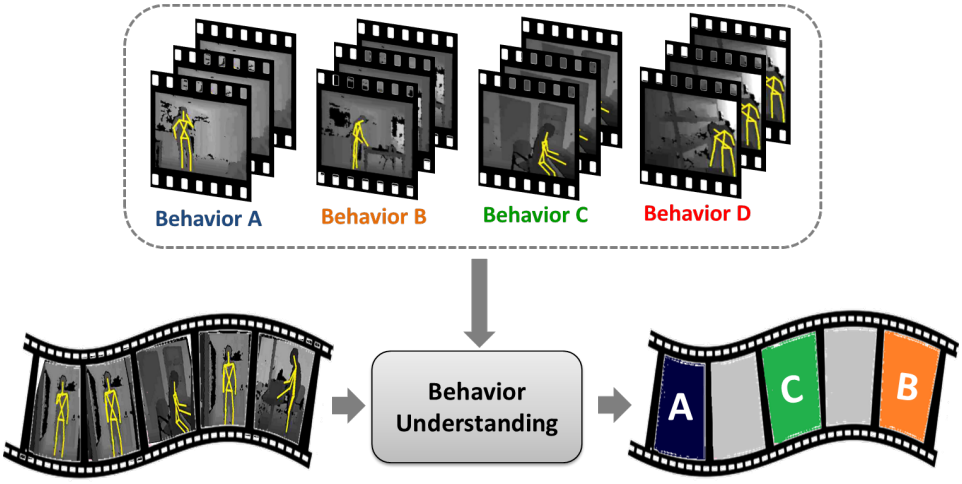


Figure 2.1 – *Illustration of the human behavior understanding problem. Given a set of learned behaviors, the system is able to recognize and localize in the time which behaviors have been performed in an unknown sequence.*

Before the release of new depth sensors, human behavior understand-

ing has been widely investigated in computer vision from 2D images or 2D videos taken from standard RGB cameras ([75, 49, 84, 77]). However, most of these methods suffer from some limitations coming from 2D videos, like the sensitivity to color and illumination changes, background clutter and occlusions.

Since the recent release of RGB-D sensors, like Microsoft Kinect [56] or Asus Xtion PRO LIVE [7], new opportunities have emerged in the field of human motion analysis and understanding. Hence, many research groups investigated data provided by such cameras in order to benefit from some advantages with respect to RGB cameras. Indeed, depth data allow better considering the 3D structure of the scene and thus easily perform background substraction and detect people in the scene. In addition, the technology behind such depth sensors allows more robustness to light variations as well as working in complete darkness.

However, the task of understanding human behaviors is still difficult due to the complex nature of the human motion. What further complicates the task is the necessity of being robust to execution speed and geometric transformations, like the size of the subject, its position in the scene and its orientation with respect to the sensor. Additionally, in some contexts, human behaviors imply interactions with objects. While such interactions can help to differentiate similar human motions, they also add challenges, like occlusions of body parts.

Moreover, the main challenge of human behavior understanding systems is the online capability. Indeed, a system able to analyze the human motion in an online way is very important for two main reasons. First, it allows a low latency making the interaction with the system more natural. Second, it allows the processing of very long motion sequences of different behaviors performed successively.

2.1.2 Behavior terminology

Before going more in detail, an initial definition of behavior terminology is necessary. During this thesis, we identify three main types of human behaviors: human gestures, human actions and human activities. Each type

of behaviors is characterized by a different degree of motion complexity, degree of human-object interaction and duration of the behavior. This is summarized in Table 2.1. However, we note that in the state-of-the-art, the boundaries between these three terminologies are often smooth as on behavior can lie between two behavior types. For instance, a simple action performed with one arm can be assimilated as a gesture. Conversely, an action performed with an object can be viewed as an activity.

Table 2.1 – *Terminology of human behaviors according to motion complexity, human-object interaction and movement duration.*

Behavior	Motion complexity	Human-object interaction	Duration
Gesture	Low	None	Short
Action	Medium	Possible	Intermediate
Activity	High	Yes	Long

Gesture

A gesture is characterized by a simple motion of only one part of the human body (often the arm). Although, due to the low level of complexity, the notion of dynamic may not be necessary and a gesture can often be characterized by a single or very few relevant human poses. The duration of a gesture does not exceed few seconds. In addition, a gesture is performed without any object. For instance gestures can be: *raising the arm up*, *bending the arm* or *pointing toward*.

Action

An action is defined as a slightly more complex movement, which can also be a combination of several gestures. Most of the time, actions are characterized by motion of several parts of the body. The duration of an action varies from few seconds to one minute approximately. In addition, some actions can be performed with an object, but in these cases, objects are held by the subject in the beginning of the action. Example of actions are: *run*, *jump*, *drink* or *phone call*.

Activity

Activities are characterized by a high level of motion complexity, where several movements or actions are performed successively. The duration of activities are of the order of minutes. In addition, what makes the activities more complex is the interaction with objects. For instance, an activity can be: *microwaving food*, which consists of the combination of *take food*, *put food into the microwave* and *take food from microwave*.

2.1.3 Applications

The main motivation behind the interest in such understanding of human behaviors, whether using standard cameras or RGB-D sensors, is the large range of applications that it offers in various fields, like entertainment, surveillance, security, motion learning and health care.

Entertainment and learning

One of the main domain in which human motion understanding approaches can be applied is the gaming/entertainment field [4, 85, 31]. Moreover, that is why the Microsoft Kinect was released as a gaming device of the video game console Microsoft Xbox.

Indeed, understanding human motions offers a new way of playing video games or interacting with the system without intermediate devices, like joysticks. In order to guarantee natural interaction with the system, the human motion analysis should be performed in real-time and also provide an effective latency by understanding the human motion before the end of its execution.

In addition, human motion analysis can be applied in motion learning system [92] in order to help the user to learn or improve a particular gesture, like sport gestures or first aid gestures. In that case, recognizing the human motion is not sufficient, but a deeper analysis is needed. The system should be able to locally detect movement mistakes, so as to provide relevant feedback to the user on how and where improving the gesture.

Surveillance and security

The field of surveillance and security has attracted many researchers in the last decades. In 2D videos, a main challenge widely investigated is the tracking of people in the scene. Thanks to RGB-D sensors, this tracking is facilitated, thus the analysis can be focused on the understanding of the scene and human behaviors [16, 26].

The goal of such surveillance systems is to observe people and detect when suspicious activities are performed. Hence, by providing an efficient human behavior understanding method, such surveillance systems are able to differentiate normal and irregular behaviors. In addition, robustness to illumination changes of RGB-D sensors, as well as their capability to work in complete darkness are very important advantages in this context with respect to standard RGB cameras.

Health care

Finally, such human behaviors analysis method can be very helpful in medical domain for both doctors and patients [66]. First, it facilitates detection and diagnosis of possible deficiency in the human motion. In addition, a deep analysis of the human motion provides to health workers accurate and relevant information, so as to observe the deficiency evolution as well as the efficiency of the treatment.

Then, it can be applied in a monitoring system, so as to detect abnormal behaviors, like falls of patients in their room [14, 55]. Hence, it allows the health workers to faster intervene in case of need.

Last, human motion analysis can also be helpful for patients. For instance, it can be applied in a rehabilitation system [70, 20, 74], so as to help the patient to perform the right rehabilitation exercises without the presence of a doctor. Such system can also compute some statistics in order to evaluate the progress of the patient in the rehabilitation task. Moreover, understanding human activity can be beneficial in robotic for helping disabled patients to achieve daily life tasks, and thus improving their quality of life.

2.2 RGB-D and 3D data

Analyzing and understanding a real-world scene observed by an acquisition system is the main goal of many computer vision systems. However, standard cameras only provide 2D information about the scene. The lack of the third dimension results in some limitations while analyzing and understanding the 3D scene. Hence, having the full 3D information about the observed scene became an important challenge in computer vision.

In order to face this challenge, research groups tried to imitate the human vision. The human perception of the scene relief is formed in the brain, which interprets the two plane images from two eyes, so as to build one single image in three dimensions. This process of reproducing relief perception from two plane images is called stereoscopic vision. It has been widely investigated in computer vision, so as to obtain the 3D information of a scene. By using two cameras observing the same scene from slightly different points of view, a computer can compare the two images to develop a disparity image and estimate the relief of the scene. For instance, this technique is currently used to create 3D movies.

For the task of human motion analysis, having 3D information about the human pose is also a challenge, which has attracted many researchers. Motion capture systems, like those from Vicon [90] are able of accurately capturing human pose, and track it along the time resulting in high resolution data, which include markers representing the human pose. Motion capture data have been widely used in industry, like in animation and video games. In addition, many datasets have been released providing such data for different human actions in different contexts, like the Carnegie Mellon University Motion Capture database [18]. However, these systems present some disadvantages. First, the cost of such technology may limit its use. Second, it implies that the subject wears some physical markers so as to estimate the 3D pose. As a result, this technology is not convenient for the general public.

More recently, new depth sensors have been released, like Microsoft Kinect [56] or Asus Xtion PRO LIVE [7]. Figure 2.2 shows pictures of these devices.

Figure 2.2 – *Example of RGB-D sensors. Left: Microsoft Kinect 2 [56]; Right: Asus Xtion PRO LIVE [7].*

In addition to standard RGB images, a depth map is also provided giving for each pixel the corresponding distance with respect to the sensor. As a result, the 3D scene can be estimated from such depth maps. Behind these depth sensors, two types of technology exist:

- **Structured light:** A known pattern is projected in the scene. Then, a sensor analyzes the distortion of the pattern in contact with object, and thus estimates the distance of each point of the pattern;
- **Time of flight:** A light signal is emitted in the scene. Knowing the speed of light, a receptor then computes the distance of the object based on the time elapsed between the emission of the signal and its reception.

Depth sensors, like Microsoft Kinect 1 or Asus Xtion PRO LIVE employ the structured light technique, while the new Microsoft Kinect 2 employs the time of light technology. In both cases, an invisible light system is used (infra-red projector and sensor).

These new acquisition devices have stimulated the development of various promising applications, including human pose reconstruction and estimation [79], scene flow estimation [30], hand gesture recognition [72], and face super-resolution [13]. A recent review of kinect-based computer vision applications can be found in [32]. The encouraging results shown in these works take advantage of the combination of RGB and depth data enabling simplified foreground/background segmentation and increased robustness to changes of lighting conditions. As a result, several software libraries make it possible to fit RGB and depth models to the data. The

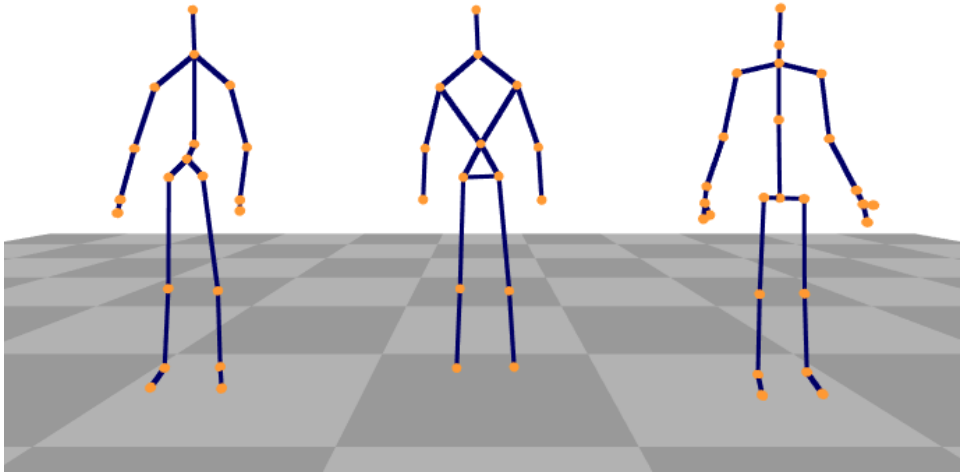


Figure 2.3 – *Three various skeletons with different number of joints obtained with Microsoft SDK 1.x (left), OpenNI SDK (middle) and Microsoft SDK 2.0 (right)*

two main software libraries are Microsoft Kinect SDK [57] and OpenNI SDK [63].

Additionnaly, Shotton et al. [79] proposed a new real-time method to accurately predict the 3D positions of body joints in individual depth maps, without using any temporal information. As a result, the human pose can be represented as a 3D humanoid skeleton with a certain number of joints. In [79], results report the prediction accuracy for 16 joints, although the tracking system developed on top of this approach by the different SDK provides different number of joints. While the OpenNI SDK provides a skeleton with 15 joints, the first version of the Microsoft Kinect SDK (1.x) was capable to estimate 3D positions of 20 joints. Recently, the new version of the Microsoft SDK (2.0) released with the Kinect 2 increased the number of estimated joints to 25 with notably 4 new joints representing two fingers per hand. Figure 2.3 shows example of the three types of skeleton.

As a result, such RGB-D sensors provide for each frame the 2D color image of the scene, its corresponding depth map and a 3D humanoid skeleton representing the subject pose, as illustrated in Figure 2.4. The combination of such data offers new opportunities for analyzing the human motion and understanding human behaviors.

Figure 2.4 – *Example of data provided by RGB-D sensors like Microsoft Kinect 2: 2D color image (left), depth map (middle) and 3D skeleton (right).*

2.3 Datasets

The emergence of RGB-D data has encouraged research groups to build new datasets for the task of human motion analysis. As a result, numerous RGB-D datasets are currently publicly available presenting various challenges and contexts. Table 2.2 summarizes the most popular RGB-D datasets publicly available for the task of human motion analysis. As shown in Table 2.2 these datasets have been collected for different contexts:

- **Single Person Recognition:** The goal is to recognize the behavior performed by one subject in front of the camera;
- **Two Persons Recognition:** In this context, the objective is to recognize the interaction between two subjects in front of the camera;
- **Robot-centric Recognition:** These recent context proposes to recognize the interaction between a subject and a robot from the point of view of the robot;
- **Multi-view:** The aim of this context is to evaluate the robustness of recognition methods to different point of views. Generally, behaviors from one point of view are used for training and the same behaviors from a different point of view are used for testing the approach;
- **Online detection:** Instead of only one behavior performed in a sequence, this more realistic context provides long sequences, where a subject is performing different behaviors successively. The goal is to recognize as well as localize in the time the performed behaviors;

- **Person re-identification:** The goal is to recognize a known person in a new sequence;
- **Fall detection:** In this context, a subject is performing some daily activities and suddenly falls down. The purpose is to detect these falls as fast as possible.

Additionally, Table 2.2 provides for each dataset its size (number of subjects and classes and total number of sequences), the type of provided data and the type of behaviors performed by the subjects. Figure 2.5 shows example frames of some datasets collected for different contexts.

In this thesis, we focus our study on human behavior recognition and understanding. In order to evaluate and compare our work for gesture recognition, we use the MSRC-12 dataset [28]. For action recognition, we employ the Florence 3D Action [22], MSR Action 3D [46], UTKinect [99] and UCF Kinect [27] datasets. For the more complex context of activity recognition, evaluations are conducted on Cornell Activity 120 [41] and Online RGBD [106] datasets. Finally, so as to analyze the online capability of our method, we adopt the Online RGBD [106] and MAD [34] datasets.

2.4 Related work

Due to the high number of potential applications and publicly available datasets, many works have been proposed in the literature to address the problem of human behavior understanding from RGB-D data. Recent surveys summarize these works [19, 21, 105]. These RGB-D based approaches benefit from the large number of works published in the last two decades on human activity recognition in 2D video sequences (see for example the recent surveys in [97, 86, 14, 68]).

While some works focus on relatively simple behaviors, like gestures or actions, other methods tackle more complex behaviors, like activities. More recently, researchers also address the issue of online detection of human behaviors.

Table 2.2 – *Summary of the most popular RGB-D datasets for human motion analysis.*

Dataset	**Size**	**Data**	**Context**	**Behavior**
Composable Activities [47]	14 subjects 16 classes 693 sequences	RGB-D Skeleton	Single person Recognition Online detection	Activity
Concurrent action [96]	12 classes 61 long videos	RGB-D Skeleton	Single person Multi-view Online detection	Activity
Cornell Activity 60 [83]	4 subjects 12 classes 60 sequences	RGB-D Skeleton	Single person Recognition	Activity
Cornell Activity 120 [41]	4 subjects 10 classes 120 sequences	RGB-D Skeleton	Single person Recognition	Activity
Florence 3D Action [22]	10 subjects 9 classes 215 sequences	RGB-D Skeleton	Single person Recognition	Action
G3DI [15]	12 subjects 15 classes	RGB-D Skeleton	Two persons Recognition	Interaction
MAD [34]	20 subjects 35 classes 40 long videos	RGB-D Skeleton	Single person Online detection	Action
MSRC-12 [28]	30 subjects 12 classes 594 sequences	Skeleton	Single person Recognition	Gesture
MSR Action 3D [46]	10 subjects 20 classes 3 tries	RGB-D Skeleton	Single person Recognition	Action
MSR Daily Activity [93]	10 subjects 16 classes 2 tries	RGB-D Skeleton	Single person Recognition	Activity
MSR Gesture 3D [5]	10 subjects 12 classes 2-3 tries	Depth	Single person Recognition	Gesture
Online RGBD [106]	36 subjects 7 classes 340 samples 36 long videos	RGB-D Skeleton	Single person Recognition Online detection	Activity
RGBD Person Re-identification [9]	79 subjects	RGB-D Skeleton	Single Person Re-identification	Walking
Robot-Centric Activity [100]	8 subjects 9 classes 366 samples	RGB-D	Robot-centric Recognition	Activity
SBU Kinect [107]	7 subjects 8 classes 300 sequences	RGB-D Skeleton	Two persons Recognition	Interaction
TST Fall detection [29]	11 subjects 240 sequences	Depth Skeleton	Fall detection	Activity
UCF Kinect [27]	16 subjects 16 classes 5 tries	Skeleton	Single person Recognition	Action
UR Fall Detection [43]	70 classes	RGB-D	Fall detection	Activity
UTKinect [99]	10 subjects 10 classes 2 tries	RGB-D Skeleton	Single person Recognition	Action
UWA3D Multiview Activity II [71]	10 subjects 30 classes 4 tries	Depth	Single person Multi-view Recognition	Activity

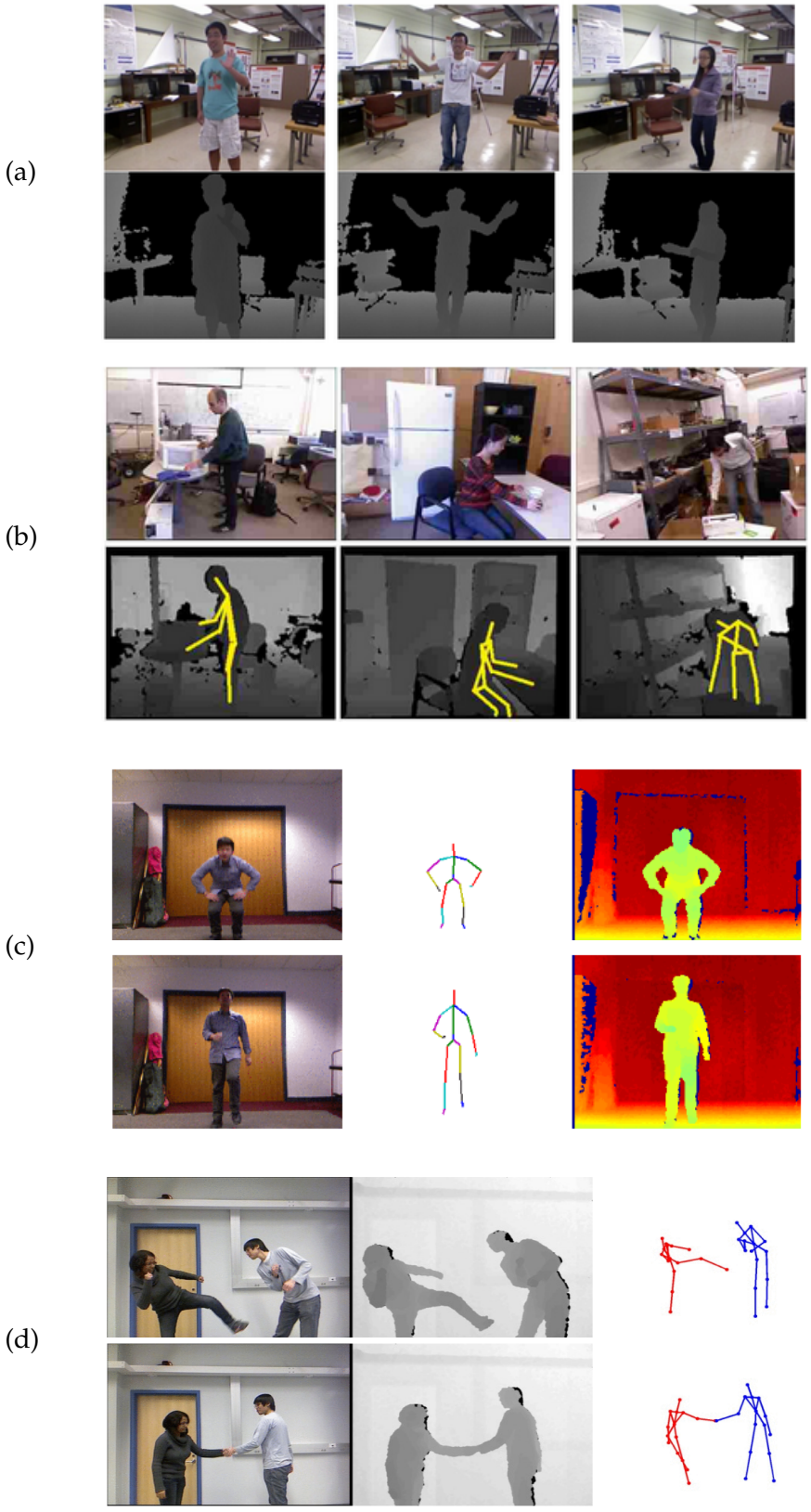


Figure 2.5 – *Example frames of various datasets collected for different contexts: (a) UTKinect; (b) Cornell Activity 120; (c) MAD; (d) SBU.*

2.4.1 Action recognition

In recent years, recognition of human actions from the analysis of data provided by RGB-D cameras has attracted the interest of several research groups. The approaches proposed so far can be grouped into three main categories, according to the way they use the depth channel: *skeleton*-based, *depth map*-based and *hybrid* approaches. Skeleton based approaches, model the human pose in subsequent frames of the sequence using the 3D position of skeleton joints. Depth map based approaches extract appearance and temporal features directly from the overall set of points of the depth maps in the sequence. In addition to these approaches, there are also some *hybrid* methods that exploit the combination of depth, skeleton or color information to improve results. Following this categorization, existing methods for human action recognition using depth information are shortly reviewed below.

Depth map-based approaches

Methods based on depth maps rely on the extraction of meaningful descriptors from the entire set of points of depth images. As depth map are like RGB images, but with a single channel, some work proposed to extend and adapt features or techniques existing in RGB images literature to depth map. Once the features are computed, different methods have been proposed to model the dynamics of the actions.

The approach in [46] proposes to project depth maps onto three orthogonal Cartesian planes (X-Y, Y-Z and X-Z) representing the front view, the side view and the top view respectively. Then, 2D contours of such projection are computed for each view and used to retrieve the corresponding 3D human silhouette for each frame of the sequence. By employing the bag-of-word paradigm on these 3D silhouettes, they can identify a set of salient postures, which are used as nodes in an action graph modeling the dynamics of the actions. Figure 2.6 illustrates the idea of 3D silhouettes. However, this representation is view dependent as different orientation of the subject would result in different projections onto the three Cartesian planes.

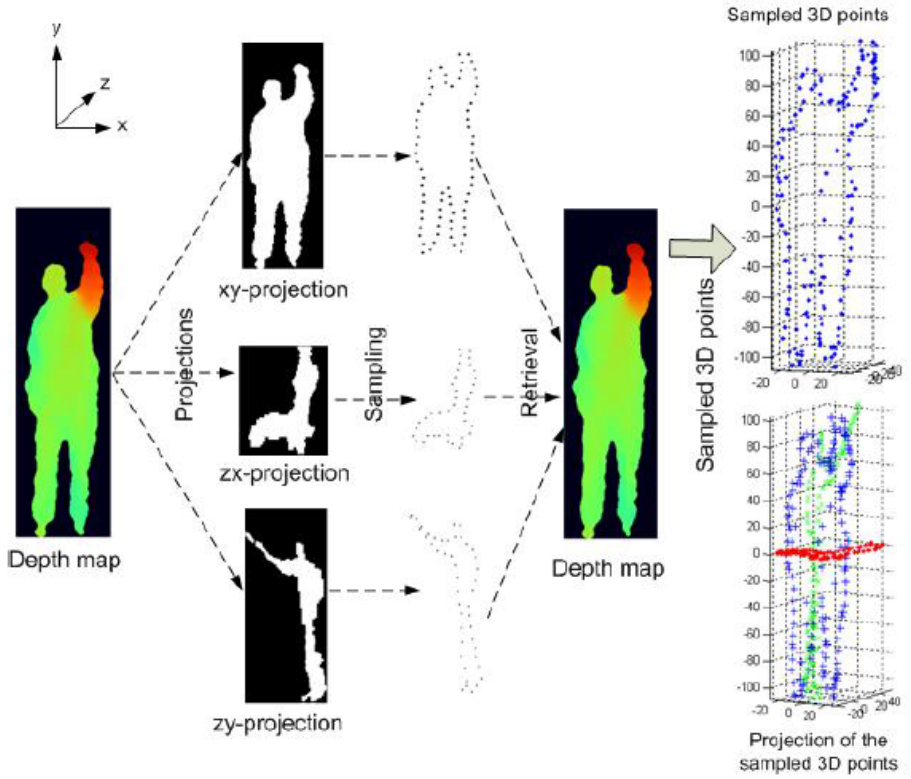


Figure 2.6 – *Computation of 3D silhouette in [46]. The depth map is projected onto three Cartesian planes to then build a 3D silhouette from 2D contours.*

The idea of projecting each depth map onto these three orthogonal planes is also employed in [104]. Then, the motion energy is obtained by computing and thresholding the difference between two successive maps. Such motion energy is then stacked through all the video sequence resulting in a Depth Motion Map (DMM) for each view. Such DMM highlights areas where main motion takes place during the action. Histogram of Oriented Gradients (HOG) is then applied to DMM maps to extract features for each view. The three HOG features are concatenated to build a single feature for each sequence. A SVM classifier is trained on this feature to perform action recognition. The process of computing this final HOG feature is displayed in Figure 2.7. Similarly to [46], this methods suffers from its view dependency.

Other works propose to extend the idea of spatio-temporal interest points (STIPs) to depth data. Indeed, its capability to handle clutter background and partial occlusions has been proven in RGB video. Hence, the work in [98] proposes to apply this idea to depth maps by extract-

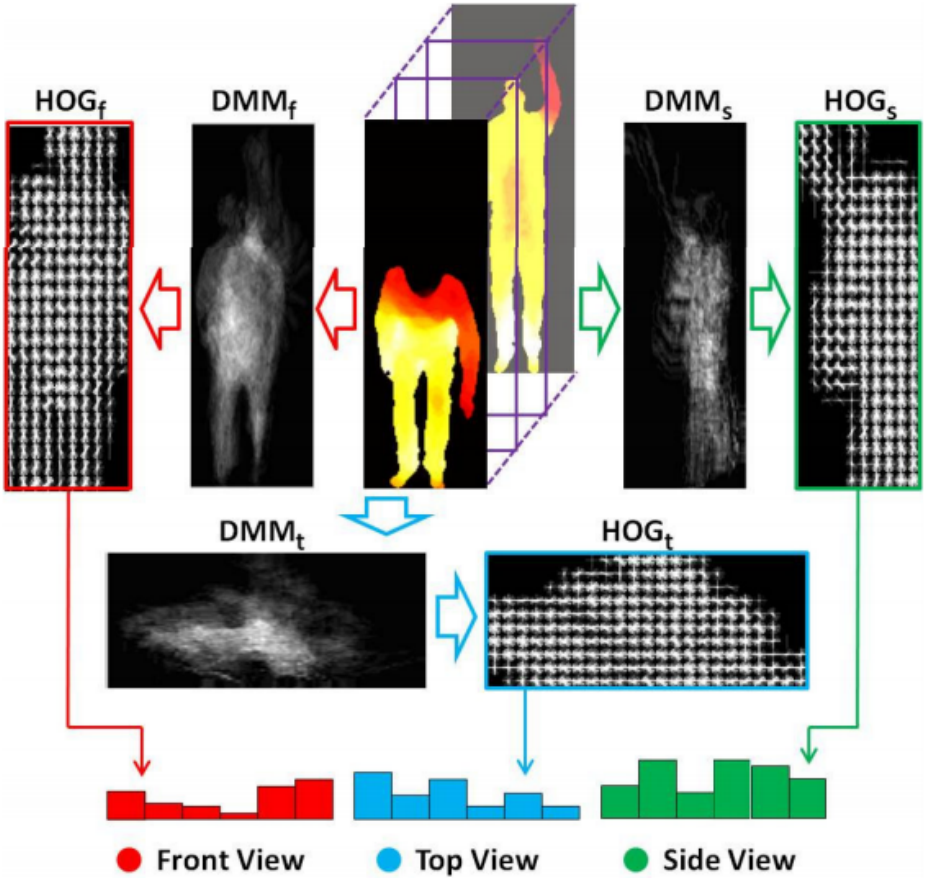


Figure 2.7 – *Process of computed HOG features on Depth Motion Maps [104].*

ing STIPs from depth video called DSTIPs. Then, the 3D cuboid around each DSTIP is considered to compute the depth cuboid similarity feature (DCSF) describing the local depth appearance within each cuboid. Finally, the bag-of-words approach is employed to identify a cuboid codebook and represent the actions. Figure 2.8 illustrates the process of DCSF computation. In [110], a novel formulation of the cuboid descriptor is proposed based on its sparsification and its quantization. This feature called 3D sparse quantization (3DSQ) is then employed in a spatial temporal pyramid (STP) [44] for hierarchically describing the action. A similar idea of STIP is proposed by Rahmani et al. [71], where key-points are detected from the 3D point cloud. Each point is then described using the Histogram of Principal Components (HOPC). The main advantage of this method is its robustness to viewpoint, scale and speed variations.

Such robustness are important challenges investigated by many researchers. For instance, a binary depth feature called range-sample depth

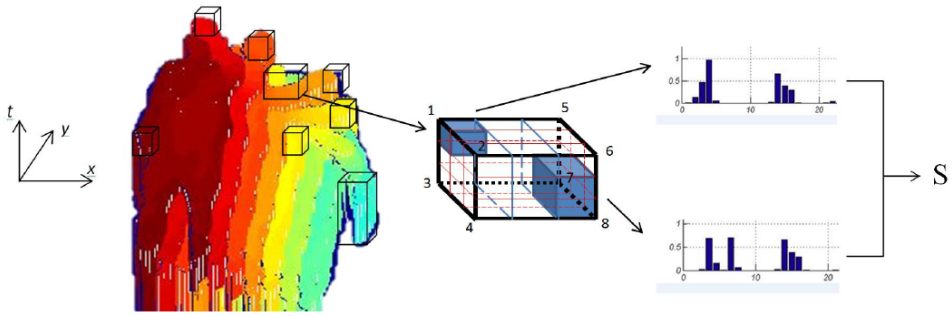


Figure 2.8 – *Illustration of the DCSF computation process [98].*

feature is proposed by Lu and Tang [50]. This feature describing both shape geometry and motion is robust to occlusion as well as possible changes in scale, viewpoint and background.

Instead of directly working on depth maps, other methods propose to consider a depth sequence as a 4D space (3D+t) divided into spatio-temporal boxes to extract features representing the depth appearance in each box. For instance, Vieira et al. [91] propose to divide the 4D space into a grid containing a certain number of 4D cells. Then the spatio-temporal occupancy pattern (STOP) is computed within each 4D cell. Such feature counts the number of points that fall into the spatio-temporal grid. Figure 2.9 illustrates such 4D space divided into 4D cells. By applying a threshold on this feature, they are able to detect which 4D cells correspond to motionless (red in the figure) and which correspond to motion (green in the figure). The concatenation of such feature of each 4D cell is used to represent the depth sequences. A similar occupancy feature called random occupancy pattern (ROP) is also employed in [93]. Differently, the 4D sub-volumes are extracted randomly at different locations and with different sizes.

The 4D space is also investigated in [64]. As shown in Figure 2.10, 4D normals over the 4D space are first computed. Then, the depth sequence is partitionned into spatio-temporal cells. Within each cell the orientation of 4D normals are quantified using 4D projectors to build a 4D histogram. This feature, called histogram of oriented 4D normals (HON4D), captures the distribution of the normal vectors for each cell. The idea of computing surface normals within spatio-temporal cells is also used by Yang and

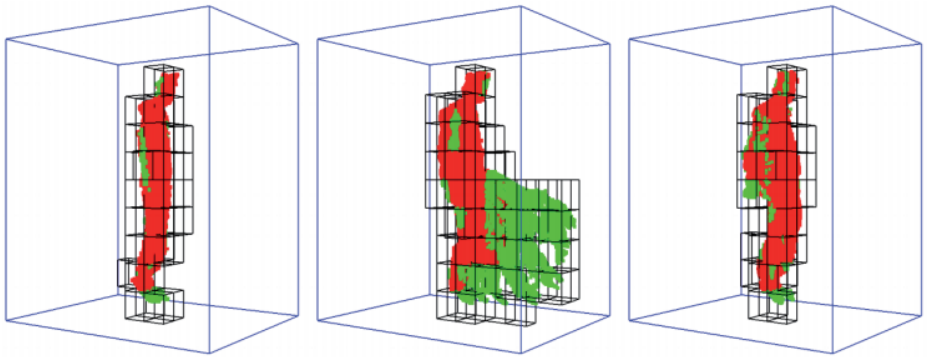


Figure 2.9 – *Illustration of the STOP feature proposed in [91].The spatio-temporal space is divided into 4D cells where motionless (red) and motion (green) regions are identifiable [91].*

Tian [103] to describe both local motion and shape information characterizing human action. However, a limitation of such normal methods is that they assume correspondence between cells across the sequence. Hence, these methods may fail when the subject significantly changes his spatial position during the performance of the action.

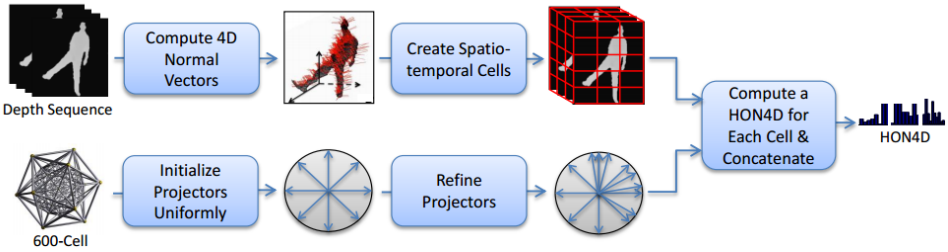


Figure 2.10 – *Overview of the approach proposed in [64] to compute the HON4D feature.*

Skeleton-based approaches

Skeleton based approaches have become popular thanks to the work of Shotton et al. [79] that makes available a representative 3D humanoid skeleton for each frame. These methods exploit such skeleton data instead of the whole depth map to describe the human pose and the action dynamics.

In [102], three features for each joint are extracted based on pair-wise differences of joint positions: in the current frame representing human posture (f_{cc}); between the current frame and the previous frame representing instantaneous motion (f_{cp}); and between the current frame and

the initial frame of the sequence representing global motion (f_{ci}). This latter is assumed to correspond to the neutral posture at the beginning of the action. Since the number of these differences results in a high dimensional feature vector, principal component analysis (PCA) is used to reduce redundancy and noise, and to obtain a compact *EigenJoints* representation of each frame. The computation of EigenJoints features is illustrated in Figure 2.11. Finally, a Naïve-Bayes nearest-neighbor classifier is used for multi-class action classification.

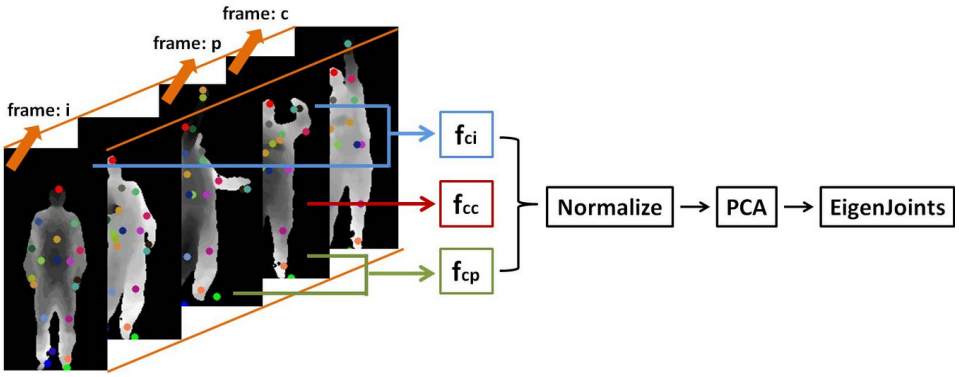


Figure 2.11 – *Computation of the EigenJoints feature proposed in [102].*

Xia et al. [99] propose to quantify the 3D space into bins using a spherical coordinate system centered at the hip joint, as shown in Figure 2.12. Then, each 3D joint is voted into the corresponding 3D bin to build the histogram of oriented joints 3D (HOJ3D) feature for each frame. Then, the resulted histograms are clustered to identify posture words representing prototypical poses of human actions. Finally, the temporal evolution of these visual words is modeled by discrete Hidden Markov Models to describe action dynamics. Such bag-of-poses idea is also employed in [76], but differently as the prototypical poses are the set of poses belonging to training sequences. A different pose description is employed by using kinematic chains. As shown in Figure 2.13a, a basis is built from torso joints and serves as the root of the kinematic tree. Then, each of the remaining joints is expressed relatively to its parent joint. Similarly to [102], a Naïve-Bayes nearest-neighbor classifier is finally employed for action classification.

In Ofli et al. [60, 61], they propose to highlight physical meaning of

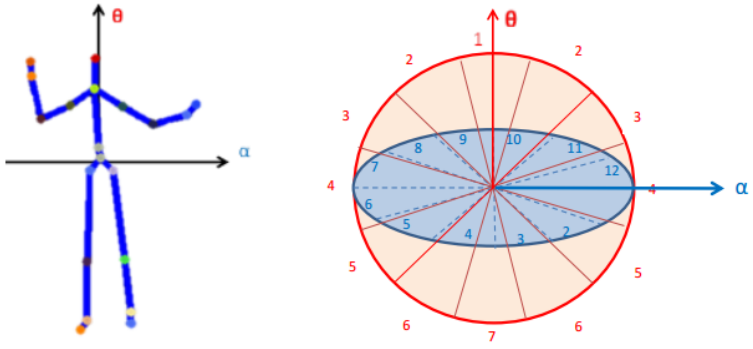


Figure 2.12 – *Spherical coordinate system employed in [99] to compute he HOJ3D feature.*

computed features. They introduce a novel representation called sequence of most informative joints (SMIJ) based on features representing angle between joints, as shown in Figure 2.13b. Hence, few informative skeletal joints are automatically selected at each time instance based on highly interpretable measures, such as mean or variance of the joint angles and maximum angular velocity of the joints. This selection varies according to the action classes. As a result, actions are represented by its corresponding SMIJ and the normalized edit distance [54] is used for comparison.

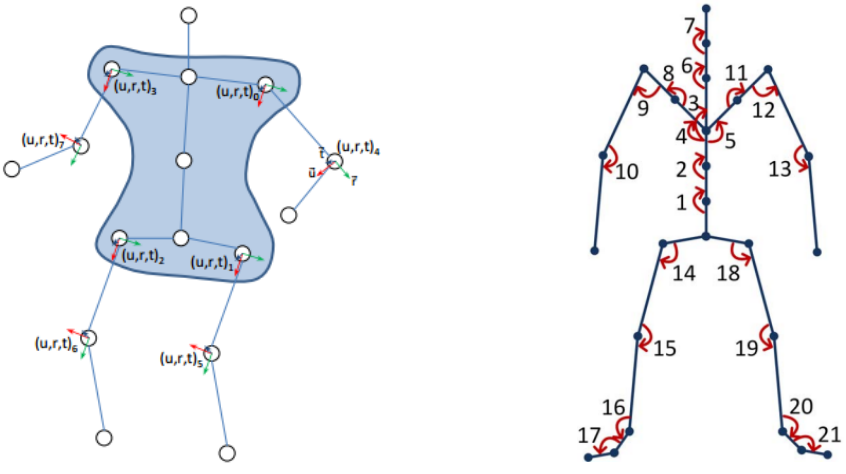


Figure 2.13 – *Example of skeletal representations used in (a) Seidenari et al. [76] and (b) Ofli et al. [60, 61].*

Other methods, like [52] and [112] propose deeper sequence representations by employing the sparse coding and dictionary learning (DL) techniques, which have been successfully applied to many computer vision

problems [2, 109, 35]. The idea behind dictionary learning is that data can be represented by a linear combination of few atoms from a learned dictionary. Existing skeletal features, like the one proposed in [102] or [108] are used in the dictionary learning algorithms to quantize the skeleton features and identify discriminative key poses for each action class. Temporal pyramid [101] is used for modeling temporal dynamics of actions and a SVM classifier is employed for action recognition.

Other approaches use differential geometry to represent skeleton data, so as to consider the non-linear nature of human motion. In [89], the authors represent each skeleton as one element on the *Lie*-group, and the sequence corresponds to a curve on this manifold, as shown in Figure 2.14. To handle rate variability among curves, Dynamic Time Warping (DTW) [11] is employed to temporally align the curves. Finally, classification is performed using linear SVM. In [80], Slama et al. express the time serie of skeletons as one point on a Grassmann manifold, where the classification is performed benefiting from Riemannian geometry of this manifold.

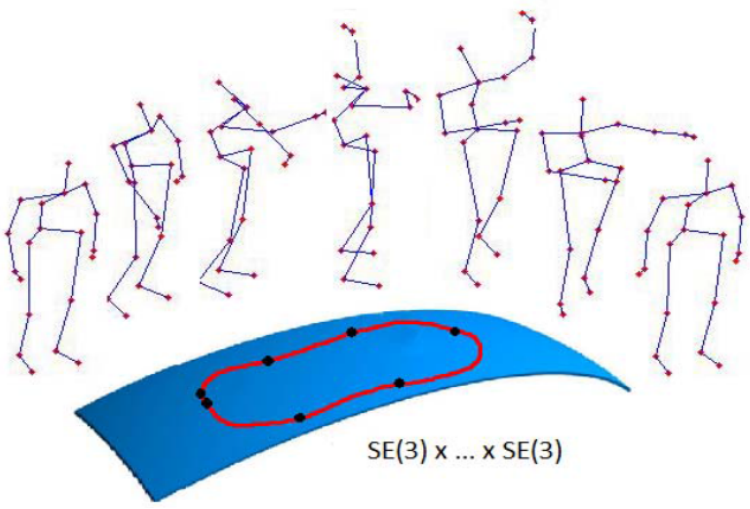


Figure 2.14 – *Representation of a skeleton as one point on a Lie Group. As a result, the full sequence is viewed as a curve [89].*

Recent works address more complex challenges in real-time action recognition systems, where a trade-off between accuracy and latency becomes an important goal. For example, Ellis et al. [27] target this trade-off by adopting a Latency Aware Learning (LAL) method for reducing la-

tency when recognizing human actions. A logistic regression-based classifier is trained on 3D joint positions sequences to search a single canonical posture for recognition. Another work investigates the challenge of low latency by proposing a 3D kinematic descriptor called the Moving Pose [108]. This feature considers both the pose information as well as differential quantities (speed and acceleration) of the body joints within a short time interval. A *k*NN-based classifier is employed for classification.

Hybrid approaches

Instead of focusing on one single channel provided by RGB-D sensors, some methods investigate the combination of these different flows, so as to benefit from their respective strengths. For instance, the work in [59] introduces two extended methods for fusing color and depth information. First, an extension of STIPs called Depth-Layered Multi-Channel STIPs (DLMC-STIPs) is employed by dividing such STIPs into several depth layers resulting in a multiple depth channel histogram representation. Second, the Three-Dimensional Motion History Images (3D-MHIs) approach, which equips the (MHIs) [23] with motion history in the depth channel is introduced. The same channels are also used in [48], where authors propose a learning method to simultaneously extract and fuse features from RGB and depth data.

Differently, the method proposed in [62] combines depth data and skeleton data to improve the action recognition accuracy. As skeleton features, similarity distances between joint angles are employed to build the joint angle similarities (JAS) feature. For depth features, HOG features are computed around each joint from each depth frame and concatenated over the time resulting in a matrix. HOG features are computed a second time on this matrix to obtain the HOG^2 feature. Finally, both features are combined and linear SVM is used for classification. This process is illustrated in Figure 2.15.

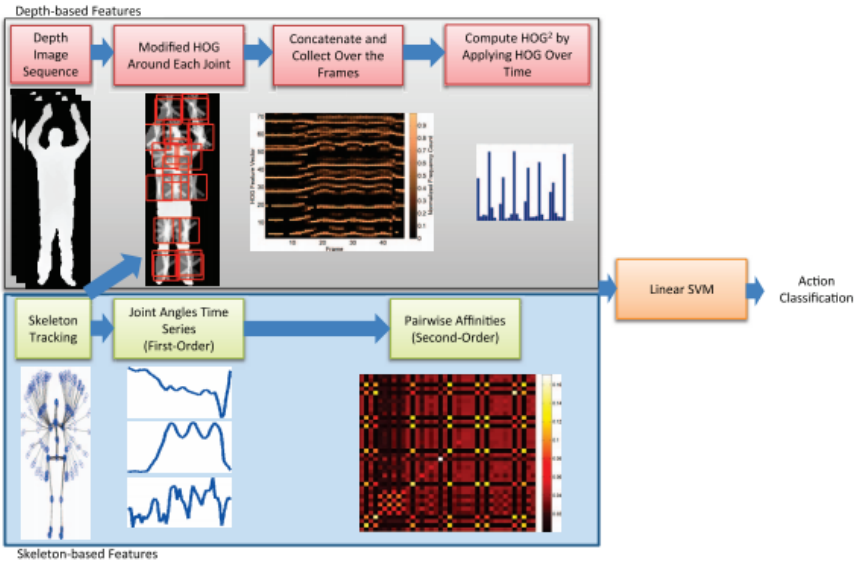


Figure 2.15 – *Illustration of the method proposed in [62] combining skeleton and depth features for action recognition.*

2.4.2 Activity recognition

As described previously, activities are mainly characterized by human-object interaction and more complex motions. In the state-of-the-art, these properties are handled differently.

In order to simultaneously represent the human motion and the interaction with objects, hybrid approaches are often employed. For instance, Ni et al. [58] propose to fuse gray-scale data with depth data to improve the detection of human-object interaction. Indeed, spatial and temporal contextual information, like relative distance or relative speed are important cues for representing human-object or human-surrounding interactions. However, due to perspective projection, extracting these information from 2D data may be inaccurate. To overcome this limitations, depth data providing 3D information about the scene is employed in addition to gray-scale data.

Other methods propose to combine depth information and skeleton data to detect and describe such manipulation of objects. Indeed, when activities mainly differ due to these manipulations, using only skeleton data is not sufficient to guarantee an effective recognition of activities. However, skeleton data provide accurate position of body parts, thus it facilitates the detection of such interaction. Indeed, it reduces the number

of possibilities where human-object interaction may take place. For instance, Wang et al. [94] propose to describe the depth appearance around each skeleton joint in addition to its position in the space. To this end, they consider the local region using a 3D spatial grid centered at the joint. Then the local point cloud is computed from depth map and the occupancy pattern feature called local occupancy pattern (LOP) is computed. This feature, similar to [91, 93], represents the spatial distribution of the point cloud in the local region. Furthermore, the temporal structure of each joint in the sequence is represented through a temporal pattern representation called Fourier Temporal Pyramid. This latter is insensitive to temporal misalignment and robust to noise, and also can characterize the temporal structure of the actions. The process is illustrated in Figure 2.16.

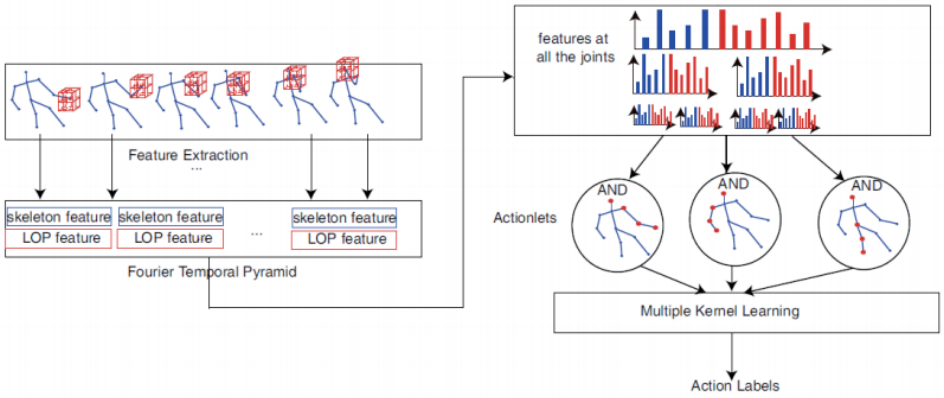


Figure 2.16 – *Illustration of the method proposed in [94] to handle simultaneously human motion and human-object-interaction.*

The same feature is also employed in [106] in addition to a set of skeleton features, like joints locations pairwise differences and temporal variations. To detect human-interaction, the authors consider the potential object positions by learning the distance between hands (both left and right) and object center from training sequences. The LOP feature is then employed to describe depth appearance in each potential object position. For classification, both skeleton and depth features are used to build middle level features called orderlets, which are robust to noise and missing joints. Similarly, Oreifej et al. [64] extend their method by computing their proposed feature HON4D, described previously, in local regions around joints to characterize manipulation of objects and recognize activities.

So as to handle the high complexity of human motion characterizing activities, methods proposed in the state-of-the-art consider the sequence more locally at the level of pose or short temporal segments.

In [47], Lillo et al. propose to decompose the motion complexity by adopting a three semantic levels hierarchical model, consisting of activities, actions and poses, as illustrated in Figure 2.17. At the lower level, body poses are encoded in a representative, but discriminative pose dictionary. At the intermediate level, simple human actions are composed by a set of successive poses. At the highest level, the model captures temporal and spatial compositions of actions into complex human activities. In addition, the authors also divide each pose into different spatial regions to capture regions that are relevant for each activity.

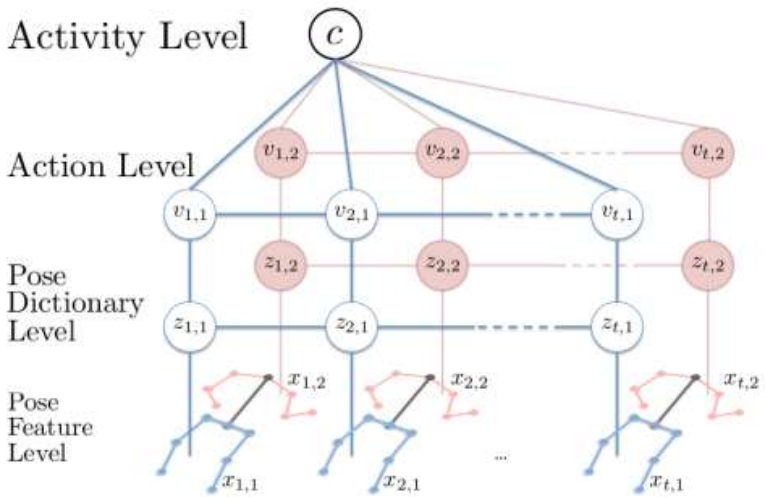


Figure 2.17 – *The hierarchical model proposed in [47] with three levels of semantic corresponding to human pose, action and activity.*

Koppula et al. [41] propose to combine both temporally local analysis of the motion and detection of possible objects to handle simultaneously the two main challenges characterizing activities. In so doing, they consider short time interval, where they describe human poses through skeleton features, like joints positions and displacements, and objects through RGB features. In order to explicitly model the human-object interaction as well as the evolution of the temporal segments, they define a Markov Random Field (MRF). In this Markov model, nodes represent objects and human motion, and edges represent the relationships between object affordances, their relations with sub-activities, and their evolution over time,

as shown in Figure 2.18. A structural SVM is employed to formulate the learning problem. An extended graphical model called Conditional Random Field (CRF) is proposed in [42], so as to detect past activities, but also anticipate which activity will happen in the future.

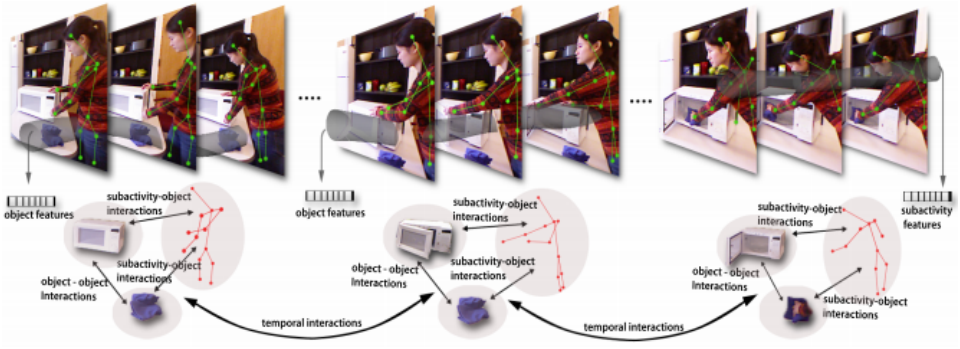


Figure 2.18 – *Pictorial representation of the different types of nodes and relationships modeled in [41].*

A graphical model is also employed by Wei et al. [95] to hierarchically define activities as combination of sub-events or atomic events including description of the human pose, the object and interaction between them. To describe human pose, differences between 3D joints coordinates of two successive frames are employed. To detect and describe possible objects in the scene, a sliding window strategy is employed with a combination of HOG features computed from both RGB and depth data. Finally, to characterize the interaction between the human and detected objects a geometric compatibility measure is computed to define if the detected objects are close enough to human arms. Figure 2.19 shows an example of a hierarchical model of an activity event.

2.4.3 Online detection

Some of the works reviewed above [95, 106], have also *online* action recognition capabilities, as they compute their features within a short sliding window along the sequence. This more complex challenge has recently been investigated on RGB-D videos. Indeed, in addition to provide fast recognition of the behavior without observing the whole sequence, such online capabilities also allow the processing of long continuous sequences,

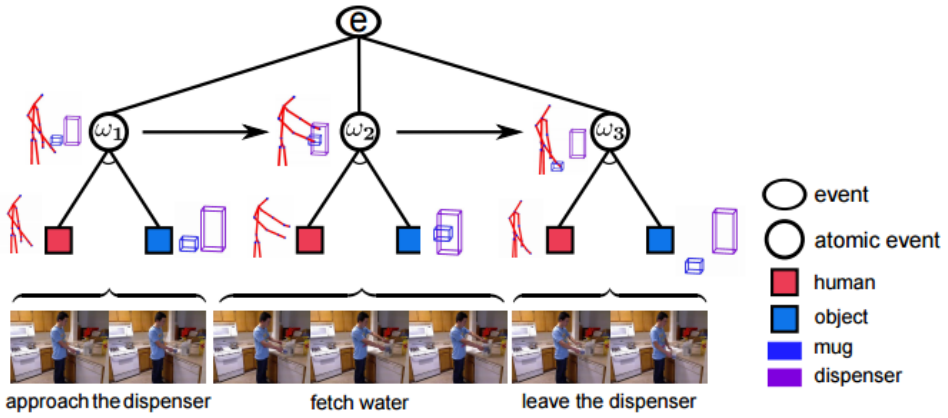


Figure 2.19 – *Hierarchical graph model of the activity event fetch water from dispenser proposed in [95].*

where several actions or activities are performed successively. This corresponds to a more realistic scenario.

For instance, Wei et al. [96] combine wavelet features for each body joint and a sequential window search algorithm, so as to detect actions in long videos in an online manner.

Huang et al. [34] proposed and applied the sequential max-margin event detector algorithm (SMMED) on long sequences comprising many actions in order to perform online detection. So as to reduce the number of possible candidates and improve computational efficiency, the proposed method allows the sequential discarding of non corresponding action classes.

2.5 Conclusion

The many advantages of RGB-D sensors over standard RGB cameras encourage researchers to investigate the new provided data for the task of human behavior recognition.

Depth map-based methods propose to exploit depth images providing natural surfaces to describe the geometry of the scene and behaviors. Extension of existing features in RGB videos are often proposed for depth sequences. While some methods focus on the description of the global sequence, other approaches detect local key-points to then describe depth appearance in these local regions. Differently, other works use the depth

information to explore the 3D point cloud of the scene or consider the sequence as a spatio-temporal 4D space.

Skeleton-based methods benefit from the 3D humanoid skeleton, which can be estimated from depth images. Such data provide accurate position of several body parts and thus allow focusing on the human motion, which mainly characterizes human behaviors. Experiments conducted in the literature demonstrated that these skeleton data are sufficient to characterize human motion and efficient for recognition of relatively simple behaviors, like gestures and actions.

However, considering the human motion may not be sufficient for more complex behaviors, like activities. Indeed, such activities are also characterized by interactions with objects. Thus, skeleton data are not suitable for describing these interactions. In this context, hybrid approaches combining strengths of skeleton and depth features are appreciated. In addition, activities also involve combinations of short movements resulting in more complex human motions. Hence, temporally local analysis of human movement is required to handle such motion complexity.

Furthermore, some emerging and interesting techniques reformulating the action recognition problem over non-linear spaces have been investigated. Motivated by their effectiveness in the 2D videos literature, such methods propose to consider geometrical spaces, like Grassmannian manifolds to represent human motion features computed from depth or skeleton data. Due to the non-linearity of human motion, such non-linear manifolds are suitable to capture the dynamic of the motion. However, only few methods employing non-linear manifolds with RGB-D data have been proposed in the literature.

Finally, the online detection challenge has been recently studied. First, such online capability allows recognizing the behavior before the end of its execution, thus it guarantees more natural interaction between the user and the system. Second, it favors the analysis of continuous sequences where several behaviors are performed successively.

All these considerations motivate us to differently address the problem

of human behavior understanding, according to the type of behavior and the context.

Hence, in the following Chapters, we first investigate the human action recognition problem by employing skeleton data only. So as to be robust to geometric transformation of the human body, we propose a translation and rotation invariant representation of the human action as the trajectory of the skeleton joints along the time. In order to consider the non-linearity of such trajectory characterizing the human motion, we propose to consider its shape in a non-linear Riemannian manifold. The comparison and classification of trajectories is performed using an elastic metric allowing robustness to different speed of executions of actions.

In a second time, we extend the method by segmenting the whole human motion into shorter motion units by analyzing the human pose shape deformation. In addition to human motion, we also consider the depth appearance around subject hands to characterize possible manipulations of objects. The sequences of motion units are modeled and classified through a Dynamic Naive Bayesian classifier. This allows us to handle more complex behaviors, like activities.

Finally, we also focus on online detection of behaviors, so as to process long sequences of several behaviors and thus meet this recent and realistic challenge.

3 Shape Analysis

Sommaire

3.1 Introduction

Over the last decades, shape analysis has been widely investigated in computer vision for different application domains, like object recognition in a scene, evolution of illness in medical imaging or face recognition in security. In the context of human motion, the effectiveness of shape information for representing human activity has been demonstrated in several state-of-the-art works, like [51, 78, 87, 1].

In this work, we investigate such shape information, so as to carry out our study on human behavior understanding.

3.1.1 Motivation of shape analysis

In order to face the issue of human behavior understanding, we focus our work on the analysis of both the human pose and the human motion that characterize such behaviors. The combination of these two analysis provides information about the human body at each time as well as its evolution along a time interval. So as to achieve such analysis, we believe that the shape cue is very important due to the geometric nature of human pose and motion.

First, a human pose can be characterized by the spatial configuration of different body parts with respect to the others in the scene. An intuitive way to capture the geometry of the human body is to consider its shape. Hence we propose to analyze the shape of such spatial configuration of body parts for human pose analysis.

Second, human motion is characterized by the evolution of the human pose along the time. In order to capture the geometric deformation of the pose along the time as well as the dynamic of the movement, we propose to consider the human motion as a trajectory of the human pose and analyze its shape.

As a result, we recast the problem of human pose and human motion analysis to a problem of shape analysis.

3.1.2 Riemannian shape analysis framework

However, in order to guarantee a robust shape analysis, the representation should be invariant to geometrical shape transformations, as well as to elastic transformation of the shape. In the case of human pose and motion analysis, these transformations are characterized by the position and the orientation of the subject in the scene, the variability of size among human people and the diverse speeds of execution of movements.

So as to analyze shape of human body and human motion while facing the constraints stated above, we employ a *Riemannian Shape Analysis* framework. Such framework allows us to capture and interpret shapes of curves in $\mathbb{R}^n$ within a Riemannian manifold and provides an elastic metric to measure the similarity between such shapes. In addition, using such manifold offers a wide variety of statistical and modeling tools that can be used to improve and deepen the analysis of human behaviors.

3.2 Mathematical framework

In the following, we introduce the mathematical framework, proposed by Joshi et al. [36], that we employ to represent, analyze and compare the shape of human poses and motion trajectories.

Note that, as explained in the following Section, human poses and motion trajectories are not lying in a same Euclidean space. Indeed, while human poses stand in the 3D space, motion trajectories are represented in a higher dimensional space. Hence, two distinct shape analysis are achieved for human pose and human motion. In the following, the generic terminology 'curve' is employed. It refers to both human poses and human motion trajectories.

3.2.1 Representation of shapes

Beforehand, in order to carry out shape analysis of a curve, we need to define it. Let $\beta : I \rightarrow \mathbb{R}^n$ be a n-dimensional curve, normalized in the interval $I = [0,1]$. Then, we represent the shape of β using the *Square-root*

Velocity Function (SRVF) defined as:

$$q(t) \doteq \frac{\dot{\beta}(t)}{\sqrt{\|\dot{\beta}(t)\|}}, \tag{3.1}$$

being $\|.\|$ the $\mathbb{L}^2$ norm. The quantity $\|q(t)\|$ is the square-root of the norm of curve derivative at time t, i.e. the square-root of the instantaneous speed. The ratio $\frac{q(t)}{\|q(t)\|}$ is the instantaneous direction along the curve. Thus, from a SRV function q representing the shape of β, we can retrieve the initial curve β (up to a translation) using:

$$\beta(t) = \int_0^t \|q(s)\| q(s)\, ds\,. \tag{3.2}$$

This is particularly useful to visualize effects of shape transformation on curve in $\mathbb{R}^n$.

The SRVF was formerly introduced in [36] to enable shape analysis. As described in [36], such representation captures the shape of a curve β and presents some advantages. First, it uses a single function to represent the curve. Then, as described later, the computation of the elastic distance between two curves is reduced to a simple $\mathbb{L}^2$ norm (equivalent to inner product in the case of functions), which simplifies the implementation and the analysis. Finally, re-parametrization of the curves acts as an isometry.

We define the set of all SRVF functions as:

$$\mathcal{C} = \{q : I \to \mathbb{R}^n | \ \|q\| = 1\} \subset \mathbb{L}^2(I, \mathbb{R}^n)\,. \tag{3.3}$$

By restricting the length of β to 1, the space $\mathcal{C}$ becomes an infinite dimensional unit-sphere in the Hilbert space $\mathbb{L}^2(I, \mathbb{R}^n)$, representing the *pre-shape space* of all curves invariant to translation and uniform scaling. Each SRVF associated to a curve is viewed as an element of $\mathcal{C}$. In addition, this makes the representation invariant to the length of the trajectory. Considering the $\mathbb{L}^2$ metric on its tangent space, $\mathcal{C}$ becomes a Riemannian manifold, as demonstrated in [36]. The tangent space at a point q is given by:

$$T_q(\mathcal{C}) = \{v \in \mathbb{L}^2(I, \mathbb{R}^n) | \ \langle v, q \rangle = 0\}\,. \tag{3.4}$$

Here, $\langle v, q \rangle$ denotes the inner product in $\mathbb{L}^2(I, \mathbb{R}^n)$.

3.2.2 Elastic distance

To compare two curves, a distance between their corresponding shape on $\mathcal{C}$ can be defined as the length of the geodesic connecting them on $\mathcal{C}$. As $\mathcal{C}$ is a hyper-sphere, the geodesic length between two elements q_1 and q_2 is defined as:

$$\theta = d_{\mathcal{C}}(q_1, q_2) = cos^{-1}(\langle q_1, q_2 \rangle) \, , \tag{3.5}$$

with the corresponding geodesic path between these two elements given by:

$$\alpha(\tau) = \frac{1}{\sin(\theta)} \left(\sin((1-\tau)\theta) q_1 + \sin(\theta\tau) q_2\right) \, . \tag{3.6}$$

Such a geodesic path represents the elastic deformation of the shape q_2 to correspond to q_1. In particular, $\tau \in [0, 1]$ in Eq. (3.6) allows us to parameterize the displacement along the geodesic path α: $\tau = 0$ and $\tau = 1$ corresponding to the shapes q_1 and q_2, respectively; an intermediate value of τ corresponding, instead, to an intermediate deformed shape between q_1 and q_2. Thus, in addition to provide a distance for measuring the similarity between two shapes, such a framework also provides the sequence of intermediate shapes that represents the optimal—in terms of minimum energy—elastic deformation between the two shapes. Note that if $q_1 = q_2$, the distance between q_1 and q_2 is equal to 0, then $sin(\theta) = 0$. In that case $\alpha(\tau) = q_1 = q_2$ for each τ.

However, shape analysis usually requires invariance to different transformations, such as translation, scale, rotation and re-parametrization. Representation of a curve using the SRVF yields invariance to translation and scaling. However, the representation is not invariant to rotation and re-parametrization. Indeed, if a curve is rotated or re-parameterized, its SRVF changes although its shape is unchanged. To cope with this, we define the rotation group $SO(n)$ and the re-parametrization group Γ. Elements $O \in SO(n)$ are rotation matrices of size $n \times n$. Elements $\gamma \in \Gamma$ are re-parametrization functions. As defined in [36], rotating a curve β by $O \in SO(n)$ and re-parametrizing it with $\gamma \in \Gamma$ yields a new curve

$\beta' = O(\beta \circ \gamma)$ equivalent to β in terms of shape. The composition $\beta \circ \gamma$ denotes the re-parametrization of the curve β. It follows the same sequence of points as β but at the evolution rate governed by γ. Hence, the SRVF of $\beta' = O(\beta \circ \gamma)$ is given by $\sqrt{\dot{\gamma}(t)}O(q \circ \gamma)(t)$. Another advantage of the SRVF is that the action of the product group $SO(n) \times \Gamma$ on $\mathcal{C}$ is on isometries, as proven in [36]. This means that the distance between two SRVF is preserved after rotation and re-parametrization have been applied. Thus, We define the equivalence class of q as:

$$[q] = \{\sqrt{\dot{\gamma}(t)}O(q \circ \gamma)(t) | O \in SO(3),\ \gamma \in \Gamma\}\ , \tag{3.7}$$

where each element of $[q]$ is equivalent up to a rotation and a re-parametrization. The set of all equivalence classes is called the *shape space* denoted as $\mathcal{S}$. To compute the geodesic distance between $[q_1]$ and $[q_2]$ on $\mathcal{S}$, we first need to find the optimal rotation O^* and re-parametrization γ^* that best register the element q_2 with respect to q_1. In practice, Singular Value Decomposition is used to find the optimal rotation, and Dynamic Programming [12] is used to find the optimal re-parametrization. Let q_2^* be the element associated with O^* and γ^*, then the distance between $[q_1]$ and $[q_2]$ is defined as:

$$d_{\mathcal{S}}([q_1], [q_2]) = d_{\mathcal{C}}(q_1, q_2^*), \tag{3.8}$$

and the corresponding geodesic between $[q_1]$ and $[q_2]$ becomes:

$$\alpha(\tau) = \frac{1}{\sin(\theta)}\left(\sin((1-\tau)\theta)q_1 + \sin(\theta\tau)q_2^*\right)\ , \tag{3.9}$$

where θ is now $d_{\mathcal{S}}([q_1], [q_2])$.

In this way, the distance between the shape of two curves in $\mathbb{R}^n$ is invariant to their translation, scale, rotation and re-parametrization. It should be noticed that, as explained in the following Sections, not all these invariances are necessary and some of them are handled differently in the context of shape analysis of human poses and human motion. Figure 3.1 illustrates the shape space $\mathcal{S}$ where the distance $d_{\mathcal{S}}([q_1], [q_2])$ is computed between two shapes q_1 and q_2^*.

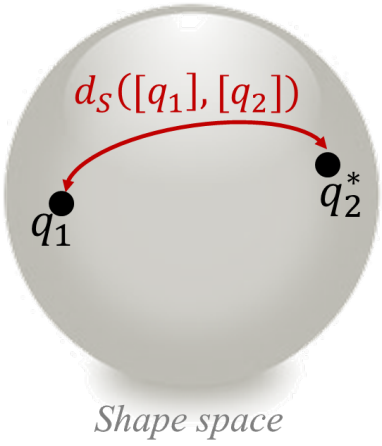


Figure 3.1 – *Schema of the shape space $\mathcal{S}$ where two shapes q_1 and q_2^* are represented. The distance $d_{\mathcal{S}}([q_1],[q_2])$ corresponds to the length of the geodesic path connecting the two shapes.*

3.2.3 Tangent space

One interesting property of such Riemannian manifold is that such space is locally similar to a linear space. One way to benefit from this property is to consider tangent spaces as defined in equation 3.4. Such tangent space is a linear vector space where more conventional statistics applies. Hence, as explained below, considering tangent space of the shape space facilitates the calculus of statistics on shapes providing deeper analysis of shapes.

So as to move from the manifold to the tangent space and vice versa, it exists interesting tools called logarithmic map and exponential map operators.

Let q_1 and $q_2^* \in \mathcal{S}$ being two shapes belonging to the shape space with q_2^* representing the optimal shape (associated with optimal orientation and re-parametrization) with respect to q_1. The logarithmic map also called inverse exponential map, $exp_{q_1}^{-1} : \mathcal{S} \rightarrow T_{q_1}(\mathcal{S})$, allows to project q_2^* into the tangent space of q_1 denoted $T_{q_1}(\mathcal{S})$. It results in a tangent vector $v_2^* \in T_{q_1}(\mathcal{S})$ called velocity vector. The computation of such velocity vector using the inverse exponential map is defined as:

$$v_2^* = exp_{q_1}^{-1}(q_2^*) = \frac{\theta}{sin\theta}(q_2^* - cos(\theta)q_1) \ , \tag{3.10}$$

where $\theta = d_{\mathcal{S}}([q_1], [q_2])$.

Such velocity vector captures the shape difference between q_1 and q_2^*. By locally analyzing $v_2^*(t)$ for each parameterized t, we obtain the local deformation needed to go from a point $q_1(t)$ of the shape q_1 to the corresponding point $q_2^*(t)$ of the shape q_2^*

Conversely, the exponential map, $exp_{q_1} : T_{q_1}(\mathcal{S}) \rightarrow \mathcal{S}$, allows to transfer a tangent vector $v_2 \in \mathcal{S}$ into the shape space $\mathcal{S}$, resulting in the shape q_2. The exponential map operator is defined as:

$$exp_{q_1}(v_2) = cos(\|v_2\|)q_1 + sin(\|v_2\|)\frac{v_2}{\|v_2\|} \, . \tag{3.11}$$

Figure 3.2 illustrates the idea of the exponential map and the inverse exponential map operators to move from the shape space to the tangent space and inversely.

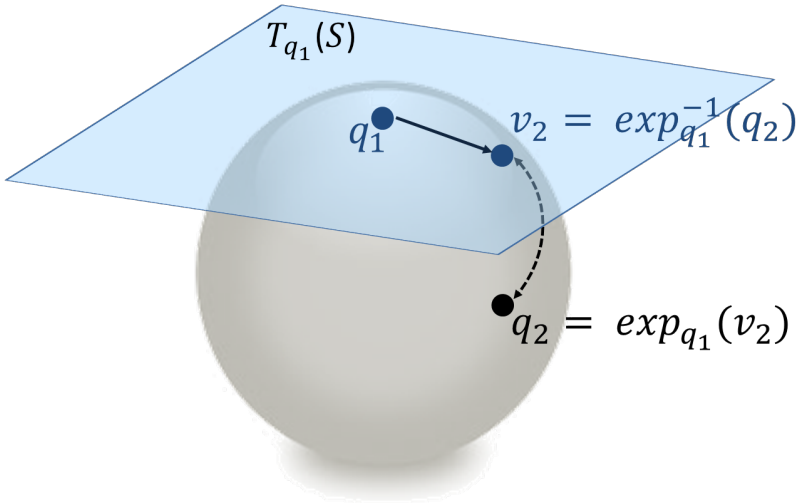


Figure 3.2 – *Illustration of the mapping between a shape $q_2 \in \mathcal{S}$ and a velocity vector $v_2 \in T_{q_1}(\mathcal{S})$ using the exponential map and inverse exponential map operators.*

3.3 Statistics on the shape space

So as to provide a deep shape analysis, it is often required to compute some statistics. Fortunately, such Riemannian manifold offers methods to compute statistical information, like the computation of mean shapes, standard deviation among a set of shapes or the learning of distribution of shapes.

In the following, we show how we compute these tools using different notion defined previously, like the elastic distance (Equation 3.8) and the inverse exponential map operator (Equation 3.10).

3.3.1 Mean shape

In any data analysis, the computation of a mean is very important as it allows to represent a set of data by a single representative template. In Riemannian geometry, one way to compute the geometric mean of a set of data sufficiently close to each other is to minimize a cost function computed from the data.

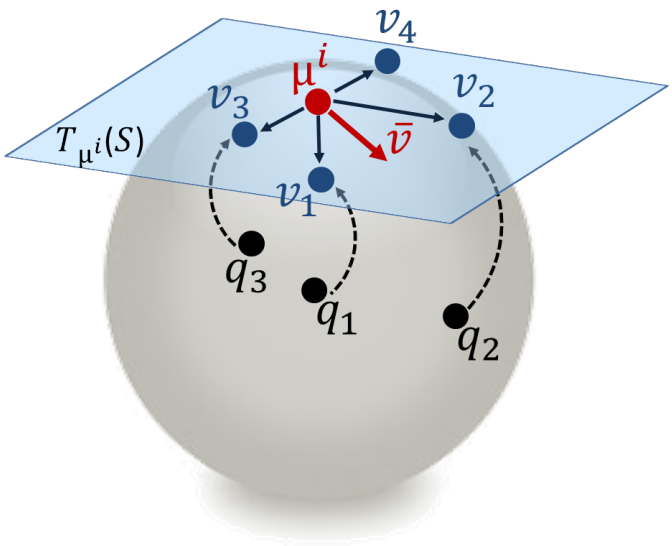
A common algorithm for such mean computation on Riemannian manifold is called the Riemannian center of mass [40] and employs as cost function the sum of squared geodesic distances between a given data and all other data. Here we propose to use this algorithm to identify a mean shape. For a given set of shapes $q_1, \ldots, q_n \in \mathcal{S}$, their Riemannian center of mass can be defined as:

$$\mu = \underset{[q]}{\operatorname{argmin}} \sum_{i=1}^{n} d_s([q], [q_i])^2 \,. \tag{3.12}$$

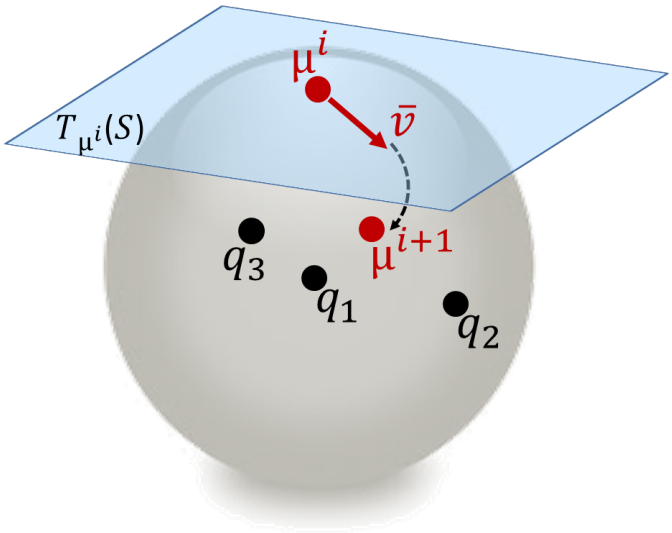
So as to minimize such cost function, the algorithm employs both the exponential map and logarithm map operators in an iterative process to update the Riemannian center of mass until convergence. More specifically, at each iteration i, shapes are first projected into the tangent space at the current mean shape μ_i using the inverse exponential map.

Based on the resulted velocity vectors, the average direction is computed and the mean shape is slightly moved in that direction. The exponential map is finally used to transfer the updated mean shape μ_{i+1} back on the shape space.

This process is summarized in Algorithm 1 and illustrated in Figure 3.3.



(a)



(b)

Figure 3.3 – *Illustration of the Riemannian center of mass computation. At each iteration i, two steps are performed: (a) Each shape q_i is projected to the tangent space of μ^i and the average direction $\bar{v}$ is computed from the set $\{v_i\}$. (b) The mean shape μ^i is moved along $\bar{v}$ by ϵ and mapped back to $\mathcal{S}$ to obtain the updated mean shape μ^{i+1}.*

Algorithm 1: Computation of Riemannian center of mass on $\mathcal{S}$

Require: Set of shapes $\{q_1, q_2, \ldots, q_N\}$; $\epsilon = 0.5$; Threshold τ for stopping criterion
Ensure: The mean shape μ_i

Select an initial mean shape μ_1 among the set of shapes
repeat

for $n = 1$ to N **do**
Compute $v_n = exp_{\mu_i}^{-1}(q_n)$
end for
Compute the average direction $\bar{v} = \frac{1}{n}\sum_{n=1}^{N} v_n$
Update μ_{i+1} by moving μ_i in the direction of $\bar{v}$ by ϵ : $\mu_{i+1} = exp_{\mu_i}(\epsilon\bar{v})$
$i = i + 1$
until $\|\bar{v}\| < \epsilon$

3.3.2 Standard deviation on the shape space

Once the mean shape is computed for a set of shapes, one can use the provided elastic distance (Equation 3.8) to compute the standard deviation so as to quantify the variation among shapes with respect to the mean. This allows evaluating the overall similarity of shapes within the set. For a given set of shapes $q_1, \ldots, q_n \in \mathcal{S}$ with corresponding mean μ, the standard deviation between this mean shape and all the shapes within the set is defined as:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n} d_{\mathcal{S}}([\mu], [q_i])^2}\ . \tag{3.13}$$

3.3.3 K-means on the shape space

In addition, we propose to also use the notion of mean shape in order to perform unsupervised learning such as clustering of shapes. Based on the provided tools described above, we propose to adapt the k-means algorithm to estimate k clusters of shapes represented by their corresponding Riemannian center of mass.

Similarly to standard k-means algorithm, the clustering problem is solved using an Expectation-Maximization (EM) approach. At the initial step, we randomly select k shapes as the cluster representatives μ_k. In the E-step, each shape q_i is assigned to a cluster based on nearest rule applied on elastic distances computed between the shape q_i and all representative shapes μ_k. Then in the M-step, the representative shapes μ_k are updated by applying the Riemannian center of mass algorithm on shapes

belonging to the cluster k, as described in Algorithm 1. These two steps are repeated until convergence.

This intrinsic k-means clustering algorithm on the shape space is summarized in Algorithm 2.

Algorithm 2: K-means clustering algorithm on $\mathcal{S}$

Require: Set of shapes $\{q_1, q_2, \ldots, q_N\}$; Number of clusters k; Maximum number of iterationsI_{max}; Threshold τ for stopping criterion
Ensure: The set of k cluster representatives$\{\mu_1, \mu_2, \ldots, \mu_k\}$

Initialize cluster representatives randomly $(\mu_1^0, \mu_2^0, \ldots, \mu_k^0)$
while $i < I_{max}$&$c > \tau$ **do**
  Compute distances from each q_n to all μ_k^i using $d_{\mathcal{S}}([\mu_k^i], [q_n])$
  Assign each shape q_n to the nearest cluster representative μ_k^i
  Update new cluster representatives $(\mu_1^{i+1}, \mu_2^{i+1}, \ldots, \mu_k^{i+1})$ using Algorithm 1
  compute the amount of change as $c = \sum_{j=1}^{k} d_{\mathcal{S}}([\mu_k^i], [\mu_k^{i+1}])$
  $i = i + 1$
end while

3.3.4 Learning distribution on the shape space

Finally, so as to expand the shape analysis, we propose to analyze the distribution within a set of shapes by learning a density function. In addition to a mean shape previously computed, these density functions also capture the variability between shapes and provide a deeper modeling of such set of shapes.

In so doing, we assume the distribution of shapes within a cluster follows a multivariate normal model. Unfortunately, learning such density functions on the shape space is not straightforward, mainly due to the non-linearity and infinite-dimensionality of such manifold. Despite of this, different methods have been proposed to deal with these two challenges [81, 6].

As explained above, a common way to circumvent the non-linearity of the manifold is to consider the tangent space to the manifold at the mean shape which is a linear vector space where conventional statistics applies. We denote $T_\mu(\mathcal{S})$ the tangent space at the mean shape. For each shape $q_i \in \mathcal{S}$, we compute its corresponding velocity vector $v_i \in T_\mu(\mathcal{S})$ using the inverse exponential map.

Then, so as to deal with the problem of infinite-dimensionality, we as-

sume the variations in tangent vectors are restricted to a m-dimensional subspace. Using tangent vectors $v_i \in T_\mu(\mathcal{S})$, we can apply Principal Component Analysis (PCA) to identify eigenvectors denoted w_i. We then select m principal eigenvectors to build the principal subspace $\mathcal{B} = \{w_1, w_2, \ldots, w_m\}$. Tangent vectors v_i are then projected into the learned subspace $\mathcal{B}$ resulting in $\tilde{v}_i$, and the covariance matrix $\Sigma \in \mathbb{R}^{m \times m}$ is computed. For a set of N projected vectors $\tilde{v}_i$, the covariance matrix Σ is defined as:

$$\Sigma = \sum_{i=1}^{N} \tilde{v}_i \tilde{v}_i^T \ . \tag{3.14}$$

Finally, we use the resulting covariance matrix Σ to learn a multivariate normal distribution of shapes around the mean shape μ. Its corresponding probability density function is defined as:

$$f(\tilde{v}) = \frac{1}{(2\pi)^{m/2}\ |\Sigma|^{1/2}} e^{-\frac{1}{2}\tilde{v}^T \Sigma^{-1} \tilde{v}} \ . \tag{3.15}$$

The process of learning the distribution on the *shape space* is illustrated in Figure 3.4 and summarized in Algorithm 3.

Algorithm 3: Learning distribution on the *shape space*

Require: Set of shapes $\{q_i\}$; N, the number of shapes
Ensure: The covariance matrix Σ used to compute the density probability function

Compute the mean shape μ from $\{q_i\}$ using Eq. (3.12)
for $i = 1$ to N **do**
  Compute the tangent vector $v_i \in T_\mu$ using Eq. 3.10
end for
Apply PCA on the set of tangent vectors $\{v_i\}$ to learn a principal subspace $\mathcal{B}$
for $i = 1$ to N **do**
  Compute the projected vector $\tilde{v}_i$ into the subspace
end for
Compute the covariance matrix Σ from the set $\{\tilde{v}_i\}$, using 3.14

3.4 Conclusion

In this Chapter, we have introduced the shape analysis framework that we employ for shape analysis of human pose and motion. Within this framework, the shape of a curve is captured using a single representation called the square-root velocity function and interpreted in a Riemannian

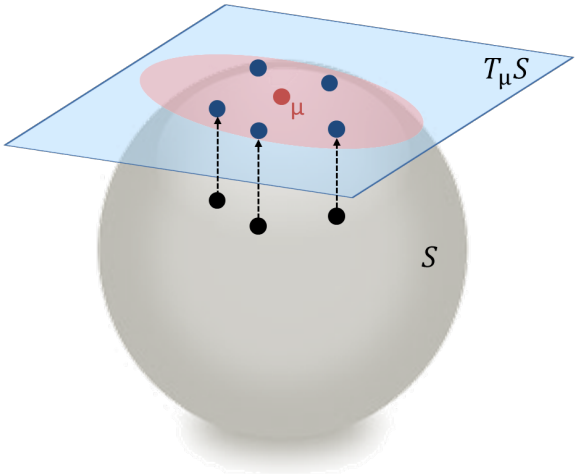


Figure 3.4 – *Learning distribution of shapes belonging to the same cluster on the shape space $\mathcal{S}$. For each cluster, the mean shape μ is computed (red) from shapes q_i belonging to the same cluster. Then, the shapes q_i (black) are projected on the corresponding tangent space $T_\mu\mathcal{S}$. Such tangent vectors v_i (blue) are used to compute the covariance matrix and learn the multivariate normal distribution (red ellipse) for each cluster.*

manifold called shape space. In order to compare shapes on such space, we exploit an elastic distance representing similarity between shapes independently to their size, location, orientation and elasticity. In addition, such Riemannian framework offers a variety of statistical tools. This is particularly useful to carry out an accurate shape analysis. As demonstrated in the following, this established shape analysis framework is the core of our study on human behavior understanding.

4 Action Recognition by Shape Analysis of Motion Trajectories

Sommaire

4.1 Introduction

Recognizing human actions in video sequences is an important open problem that is currently at the heart of many research domains, including surveillance, natural interfaces and rehabilitation. The recent release of depth sensors, like Microsoft Kinect has allowed a rapid advancement in these domains. Indeed, the depth information provided in addition to RGB data allows facing important challenges in 2D videos, like the invariance to illumination changes. Moreover, depth data facilitate the segmentation of objects in the scene (like humans) as well as the background subtraction. However, the design and development of models for action recognition that are both accurate and efficient is a challenging task due to the variability of the human pose, clothing and appearance.

In the state-of-the-art, different methods have been proposed for the problem of human action recognition using depth data, thus benefiting of advantages enunciated before. These methods rely on the extraction of meaningful descriptors from the entire set of points of depth images in order to describe the human pose, like 3D human silhouettes [46] or Histogram of Oriented Gradient (HOG) features computed on the depth image [62].

Another advantage of using depth data is that it becomes easier to obtain meaningful information about the human pose. Indeed, thanks to the work of Shotton et al. [79], the 3D positions of body joints can be easily and accurately extracted in real-time from individual depth images, without using any temporal information. This set of 3D joints forming a humanoid provides useful information about the subject pose. As a result, several methods have been proposed to compute representative features based on these 3D joints, like pairwise distances [102, 94] or histogram of joints position [99].

Nevertheless, a human action is naturally characterized by the evolution of the pose of the human body over the time. Hence, only describing the human pose may not be sufficient and a relevant spatio-temporal representation of the dynamic that defines the human action is necessary. So as to model dynamic of actions, various approaches have been investi-

gated, like discrete Hidden Markov Models [99], Depth Motion Maps [104] highlighting spatio-temporal areas where motion happens. Other methods consider the video sequence as a 4D space [91, 93, 98].

Additionally, some emerging and interesting techniques reformulate computer vision problems, like action recognition, over non-linear spaces [88, 33, 51], so as to better characterize and analyze the non-linearity of the problem. Their effectiveness in 2D videos motivated us to investigate such non-linear spaces, like Riemannian manifolds, for action recognition from 3D videos.

4.1.1 Constraints

Even if depth data significantly reduce certain difficulties affecting action recognition performance in 2D videos, some important challenges directly related to the human action recognition problem remain.

First, an effective action recognition approach should be robust to any geometric variability among the subjects. Particularly, the various possible size of subjects performing action should not affect the effectiveness of the method. In addition, in a real-world and non cooperative context like surveillance, actions may not specifically be performed in the center of the scene and in front of the sensor. Hence, the method should be able to recognize human action independently to the position and orientation of the subject.

Second, actions can be performed at different speed among the subject. While such speed or acceleration information may be meaningful to a deeper analysis of human motion like intention recognition, it should not be considered for the case of action recognition. Indeed, actions are mainly characterized by a certain motion of the human body and not related to any speed of execution. Hence, an approach robust to the speed of execution of actions is necessary.

These constraints can be handled in two distinct ways. The first way is to propose a robust representation of actions so that a same representation is employed to describe the same actions with all possible variations. Conversely, the second manner employs an invariant metric that compare

two actions independently to the possible variations stated before. As explained later, in order to face these challenges, we use a different manner according to the constraints.

However, what further complicates the task of human action recognition is the high variability human motion. Indeed, if we ask to a subject to perform several time the same action, he will not perform it exactly in the same way as previous attempts. Such variability may be increased when analyzing the same action performed by different subjects. Hence an efficient action recognition method need to be robust to such motion variations.

4.1.2 Overview of our approach

In this Chapter, we propose an original approach to extract a compact representation of a human action captured through a depth sensor, and enable accurate action recognition .

Our proposed method is a skeleton-based approach since we only use skeleton features to represent the human pose. Indeed, we consider that skeleton data containing the 3D positions of different parts of the body are sufficient to provide an accurate representation of the human pose. In addition, these skeleton features are directly provided by depth sensors and also provide local information about the human body. This makes it possible to analyze only some parts of the human body instead of the global pose. However, even if accurate 3D joint positions are available, the action recognition task is still difficult due to significant spatial and temporal variations in the way of performing an action.

In order to enable our method to be invariant to geometric transformation of the subject and thus to face one of the main constraints stated before, we propose a translation and rotation invariant representation of the skeleton sequences.

So as to capture the dynamics of human motion characterizing the action, we model the sequence of frame features as trajectory representing the evolution of the human body along the time. To this end, the full skeleton is modeled as a multi-dimensional vector obtained by concate-

nating the three-dimensional coordinates of its joints. Then, the trajectory described by this vector in the multi-dimensional space is regarded as a signature of the temporal dynamics of the movements of all the joints.

These trajectories are then interpreted in a Riemannian manifold, so as to model and compare their shapes using elastic registration and matching in the shape space. In so doing, we recast the action recognition problem as a statistical analysis on the shape space manifold. Furthermore, by using an elastic metric to compare the similarity between trajectories, robustness of action recognition to the execution speed of the action is improved. Figure 4.1 summarizes the proposed approach.

4.1.3 Motivation

The main considerations that motivated our solution are:

- The fact that many feature descriptors typically adopted in computer vision applications lie on curved spaces due to the geometric nature of the problems;
- The shape and dynamic cues are very important for modeling human action, and their effectiveness have been demonstrated in several state-of-the-art works [51, 78, 87, 1];
- Using such manifold offers a wide variety of statistical and modeling tools that can be used to improve the accuracy of action recognition.

4.2 Shape analysis of motion trajectories

In this section, we first describe our representation of action sequences using skeleton data provided by depth sensors. Then, we present how we analyze these sequences so as to guarantee future robust action recognition.

4.2.1 Spatio-temporal representation of human motion

Using RGB-D cameras, such as the Microsoft Kinect, a 3D humanoid skeleton can be extracted from depth images in real-time by following the ap-

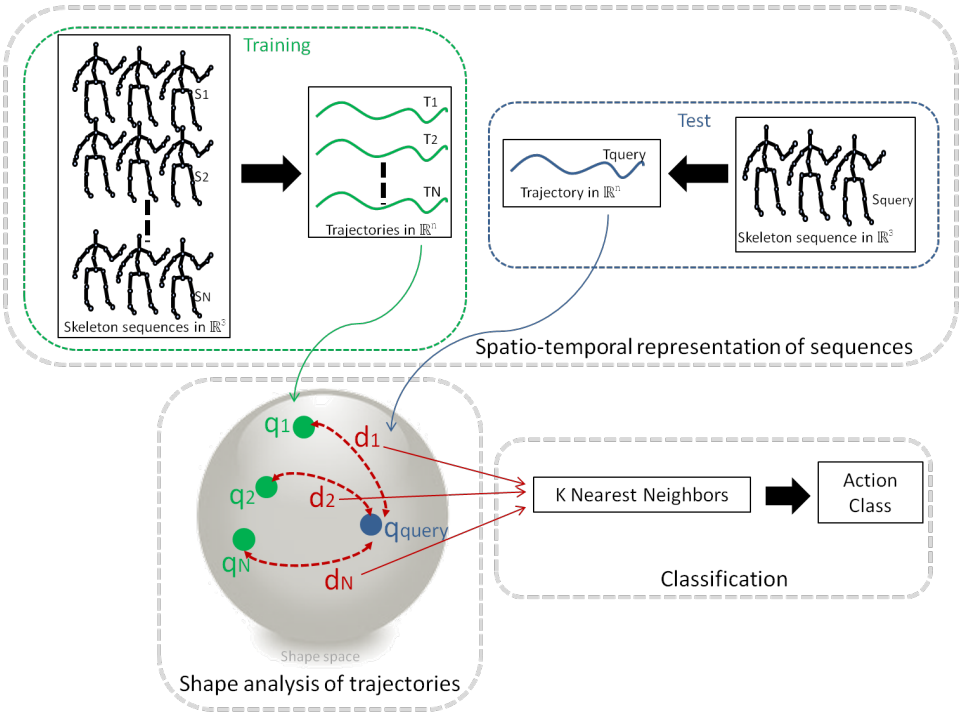


Figure 4.1 – *Overview of our approach: First, skeleton sequences are represented as trajectories in a n-dimensional space; These trajectories are then interpreted in a Riemannian manifold (shape space); Recognition is finally performed using kNN classification on this manifold.*

proach of Shotton et al. [79]. This skeleton contains the 3D position of a certain number of joints representing different parts of the human body. The number of estimated joints depends on the SDK used in combination with the device. Skeletons extracted with the Microsoft Kinect SDK contain 20 joints, while 15 joints are estimated with the PrimeSense NiTE.

For each frame t of a sequence, the real-world 3D position of each joint i of the skeleton is represented by three coordinates expressed in the camera reference system $p_i(t) = (x_i(t), y_i(t), z_i(t))$. Let N_j be the number of joints the skeleton is composed of, the posture of the skeleton at frame t is represented by a $3N_j$ dimensional tuple:

$$v(t) = [x_1(t)\ y_1(t)\ z_1(t)\ \ldots\ x_{N_j}(t)\ y_{N_j}(t)\ z_{N_j}(t)]^T\ . \tag{4.1}$$

For an action sequence composed of N_f frames, N_f feature vectors are extracted and arranged in columns to build a feature matrix M describing

the whole sequence:

$$M = \begin{pmatrix} v(1) & v(2) & \dots & v(N_f) \end{pmatrix} . \tag{4.2}$$

This feature matrix represents the evolution of the skeleton pose over the time. Each column vector v is regarded as a sample of a continuous trajectory in $\mathbb{R}^{3N_j}$ representing the action in a $3N_j$ dimensional space called *action space*. The size of such feature matrix is $3N_j \times N_f$.

Figure 4.2 illustrates the representation of a human action sequence as a spatio-temporal trajectory of skeleton coordinates in a $3N_j$-dimensional space.

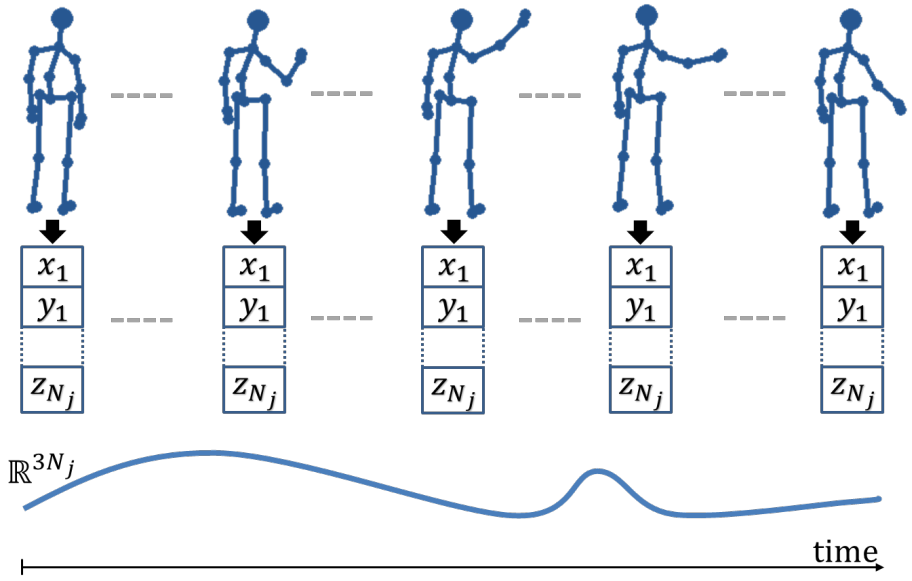


Figure 4.2 – *Spatio-temporal representation of human action. First row: Skeletal representation of the action along the time; Second row: Each skeleton is represented as a vector built from the three coordinates of the N_j skeleton joints; Third row: The concatenation of such vectors along the time results in a spatio-temporal trajectory in $\mathbb{R}^{3N_j}$, representing the human motion during the sequence.*

To reduce the effect of noise that may affect the coordinates of skeleton joints, a smoothing filter is applied to each sequence. This filter weights the coordinates of each joint with the coordinates of the same joint in the neighboring frames. In particular, the amount of smoothing is controlled by a parameter σ that defines the size $W_s = 1 + 2 \times \sigma$ of a temporal window centered at the current frame. For each joint $i = 1, \dots, N_j$ at frame

$t = 1 + \sigma, \ldots, N_f - \sigma$ the new x coordinate is:

$$x_i(t) = \frac{1}{W_s} \sum_{\tau=t-\sigma}^{t+\sigma} x_i(\tau) \ . \tag{4.3}$$

The same applies to y and z. The value of σ is selected by performing experiments on a set of training sequences. The best accuracy is obtained for $\sigma = 1$, corresponding to a window size of 3 frames. We note that such smoothing of motion trajectories also allows to decrease the variability among several performances of a same action by a same subject.

4.2.2 Invariance to geometric transformation

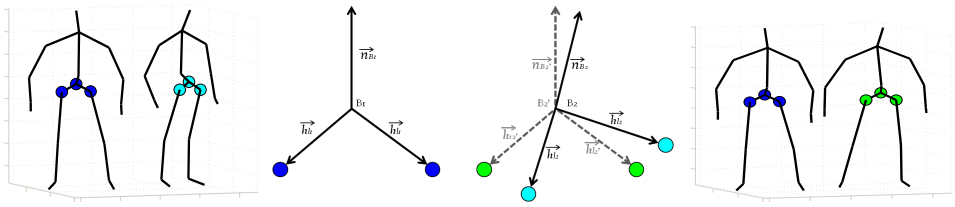
A key feature of action recognition systems is the invariance to the translation and rotation of the subject in the scene. Two instances of the same action differing only by the position and orientation of the person with respect to the acquisition device should be recognized as belonging to the same action class. This goal can be achieved either by adopting a translation and rotation invariant representation of the action sequence or providing a suitable distance measure that copes with translation and rotation variations. We adopt the first approach by normalizing the position and the orientation of the subject in the scene before the extraction of the joint coordinates.

For this purpose, we first define the spine joint of the initial skeleton as the center of the skeleton (*root joint*). Then, a new base B is defined with origin in the root joint: it includes the left-hip joint vector $\overrightarrow{h_l}$, the right-hip joint vector $\overrightarrow{h_r}$, and their cross product $\overrightarrow{n_B} = \overrightarrow{h_l} \times \overrightarrow{h_r}$.

This new base is then translated and rotated, so as to be aligned with a reference base B_0 computed from a reference skeleton (selected as the neutral pose of the sequence). The calculation of the optimal rotation between the two bases B and B_0 is performed using *Singular Value Decomposition* (SVD). For each sequence, once the translation and the rotation of the first skeleton is computed with respect to the reference skeleton, we apply the same transformations to all other skeletons of the sequence.

This makes the representation invariant to the position and orientation

of the subject in the scene. Figure 4.3a shows an example of two different skeletons to be aligned. The bases B_1 and B_2 computed for the two skeletons are shown in Figure 4.3b, where the rotation required to align B_2 to B_1 is also reported. In Figure 4.3c, the two aligned skeletons are shown.



(a) (b) (c)

Figure 4.3 – *Invariance to geometric transformations: (a) Two skeletons with different orientations. The skeleton on the left is the reference one. The skeleton on the right is the first skeleton of the sequence that should be aligned to the reference skeleton; (b) Bases B_0 and B are built from the two corresponding hip vectors and their cross product. The base B' corresponds to B aligned with respect to B_0; (c) The resulting skeleton (right) is now aligned with respect to the first one (left). The transformations computed between these two bases are applied to all skeletons of the sequence.*

4.2.3 Body part representation

In addition to enable the representation of the action using the whole body, the proposed solution also supports the representation of individual body parts, such as the legs and the arms.

There are several motivations for focusing on parts of the body. First of all, many actions involve motion of only some parts of the body. For example, when subjects answer a phone call, they only use one of their arms. In this case, analyzing the dynamics of the arm rather than the dynamics of the entire body is expected to be less sensitive to the noise originated by the involuntary motion of the parts of the body indirectly involved in the action.

Furthermore, during the actions some parts of the body can be out of the field of view of the camera or occluded by objects or other parts of the body. This can make the estimation of the coordinates of some joints inaccurate, compromising the accuracy of action recognition.

Finally, due the symmetry of the body along the vertical axis, one same action can be performed using one part of the body or another. With reference to the action "answer phone call", the subject can use his left arm

or right arm. By analyzing the whole body, we can not detect such variations. Differently, using body parts separately, simplifies the detection of this kind of symmetrical actions.

To analyze each part of the body separately, we represent a skeleton sequence by four feature sets corresponding to the body parts. Each body part is associated with a feature set composed by the 3D normalized position of the joints that are included in that part of the body. Let N_{j_p} be the number of joints of a body part, the skeleton sequence is now represented by four trajectories in $3 \times N_{j_p}$ dimensions instead of one trajectory in $3 \times N_j$ dimensions. The actual number of joints per body part can change from a dataset to another according to the SDK used for estimating the body skeleton.

In all the cases, $N_{j_p} < N_j$ and the body parts are disjoint (i.e., they do not share any joint). Figure 4.4 illustrates the representation of separated body parts where $N_{j_p} = 4$ for each body part representing the four limbs. As a result, the sequence is represented by four trajectories in $\mathbb{R}^{12}$.

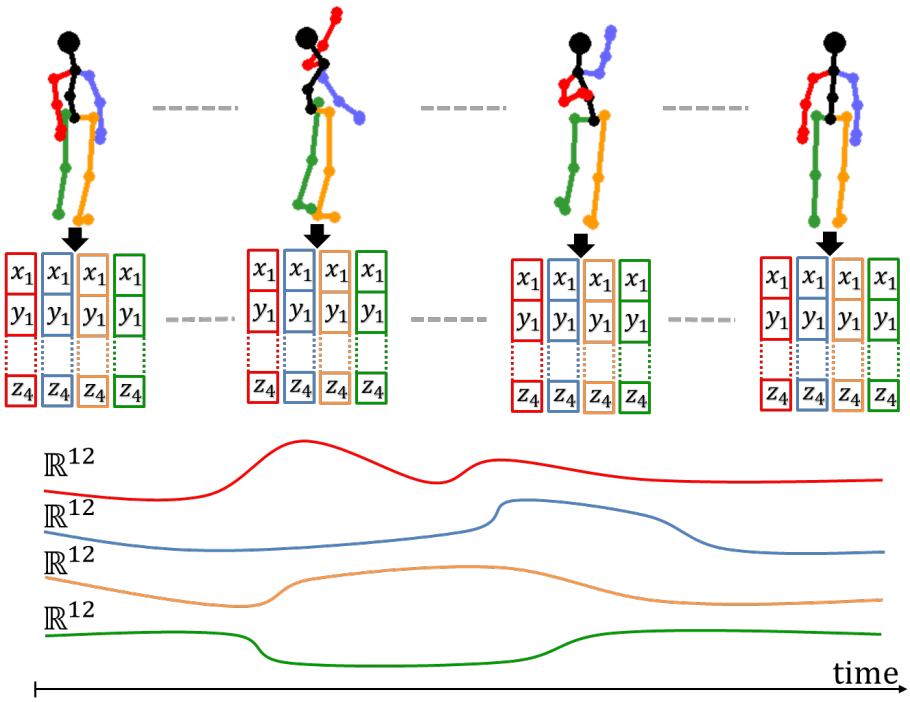


Figure 4.4 – *Spatio-temporal representation of each body part separately. First row: Skeletal representation of the action along the time where each limb is displayed in a different color; Second row: Each limb of each skeleton is represented as a vector built from the three coordinates of the corresponding joints; Third row: The concatenation of such vectors along the time results in four spatio-temporal trajectories in $\mathbb{R}^{12}$, representing the human motion of each body parts separately.*

4.2.4 Trajectory shape representation

An action is a sequence of poses which can be regarded as the result of sampling a continuous trajectory in the $3N_j$-dimensional action space. The trajectory is defined by the motion over the time of the feature point encoding the 3D coordinates of all the joints of the skeleton (or by all the feature points coding the body parts separately). According to this, two instances of the same action are associated with two trajectories with similar shape in the action space.

Hence, action recognition can be regarded and formulated as a shape matching task. Figure 4.5 provides a simplified example of action matching by shape comparison. The plot displays five trajectories corresponding to the coordinates of the left hand joint in five different actions. Three trajectories correspond to three instances of the action *drawing circle*. The remaining two trajectories correspond to the actions *side boxing* and *side kick*.

This simplified case, in which each trajectory encodes the coordinates of just one joint, makes it clear that similar actions yield trajectories with similar shapes in the action space.

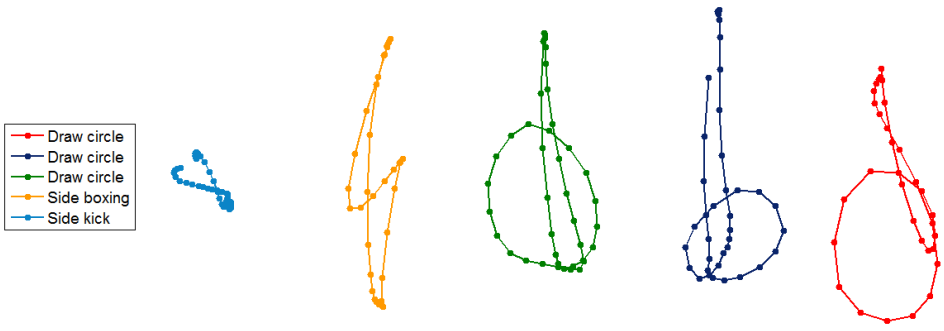


Figure 4.5 – *Trajectories representing the coordinates of the left arm joint for five actions: From left to right, side kick, side boxing, and draw circle (three different instances). Points displayed in bold represent the sample frames along the trajectories.*

Figure 4.5 also highlights some critical aspects of representing actions by trajectories. Assuming the actions are sampled at the same frame rate, performing the same action at two different speeds yields two curves with a different number of samples. This is the case of the red and blue trajectories in Figure 4.5, where samples are highlighted by bold points along the trajectories. Furthermore, since the first and the last poses of an action

are not known in advance and may differ even for two instances of the same action, the measure of shape similarity should not be biased by the position of the first and last points of the trajectory.

4.2.5 Trajectory shape analysis

Once the actions executed during the sequences are represented as spatio-temporal trajectories in $\mathbb{R}^n$, we propose to perform shape analysis of such trajectories in order to compare them and perform action recognition. In so doing, the shape of a trajectory is captured using the square-root velocity function (Equation 3.1 defined in Chapter 3). Then, we employ the elastic metric provided by the Riemannian shape analysis framework so as to compute similarity between shapes.

As demonstrated in Chapter 3, such shape analysis of trajectories is carried out independently to geometric transformation and re-parameterization of trajectories. Thus, it allows us to face the main challenges previously stated for the task of action recognition.

Nonetheless, as explained in Section 4.2.2, the spatio-temporal representation of sequences that we propose is invariant to the subject position in the scene as well as its orientation with respect to the camera. Hence, a shape analysis invariant to the rotation is not necessary. More particularly, we do not need to find the optimal rotation between two shapes before the computation of the distance.

As a result, by representing action sequences by spatio-temporal trajectories and by using an elastic distance to compare shapes of trajectories, we propose to address the issue of human action recognition as a problem of shape analysis.

4.3 Action recognition

Once a metric to compare two action sequences is defined, we propose to employ it for the task of human action recognition. The proposed action recognition approach is based on the K-Nearest Neighbors (kNN) algorithm applied both to full-body and separate body parts.

4.3.1 KNN classification

Let $\{(X_i,y_i)\}$, $i = 1,\dots,N$, be the training set with respect to the class labels, where X_i belongs to a Riemannian manifold $\mathcal{S}$, and y_i is the class label taking values in $\{1,\dots,N_c\}$, with N_c the number of classes. The objective is to find a function $F(X) : \mathcal{S} \longmapsto \{1,\dots,N_c\}$ for clustering data lying in different submanifolds of a Riemannian space, based on the training set of labeled items of the data.

To this end, we propose a kNN classifier on the Riemannian manifold, learned by the points on the open curve shape space representing trajectories. Such learning method exploits geometric properties of the shape space, particularly its Riemannian metric. This relies on the computation of the geodesic distances to the nearest neighbors of each data point of the training set.

The action recognition problem is reduced to nearest neighbor classifier in the Riemannian space. More precisely, given a set of training trajectories $X_i : i = 1,\dots,N$, they are represented by the underlying points $q_i : i = 1,\dots,N$, which map trajectories on the shape space manifold (see Figure 4.1). Then, any new trajectory X_n is represented by its SRVF q_n. Finally, a geodesic-based classifier is used to find the K-closest trajectories to q_n using the elastic metric given by Equation (3.5).

4.3.2 Average trajectories

An important advantage of using such Riemannian approach is that it provides tools for the computation of statistics of the trajectories. For example, we can use the notion of Riemannian Center of Mass [40] to compute an average trajectory from several trajectories. The average trajectory among a set of different trajectories can be computed to represent the intermediate one, or between similar trajectories obtained from several subjects to represent a template, which can be viewed as a good representative of a set of trajectories.

To classify an action trajectory, represented as a point on the manifold, we need to compute the total warping geodesic distances to all points from training data. For a large number of training data this can be associated

to a high computational cost. This can be reduced by using the notion of "mean" of class action, and computing the mean of a set of points on the manifold.

As a result, for each action class we obtain an average trajectory, which is representative of all the actions within the class. According to this, the mean can be used to perform action classification by comparing the new action with all the cluster means using the elastic metric defined in equation (3.5). For a given set of training trajectories $q_1, \ldots, q_n$ on the shape space, their Riemannian center of mass is obtained using Algorithm 1.

As an example, Figure 4.6 shows the skeleton representation of three actions sequences as well as the resulting average trajectory in the action space. From shapes in the shape space, the corresponding trajectories can be retrieved using Equation 3.2. Then, as trajectories are built from joint coordinates, we can easily obtain the entire skeleton sequence corresponding to a trajectory. Figure 4.6 shows ten skeletons for each sequence.

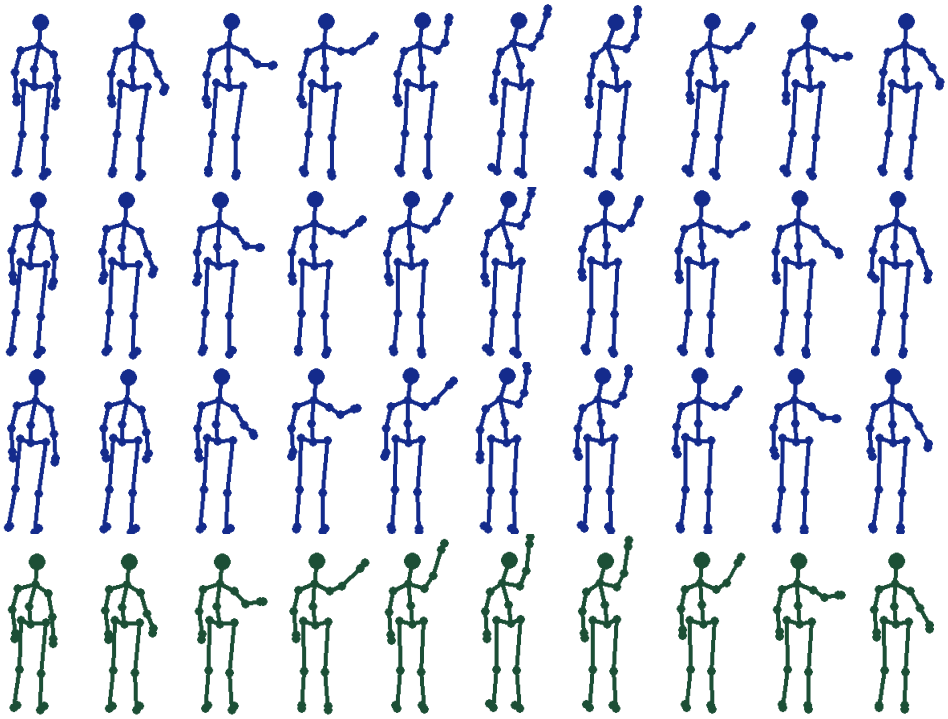


Figure 4.6 – *Computation of the average trajectory between three action trajectories. Among shapes on the shape space, the corresponding mean shape is obtained by applying the Riemannian center of mass algorithm. The corresponding skeleton sequences in the action space are displayed. The fourth row correspond to the average trajectory of the three first sequences.*

By computing such average trajectories for each action class, we im-

plicitly assume that there is only one way to perform each action. Unfortunately, this is not the case. In fact, two different subjects can perform the same action in two different ways. This variability in performing actions between different subjects can affect the computation of average trajectories and the resulting templates may not be good representatives of the action classes.

For this reason, we compute average trajectories for each subject, separately. Instead of having only one representative trajectory per action, we obtain one template per subject per action. In this way, we keep separately each different way of performing the action and the resulted average trajectories are not any more affected by such possible variations. As a drawback, with this solution the number of template trajectories in the training set increases. Let N_c be the number of classes and N_{Str} the number of subjects in the training set, the number of training trajectories is $N_c \times N_{Str}$. However, as subjects perform the same action several times, the number of training trajectories is still lower than using all trajectories.

4.3.3 Body part-based classification

In the classification step, we compute distances between corresponding parts of the training sequence and the new sequence. As a result, we obtain four distances, one for each body part. The mean distance is computed to obtain a global distance representing the similarity between the training sequence and the new sequence.

We keep only the k smallest global distances and corresponding labels to take the decision and associate the most frequent label to the new sequence. Note that in the case where some labels are equally frequent, we apply a weighted decision based on the ranking of the distances. In that particular case, the selected label corresponds to the smallest distance. However, one main motivation for considering the body parts separately is to analyze the moving parts only.

To do this, we quantify the total motion of each part over the sequence. We cumulate the Euclidian distances between corresponding joints in two consecutive frames for all the frames of the sequence. The total motion of

a body part is the cumulated motion of the joints forming this part. We compute this total motion on the re-sampled sequences, so that it is not necessary to normalize it. Let $j^k : k = 1, \ldots, N_{j_p}$, be a joint of the body part, and N_f be the frame number of the sequence, then the total motion m of a body part for this sequence is given by:

$$m = \sum_{k=1}^{N_{jp}} \sum_{i=1}^{N_f - 1} d_{Euc}(j_i^k, j_{i+1}^k) \, , \tag{4.4}$$

where $d_{Euc}(j_1, j_2)$ is the Euclidian distance between the 3D joints j_1 and j_2, and N_{j_p} is the number of joints per body part (i.e., this number can change from a dataset to another according to the SDK used for the skeleton estimation).

Once the total motion for each part of the body is computed, we define a threshold m_0 to separate moving and still parts. We assume that if the total motion of a body part is below this threshold, the part is considered to be motionless during the action.

In the classification, we take into consideration a part of the body only if it is moving either in the training sequence or the test sequence. If one part of the body is motionless in both actions, this part is ignored and does not concur to compute the distance between the training and test sequences. For instance, if two actions are performed only using the two arms, the global distance between these two actions is equal to the mean of the distances corresponding to the arms only. We empirically select the threshold m_0 that best separates moving and still parts with respect to a labeled training set of ground truth sequences. To do that, we manually labeled a training set of sample sequences by assigning a motion binary value to each body part. The motion binary value is set to 1 if the body part is moving and set to 0 otherwise.

We then compute the total motion m of each body part of the training sequences and give a motion decision according to a varying threshold.

We finally select the threshold that yields the decision closest to the ground truth. In the experiments, we notice that defining two different thresholds for the upper parts and lower parts slightly improves the accuracy in some cases.

4.4 Experimental evaluation

The proposed action recognition approach is evaluated in comparison to state-of-the-art methods using three public benchmark datasets. We also propose to deeply analyze the effect of different skeletal representations as well as the robustness of the method regarding different invariances. In addition, we measure the capability of our approach to reduce the latency of recognition by evaluating the trade-off between accuracy and latency over a varying number of actions.

4.4.1 Action recognition analysis

The three benchmark datasets that we use to evaluate the accuracy of action recognition differ in the characteristics and difficulties of the included sequences. This allows an in depth investigation of the strengths and weaknesses of our solution. For each dataset, we compare our approach to state-of-the-art methods.

MSR Action 3D dataset

This public dataset was collected at Microsoft research [46] and represents a commonly used benchmark. It includes 20 actions performed by 10 persons facing the camera. Each action is performed 2 or 3 times. In total, 567 sequences are available. The different actions are *high arm wave*, *horizontal arm wave*, *hammer*, *hand catch*, *forward punch*, *high throw*, *draw X*, *draw tick*, *draw circle*, *hand clap*, *two hand wave*, *side-boxing*, *bend*, *forward kick*, *side kick*, *jogging*, *tennis swing*, *tennis serve*, *golf swing*, *pick up & throw*.

These game-oriented actions cover different variations of the motion of arms, legs, torso and their combinations. Each subject is facing the camera and positioned in the center of the scene. Subjects were also advised to use their right arm or leg when actions are performed with a single arm or leg. All the actions are performed without any interaction with objects. Two main challenges are identified: the high similarity between different group of actions and the changes of the execution speed of actions.

For each sequence, the dataset provides depth, color and skeleton information. In our case, we only use the skeleton data. As reported in [94],

10 actions are not used in the experiments because the skeletons are either missing or too erroneous. For our experiments, we use 557 sequences.

For this experiment, we test our approach with the variations mentioned in Section 4.3 related to the body parts and Riemannian center of mass. As the subjects in this dataset are always facing the camera, the normalization of subjects orientation before computing features is not necessary. The results are reported in Table 4.1.

First, it can be noted that the best accuracy is obtained using the full skeleton and the Riemannian center of mass algorithm applied per action and per subject (92.1%). In this case, we use $k = 4$ in the kNN classification process.

Table 4.1 – *MSR Action 3D. We test our approach with its different variations (full skeleton, body parts without and with motion thresholding), and classification methods (kNN only, kNN and Riemannian center of mass (Rc) per action, kNN and Riemannian center of mass per action and per subject).*

Method	Acc. (%)
Full Skeleton & kNN	88.3
Full Skeleton & kNN & Rc per action	89.0
Full Skeleton & kNN & Rc per action/subject	**92.1**
Body Parts & kNN	80.8
Body Parts & kNN & Rc per action	87.6
Body Parts & kNN & Rc per action/subject	89.7
Body parts + motion thres. & kNN	91.1
Body parts + motion thres. & kNN & Rc per action	89.7
Body parts + motion thres. & kNN & Rc per action/subject	91.8

Note that this improvement of the accuracy using the Riemannian center of mass is not expected. Indeed, the computation of average trajectories can be viewed as an indexing of available sequences and should not add information facilitating the classification task. An explanation of accuracy improvement can be given for the case of two similar action classes. In that case, a sequence belonging to a first class can be very similar to sequences belonging to a second class, and thus selected as false positive during classification. Computing average trajectories can increase the inter-class distance and thus improve the classification accuracy. For instance, the first two actions (*high arm wave* and *horizontal high arm wave*) are very similar. Using such average trajectories reduces the confusion between these two actions, thus improving the accuracy.

Second, these results also show that the analysis of body parts sepa-

rately improves the accuracy from 88.3% to 91.1%, in the case where only the *k*NN classifier is used. When the Riemannian center of mass algorithm is used in addition to *k*NN, the values of the accuracy obtained by analyzing body parts separately or analyzing the full skeleton are very similar.

Table 4.2 reports results of the comparison of our approach to some representative state-of-the-art methods. As demonstrated in [65], various protocols have been proposed in the state-of-the-art to evaluate effectiveness of methods on MSR Action 3D dataset. Hence the comparison between methods is not always fair. Here, we followed the same experimental setup as in Oreifej et al. [64] and Wang et al. [94], where the actions of five actors are used for training and the remaining actions for test. This protocol is more realistic as a same subject is not in both training and testing sets. Our approach outperforms the other methods except the one proposed in [62]. However, this approach uses both skeleton and depth information. They reported that using only skeleton features, an accuracy of 83.5% is obtained, which is lower than our approach.

Table 4.2 – *MSR Action 3D. Comparison of the proposed approach with the most relevant state-of-the-art methods.*

Method	Accuracy (%)
EigenJoints [102]	82.3
STOP [91]	84.8
DMM & HOG [104]	85.5
Random Occupancy Pattern [93]	86.5
Actionlet [94]	88.2
DCSF [98]	89.3
JAS & HOG^2 [62]	**94.8**
HON4D [64]	88.9
Ours	**92.1**

Furthermore, following a cross validation protocol, we perform the same experiments exploring all possible combinations of actions used for training and for test. For each combination, we first use only *k*NN on body parts separately. We obtain an average accuracy of 86.09% with standard deviation 2.99% ($86.09 \pm 2.99\%$). The minimum and maximum values of the accuracy are, respectively, 77.16% and 93.44%. Then, we perform the same experiments using the full skeleton and the Riemannian center of mass per action and per subject, and obtain an average accuracy

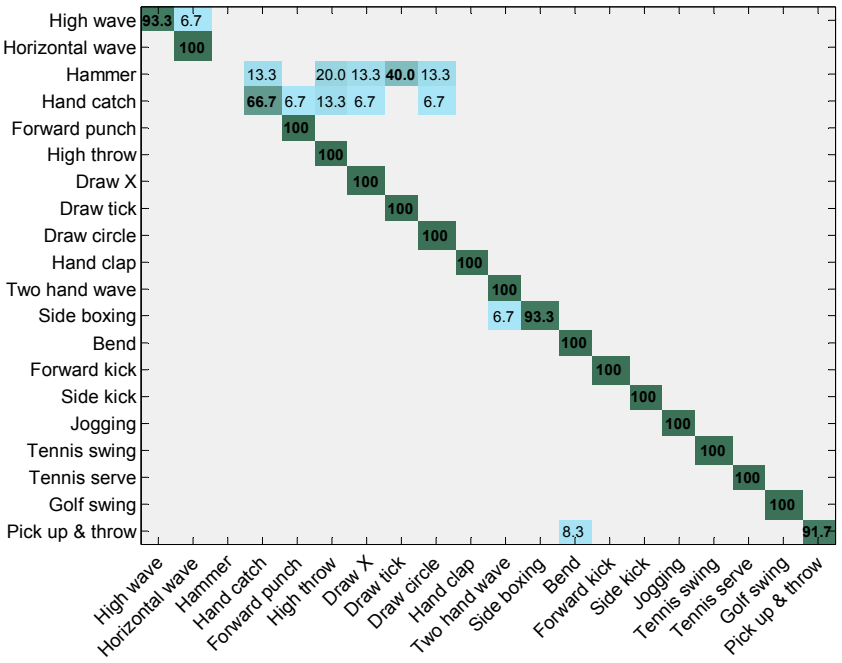
of $87.28 \pm 2.41\%$ (*mean* $\pm$ *std*). In this case, the lowest and highest accuracy are, respectively, 81.31% and 93.04%.

Compared to the work in [64], where the mean accuracy is also computed for all the possible combinations, we outperform their result ($82.15 \pm 4.18\%$). In addition, the small value of the standard deviation in our experiments shows that our method has a low dependency on the training data.

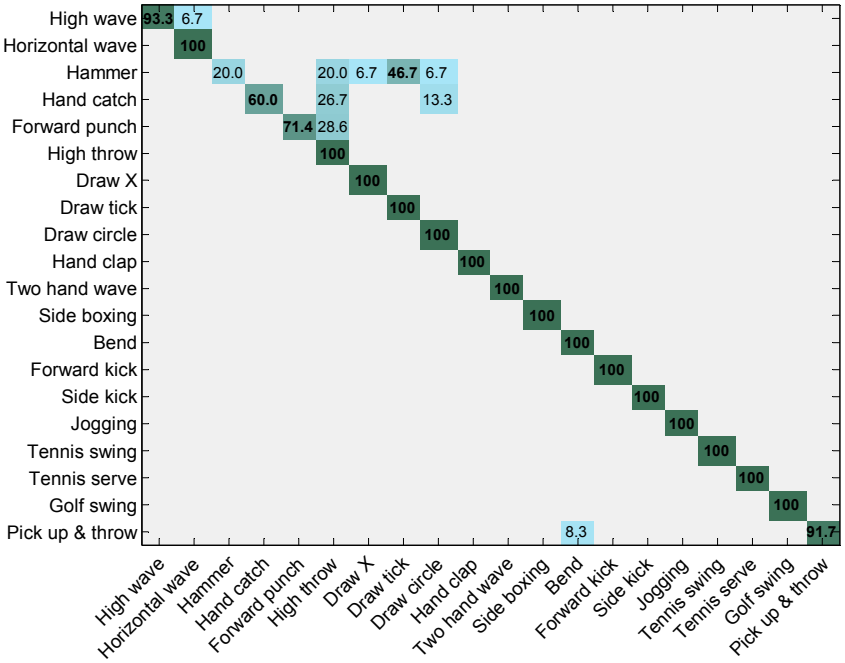
In order to show the accuracy of the approach on individual actions, the confusion matrix is also computed. Figure 4.7 shows the confusion matrix when we use the *k*NN and the Riemannian center of mass per action and per subject with the full skeleton (Figure 4.7a) and with body parts (Figure 4.7b).

It can be noted that for all variations of our approach, we obtained very low accuracies for the actions *hammer* and *hand catch*. This can be explained by the fact that these actions are very similar to some others. In addition, the way of performing these two actions varies a lot depending on the subject. For example, for the action *hammer*, subjects in the training set perform it only once, while some subjects in the test set perform it more than once (cyclically). In this case, the shape of the trajectories is very different. Our method does not deal with this kind of variations. Figure 4.9 illustrates an example of this failure case. As action sequences are represented in high dimension space, trajectories corresponding to only one joint (the right hand joint) are plotted. Indeed, the trajectories of four different samples of the action *hammer* are illustrated, where only one hammer stroke or two hammer strokes are performed. Figures 4.8 shows example sequences of one of each case.

It can be observed in Figure 4.9 that the shape of the trajectories is different in the two cases. In order to visualize samples of three different classes in a two-dimensional space, the Multidimensional scaling (MDS) technique [17] is applied using distance matrix computed on the shape space. These classes are shown in the right part of the figure: *horizontal arm wave* (clear blue), *hammer* (dark blue) and *draw tick* (green). We can see that samples of the action *hammer* are split in two different clusters corre-



(a)



(b)

Figure 4.7 – *MSR Action 3D. Confusion matrix for two variations of our approach: (a) Full skeleton with kNN and Karcher mean per action and per subject; (b) Body parts with kNN and Karcher mean per action and per subject.*

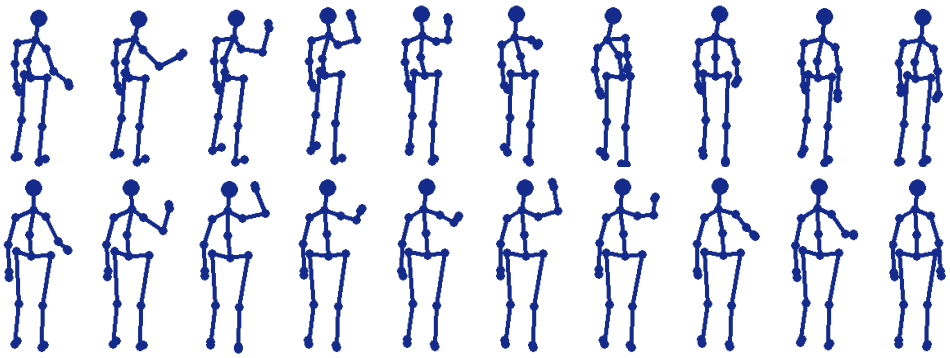


Figure 4.8 – *Visualization of two hammer sequences. In the first sequence, one hammer strocke is performed (sample (a) in Figure 4.9). In the second sequence, two hammer strokes are performed (sample (d) in Figure 4.9).*

sponding to two different ways of performing the action. The distribution of data in the *hammer* cluster is partly overlapped to data in the *draw tick* cluster yielding inaccurate classification of these samples.

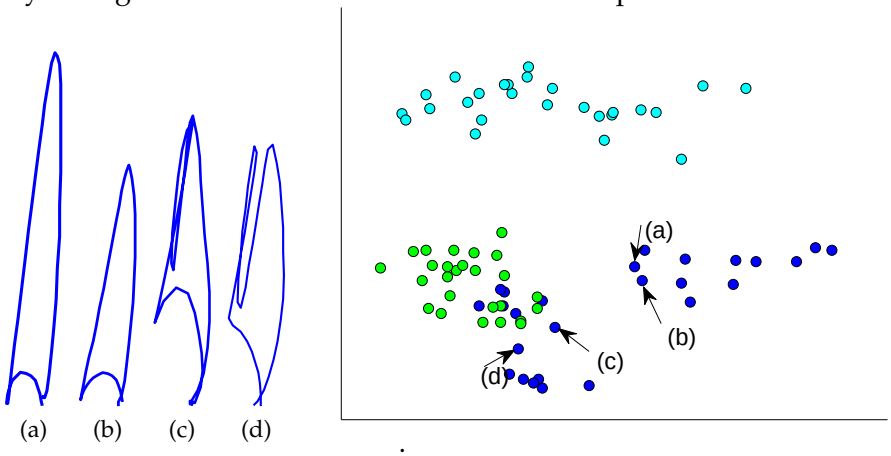


Figure 4.9 – *Visualization of a failure case for the action hammer. Sample trajectories of the right hand joint are shown on the left: (a-b) one hammer stroke; (c-d) two hammer strokes. On the right, clustering of action samples using MDS in a 2D space is reported for three different classes: horizontal arm wave (clear blue), hammer (dark blue) and draw tick (green). The samples of the action hammer are split in two clusters corresponding to the two different ways of performing the action. The distribution of data of the hammer cluster is partly overlapped to data of the draw tick cluster*

Florence 3D Action dataset

This dataset was collected at the University of Florence using a Kinect camera [22]. It includes 9 actions: *arm wave*, *drink from a bottle*, *answer phone*, *clap*, *tight lace*, *sit down*, *stand up*, *read watch*, *bow*. Each action is performed by 10 subjects several times for a total of 215 sequences. The sequences are acquired using the OpenNI SDK, with skeletons represented

by 15 joints instead of 20 as with the Microsoft Kinect SDK. The main challenges of this dataset are the similarity between actions, the human-object interaction, and the different ways of performing a same action.

Results obtained for this dataset are reported in Table 4.3. It can be observed that the proposed approach outperforms the results obtained in [76] using the same protocol (leave-one-subject-out cross validation), even if we do not use the body parts variation.

Table 4.3 – *Florence 3D Action. We compare our method with the one presented in [76].*

Method	Accuracy (%)
NBNN + parts + time [76]	82.0
Our Full Skeleton	85.85
Our Body part	**87.04**

By analyzing the confusion matrix of our method using body parts separately (see Figure 4.10), we can notice that the proposed approach obtains very high accuracies for most of the actions.

However, we can also observe some confusion between similar actions using the same group of joints. This can be observed in the case of *read watch* and *clap hands*, and also in the case of *arm wave*, *drink* and *answer phone*. For these two groups of actions, the trajectories of the arms are very similar. For the first group of actions, in most of the cases, *read watch* is performed using the two arms, which is very similar to the action *clap hands*. For the second group of actions, the main difference between the three actions is the object held by the subject (no object, a bottle, a mobile phone). As we use only skeleton features, we cannot detect and differentiate these objects.

As an example, Figure 4.11 shows two different actions, *drink* and *phone call*, that in term of skeleton are similar and difficult to distinguish.

UTKinect dataset

In this dataset, 10 subjects perform 10 different actions two times, for a total of 200 sequences [99]. The actions include: *walk*, *sit-down*, *stand-up*, *pick-up*, *carry*, *throw*, *push*, *pull*, *wave* and *clap-hand*. Skeleton data are gathered using Kinect for Windows SDK. The actions included in this dataset

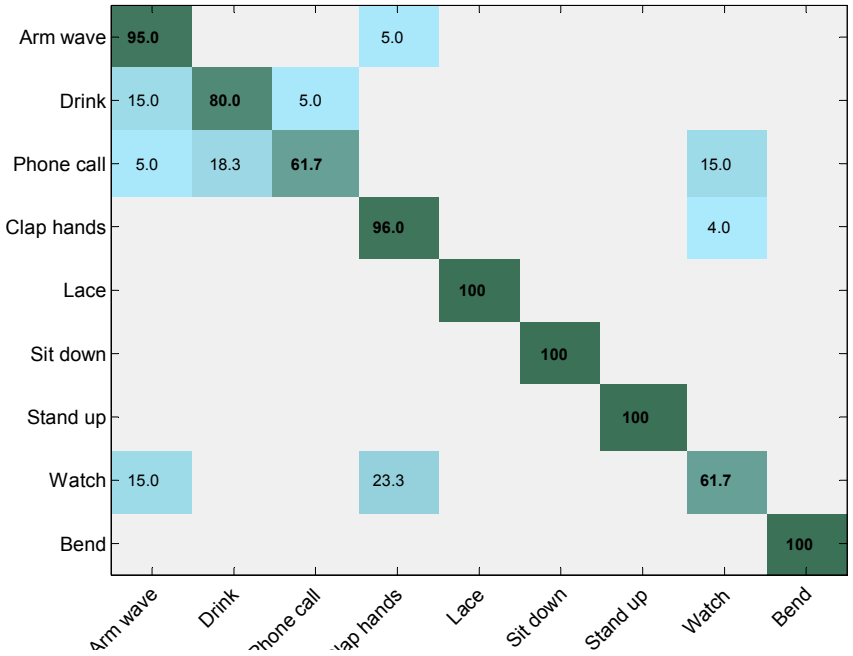


Figure 4.10 – *Confusion matrix obtained by our approach on Florence 3D Action. We can see that similar actions involving different objects are confused.*

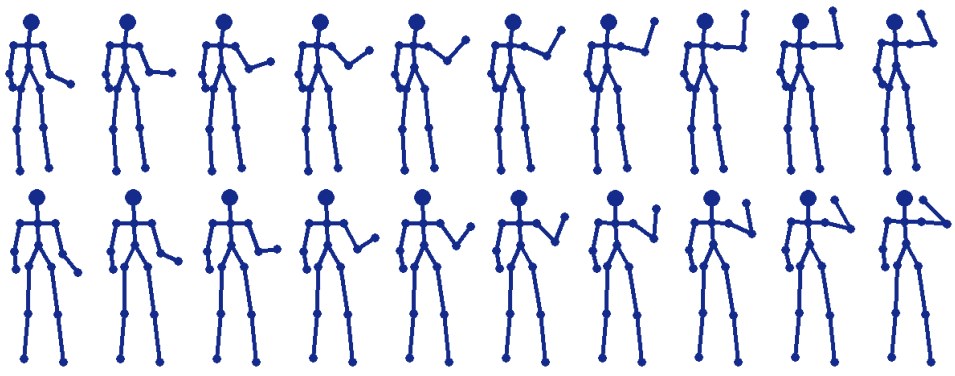


Figure 4.11 – *Example of similar actions from Florence 3D Action dataset: First row corresponds to drink action where the subject holds a bottle; Second row corresponds to phone call action, where the subject holds a phone.*

are similar to those from MSR Action 3D and Florence 3D Action, but they present some additional challenges: they are registered from different views; and there are occlusions caused by human-object interaction or by the absence of some body parts in the sensor field of view.

In order to compare to the work in [99], we follow the same experimental protocol (leave one sequence out cross validation method). For each iteration, one sequence is used as test and all the other sequences are used as training. The operation is repeated such that each sequence is used once as testing. We obtained an accuracy of 91.5%, which improves the accuracy of 90.9% reported in [99]. This shows that our method is robust to different points of view and also to occlusions of some parts of the body.

However, by analyzing the confusion matrix in Figure 4.12, we can notice that lower accuracies are obtained for those actions that include the interaction with some object, for instance the *carry* and *throw* actions. These actions are not always distinguished from actions that are similar in terms of dynamics yet not including the interaction with some object, like *walk* and *push*, respectively. This result is due to the fact that our approach does not take into account any informative description of objects.

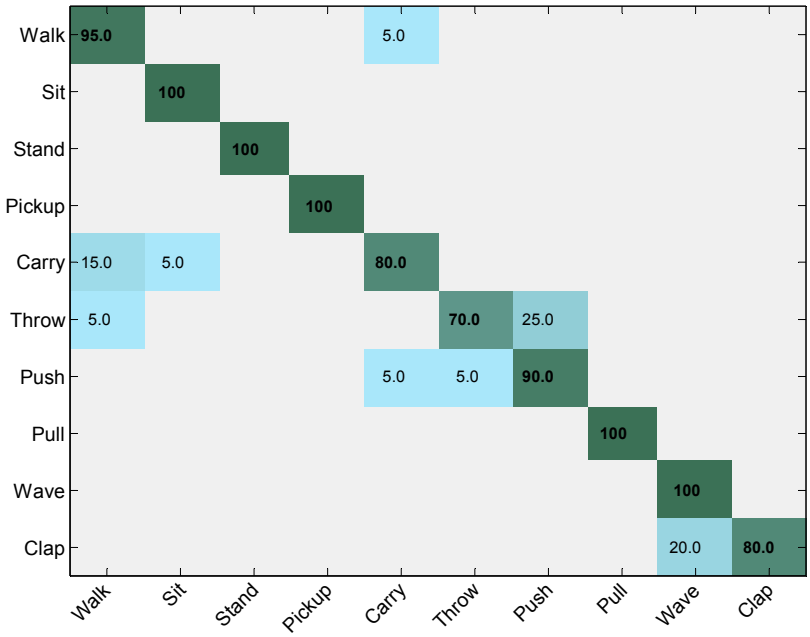


Figure 4.12 – *Confusion matrix obtained by our approach on UTKinect. We can see that similar actions involving different objects are confused.*

Discussion

Results on different datasets show that our approach outperforms most of the state-of-the-art methods.

First, some skeleton based methods like [102] use skeleton features based on pairwise distances between joints. However, results obtained on MSR Action 3D dataset show that analyzing how the whole skeleton evolves during the sequence is more discriminative than taking into consideration the joints separately. In addition, the method proposed in [102] is not invariant to the execution speed. To deal with the execution speed, in [76] a pose-based method is proposed. However, the lack of information about temporal dynamics of the action makes the recognition less effective compared to our method, as shown in Table 4.3.

Second, the comparison with depth-map based methods shows that skeleton joints extracted from depth-maps are effective descriptors to model the motion of the human body along the time. However, results also show that using strength of both depth and skeleton data may be a good solution as proposed in [62]. The combination of both data can be very helpful especially for the case of human-object interaction, where skeleton based methods are not sufficient as shown by the experiments on UTKinect dataset.

4.4.2 Representation and invariance analysis

In the following, we analyze the results obtained by our method with different representation of the humanoid skeleton. Then, we also evaluate the robustness of the method regarding different invariances as geometric transformation and execution speed of actions.

Body parts analysis

The experiments above show that using only the moving parts of the body yields an improvement of the recognition accuracy. In addition, it allows the reduction of the dimensionality of the trajectories and thus the computational costs for their comparison. As we do not use the spine joints of the skeleton, the dimensionality is reduced at least to 48D instead of 60D.

Furthermore, for the actions that are performed with only one part of the body, the dimensionality is reduced to only 12D (in the case of skeletons with four joints per limb).

Invariance to geometric transformations

To demonstrate the effectiveness of our invariant representation against translation and rotation, we analyze the distance between sequences representing the same action class, but acquired from different viewpoints. To this end, we select two samples from the UTKinect dataset corresponding to the action *wave*, and compute the distance between them with and without our invariant representation.

We can see in Table 4.4 that the distance drops from 1.1 to 0.6 if we use our invariant representation. We also compute the distance between actions belonging to similar classes, like *wave* and *clap*. It can be noticed that if we do not use the invariant representation, the nearest sample to the test sample belongs to the class *clap*; however, if the invariant representation is used, the nearest sample belongs to the class *wave*, the same as the test sample.

Table 4.4 – *Distances between a wave sample and two samples of the actions wave and clap acquired from different viewpoints. The columns 'aligned' and 'non-aligned' report the distance value computed with the invariant representation or without it, respectively.*

	wave sample		*clap* sample	
	non-aligned	aligned	non-aligned	aligned
wave sample	1.1	0.6	1.0	0.9

Figure 4.13 shows the corresponding sequences with and without our invariant representation. We can see that the first (blue) and third (green) sequences belonging to classes *wave* and *clap* have similar orientation. Moreover, the two first sequences (blue and orange) belong to the same class *wave* but have different orientation. This shows that different orientation may affect the distance and thus the classification of sequences.

Rate invariance

One main challenge in action recognition is robustness to variations in the execution speed of the action. Without this invariance, two instances of

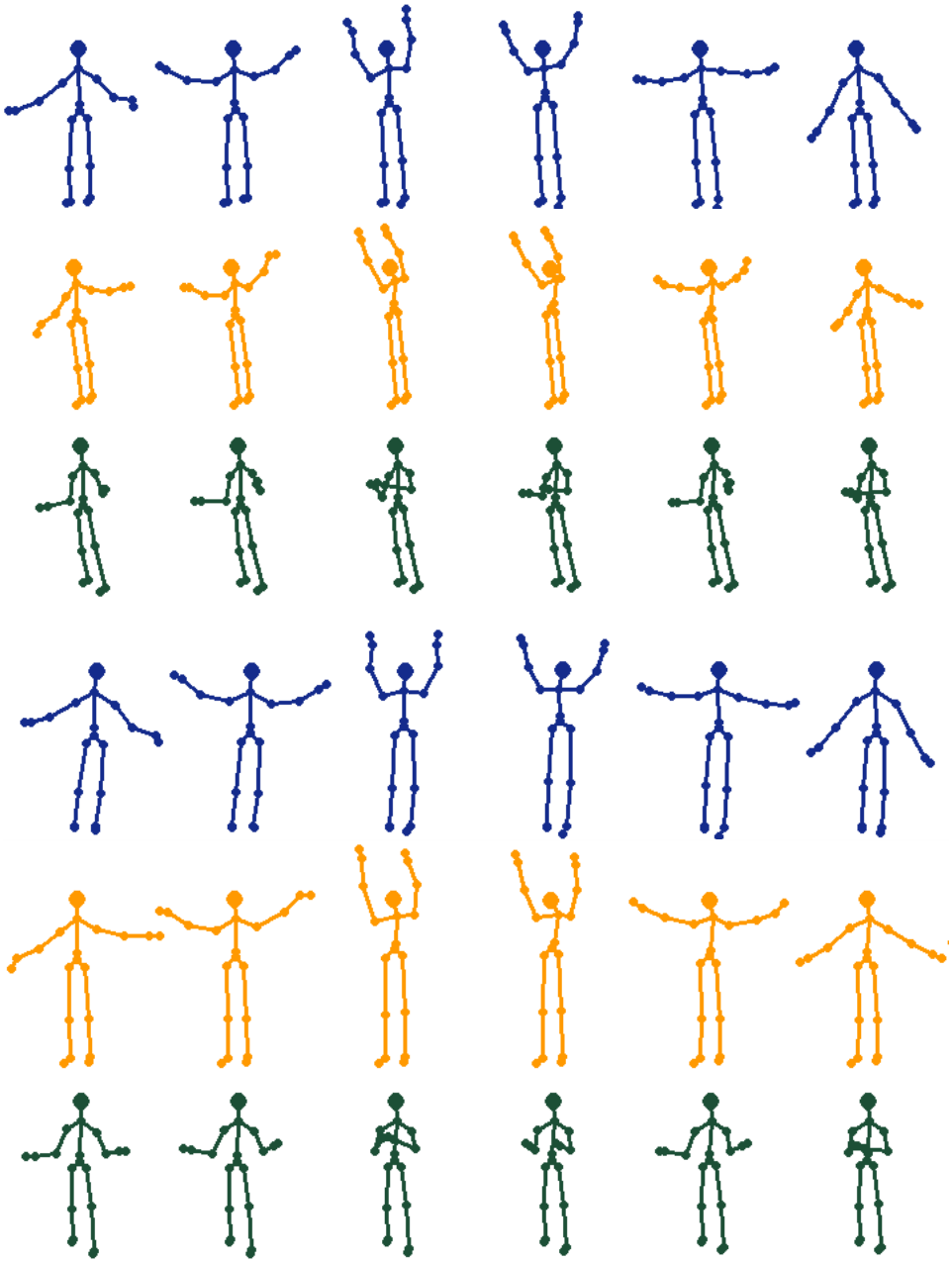


Figure 4.13 – *Illustration of orientation invariance. The first three sequences correspond to classes wave (blue and orange) and clap (green) without our invariant representation. The last three rows show the same sequences with our invariant representation.*

the same action performed at different velocities can be miss-classified. That is why temporal matching between two trajectories is decisive before computing their distance.

The Dynamic Time Warping algorithm is usually employed to solve this problem. It is a popular technique in temporal data analysis, which is used in several applications, including activity recognition by video comparison [88]. In our case, a special version of this algorithm is used to warp similar poses of two sequences at different time instants.

Before computing the distance between two trajectories, we search for the optimal re-parametrization of the second trajectory with respect to the first one. This registration allows us to compare the shape of two trajectories regardless of the execution speed of the action. In practice, we use Dynamic Programming [12] in the shape space to find the optimal re-parametrization and perform registration. The key idea behind dynamic programming is to solve a given problem by considering a collection of simpler sub-problems. As many of these sub-problems are often the same, the dynamic programming approach seeks to solve each sub-problem only once, thus saving a lot of computation. This is especially useful when the number of repeating sub-problems is exponentially large.

To show the importance of this step, we performed the same experiments presented above for two datasets, but without considering the registration step before comparison. The obtained results are presented in Table 4.5.

Table 4.5 – *Results of the proposed method in the case the registration step is considered (R) or not (NR).*

Method	MSR Act. 3D (%)	Florence Act. 3D (%)
kNN Full Skeleton - NR	73.9	82.1
kNN Full Skeleton - R	88.3	85.9
kNN Body parts - NR	73.5	84.7
kNN Body parts - R	91.1	87.0

We can notice that skipping the registration step makes the accuracy much lower, especially for the MSR Action 3D dataset, where the accuracy drops of about 20%. In this dataset, actions are performed at very different speed.

Figure 4.14 shows an example of the action *high throw* performed by two different subjects at different speed. The first row represents eight frames of a training sequence; The second row represents the same eight frames of a new sequence performed at different speed without registration; The third row represents the new sequence after registration with respect to the training sequence. In the reported case, the distance between sequences decreases from 1.31 (without registration) to 0.95 (with registration).

4.4.3 Latency analysis

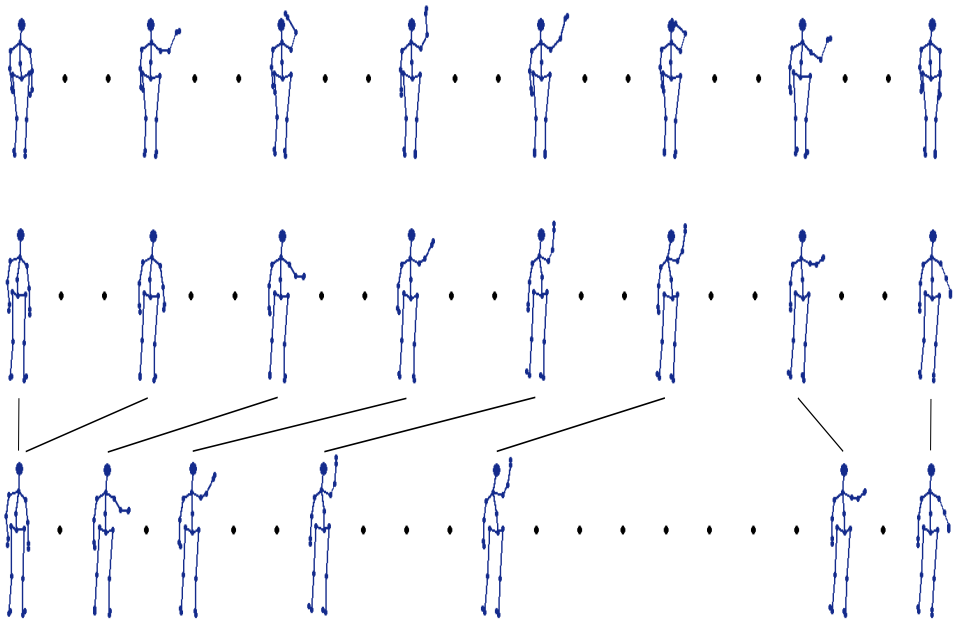
The latency is defined as the time lapse between the instant when a subject starts an action and the instant when the system recognizes the performed action. The latency can be separated into two main components: the *computational* latency and the *observational* latency. The computational latency is the time the system takes to compute the recognition task from an observation. The observational latency represents the amount of time an action sequence needs to be observed in order to gather enough information for its recognition.

Computational latency

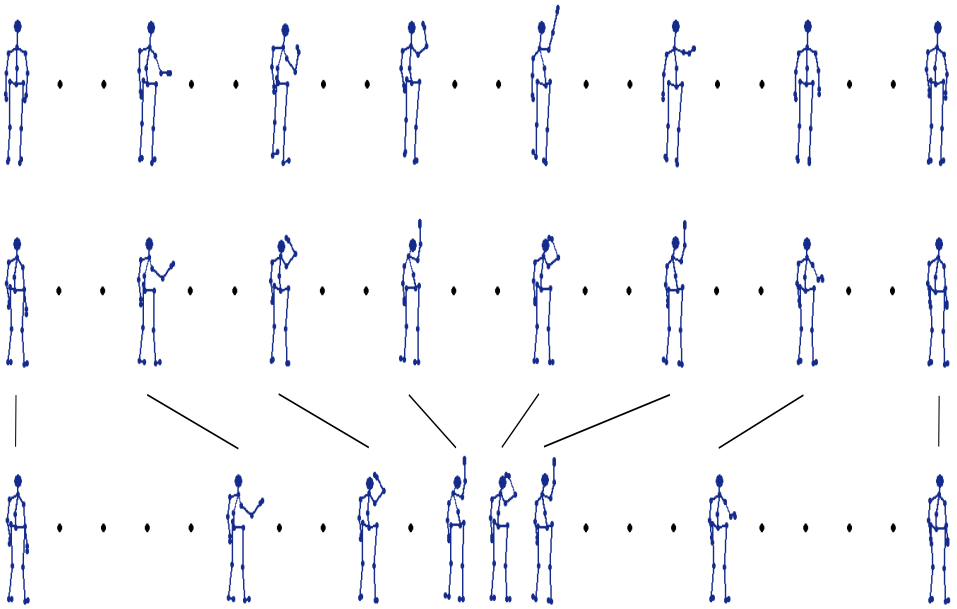
We evaluate the computational latency of our approach on the MSR Action 3D dataset. Using a Matlab implementation with an Intel Core i-5 2.6GHz CPU and a 8GB RAM, the average time required to compare two sequences is 50 *msec* (including trajectories representation in shape space, trajectories registration, distance computation between trajectories, and sequence labeling using kNN).

For a given new sequence, the total computational time depends on the number of training sequences. Indeed, distances between the new sequence and all other training sequences have to be computed, and the k shortest distances are used to label the new sequence.

For example, using the 50-50 *cross subject* protocol on the MSR Action 3D dataset, and using only the kNN approach, classification of an unknown sequence requires comparison to 266 training sequences. Thus,



(a)



(b)

Figure 4.14 – *Temporal registration for actions (a) high arm wave and (b) high throw. From the top: the initial sequence; the sequence to be registered with respect to the initial sequence; the resulting registered sequence. Black lines connect corresponding poses showing how the sequence has been stretched and bent.*

with our approach, the system takes $266 * 0.05 = 13.3$ *sec* to label a new sequence. This computational time is large and thus not suitable for real-time processing.

If we use the Riemannian center of mass per class to have only one representative sequence per class, the number of training sequences is reduced to 20 and the computational time decreases to 1 *sec*, which is more adequate for real-time applications.

As shown in Table 4.1, we obtain our best accuracy for this dataset using riemannian center of mass per action per subject. In that case, the resulted number of training trajectories is 91. Thus, the computational latency becomes $91 * 0.05 = 4.55$ *sec*.

Table 4.6 – *Average computational time to compare two sequences of the MSR Action 3D dataset (the average length of sequences in this dataset is 38 frames). It results that more than 60% of the time is spent in the registration step.*

Step	shape-space representation	registration	distance	kNN labeling	Total
Time (s)	0.011	0.032	0.002	0.005	0.05

Observational latency

To analyze the observational latency of our approach, we show how the accuracy depends on the duration of observation of the action sequence. In the first experiment, the observational latency is analyzed on the MSR Action 3D dataset, where the accuracy is computed by processing only a fraction of the sequence. In each case, we cut the training sequences into shorter ones to create a new training set. During the classification step, we also cut test sequences to the corresponding length and apply our method. We performed experiments using only kNN and also using Riemannian center of mass per action and per subject.

In Figure 4.15, we can see that an accuracy closed to the maximum one is obtained even if we use only half of the sequences. This shows that the computational latency can be masked by the observational latency in the cases where sequences are longer than twice the computational latency. In these cases, the action recognition task can be performed in real-time. This is particularly convenient for applications like video games that require

fast response of the system before the end of the performed action to support real-time interaction.

We also note that the approach using only *k*NN results in slightly higher accuracies when small portions of the sequences are used, with respect to the method also using Riemannian center of mass per action and per subject. This can be explained by the fact that in some cases, subjects do not start the action in the beginning of the sequence but later. Thus, short parts of the sequences may correspond to different parts of the action and even to motionless parts. Hence, the resulted average trajectories may not be very representative of the action.

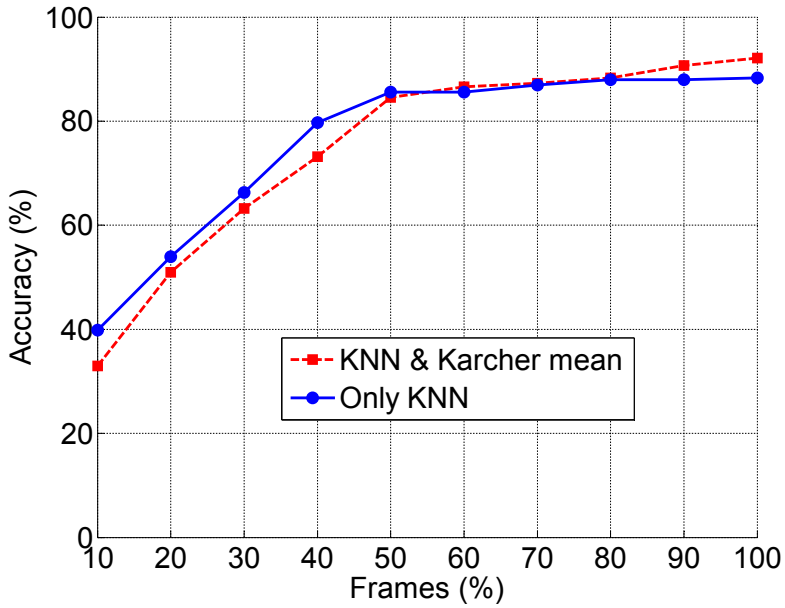


Figure 4.15 – *Latency analysis on MSR Action 3D: Our approach is performed using only the kNN (blue curve), and then using the Karcher mean (red curve). The accuracy at each point of the curves is obtained by processing only the number of frames shown in the x-axis.*

To compare the observational latency of our approach, we perform experiments on the UCF-Kinect dataset [27], where the observational latency of other methods is also evaluated. This dataset consists of 16 different gaming actions performed by 16 subjects five times for a total of 1280 sequences [27]. All the actions are performed from a rest state, including *balance*, *climb up*, *climb ladder*, *duck*, *hop*, *vault*, *leap*, *run*, *kick*, *punch*, *twist left*, *twist right*, *step forward*, *step back*, *step left*, *step right*. The locations of 15 joints over the sequences are estimated using Microsoft Kinect sensor

and the PrimeSense NiTE. This dataset is mainly used to evaluate the ability of our approach in terms of accuracy/latency for a low-latency action recognition system.

The same experimental setup as in Ellis et al. [27] is followed. To do that, we use only the *k*NN and a *4-fold cross validation* protocol. Four subjects are selected for test and the others for training. This is repeated until each subject is used once. Actually, since there are 16 subjects, four different test folds are built and the mean accuracy of the four folds is reported. For a fair comparison to Ellis et al. [27], the obtained accuracy is reported with respect to the maximum number of frames (and not to a percentage of sequences). For each step, a new dataset is built cutting the sequences to a maximum number of frames. The length of the sequences varies from 27 to 269 frames with an average length equal to 66.1 ± 34 frames. It should be noticed that, if the number of frames of a sequence is below the maximum number of frames used in experiments, the whole sequence is treated.

We compare our results with those reported in [27], including their proposed approach *Latency Aware Learning* (LAL), and two baseline solutions: *Bag of Words* (BoW) and *Conditional Random Field* (CRF). The observational latency on this dataset is also evaluated in [62], but following a different evaluation protocol (i.e., a *70/30 split* protocol instead of the *4-fold cross validation* proposed in [27]), so their results are not reported here.

The curves in Figure 4.16 and the corresponding numerical results in Table 4.7 show that our approach clearly outperforms all the baseline approaches reported in [27]. This significant improvement is achieved either using a small or a large number of frames (see the red curve in Figure 4.16).

Table 4.7 – *Numerical results at several points along the curves in Figure 4.16.*

	#Frames						
Method	10	15	20	25	30	40	60
CRF	14.5	25.5	46.9	67.3	80.7	91.4	94.3
BoW	10.7	21.2	43.5	67.6	83.2	91.9	94.1
LAL	13.9	37.0	64.8	81.6	90.6	95.2	95.9
Our	**30.5**	**60.9**	**79.9**	**91.1**	**95.1**	**97.8**	**99.2**

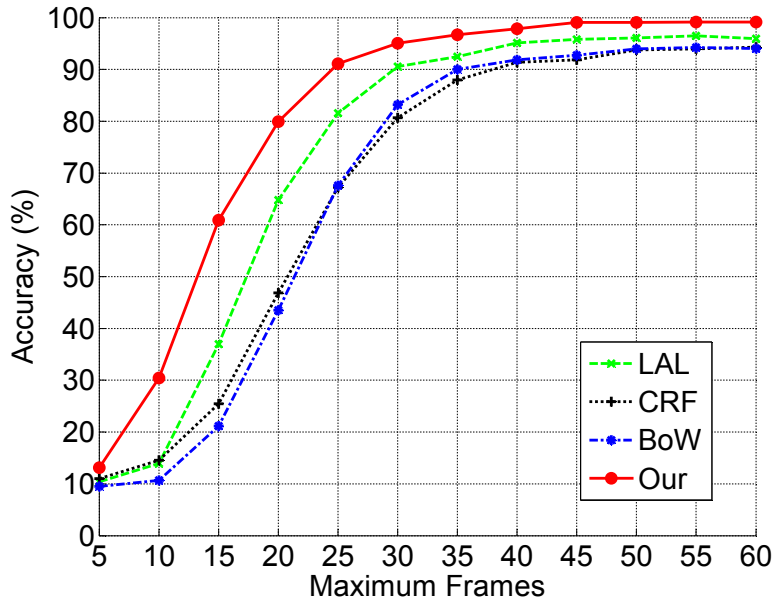


Figure 4.16 – *Latency analysis on UCF-Kinect. Values of the accuracy obtained by our approach using only the kNN, compared to those reported in [27]. The accuracy at each point of the curves is obtained by processing only the number of frames shown in the x-axis.*

We can also notice that only 25 frames are sufficient to guarantee an accuracy over 90%, while BoW and CRF show a recognition rate below 68%, and LAL achieves 81.65%. It is also interesting to notice that using the whole sequences, we obtain an accuracy of 99.15%, and the same accuracy can be obtained by processing just 45 frames of the sequence.

4.5 Conclusion

In this Chapter, an effective human action recognition approach is proposed using a spatio-temporal modeling of motion trajectories in a Riemannian manifold. The evolution of the 3D position of each skeleton joint along the sequence is represented as a motion trajectory in the action space. The shape of each motion trajectory is then expressed as a point in the shape space. Thanks to the Riemannian geometry of this manifold, action classification is solved using the nearest neighbor rule, by warping all the training points to the new query trajectory and computing an elastic metric between the shape of trajectories.

The experimental results on the MSR Action 3D, Florence 3D Action

and UTKinect datasets demonstrate that our approach outperforms the existing state-of-the-art methods in most of the cases.

Furthermore, the evaluation in terms of latency clearly demonstrates the efficiency of our approach for a rapid recognition. In fact, 90% action recognition accuracy is achieved by processing just 25 frames of the sequence. Thereby, our approach could be used for applications of human action recognition in interactive systems, where a robust real-time recognition at low latency is required.

However, experiments also demonstrated some limits of our approach. Firstly, we identify a failure case when actions can be characterized by a different number of repetitions of a single gesture. In that case the shape of the resulted motion trajectories may differ and the recognition effectiveness can be affected.

Secondly, as we are using only skeleton data, we only have information about the human pose and its evolution along the time. The analysis of obtained results on benchmark action dataset shown that some different actions may be very similar in term of human motion. What differentiate such similar actions is the object held by the subject. By only using skeleton data, we are unable to describe such interaction with objects and thus to differentiate these similar actions.

Finally, our proposed method is a sequence-based approach: we analyze and classify a full delimited sequence. Indeed, during experiments, we consider that each sequence contains only one action starting in the beginning of the sequence and finishing at its end. However, this consideration does not reflect a real-world context in which a camera is continuously observing a scene. Hence, the subject may perform different actions successively as well as remain still during a certain time interval. Thus our method is not appropriate for this real-world context.

In the following Chapter, we investigate a way to deal with these limits and thus we propose a method suitable for more complex cases.

5 Behavior Understanding by Motion Unit Decomposition

Sommaire

5.1 Introduction

In this Chapter, we propose to extend our analysis so as to define a method suitable for not only human actions, but all kind of human motions that we define as human behaviors.

In recent years, recognition and understanding of human behaviors by analyzing data provided by depth cameras has attracted the interest of several research groups (see [105] for a recent review). While some methods focus on the analysis of human motion in order to recognize human *gestures* or *actions*, other approaches try to also model interactions with objects, so as to analyze more complex behaviors, like *activities*.

Hybrid solutions are often proposed, which use depth maps for modeling scene objects, and body skeleton for modeling the human motion. For example, Wang et al. [94] used Local Occupancy Patterns to represent the observed depth values in correspondence to skeleton joints. Other methods propose to describe and model spatio-temporal interactions between human and objects characterizing the activities, using Markov Random Field (MRF) [41] or Conditional Random Field (CRF) [42].

Whereas these solutions study short sequences, where one single behavior is performed along the sequence, additional challenges appear when several different behaviors are executed sequentially over a long sequence. In order to face these challenges, methods based on *online detection* are proposed. Such methods can recognize behaviors before the end of their execution by analyzing short parts of the observed sequence. Thus, these methods are able to recognize multiple behaviors within a long sequence, which may not be the case for methods analyzing the entire sequence directly.

5.1.1 Challenges

While constraints defined in Chapter 4 for action recognition, like robustness to geometric transformations remain, some additional challenges appear when it comes to study more complex human motions, like activities.

Indeed, the high complexity of human motions and the variability of gesture combinations that can characterize the human behavior signifi-

cantly complicate the task. Local (over time) analysis of the human motion is often necessary in order to have a more accurate and thorough understanding of what the subject is doing.

In addition, human behaviors often involve interactions with the real-world environment and manipulations of objects in the scene. While such information about the context may help the understanding of what the subject is doing, it also involves possible occlusions of parts of the human body, resulting in missing or noisy data.

Finally, as mentioned above, an interesting challenge relates to the online capability of methods favoring not only to perform real-time analysis but also to detect successive human behaviors in a long motion sequence. In sum, such online feature allows to answer to a more realistic need.

5.1.2 Overview of the approach

So as to face the challenges stated above, we propose a framework based on a local analysis of motion sequences and a combination of skeleton and depth features to describe both human motion and interaction with objects.

First, we represent the skeleton of each frame by a 3D curve describing the human pose. These curves are then interpreted in a Riemannian manifold, which defines a shape space where shapes of the curves can be modeled and compared using elastic registration and matching. Such shape analysis allows the identification and grouping of the human poses belonging to a same elementary motion. As a result, a motion sequence is temporally segmented into a set of successive sub-sequences of elementary motions, called *Motion Units* (MU).

A MU is thus characterized by a sequence of skeletons, each of which is in turn modeled as a multi-dimensional vector by concatenating the three-dimensional coordinates of its joints. Then, the trajectory described by this vector in the multi-dimensional space is regarded as a signature of the temporal dynamics of the movements of all the joints. Similarly to action trajectories in Chapter 4, the shapes of such motion trajectories are studied and compared in a Riemannian shape space. The elastic metric provided

in this framework allows us to compare motion trajectories independently to their elasticity, i.e., the execution speed of motions. A statistical analysis on this manifold allows us to identify representative shapes characterizing a set of MUs.

However, skeleton data are not sufficient to describe human behaviors, when objects are manipulated. This motivated us to describe, in each MU, the depth appearance around subject hands providing information about possible human-object interactions.

Finally, the sequence of MUs is modeled through a Dynamic Naive Bayesian classifier, in order to combine both skeleton and depth features and captures the dynamics of the human behavior. Figure 5.1 summarizes the proposed solution.

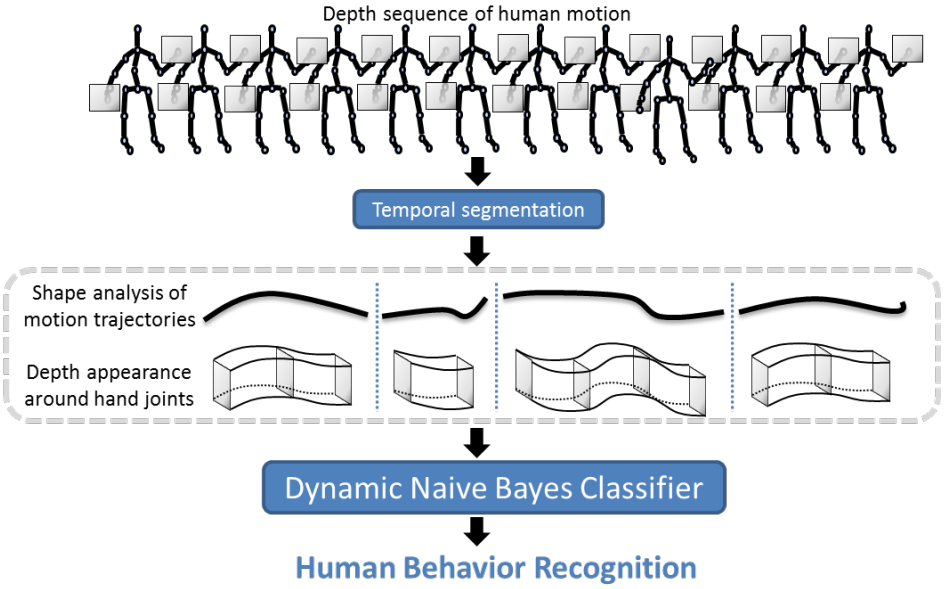


Figure 5.1 – *Overview of our approach. Shape analysis of human poses allows us to identify temporal segments of elementary motions (i.e., MU). Each MU is described using the trajectory of the joints of the skeleton regarded as a multidimensional vector, and the depth appearance around subject hands. A Dynamic Naive Bayesian classifier is then used to model the sequence of temporal segments and recognize the human behavior.*

5.1.3 Motivations

The main factors that motivate us to extend our study and to propose this solution are:

- Experiments in Chapter 4 highlighted some limits when the complexity of the human action is increased, like the repetition of gestures;
- Experiments also demonstrated a limitation of skeleton data which are insufficient to model human-object interactions;
- Local (over time) analysis of the human motion is often necessary in order to have a more accurate and thorough understanding of what the subject is doing;
- Online capability is an important aspect in order to fit to more realistic scenarios.

5.2 Segmentation of motion sequences

Due to the complexity of the human motion characterizing activities, we propose to decompose the problem by segmenting the full motion into shorter MU, which are easier to analyze. This process is based on the analysis of the human pose at each frame of the sequence.

5.2.1 Pose representation

The human body is represented by a set of 3D joints, which are located in correspondence to the different body parts. Thus, a human pose is characterized by a certain spatial configuration of these 3D joints. In order to describe human poses, we propose to analyze the shape of the spatial configuration of 3D joints. By connecting the 3D joints, the human pose can be viewed as a 3D curve representing the shape of the human body.

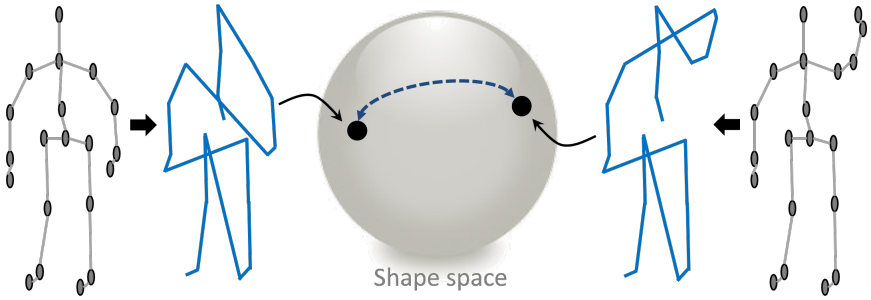
As shown in Figure 5.2, in order to keep natural information about the human shape represented by the limbs, we keep a coherent structure linking together joints belonging to the same limb. Thus, a 3D curve representing the human pose connects successively the spine's joints, then the arms's joints (left/right) and finally the legs's joints (left/right). This is illustrated in Figure 5.2.

In this way, a human pose is represented by a 3D curve instead of a 3D skeleton. We can now perform shape analysis of curves using the shape

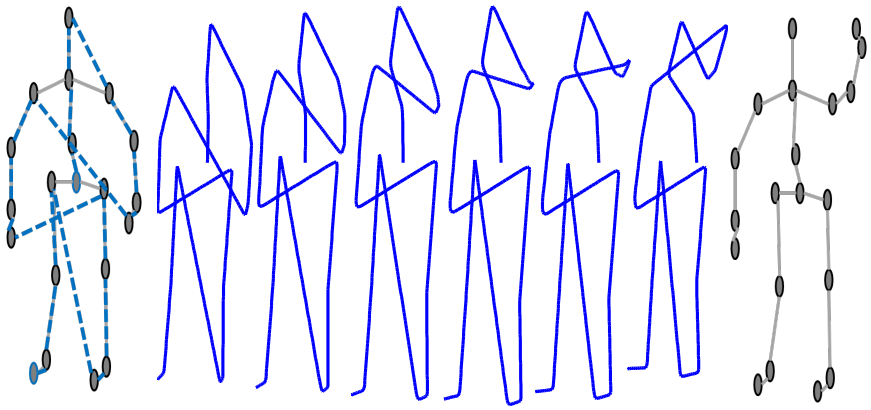
analysis framework and the provided distance, as described in Chapter 3, for $n = 3$, as each joint is represented by three coordinates x, y and z.

Note that, as explained later, we need to compare successive human poses from a same sequence (same subject). Hence, we can assume that the scale of skeletons is unchanged during a sequence as well as the orientation of the subject between two successive poses. Likewise, as a 3D curve connects joints in a predefined order, the parametrization of curves remains the same along a single sequence. Since it is not necessary to find the optimal re-parametrization between two shapes, the analysis of the shape of the 3D curves is simplified.

Figure 5.2 shows a geodesic path between two human poses represented by their 3D curve.



(a)



(b)

Figure 5.2 – *Shape analysis of human poses in the shape space. (a) Human poses are represented by 3D curves. The shape of a curve is an element of the shape space. The distance between two shapes is measured through the geodesic distance (length of the minimum path) between their representations in the shape space. (b) Visualization of the geodesic path representing a natural deformation of the shape of the initial pose to the shape of the target pose.*

5.2.2 Motion segmentation

Once a distance measuring the similarity between the shape of two poses is defined, we can use it to analyze the deformation of the human body along an activity sequence. Hence, in order to divide the continuous sequence into segments exhibiting elementary motion, i.e., MUs, we detect when the motion is changing.

We consider that when a human is performing two successive motions, its speed is higher during the performance of motions. Conversely, its speed becomes slower at the end of the first motion and at the beginning of the second one. Hence, we identify MU by breaking the sequence in correspondence to points where the speed of change of the 3D curve has a local minimum. Such considerations are motivated by similar and efficient assumptions employed in [37, 38] on the speed of human hands for task segmentation.

The notion of speed of motion can be viewed as the similarity or dissimilarity among human poses within a short time interval. A time interval where motion happens corresponds to different poses within the time interval. Conversely, similar poses in the time interval correspond to the transition between the two motions. Our goal is to detect such transitions in an activity sequence in order to decompose it into elementary motions.

For that, we propose to analyze the human pose shape within a sliding window along the sequence and take advantage of the shape analysis framework that enables the computation of statistics, like the mean and the standard deviation, on the manifold. Hence, given the poses $p_1, \dots, p_n$ observed over a temporal window of predefined duration, the average pose shape μ is computed as the Riemannian center of mass [40] of the pose shapes $q_1, \dots, q_n$ on the shape space (see Algorithm 1 in Chapter 3 for more details). For this purpose, the distance $d_{\mathcal{S}}$ described in Chapter. 3 is used according to the Riemannian center of mass following expression:

$$\mu = \arg\min_{[q]} \sum_{i=1}^{n} d_{\mathcal{S}}([q], [q_i])^2 \,. \tag{5.1}$$

Once the mean pose shape is computed, the standard deviation be-

tween this mean shape and all the shapes within the window is estimated using the distance between the mean shape and the sample shapes (see Equation 3.13 in Chapter 3). Such standard deviation value, denoted σ, represents the amount of dissimilarity or motion among human poses within the window.

Figure 5.3 illustrates the process of computing σ on the shape space. Higher values of σ correspond to faster motion, while lower values correspond to slower motion, i.e., transition intervals. By detecting local minima along the sequence, we are able of temporally localizing the motion transitions, and thus of decomposing the sequence into MUs.

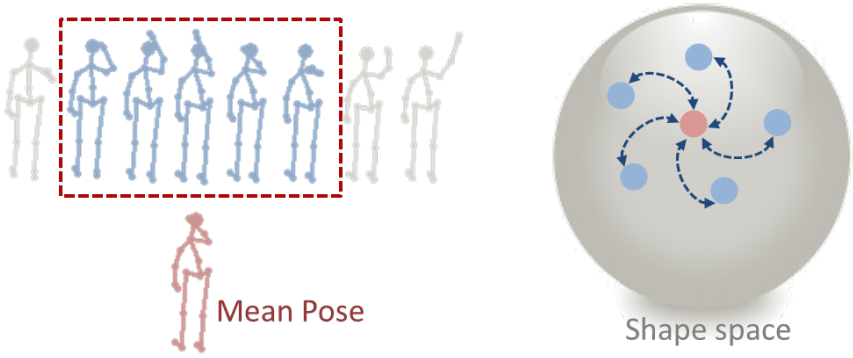


Figure 5.3 – *Computation of the standard deviation on the shape space. The shapes of poses in a window are represented as elements on the shape space. The mean shape and standard deviation are computed in the shape space.*

As an example, Figure 5.4 shows the evolution of the standard deviation σ along a sequence and the different MUs identified by breaking the sequence in correspondence to local minima of σ.

5.3 Segment description

Once an activity sequence is segmented, we propose to analyze the resulting MUs in order to describe the whole sequence. First, we propose to explore the motion performed by the subject during each MU. Secondly, we consider depth appearance around hands in order to characterize possible objects held by the subject. Additionally, we investigate a bag-of-word paradigm so as to describe a complex motion sequence by a set of exemplar MUs.

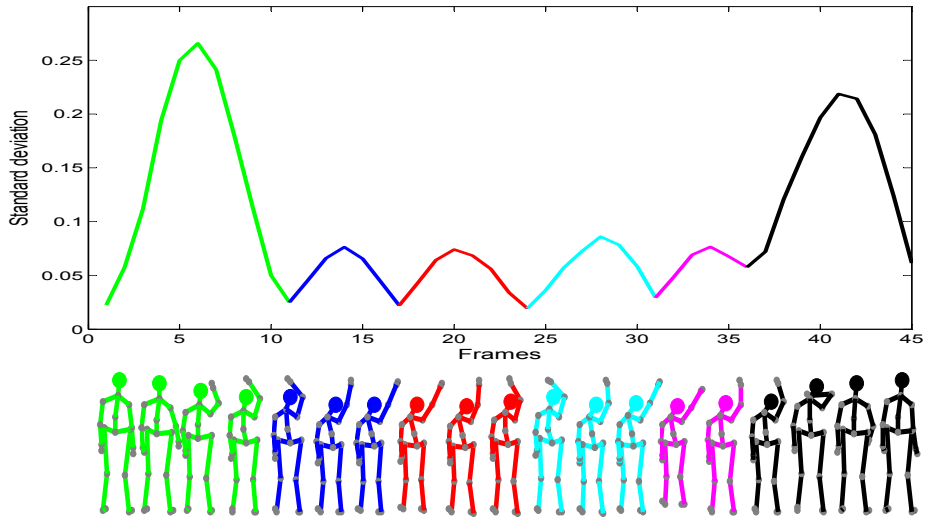


Figure 5.4 – *Segmentation of a sequence based on minima of the standard deviation σ. Different MUs are displayed with different colors. The corresponding poses are displayed under the plot of the standard deviation.*

5.3.1 Human motion description

First, we propose to describe the human movement during a MU by interpreting the motion of the pose along the time interval corresponding to a MU. We proceed similarly to human motion in the whole sequence in Chapter 4. For each frame included in the MU, we concatenate the three real world coordinates of each joint to build a feature vector. Let N_j be the number of joints the skeleton is composed of, the posture of the skeleton at frame t is represented by a $3N_j$ dimensional tuple:

$$v(t) = [x_1(t)\ y_1(t)\ z_1(t), \ldots, x_{N_j}(t)\ y_{N_j}(t)\ z_{N_j}(t)]^T\ . \tag{5.2}$$

For a MU composed of N_f frames, N_f feature vectors are extracted and arranged in columns to build a feature matrix M describing the whole segment:

$$M = \begin{pmatrix} v(1) & v(2) & \ldots & v(N_f) \end{pmatrix}\ . \tag{5.3}$$

This feature matrix represents the evolution of the skeleton pose over the time. Hence, it can be viewed as a continuous trajectory in R^{3N_j} representing the motion in a $3N_j$ dimensional space. The size of such feature matrix is $3N_j \times N_f$.

Note that, in order to guarantee the analysis of MU to be invariant

to translation and rotation, we normalize the position and the orientation of the subject before extracting the features. We use the same method described in Chapter 4. This makes the representation invariant to the position and orientation of the subject in the scene.

With this representation, an activity sequence can be viewed as a set of short spatio-temporal trajectories in R^{3N_j} representing MUs, as illustrated in Figure 5.5.

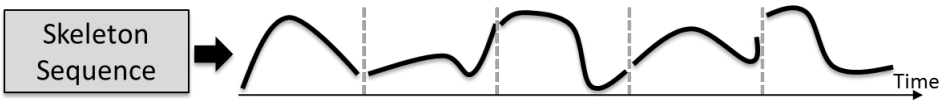


Figure 5.5 – *An activity sequence can be viewed as a set of successive spatio-temporal trajectories in R^{3N_j} representing MUs performed by the subject.*

In order to analyze these MUs represented as trajectories, we propose to analyze their shape, assuming that similar MUs are represented by trajectories with similar shapes, while different MUs are represented by trajectories with different shapes. We use the shape analysis framework described in Chapter 3 to capture and analyze the shapes of trajectories represented in a high dimensional space ($n = 3N_j$). Shapes are represented as elements on the shape space and the similarity measure between two shapes is the elastic metric $d_{\mathcal{S}}$ (Equation 3.8 on this shape space.

5.3.2 Depth appearance

In order to perform robust and discriminant recognition of human activity, descriptors of the motion of the body skeleton should be complemented with descriptors of the objects the user is interacting with, if any. Combination of motion and object descriptors not only improves the robustness of the activity recognition scheme, but is also necessary to discriminate between actions that would be almost identical in terms of motion patterns.

For instance, discriminating activities like *Drink* and *Phone call* based on the analysis of the sole motion patterns would require a description framework capable of accurately distinguishing whether the user hand (i.e., the joint of the skeleton corresponding to the user hand) is closer to the mouth than to the ear. This level of accuracy is generally beyond the capability of commercial low-cost scanners, unless the user is very

close to the sensor. Differently, two such actions can be much more easily discriminated by considering the objects the user is interacting with.

Based on these premises, we propose to describe the depth appearance around subject hands in addition to their motion. To this end, we use the Local Occupancy Pattern (LOP) [93] feature around the two hands of the subject and adapt it to our method.

This feature represents the distribution of depth pixels within a local region. As depth images can be viewed as 3D point clouds, the local regions are represented by 3D bounding boxes centered at the hand joints. Each bounding box is partitioned into a grid with non overlapping 3D cells. The number of cells is $N_c = N_x \times N_y \times N_z$, where N_x, N_y and N_z are the number of divisions in the corresponding dimension. The number of 3D points that fall in each cell is counted.

Let P_i be the number of points within a cell i, a sigmoid normalization is employed to obtain the occupancy information O_i of the cell [93]:

$$O_i = \frac{1}{1 + exp(-\lambda P_i)}\ , \tag{5.4}$$

where λ is used to parametrize the slope of the sigmoid function. The LOP feature l for the local region is the concatenation of occupancy information of all cells: $l = [O_1, O_2, \ldots, O_{N_c}]$.

As shown in Figure 5.6, the LOP feature can be viewed as a histogram representing the distribution of depth pixels in the local region around hand joint, thus describing the depth appearance of the object held by the subject.

In order to guarantee compatibility with motion description presented above, we keep the representation of an activity as a sequence of successive segments representing MUs.

For each frame of a MU, we compute the LOP feature for each hand joint (l_l and l_r) and concatenate them to form one global LOP feature vector $L_f = [l_l, l_r]$ for the frame f. The length of such feature vector is $2 \times N_c$. The same is repeated for all frames of the MU, in order to describe the depth appearance during the whole MU. As a result, each

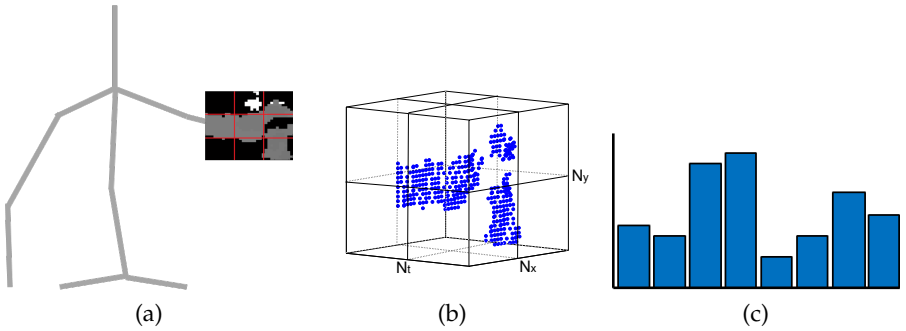


Figure 5.6 – *LOP feature computation for one frame, where the subject is holding a bottle. (a) A bounding box is extracted from the depth image around the hand joint. (b) The bounding box can be viewed as a 3D cuboid divided into 3D cells. Here* $N_x = N_y = N_z = 2$. *(c) The number of 3D points within each 3D cell is counted. As a result, the LOP feature can be viewed as a histogram, where each bin represents a 3D cell.*

MU is described by a set of N_f LOP features, N_f being the number of frames of the MU.

However, MU segments can have different duration N_f. As a result, MU are described with a different number of LOP features, which is not convenient to compare them. To deal with duration variability, we propose a compact representation of the depth appearance, which is independent of its duration.

First, we assume the object held by the subject during the time interval corresponding to a MU does not change considerably. Thus, we consider that the distribution of depth pixels around hand joint is stable and we compute the mean of the LOP features among frames of a MU. As a result, we have one single feature describing average depth appearance during a MU, that we call Mean LOP feature (MLOP).

Then, we consider the manipulation of the object during MU can change the depth appearance around hand joint. For instance, for the activity *Drink*, a MU would consist of bringing the container to the mouth. In that case, the support where the object is located may appear in the local region around the hand, in the first part of the MU, but at the end of the MU, the face of the subject may be present in this local region. This variability of depth appearance may add useful information compared to the average depth appearance along the MU.

To represent this depth variation, we adopt an extension of LOP fea-

ture in four dimensions called 4DLOP. The spatio-temporal volume representing the evolution of the local region around hands along the MU is also partitioned in the time dimension. As a result, the number of cells of this 4D volume is now $N_c = N_x \times N_y \times N_z \times N_t$, where N_t is the number of divisions in the temporal dimension. The occupancy information is computed for each 4D cell using Eq. (5.4) to build the 4DLOP feature.

Note that, differently to [91], which analyzes depth variation in fixed 4D boxes, we consider depth variation in a moving spatio-temporal region following the motion of human hands. This idea is illustrated in Figure 5.7.

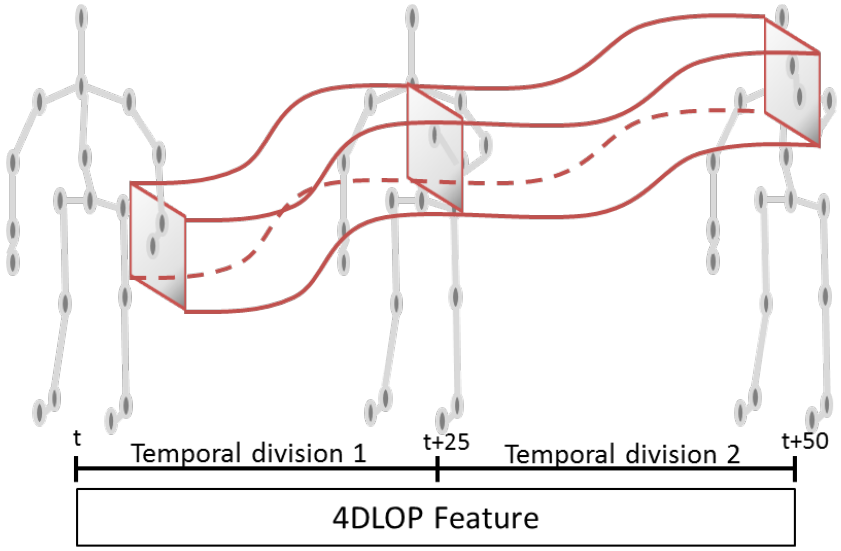


Figure 5.7 – *Schema of the 4DLOP feature. Here, the duration of the MUs is 50 frames. The spatio-temporal volume along the time is divided in two intervals ($N_t = 2$). As a result the 4DLOP feature vector represents the depth appearance evolution along the MU in two time steps.*

As final result, each MU segment is represented by a feature vector describing the depth appearance along the interval independently to its duration (either MLOP or 4DLOP).

5.3.3 Vocabulary of exemplar MUs

Additionally, we propose to investigate the bag-of-word paradigm so as to identify codebook of exemplar MUs (symbols) to be used as a reference dictionary to describe human behaviors. As usual, such codebook is learned from training sequences. Then our idea is that unknown complex

motion sequences can be represented as a set of generic MUs from the learn codebook, thus facilitating their analysis and understanding.

As each MU is described by two types of features representing human motion and depth appearance, we identify two distinct codebooks for each of the feature.

Human motion codebook

For the case of human motion, each MU is represented by the shape of the corresponding motion trajectory in the shape space. To learn the codebook of exemplar shapes, we perform clustering of shapes using the k-means clustering algorithm (Algorithm 2) described in Chapter 3 on the shape space. Such clustering provides a mapping between trajectory shapes represented on the shape space and a finite set of symbols corresponding to clusters.

In order to describe each cluster by using only its corresponding exemplar shape, we propose to learn a density function for each cluster. These density functions capture the variability between shapes belonging to the same cluster and provide a deeper modeling of each cluster. In so doing, we assume the distribution of shapes within a cluster follows a multivariate normal model.

The process of learning such multivariate distribution on the shape space is described in Algorithm 3 in Chapter 3. We learn a distribution for each cluster k by considering the tangent space at the cluster mean μ_k. This approximation is valid because samples belong to the same cluster. Thus, we can assume that they lie in a small neighborhood around the mean shape μ_k.

As a result, each cluster of shapes is represented by a probability density function defined as:

$$f(\tilde{v}) = \frac{1}{(2\pi)^{n/2}\ |\Sigma|^{1/2}} e^{-\frac{1}{2}\tilde{v}^T\Sigma^{-1}\tilde{v}}\ . \tag{5.5}$$

where Σ corresponds to the covariance matrix computed on the learned principal subspace (see Algorithm 3 in Chapter 3 for more details).

The process of learning the distribution on the shape space is illustrated in Figure 5.8.

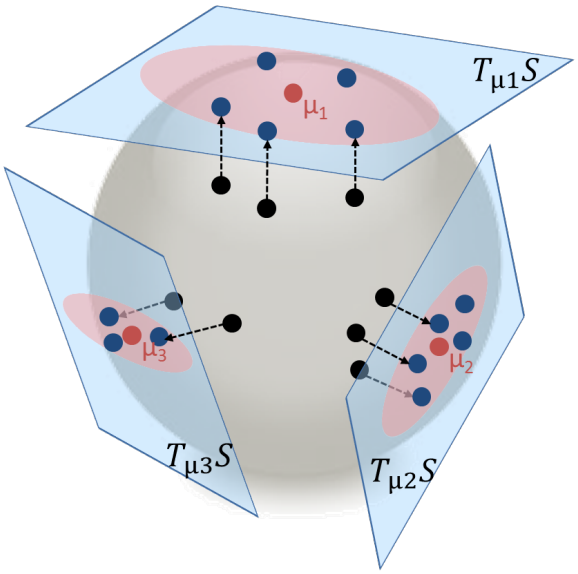


Figure 5.8 – *Learning distribution of shapes belonging to the same cluster on the shape space $\mathcal{S}$. For each cluster, the mean shape μ is computed (red) from shapes q_i belonging to the same cluster. Then, the shapes q_i (black) are projected on the corresponding tangent space $T_\mu\mathcal{S}$. Such tangent vectors v_i (blue) are used to compute the covariance matrix and learn the multivariate distribution for each cluster.*

As mentioned above, the codebook is learned only from MUs belonging to training sequences. Such learned codebook is used to label a MU of a test sequence, characterized by its trajectory shape on the shape space. The test shape is first projected into the learned subspace of a cluster k. Then, using the corresponding covariance matrix, we can compute the probability that the test shape has been generated by the learned density function corresponding to the cluster k. We do the same for each cluster and assign the test shape to the cluster giving the highest probability.

To compute such probability, a common way is to use the log probability. Let $\tilde{v}_k$ being a test shape q projected in the principal subspace of the cluster k with a corresponding covariance matrix Σ_k. Then the log probability that the shape q belongs to the cluster k is defined as:

$$L = -\frac{1}{2}ln(|\Sigma_k|) \ -\frac{1}{2}\tilde{v}_k^T\Sigma_k^{-1}\tilde{v}_k \ -\frac{n}{2}ln(2\pi) \tag{5.6}$$

Depth appearance codebook

For the case of depth appearance, each MU is described by a LOP feature representing depth appearance around subject hands. Similarly to motion trajectories, we use the k-means algorithm to cluster LOP features and build a codebook of exemplar LOP. The distance d_l that we use to compare two LOP feature vectors l_A and l_B is the l^2-norm:

$$d_l = \|l_A - l_B\|^2_{N_c} = \sum_{i=1}^{N_c} (a_i - b_i)^2 \,, \tag{5.7}$$

where a_i and b_i are the i-th components of l_A and l_B, respectively. Such clustering provides a mapping between LOP feature vectors and a finite set of LOP symbols represented by the cluster centroids.

Similarly to human motion, the codebook is learned from MU segments of training sequences. For MU segments of test sequences, we first compute the distance d_l between the corresponding LOP feature and all the exemplar LOP. Finally, the labeling is done using the nearest rule.

5.4 Detection of repetitions and cycles

In a first step, we propose to use the decomposition of a sequence into MUs to detect some repeated gestures or cyclic motion within the sequence. This is achieved by analyzing and comparing shapes of MU trajectories. Such detection provides deeper analysis of a motion sequence and thus facilitates the human behavior understanding.

For instance, detecting cyclic movement allows to group them in the same cluster and thus segment a long sequence in distinct clusters representing different actions. In addition, we observed in the Chapter 4 that actions characterized by repetitions of a single gesture may not be easily recognized. As a result, detecting such repetitions may facilitate the recognition task.

5.4.1 Detection of periodic movement

In order to analyze MUs and detect periodic movement, we employ the human motion description introduced in Section 5.3.1. Hence, the elastic distance $d_{\mathcal{S}}$ is used to compare two MUs.

As MUs are not necessarily repeated successively, but instead periodically, we search for different length of periodicity. Let MU_i be the i-th MU of a sequence and q_i its corresponding shape on the shape space. We define $P(\omega, i)$ the periodicity value of length ω for the i-th MU as:

$$P(\omega, i) = \frac{1}{\omega} \sum_{f=i-\omega+1}^{i} \phi(MU_f, MU_{f-\omega}) \ , \tag{5.8}$$

where:

$$\phi(MU_i, MU_j) = \begin{cases} 1 & \text{if } d_{\mathcal{S}}([q_i], [q_j]) < \text{threshold} \\ 0 & \text{otherwise} \end{cases} \ . \tag{5.9}$$

If $P(\omega, i) = 1$, a periodicity of length ω is detected at the i-th MU. The threshold is used to separate similar MUs from dissimilar MUs.

In the following, we employs this periodic detection for two different tasks:

- Grouping periodic movements to segment a long sequence into distinct cyclic actions;
- Removing repeated gestures to facilitate the action recognition of the whole sequence.

5.4.2 Action segmentation

In the first case, a sequence contains successive periodic actions, such as *walking*, *running*, *boxing*, etc. The periodicity of the action is an important characteristic that allows us to perform segmentation. For instance, the action *walking* is a succession of *left step* and *right step*.

Our method described in Section 5.2.2 allows the segmentation of the sequence in MUs corresponding to *left step* and *right step*. Once the segmentation is performed, we detect periodic MUs in order to group them in the same action cluster, e.g., *walking*. Detecting such periodic MUs along

the whole sequence and grouping them together result in a segmentation of the whole sequence into different clusters corresponding to distinct actions, as illustrated in Figure 5.9.

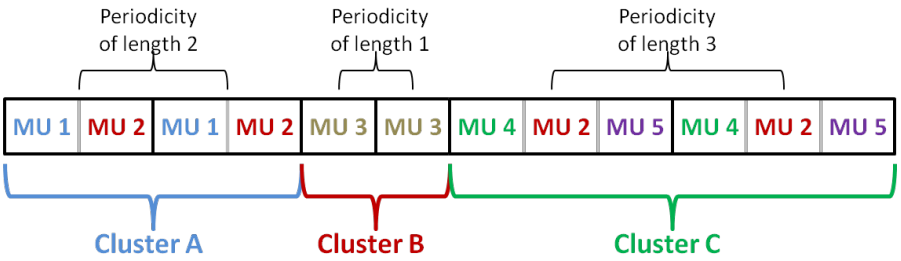


Figure 5.9 – *Clustering of periodic MUs.*

5.4.3 Repetitions removal

In the second case, a sequence contains one single action possibly characterized by repetitions of a single gesture.

In Chapter 4, results demonstrated that the proposed approach is unable to manage repetitions within a sequence when only sequences with one gesture instance are learned.

The method described above allows us to detect such repetitions. During the analysis of the sequence, if repetitions are detected, only the first instance is kept to represent the sequence. As a result, every sequence from the training or test set contains only one instance of the gesture.

However, when repetitions are removed, we may loose the continuity of the action between the two remaining extreme parts of the sequence. In order to keep continuity, we consider the shape of the two extreme poses (ending pose of the first part, and starting pose of the second part), represented in the shape space for $n = 3$. Then, we estimate the deformation between these two poses using the geodesic path (Equation. 3.9). We discretize the path with a small number of steps representing the deformation between the two extreme poses. This process is illustrated in Figure 5.10. Note that, the removed part of the sequence is a repetition of a previously observed MU. Thus, the ending poses should not differ a lot.

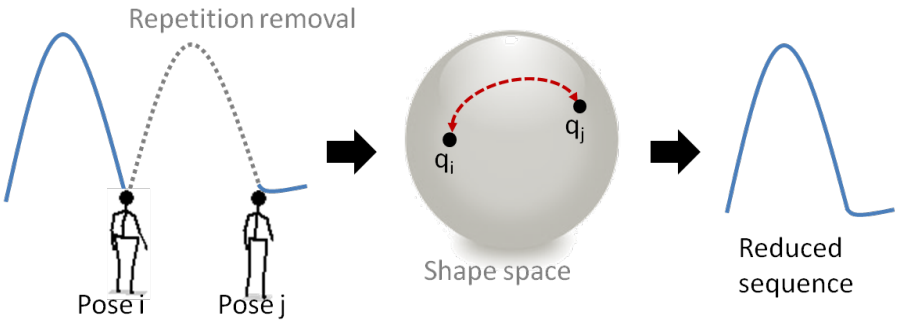


Figure 5.10 – *Removal of repeated MUs keeping continuity of the action sequence. Deformation between the two extreme poses $Pose_i$ and $Pose_j$ is estimated using the geodesic path on the shape space.*

5.5 Experimental evaluation of periodic movement detection

In this section, we propose to evaluate the approach of detecting periodic movement and demonstrate its usefulness for two different tasks stated above: action segmentation and action recognition. First, we evaluate the resulted action segmentation based on the detection of periodic MUs. Second, we analyze how the removal of repeated MUs improve our previous action recognition approach described in Chapter4.

The experiments are performed on two datasets, which provide data of two types: motion capture (mocap) data in the CMU dataset [18]; skeleton data captured with Microsoft Kinect in the MSR Action 3D [46].

5.5.1 Action segmentation

We evaluate the performance of our approach for the task of action segmentation using samples of the CMU dataset and compare it with the method proposed in [111] called HACA. Similarly to [111], we use 14 sequences performed by the subject #86.

We evaluate the resulted segmentation in comparison with the ground truth by computing the confusion matrix between the segmentation obtained with our method and the ground truth. Then, we use the same metric used in [111] to compute the segmentation accuracy. Figure 5.11 shows the segmentation accuracy for the 14 sequences compared to [111].

It can be observed that we obtain competitive accuracies compared to

HACA. Note that, a single sequence can include the same action several times at several time intervals, like *walking*. With our method, if a second instance of the same action happens, we view it as a new cluster. In order to handle this characteristic and be comparable with HACA, we assign the same label to similar clusters using the distance d_S. Indeed, each cluster can be viewed as a longer motion trajectory. The distance d_S is computed between shapes of two cluster trajectories. If the distance is below a threshold, the same label is assigned to the two clusters.

We note that without this constraint, our approach segments a sequence in an online way parcouring only once the sequence with the sliding window method. In comparison, the offline method proposed in [111] needs a first initialization of the segmentation and then performs optimization in several iterations.

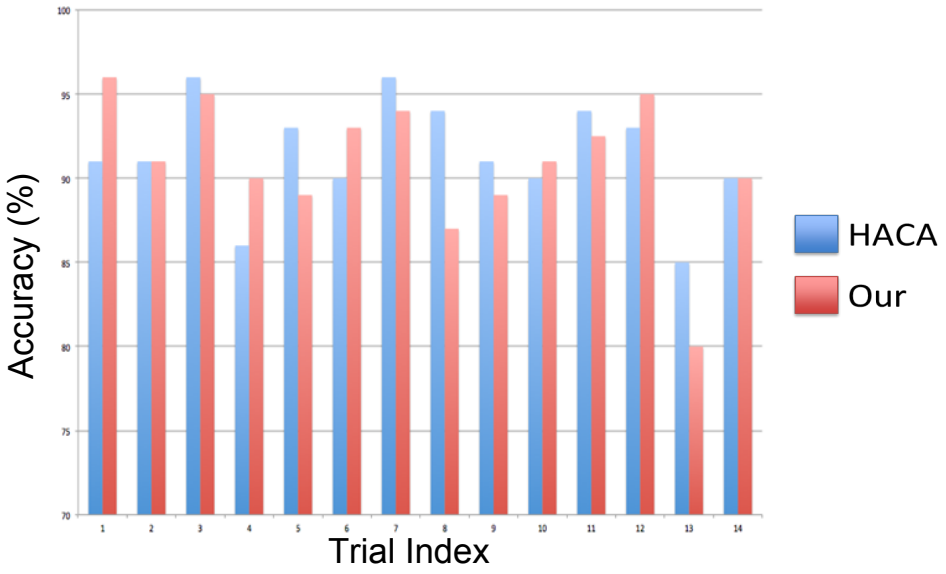


Figure 5.11 – *CMU dataset: Segmentation accuracy for 14 sequences.*

Figure 5.12 shows the segmentation results of the fourth sequence obtained by HACA [111] (second row) and our method (third row), in comparison with ground truth segmentation (first row). Different colors correspond to different actions. The white bars within the same color represent the detected periodic movements. For instance, the first action in red (walking) is composed of five movements, each representing a walk cycle (one left step and one right step).

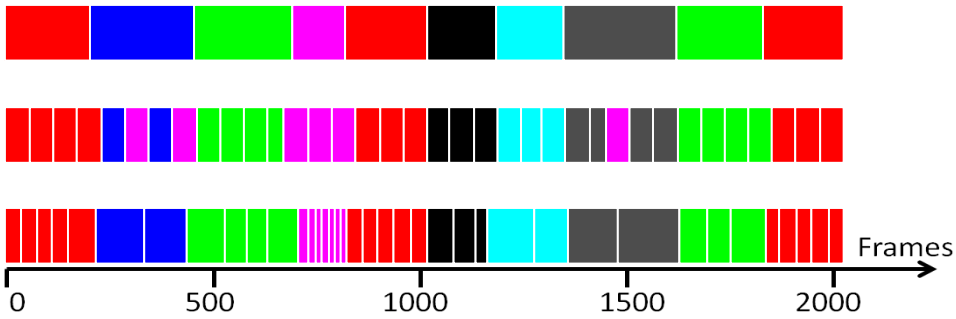


Figure 5.12 – *Segmentation results obtained for a sequence. 1st row corresponds to ground truth, 2nd row to HACA [111] and 3rd row to our approach.*

Table 5.1 – *MSR Action 3D: Comparison of the proposed approach with the most relevant state-of-the-art methods. Our previous method described in Chapter 4 is refereed as Devanne et al. [24].*

Method	Accuracy (%)
Actionlet [94]	88.2
DCSF [98]	89.3
JAS & HOG^2 [62]	**94.8**
HON4D [64]	88.9
Moving Pose [108]	91.7
Devanne et al. [24]	92.1
Our	**94.3**

5.5.2 Action recognition

We demonstrate the usefulness of our approach to improve the action recognition of our previous method, described in Chapter 4, evaluated on the MSR Action 3D dataset. We observed that if an action is repeated more than once within a sequence, it affects the shape of the corresponding trajectory, and thus the accuracy of the action recognition.

We use the proposed segmentation approach to detect and remove such repetitions within a sequence. The overall accuracy is increased from 92.1% to 94.3%. However the experiments in Chapter 4 demonstrated that only one action class among 20 was mainly affected by this repetition variability (*hammer*). Table 5.1 shows that compared to state-of-the-art's method, such improvement allows us to obtain competitive accuracy.

In addition, the confusion matrix in Figure 5.13 shows that, in comparison with the previous approach, the proposed method allows us to improve the recognition of the action *hammer* without affecting the recognition of the other classes. Nevertheless, this improvement is not as high

as expected (33.3%). This shows that even if only one instance of the gesture is kept, this action remains similar to other like *high throw* or *draw tick*.

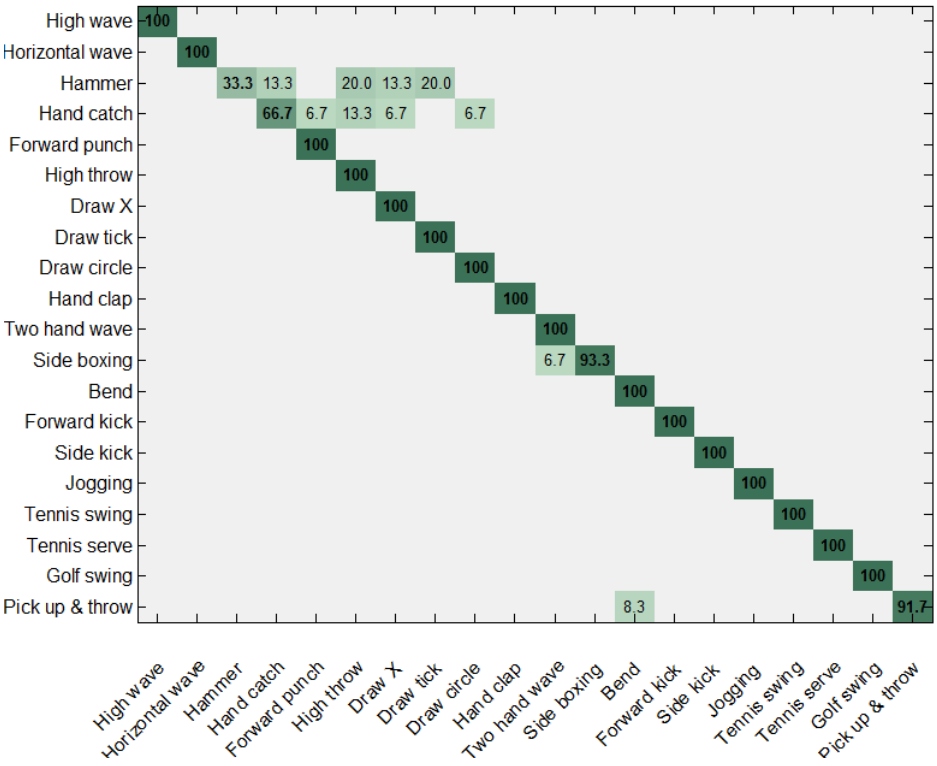


Figure 5.13 – *Confusion matrix obtained by removing repeated gestures on MSR Action 3D. We can notice a higher accuracy for the action hammer without affecting accuracy of other action classes.*

5.6 Modeling complex motion sequences

So as to deepen the analysis and understanding of human behavior, we propose in a second step to represent and model complex motion sequences as a set of exemplar MUs using bag-of-words paradigm so as to identify codebooks of MUs.

As explained in Section 5.3.3, a sequence is decomposed into several MUs and each MU is described in terms of human motion and depth appearance around subject hands by its corresponding symbol from a learned codebook.

As a result, each sequence can be viewed as a combination of two sequences of successive symbols, one corresponding to human motion

extracted from skeleton data, and the other corresponding to depth appearance around hands. Such temporal evolution of symbols represents the dynamic of the action or activity.

In so doing, we assume that actions or activities of the same class are represented with similar arrangements of MUs. Conversely, different sequences of symbols should represent actions or activities of different classes. Hence, we need a methodology allowing us to analyze the temporal evolution of symbols and recognize different arrangements of MUs representing different actions or activities.

To this end, we propose to use the Dynamic Naive Bayesian classifier (DNBC) [53] as statistical model.

5.6.1 Dynamic naive bayesian classifier

In the literature, Hidden Markov Models (HMM) have been widely used to statistically represent the dynamics of a process [69, 39]. Their effectiveness has been proved for human behavior recognition from RGB videos [82, 3] or depth data [25, 99], where the human poses or human motion are represented by only one single observation of attributes. In our case, sequences are represented by two channels of attributes. Several works have been proposed to deal with multiple attributes by mixing them. As a result, one observation is the conjunction of several attributes and thus implies conditional dependence among attributes given the state.

In order to face this issue, we propose to use DNBC, which can be viewed as an extension of HMMs. The main difference is that the observation node is decomposed into several attributes that can be considered as independent given the state. This independence assumption is particularly suitable for our problem, where we have two different attributes describing one MU. Different strategies of fusing multiple features from three different modalities have also been studied in [67] where a hierarchical HMM is employed. Experiments carried out in [67] demonstrated that the intermediate fusion gives better results. Such fusion strategy is similar to DNBC as it considers each modality separately. As a result, an

observation is a set of modalities, similarly to our DNBC model where each observation is a set of attributes.

Figure 5.14 depicts an example of a DNBC with two attributes per state.

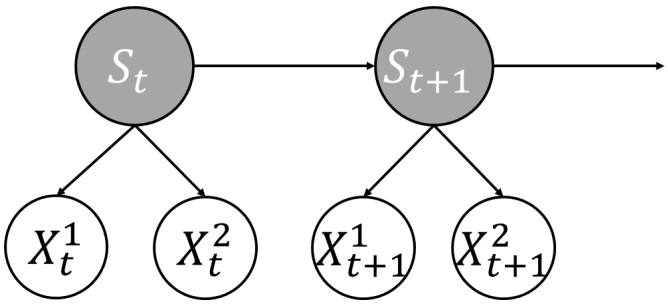


Figure 5.14 – *Example of Dynamic Naive Bayesian Classifier with two time intervals (t and $t+1$). For each time interval t, the process is at state S_t and emits the two attributes X_t^1 and X_t^2. In our case, the two attributes X^1 and X^2 correspond to human motion and depth appearance, respectively.*

In order to model the dynamics of the process describing activity sequences with T MU (time intervals), we first consider a realization of the states of this process as $S = \{S_t | t = 1, \ldots, T\}$. Let N_s be the number of possible states of the process, $1 \leq S_t \leq N_s$. The sequence of observations is denoted as $X = \{X_t^a \mid t = 1, \ldots, T \ , \ 1 \leq a \leq A\}$, where A is the number of attributes. In our case $A = 2$, with $a = 1$ and $a = 2$ corresponding to the human motion and depth appearance attribute, respectively. Each attribute X_t^a takes a finite discrete value k with $1 \leq k \leq K^a$, from its corresponding learned codebook V^a of size K^a. Hence, the joint probability function of the DNBC is:

$$P(X, S) = P(S_1) \prod_{t=1}^{T-1} P(S_{t+1}|S_t) \prod_{t=1}^{T} \prod_{a=1}^{A} P(X_t^a|S_t) \ , \tag{5.10}$$

where $P(S_1)$ is the prior probability specifying the probability distribution over the initial state (at $t = 1$); $P(S_{t+1}|S_t)$ is the transition probability between states S_t and S_{t+1} defined by a transition matrix of size $N_s \times N_s$; $P(X_t^a|S_t)$ is the emission probability corresponding to the probability function of the observed attribute a at time t in the state S_t defined by an observation matrix of size $N_s \times K^a$.

The main difference between DNBC and HMM is the product

$\prod_{a=1}^{A} P(X_t^a|S_t)$ representing independence among attributes given the state. If $A = 1$, the resulted joint probability function is the same as for HMM.

In addition, as usual with HMM, we assume that our DNBC follows two standard properties: the Markov property and the stationary property: The first one establishes that future states of the process only depend on the current states, not on past states; The second one defines the transition probabilities among states are unchanged along the time.

5.6.2 Learning

Like in HMM applications, we only know X, while S is not available. Thus, for estimating the model parameters, i.e, the *prior*, *transition* and *emission* probabilities, some tools are desirable.

The prior probability represents the initial state of the process. The transition probability is the probability to transit from one state to another state of the process. The emission probability represents, for each state, the probability of generating each attribute.

Similarly to HMM, a common way to learn such parameters from training sequences of observed symbols is to use the Baum-Welch algorithm [10]. For the case of DNBC, the process of estimating parameters is slightly modified due to the model setting allowing the emission of several attributes per state (more details on this can be found in [8]).

For our task of activity recognition, we assume each activity class is modeled with a different DNBC. Let the activity class $c \in \{1, \ldots, C\}$ with C being the number of activity classes, we learn one DNBC denoted λ_c for each class c using the training sequences belonging to the class c.

5.6.3 Classification

Offline classification

The classification process of an observation sequence X is the following. First, the sequence is presented to each of the trained DNBC λ_c modeling different activity classes. Then, the likelihood $P(X|\lambda_c)$ that the sequence

X has been generated by the DNBC λ_c is computed using the Forward algorithm.

Finally, the sequence is classified as the activity whose corresponding DNBC gives the highest log-likelihood:

$$activity(X) = \underset{c}{\text{argmax}} \ \log(P(X|\lambda_c)) \ . \tag{5.11}$$

Online classification

The classification process is then extended to be able of working in an online manner. In that case, we do not need to wait for the end of the sequence to give a classification decision. This is particularly convenient for real-time purpose, so as to guarantee natural interaction with the system. In addition, it allows us to process a sequence as a continuous stream where several activities can be performed successively, which is often the case in a real-world context.

As shown in Section 5.2.2, the segmentation process is based on a sliding window technique. Hence, it can also be applied in an online manner, so as to detect MU segments from a continuous stream. Each new frame of the sequence is given as input to the segmentation process. When a MU is detected, we compute the corresponding human motion and depth appearance features and assign a symbol to each, as described in Section 5.3.3. The resulted observation sequence of length-1 is then presented to each trained DNBC in order to compute the corresponding log-likelihoods.

For each new detected MU, the same is performed. Thus, the length of the observation sequence is increased by one, and the log-likelihoods are updated. If the log-likelihood of a class falls below a threshold, we discard the activity class. This allows us to gradually reduce the set of possible classes. The process is repeated until all classes are discarded. Then, among the remaining classes in the previous iteration, we keep the class with the highest log-probability as the detected activity.

Transitions between activities are often smooth. Thus, when an activity is finished, its corresponding log-probability may not considerably decrease and directly fall below the threshold. In order to take into account

this smooth transition, we select as the ending boundary of the activity the time step when its corresponding log-probability starts to decrease instead of the time step when it falls below the threshold.

Finally, we restart the detection process from the successive time step using all the classes. This is repeated until the end of the sequence. As a result, we obtain the set of detected activities along the sequence with corresponding starting and ending boundaries. This online activity detection is illustrated in Figure 5.15.

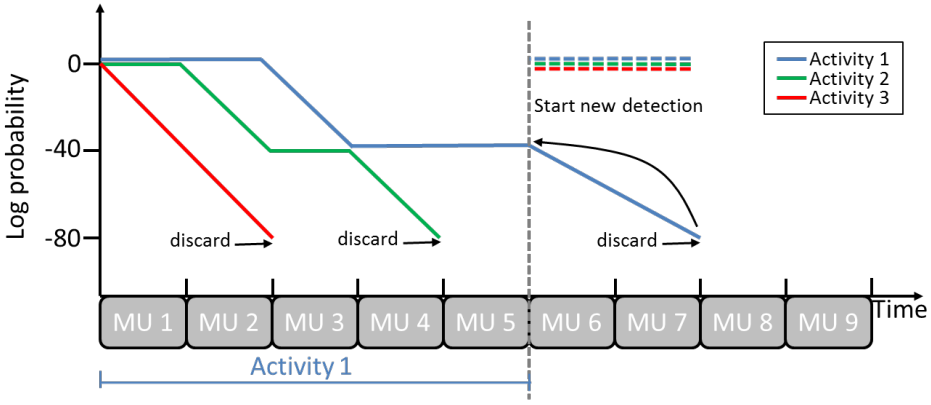


Figure 5.15 – *Online detection method. The Activity-2 and Activity-3 are discarded after the fourth and second time interval, respectively, as their log-probability fall below -80. The remaining Activity-1 is discarded after the seventh time interval. As a result, the five first time intervals are classified as Activity-1, and a new detection is started from the sixth time step.*

5.7 Experimental evaluation of modeling for recognition

We evaluate the proposed approach in comparison with state-of-the-art methods using four public benchmark datasets, each of which allows testing different tasks: human gesture recognition with repetitions; activity recognition; online detection of successive simple actions; and activities involving interactions with objects.

5.7.1 Gesture recognition

In a first time, we evaluate our method in the task of human gesture recognition. The main goal of this experiment is to show how the proposed

method deals with actions characterized by repetitions of a single gesture. In particular, we want to highlight that the proposed decomposition of a sequence into a set of MUs allows to handle such variability.

We perform this experiment on the Microsoft Research MSRC-12 [28] dataset, which includes sequences of subjects performing 12 gestures. Following the same protocol as in Lehrmann et al. [45], only six gestures are considered: *Duck*, *Goggles*, *Shoot*, *Throw*, *Change weapon* and *Kick*.

Each action is performed several times by 30 subjects, for a total of about 50 sequences per class. This results in 296 sequences of about 1000 frames each, where a single gesture is performed several times (10 times in most of the cases, but this number may vary from 2 to 15). This variability is indeed important to show how it can affect the recognition accuracy.

Only skeleton data are provided in this dataset, so we only use the motion features to describe each segment. As a consequence, the DNBC which models the sequences has only one attribute, thus becoming equivalent to a HMM.

In order to fairly compare our method to [45], we follow the same protocol and employ a 5-fold leave-one-person-out cross validation, where each fold consists of 24 persons for training and 6 persons for test. Results are reported in Table 5.2 as average accuracy of each fold in comparison to [45], and our previous method described in Chapter 4 (Devanne et al. [24]).

Table 5.2 – *MSRC-12. Comparison of the proposed approach with DFM [45] and [24]. Accuracies per class as well as mean accuracies are reported in percentage.*

Class	DFM [45]	Devanne et al. [24]	Our
Duck	96.0	**100**	**100**
Goggles	88.0	82.0	**91.6**
Shoot	**85.7**	73.5	83.0
Throw	**90.0**	88.0	**90.0**
Change weapon	87.5	**89.6**	**94.0**
Kick	**98.0**	**98.0**	**98.2**
Mean	90.9	88.5	**92.8**

From Table 5.2, we can notice that the proposed approach outperforms

the work in [45] for all gesture classes except one (*Shoot*), with an overall accuracy of 92.8%, compared to 90.9% reported in [45].

In addition, we can see that in comparison with our previous method [24], the overall accuracy is increased by about 4%. Analyzing the failure cases, we notice that the different number of repetitions in the sequences affects the accuracy of the previous method. This is for instance the case of similar actions, like *Goggles* and *Shoot*. If a test *Shoot* sequence includes a number of repetitions, which is not frequent in the training *Shoot* sequences, but frequent in the training *Goggles* sequences, it may be assigned to a *Goggles* sequence and thus poorly classified. Hence, the effectiveness of our previous method [24] is related to the number of gesture instances.

To emphasize this latter aspect, we run an experiment on the sequences that belong to the *Goggles* and *Shoot* classes only. In the training set, we include *Goggles* sequences with exactly 10 repetitions of the gesture, plus all *Shoot* sequences except those with exactly 10 repetitions of the gesture (this latter sequences are included in the test set).

We then evaluate the recognition accuracy of our previous method [24], and the approach presented in this Chapter. We observe that the recognition accuracy of the class *Shoot* is increased from 39.4% to 80.2%. This shows that our method allows us to improve the recognition accuracy when the number of repetitions of a single gesture can vary within a sequence.

Indeed, as we use DNBC to model the sequences, repetitions of gestures are characterized by repetitions of the process without changing the structure of the model. The same model is used to generate a sequence with any number of gesture instances, thus allowing robustness to repetition variability.

5.7.2 Activity recognition

In a second time, we propose to evaluate the approach for the recognition of more complex motion sequences like activities. The evaluation is per-

formed on two different datasets: Cornell Activity dataset 120 [41] and Online RGB-D dataset [106].

Cornell Activity 120 dataset

We use the Cornell Activity dataset 120 [41] (CAD120) to test our approach in the context of human activity recognition. This dataset contains 120 RGB-D sequences of ten high-level activities performed by four different subjects three times each. Activities involving objects manipulation are: *making cereal*, *taking medicine*, *stacking objects*, *unstacking objects*, *microwaving food*, *picking objects*, *cleaning objects*, *taking food*, *arranging objects* and *having a meal*. Activities belonging to the same class may be performed with different types of objects. For instance, the activity *stacking objects* may be executed with either *pizza boxes*, *plates* or *bowls*.

Three main challenges are identified for this dataset: 1) the variability in performing activities among subjects; 2) the orientation of the subjects in the scene: while subjects are always in the field of view of the camera, they are not always facing the camera; 3) manipulation with the environment results in occlusions of part of the body. This is especially the case for legs, which are often occluded by the table where activities are executed (to better face this problem, we use only the upper body skeleton to represent the human pose). Figure 5.16 illustrates these challenges by showing example frames of three sequences.

For a fair comparison with state-of-the-art methods, the *leave-one-person-out* cross protocol is used where, for each fold, one subject is used for test and the three remaining for training. The average accuracy among the four folds is finally computed.

Table 5.3 reports results obtained by our method in comparison to several state-of-the-art approaches. In particular, methods are compared by separating the case in which only the human skeleton is used, from the case in which both skeleton and depth data are considered. In our case, this results to learn a DNBC for each activity class using only one attribute (human motion) or two attributes (human motion and depth appearance),

(a) (b) (c)

Figure 5.16 – *Illustration of the challenges identified in CAD 120 dataset. We can observe that the two first activities (a) and (b), corresponding to the same class arranging object, are performed differently by the two subjects at different location and orientation with respect to the camera. In addition, the third activity (c) shows an example of legs occlusion resulting in a noisy skeleton.*

respectively. Further, our method is evaluated using both the MLOP and LOP4D depth appearance features.

Table 5.3 – *Cornell Activity dataset 120. Comparison between our approach and state-of-the-art methods.*

Method	Accuracy (%)
Skeleton Only	
Koppula et al. [41]	27.4
Devanne et al. [24]	48.3
Our	**69.4**
Skeleton + Depth	
Koppula et al. [41]	80.6
Koppula and Saxena [42]	**83.1**
Rybok et al. [73]	78.2
Our (Skel + MLOP)	79.0
Our (Skel + LOP4D)	**82.3**

From the results, we can first notice that our method significantly outperforms the other approaches when only skeleton data are used. More specifically, in comparison with our previous method described in Chapter 4 [24], which represents each activity by spatio-temporal trajectory only, the recognition accuracy is improved by more than 20%. This shows that when activities involve complex motions, it is not sufficient to analyze

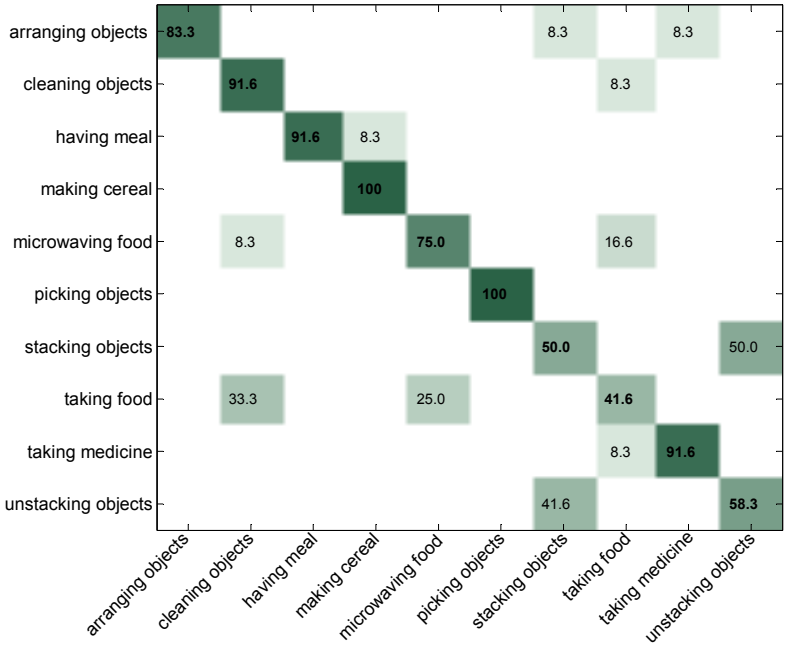
the global motion. Indeed, local analysis and decomposition of the activity into MUs provides a better representation of activities, thus allowing a better understanding of the human behavior.

In addition, the accuracy of 69.4% obtained by our method using only skeleton data shows that the decomposition of the sequence allows us to quite well recognize activity sequences involving objects manipulation, even without describing any explicit information about objects held by the subject.

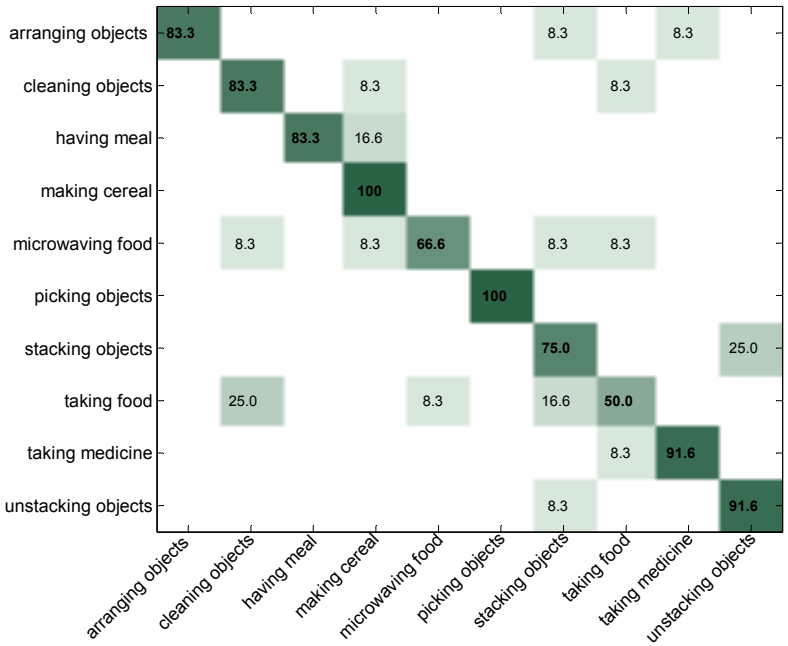
However, results show that using the skeleton only is not sufficient to be competitive with state-of-the-art methods. As we can see in Table 5.3, using depth appearance features in addition to skeleton in our DNBC allows us to improve the recognition by about 13%. As a result, we obtain competitive accuracy in comparison with other approaches. Indeed, only [42] is above by less than 1%. Note that methods in [41] and [42] use ground truth object bounding box in the training process. In our case, we do not need this information.

Finally, by comparing the results obtained with our two different depth appearance features, we can notice that analyzing depth appearance evolution along short intervals allows us to better characterize interactions with objects. This analysis is strengthened by analyzing confusion matrices in Figure 5.17. We compute the confusion matrices obtained by our method for both depth appearance features.

It can be noticed that our method scores a lot of confusion for the pair of activities *stacking objects* and *unstacking objects*. As these activities are opposite, they may be represented by the same sequence of MUs. The variation of depth appearance around subject hands may differentiate these activities. Indeed, when a subject takes an object or puts an object, its motion is similar but the depth appearance around joints may be different. We can see that using the LOP4D feature results in less confusion between the two activities than using the MLOP feature. Indeed, in this particular case, the average depth appearance of *putting the object* and *taking the object* may be very similar and represented by the same symbol from the codebook. The 4DLOP feature capturing the variation of depth



(a)



(b)

Figure 5.17 – *Confusion matrices obtained with our method on CAD-120 using MLOP (a), and 4DLOP (b).*

appearance is more suitable to discriminate the two elementary motions, and thus the two activities.

Additionally, we can see the activities *cleaning objects*, *microwaving food* and *taking food* are confused. These three activities are very similar as they involve similar elementary motion, like *open the microwave*, *take or put an*

object and *close the microwave*. Using our method, we are not accurately able of differentiating when the subject put the food or take it from microwave and when s/he is cleaning the microwave.

On this dataset, we also evaluate and analyze the effectiveness of our method when the value of parameters (size of the codebook and number of states used in DNBC) is changed. The evolution of the accuracy with respect to both parameters is displayed in Figure 5.18 for both MLOP and LOP4D features.

First, it can be observed the proposed method obtains the best accuracy using both features, when a DNBC with 10 states is trained. It can be also observed the accuracy is relatively independent from the number of states (except when only three states are used).

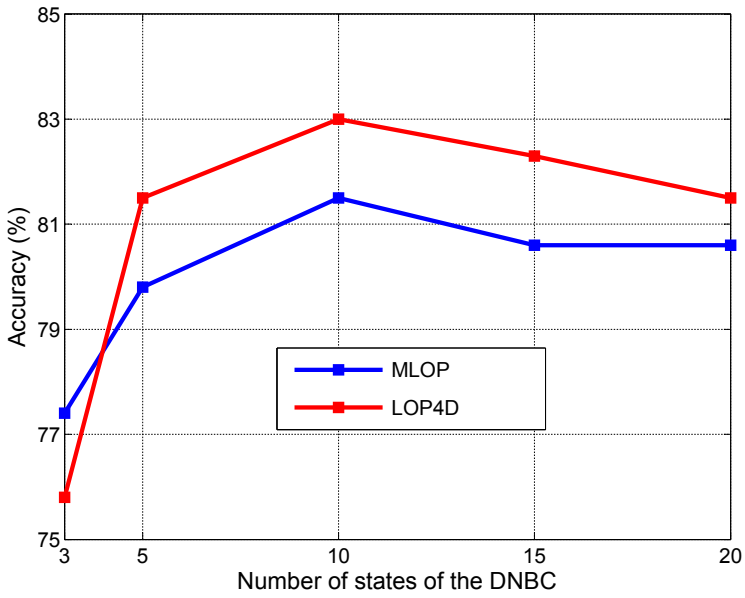
Second, we can notice the best accuracy is obtained with a codebook of size 50 for the MLOP feature, and a codebook of size 100 for the LOP4D feature. In addition, if too much exemplar features (i.e., 200) are used, we observe the accuracy falls down. Indeed, learning a codebook with too much symbols may result in similar activities represented by different symbols. Hence, sequences only represent a particular sequence performed by one subject, rather than a generic template of one activity class.

Online RGB-D dataset

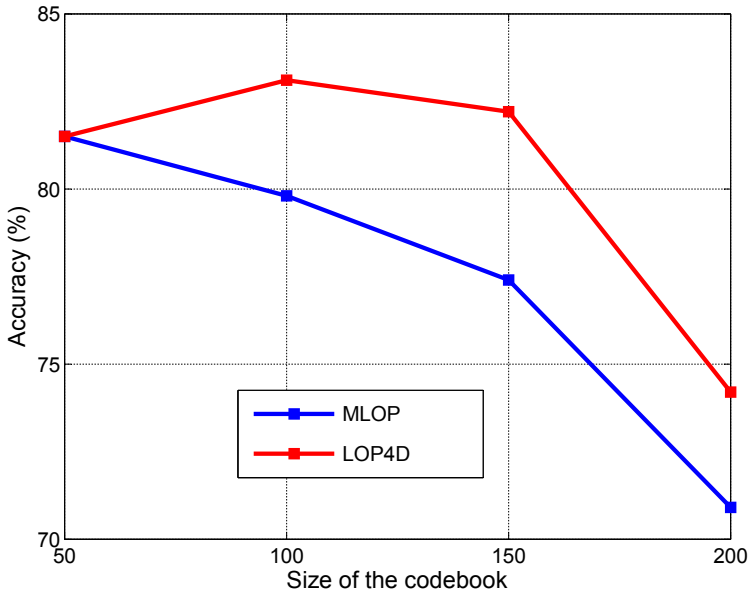
The Online RGB-D dataset [106] is interesting because it proposes different types of sequences, which allow evaluation in different contexts, like activity recognition and online activity detection. The dataset contains RGB-D sequences of seven activities: *drinking*, *eating*, *using laptop*, *picking up phone*, *reading phone (sending SMS)*, *reading book* and *using remote*.

On this dataset, we first evaluate the effectiveness of our method for activity recognition. To this end, we use a first set of sequences, where one single activity is performed by sixteen different subjects. For a fair comparison, we follow the same procedure as in [106]. Half of the subjects are used for training and the other half for test. Then, a 2-fold cross validation is used and the mean accuracy of the two folds is computed.

We propose to compare our approach with state-of-the-art methods



(a)



(b)

Figure 5.18 – *Accuracy evolution of our method with respect to varying parameters: the number of states of DNBC (a), and the size of the codebooks (b).*

and differentiate between approaches that use skeleton features only or depth features only, and approaches that use both skeleton and depth features.

We model the activities by DNBC with one or two attributes according to the number of features we use. When we use depth features within

our method, we use the 4DLOP feature and learn a codebook of size 100. Results are reported in Table 5.4. We note that the dataset also provides object location used for training in the work [106]. In our method, we do not need such information.

Table 5.4 – *Online RGB-D dataset. Comparison of our approach with state of the art methods for the task of activity recognition*

Method	Accuracy (%)
Depth Only	
DSTIP + DCSF [98].	61.7
DOM [106]	46.4
Our	**64.5**
Skeleton Only	
EigenJoints [102]	49.1
Moving Pose [108]	38.4
DOM [106]	63.3
Our	**71.8**
Skeleton + Depth	
Actionlet [94]	66.0
DOM [106]	71.4
Our	**80.9**

By analyzing Table 5.4, it can be noticed that the proposed approach outperforms the state-of-the-art methods for every combination of features.

It should be also noted, in the case only depth features are used, our method is not fairly comparable to the others. In fact, our approach is based on a decomposition of activities into MUs, but this decomposition is computed using skeleton information. Thus, even if we only use depth features to describe MUs, our method still needs skeleton data in order to identify MUs. Hence, we are not only using depth features, like the method in [98]. Nevertheless, we can see that our segmentation approach allows a good recognition of activities when each segment is only described by depth appearance feature.

Compared to skeleton-based methods, our proposed approach significantly outperforms other solutions. This shows that our segmentation approach combined with shape analysis of the human motion allows us

to efficiently recognize activities involving manipulation of objects. Even without considering any information about objects held by the subject, we are able to recognize 71.8% of the activities. This accuracy is higher than for methods in [94] and [106], which combine both skeleton and depth features.

Finally, if we add depth features to the skeleton, the recognition accuracy is significantly increased to 80.9%, which is almost 10% above the best state-of-the-art method [106].

We evaluate also the latency of our approach in comparison to stat-of-the-art methods. The latency measures the ability of recognizing the activity without observing the whole sequence. Hence, the average recognition accuracy is computed on different observed portions of the activity sequence. The evolution of accuracy is reported in Figure 5.19 in comparison to state-of-the-art methods.

It can be noticed that the proposed approach outperforms the two methods in [98] and [108] for every observation sequence. However, we can see that the accuracy of our method exceeds the method [106] only from 40% of observation. Indeed, when we only observe 10%, 20% or 30% of the sequence and we apply our segmentation method, it often results in activity sequences represented by one or two temporal segments. In these cases, the dynamic of the activity is null (one observation) or very small (two observations). Hence, the use of statistical models, like DNBC is not appropriate and efficient for modeling short portions of the activity sequence. This is the main reason motivating lower accuracy of our approach with respect to [106]. When more information is available, we are able to capture and model the dynamic of the activity.

Finally, we can see that our method allows efficient recognition (accuracy of 75.6%) when only half of the sequence is observed. When 60% of the sequences is observed, we obtain an accuracy of 79.5%, which is close to the accuracy of 80.9% obtained for the whole sequence. This shows that even if our method is not suitable for very early detection of activities (less than 30% of observation), we are able of guaranteeing a good recognition accuracy when only half of the sequence is observed.

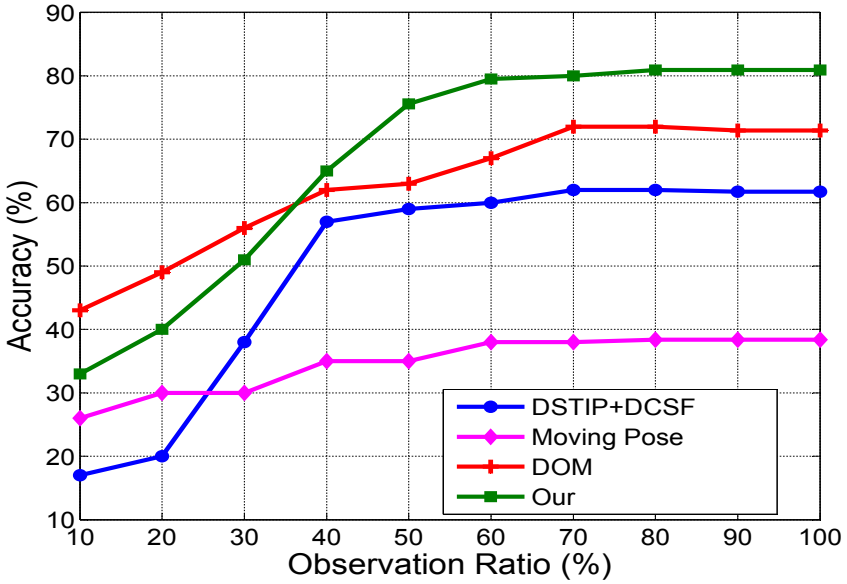


Figure 5.19 – *Latency analysis on Online RGB-D dataset. Accuracy obtained for different portion of the sequences is compared for our approach and state of the art methods.*

5.7.3 Online detection of actions/activities

In a last time, we propose to evaluate the capability of our approach for online detection of human behaviors in a long motion sequence. The evaluation is performed on the Multi-Modal Action Detection dataset [34] for action detection and on an extended version of the Online RGB-D dataset [106] for activity detection.

Multi-Modal Action Detection

We evaluate our method on the Multi-Modal Action Detection (MAD) database [34], for the task of online detection. This RGB-D dataset has the advantage over most available human action datasets of including not isolated clips.

Indeed, previous datasets provide only distinct sequences of actions. As a consequence, the online detection capability in these datasets, can be evaluated only by manually concatenating isolated clips. This results in discontinuities between actions (i.e., transitions are not naturally smooth), which does not correspond to a real case. Thanks to the MAD dataset pro-

viding long sequences of different actions, the online detection efficiency can be evaluated in a more realistic scenario.

In particular, the MAD database contains 40 sequences of 20 subjects (2 sequences per subject) performing 35 actions in each sequence. The length of the sequences is from 2 to 4 minutes. The 35 actions include full-body motion (*Running*, *Crouching*), upper-body motion (*Throw*, *Baseball swing*), and lower-body motion (*Kicking*). Each subject performs the 35 actions continuously, and the segments between two actions are considered as the null class.

Since actions are performed without objects, and for a fair comparison with state-of-the-art-methods, we use skeleton data only in these experiments. Hence, sequences are described only with human motion features and represented by models with one attribute. A five-fold-cross-validation over the 20 subjects (4 subjects per fold) is used as evaluation protocol. In each iteration, the labeled sequences of four folds are used to build the vocabulary of motion units and train the DNBCs. We used the ground truth segmentation in order to separate each action of the training sequences and learn one DNBC per action. One model corresponding to the null class is also learned from transition intervals when the human is standing.

Our method is run in an online way as described in Section 5.6.3. As a result, we obtain a segmented sequence with an action label for each temporal segment corresponding to the action we detected. In order to evaluate the method and compare it with the state-of-the-art, we compute two measures:

- *Precision*, which corresponds to the percentage of correctly detected actions over all the detected actions;
- *Recall*, that is the percentage of correctly detected actions over all the ground truth actions.

A detected action is considered as correctly detected if it overlaps with 50% of the segments of the ground truth action. We compare these two measures with the SMMED and MSO-SVM methods, both proposed

in [34]. The results are reported in Table 5.5. We can see that our method outperforms the state-of-the-art approaches for both the measures.

Table 5.5 – *MAD. Comparison of the proposed online detection approach with SMMED [34] and MSO-SVM [34]. The precision and recall measures are computed with the assumption that a detected action is correct if it overlaps with* 50% *of the ground truth segment.*

Measure (%)	MSO-SVM [34]	SMMED [34]	Our
Recall	51.4	57.4	**79.7**
Precision	28.6	59.2	**72.1**

Figure 5.20 also shows the detection results of one sequence in comparison with the ground truth and the best state of the art method, SMMED, proposed in [34].

We can see that the duration of the detected actions by our method is more similar to the ground truth compared to [34]. While both our method and [34] are able to accurately detect actions along the time, our method detect more efficiently the end of actions resulting to duration of detected actions closer to the ground truth. With the approach proposed in [34], most of the actions are well detected, but in most of the cases these correspond to a shorter time interval than the real one. As an overlap of 50% with ground truth is considered as the criterion of good detection, our method obtains higher values of recall and precision.

Online RGB-D dataset

Finally, we propose to evaluate our approach for online activity detection on the Online RGB-D dataset [106]. The same set of activities as for activity recognition is used to train one DNBC for each activity class. In addition, we use a set of background activities provided by the dataset, so as to learn the null class.

We run our detection method on a new set of sequences. It includes 36 long sequences from 30 seconds to two minutes, where 12 new subjects are successively performing different activities. In total, 123 activities are performed during these 36 sequences, with an average percentage of background frames of about 30%. Manual labeling provided by the dataset is used as ground truth.

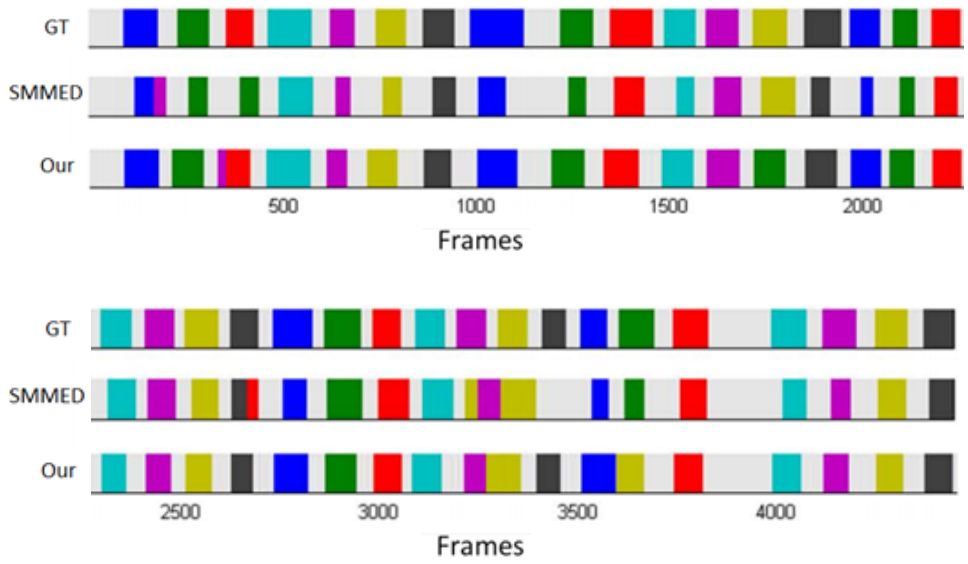


Figure 5.20 – *Action detection result for the sequence-1 of subject-1 from the MAD database. The full sequence is divided in two parts to guarantee a clearer visualization. The first row corresponds to the ground truth, the second row to the SMMED method [34], and the third row to the proposed method. We can see that our method provides segments whose duration is closer to ground truth compared to [34].*

Differently to experiments on the MAD dataset, detection is evaluated using a frame-level accuracy as in [106], which is computed by averaging the number of well classified frames out the all set of frames in the test sequences.

Results are reported in Table 5.6. Using this metric, a detected activity overlapping 50% with the ground truth has a score of 50% and is not considered as entirely well detected conversely to the metric used in experiments on MAD dataset. We can see that our method performs better than state-of-the-art approaches to detect activity in an online manner.

Table 5.6 – *Online RGB-D Dataset. Comparison of our approach with state of the art methods for the task of online activity detection.*

Method	Accuracy (%)
EigenJoints [102]	23.6
DSTIP + DCSF [98]	32.1
Moving Pose [108]	50.0
DOM [106]	56.4
Our	**60.9**

5.8 Conclusion

In this Chapter, we propose an effective method for modeling and understanding different kind of human behaviors, like gestures, actions and activities.

We first propose a pose-based analysis in order to decompose a sequence into representative elementary motion units. In one hand, such temporal segments are represented as motion trajectories and interpreted in the shape space. Thanks to the Riemannian geometry of this manifold, shape analysis is performed, so as to identify representative shapes characterizing elementary motions. On another hand, we add depth appearance information in order to characterize possible object manipulation along temporal segments.

The combination of skeleton and depth data, as well as the modeling of the dynamic of the sequence of segments is done through a Dynamic Naive Bayesian Classifier.

Experiments on the MSRC-12 dataset [28] demonstrate that the proposed approach is capable of recognizing human gestures and improve our previous method described in Chapter 4 by handling gesture repetitions. Evaluations performed on the CAD-120 dataset [41] and ORGBD dataset [106] show that our method performs well for the task of human activity recognition.

Finally, we adapt our method to allow online detection in long sequences of several behaviors, which is an important challenge in real-world context. Evaluation on MAD dataset [34] and ORGB-D dataset [106] demonstrate that the proposed approach outperforms state-of-the-art methods for online action detection and online activity detection, respectively.

However, experiments also show that the online detection problem need to be more investigated and more specifically the early detection of behaviors. Indeed, our method is currently not able to detect human behaviors with a very short observation of the motion. Such early detection is an important challenge for real-world applications, so as to guarantee real-time and natural human-machine interaction.

6 Conclusion

Sommaire

6.1 Summary

In this thesis, we addressed the issue of human behavior understanding from RGB-D data, a widely investigated topic due to its large panel of potential applications. We differentiate the study of behaviors according to their complexity.

In a first time, we focused on the recognition of relatively simple behaviors, like actions. To this end, we employed skeleton data provided by RGB-D sensors, which represent the human body pose as a set of connected 3D joints for each frame of the sequence. In order to analyze the action performed by the subject during the sequence, we proposed a rotation and translation invariant representation of the skeleton sequence as a spatio-temporal trajectory in a high dimensional space. This representation simultaneously encodes the human pose shape and captures the dynamics of the human motion.

In order to analyze and compare such trajectories, we considered their shape within a Riemannian framework. Each shape is interpreted on a Riemannian manifold and an elastic metric provided by the framework, corresponding to the geodesic length between two elements, is used to compute shape similarity independently to the elasticity of trajectories. Hence, it allows the motion trajectory analysis to be robust to the execution speed of actions. Such elastic metric is employed within a kNN classifier to perform action recognition. The efficiency of the proposed method is verified on three benchmark datasets.

Experimental results demonstrated that we obtain competitive results in comparison with state-of-the-art approaches. In addition, we proposed to evaluate the latency of our solution on two datasets. Results shown that we are able to guarantee high action recognition accuracy by observing only half of the sequence. However, experiments also demonstrated that the proposed solution suffers from limitations when actions involve variable repetitions of a single gesture or manipulations of objects.

In a second time, we extended our study, so as to simultaneously handle these limitations and considered more complex behaviors, like activities. Hence, we proposed a segmentation method in order to decompose a

motion sequence into a set of short elementary motion called motion units (Mus). In so doing, we represented each 3D skeleton as a curve representing the human body. The shape of curves is then analyzed within the Riemannian shape analysis framework. By combining a sliding window technique and statistical tools on such manifold, we were able to analyze pose variations and thus identify different MUs. In order to describe and investigate these MUs, we employ the same idea used previously for action recognition by considering the shape of the spatio-temporal trajectory of each MU.

In a first step, we applied this MU decomposition so as to detect cyclic movements. Experiments demonstrated the usefulness of the decomposition for the task of action segmentation and action recognition. Indeed, such cyclic MUs are either grouped together for the task of action segmentation of long sequences or conversely removed for improving action recognition efficiency.

In a second step, we considered the sequence of such MUs for temporally modeling more complex behaviors, like activities. In order to also describe manipulation of objects characterizing activities, the depth appearance around subjects hands is considered in addition to human motion. Codebooks for both descriptors are then learned, so as to identify representative MUs. Finally, these exemplar MUs are used to represent a sequence, which is interpreted through a Dynamic Naive Bayesian classifier, so as to capture the dynamics and achieve human behavior recognition. Experiments on four representative datasets evaluated the potential of the proposed solution in different contexts including gesture or activity recognition. Competitive results in comparison with state-of-the-art methods are reported.

Finally, the challenge of online detection is addressed for long sequences where several behaviors are performed successively. Experimental results on two datasets demonstrated that the proposed model is able to efficiently detect behaviors in an online manner.

6.2 Limitations and future work

Experiments shown that the proposed solutions guarantee an acceptable behavior recognition rate from about 50% of observation of a sequence. However, for applications requiring an earlier recognition by the analysis of very few observations, the proposed solutions present some limitations. This challenge is very important for real-time purpose and constitute a future research direction that we would like to investigate.

In addition, our method employs a depth appearance description around hand joints to characterize human-object interactions. While it allows us to differentiate similar activities in terms of human motion, such method does not recognize the objects. Information about detected objects in the scene could add more useful knowledge about the context and the environment so as to achieve a deeper understanding of the behavior.

Finally, this study focus on the understanding and recognition of human behaviors. As future work, we would like to explore a deeper analysis of the human motion so has to obtain information on how two human movements are different in addition to how much. This could be particularly useful to asses the quality of a human motion with respect to a template motion. For instance, in a sport context, such quality assessment would allow a subject to train or improve a particular gesture using corresponding feedback of the system on where the gesture can be perfectible. The motion analysis could also be useful in the medical or security domain in order to detect movement deficiency, like abnormal gait, or uncommon behavior, like fall of people.

Bibliography

[1] M. F. Abdelkader, 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*, 115(3):439–455, 2011. (Cited pages 38 and 55.)

[2] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing over-complete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006. (Cited page 27.)

[3] M. Ahmad and S.W. Lee. Human action recognition using shape and clg-motion flow from multi-view image sequences. *Pattern Recognition*, 41(7):2237–2252, 2008. (Cited page 110.)

[4] K. Aitpayev and J. Gaber. Collision avatar (ca): Adding collision objects for human body in augmented reality using kinect. In *International Conference on Application of Information and Communication Technologies (AICT)*, Stuttgart, Germany, May 2012. (Cited page 11.)

[5] A.Kurakin, Z. Zhang, and Z. Liu. A real-time system for dynamic hand gesture recognition with a depth sensor. In *European Signal Processing Conference (EUSIPCO)*, Bucharest, Romania, August 2012. (Cited page 18.)

[6] A.Srivastava, S. Joshi, W. Mio, and X. Liu. Statistical shape analysis: Clustering, learning, and testing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):590–602, April 2005. (Cited page 48.)

[7] ASUS Xtion PRO LIVE. `http://www.asus.com/Multimedia/Xtion_PRO/`, 2013. (Cited pages 9, 13, and 14.)

[8] H.H. Avilés-Arriaga, L.E. Sucar-Succar, C.E. Mendoza-Durán, and L.A. Pineda-Cortés. A comparison of dynamic naive bayesian classifiers and hidden markov models for gesture recognition. *Journal of Applied Research and Technology*, 9(1):81–102, Apr 2011. (Cited page 112.)

[9] B. I. Barbosa, M. Cristani, A. Del Bue, L. Bazzani, and V. Murino. Re-identification with rgb-d sensors. In *ECCV First International Workshop on Re-Identification*, Florence, Italy, October 2012. (Cited page 18.)

[10] L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970. (Cited page 112.)

[11] R. Bellman and R. Kalaba. On adaptive control processes. *IRE Transactions on Automatic Control*, 4(2):1–9, 1959. (Cited page 27.)

[12] R. E. Bellman and E. D. Stuart. Applied dynamic programming. In *RAND Corporation*, 1962. (Cited pages 42 and 79.)

[13] S. Berretti, A. Del Bimbo, and P. Pala. Superfaces: A super-resolution model for 3D faces. In *Proc. Work. on Non-Rigid Shape Analysis and Deformable Image Alignment*, pages 73–82, Florence, Italy, Oct. 2012. (Cited page 14.)

[14] W. Bian, D. Tao, and Y. Rui. Cross-domain human action recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(2):298–307, April 2012. (Cited pages 12 and 17.)

[15] V. Bloom, V. Argyriou, and D. Makris. G3di: A gaming interaction dataset with a real time detection and evaluation framework. In *Workshop of European Conference on Computer Vision (ECCV)*, Zurich, Swiss, September 2014. (Cited page 18.)

[16] E. Bondi, L. Seidenari, A.D. Bagdanov, and A. Del Bimbo. Real-time people counting from depth imagery of crowded environments. In

International Conference on Advanced Video and Signal Based Surveillance (AVSS), Seoul, South Korea, August 2014. (Cited page 12.)

[17] I. Borg and P.J.F. Groenen. *Modern Multidimensional Scaling: theory and applications (2nd ed.)*. New York: Springer-Verlag, 2005. (Cited page 70.)

[18] Carnegie Mellon University Motion Capture Database. `http://mocap.cs.cmu.edu`, 2012. (Cited pages 13 and 106.)

[19] A. A. Chaaraoui, P. Climent-Perez, and F. Florez-Revuelta. A review on vision techniques applied to human behaviour analysis for ambient-assisted living. *Expert Systems with Applications*, 39(12):10873–10888, 2012. (Cited page 17.)

[20] C.-Y. Chang, B. Lange, M. Zhang, S. Koenig, P. Requejo, N. Somboon, A. Sawchuk, and A. Rizzo. Towards pervasive physical rehabilitation using microsoft kinect. In *In Proceedings of the 6th International Conference on Pervasive Computing Technologies for Healthcareon*. (Cited page 12.)

[21] X. Chen and M. Koskela. Skeleton-based action recognition with extreme learning machines. *Neurocomputing*, 149:387–396, 2013. (Cited page 17.)

[22] Florence 3D Action dataset. `http://www.micc.unifi.it/vim/datasets/3dactions`, 2012. (Cited pages 17, 18, and 72.)

[23] J. W. Davis and A. F. Bobick. The representation and recognition of action using temporal templates. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Juan, USA, June 1997. (Cited page 28.)

[24] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. Del Bimbo. 3D human action recognition by shape analysis of motion trajectories on riemannian manifold. *IEEE Trans. on Cybernetics*, 45(7):1340–1352, 2014. (Cited pages 108, 115, 116, and 118.)

[25] A. Dubois and F. Charpillet. Human activities recognition with rgb-depth camera using hmm. In *International Conference of the IEEE Engineering in Medicine and Biology Society*, Osaka, Japan, July 2013. (Cited page 110.)

[26] V. Elangovan, V.K. Bandaru, and A. Shirkhodaie. Team activity analysis and recognition based on kinect depth map and optical imagery techniques. In *Signal Processing, Sensor Fusion, and Target Recognition XXI*, Baltimore, USA, April 2012. (Cited page 12.)

[27] C. Ellis, S. Z. Masood, M. F. Tappen, J. J. La Viola Jr., and R. Sukthankar. Exploring the trade-off between accuracy and observational latency in action recognition. *Int. Journal on Computer Vision*, 101(3):420–436, 2013. (Cited pages 17, 18, 27, 83, 84, and 85.)

[28] S. Fothergill, H. M. Mentis, P. Kohli, and S. Nowozin. Instructing people for training gestural interactive systems. In Joseph A. Konstan, Ed H. Chi, and Kristina Höök, editors, *CHI*, pages 1737–1746. ACM, 2012. (Cited pages 17, 18, 115, and 129.)

[29] S. Gasparrini, E. Cippitelli, S. Spinsante, and E. Gambi. A depth-based fall detection system using a kinect sensor. *Sensors*, 14(2):2756–2775, 2014. (Cited page 18.)

[30] S. Hadfield and R. Bowden. Kinecting the dots: Particle based scene flow from depth sensors. In *Proc. Int. Conf. on Computer Vision*, pages 2290–2295, Barcelona, Spain, Nov. 2011. (Cited page 14.)

[31] N. Hadjiminas and C.H.T Child. Be the controller: A kinect tool kit for video game control - recognition of human motion using skeletal relational angles. In *International Conference On Computer Games, Multimedia And Allied Technology (CGAT)*, Bali, Indonesia, May 2012. (Cited page 11.)

[32] J. Han, L. Shao, D. Xu, and J. Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE Transactions on Cybernetics*, 43(5):1318–1334, 2013. (Cited page 14.)

[33] M. T. Harandi, C. Sanderson, A. Wiliem, and B. C. Lovell. Kernel analysis over Riemannian manifolds for visual recognition of actions, pedestrians and textures. In *Proc. IEEE Work. on the Applications of Computer Vision*, WACV'12, pages 433–439, Washington, DC, USA, 2012. IEEE Computer Society. (Cited page 53.)

[34] D. Huang, Y. Wang, S. Yao, and F. De la Torre. Sequential max-margin event detectors. In *European Conference on Computer Vision (ECCV)*, Zurich, Swiss, September 2014. (Cited pages 17, 18, 33, 125, 127, 128, and 129.)

[35] Z. Jiang, Z. Lin, and L. S. Davis. Learning a discriminative dictionary for sparse coding via label consistent k-svd. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, USA, June 2011. (Cited page 27.)

[36] S. H. Joshi, E. Klassen, A. Srivastava, and I. Jermyn. A novel representation for Riemannian analysis of elastic curves in R^n. In *Proc IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 1–7, Minneapolis, MN, USA, June 2007. (Cited pages 39, 40, 41, and 42.)

[37] S.B. Kang and K. Ikeuchi. Temporal segmentation of tasks from human hand motion. Technical report, Carnegie Mellon University, April 1993. (Cited page 94.)

[38] S.B. Kang and K. Ikeuchi. Determination of motion breakpoints in a task sequence from human hand motion. In *IEEE International Conference on Robotics and Automation*, San Diego, USA, May 1994. (Cited page 94.)

[39] S. Karaman, J. Benois-Pineau, V. Dovgalecs, R. Mégret, J. Pinquier, R. André-Obrecht, Yann Gaëstel, and J. Dartigues. Hierarchical hidden markov model in detecting activities of daily living in wearable videos for studies of dementia. *Pattern Recognition*, 41(7):2237–2252, 2008. (Cited page 110.)

[40] H. Karcher. Riemannian center of mass and mollifier smoothing.

Comm. on Pure and Applied Math., 30:509–541, 1977. (Cited pages 45, 63, and 94.)

[41] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *International Journal of Robotics Research*, 32(8):951–970, Jul 2013. (Cited pages xi, 17, 18, 31, 32, 89, 117, 118, 119, and 129.)

[42] H. S. Koppula and A. Saxena. Learning spatio-temporal structure from rgb-d videos for human activity detection and anticipation. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, Atlanta, USA, June 2013. (Cited pages 32, 89, 118, and 119.)

[43] B. Kwolek and M. Kepski. Human fall detection on embedded platform using depth maps and wireless accelerometer. *Computer Methods and Programs in Biomedicine*, 117(3):489–501, 2014. (Cited page 18.)

[44] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, New York, USA, June 2006. (Cited page 22.)

[45] A.M. Lehrmann, P.V. Gehler, and S. Nowozin. Efficient nonlinear markov models for human motion. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1314–1321, Columbus, OH, USA, June 2014. (Cited pages 115 and 116.)

[46] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3D points. In *Proc. Work. on Human Communicative Behavior Analysis*, pages 9–14, San Francisco, California, USA, June 2010. (Cited pages xi, 17, 18, 20, 21, 52, 67, and 106.)

[47] I. Lillo, A. Soto, and J. C. Niebles. Discriminative hierarchical modeling of spatio-temporally composable human activities. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Columbus, USA, june 2014. (Cited pages xi, 18, and 31.)

[48] L. Liu and L. Shao. Learning discriminative representations from RGB-D video data. In *Proc. of the Twenty-Third Int. Joint Conf. on Artificial Intelligence*, IJCAI'13, pages 1493–1500. AAAI Press, 2013. (Cited page 28.)

[49] L. Liu, L. Shao, X. Zhen, and X. Li. Learning discriminative key poses for action recognition. *IEEE Transactions on Cybernetics*, 43(6):1860–1870, Dec 2013. (Cited page 9.)

[50] C. Lu, J. Jia, and C. Tang. Range-sample depth feature for action recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Columbus, USA, june 2014. (Cited page 23.)

[51] Y. M. Lui. Tangent bundles on special manifolds for action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 22:930–942, 2012. (Cited pages 38, 53, and 55.)

[52] J. Luo, W. Wang, and H. Qi. Group sparsity and geometry constrained dictionary learning for action recognition from depth maps. In *IEEE Int. Conf. on Computer Vision (ICCV)*, Sydney, Australia, December 2013. (Cited page 26.)

[53] M. Martinez and L. E. Sucar. Learning dynamic naive bayesian classifiers. In *Proceedings of the Twenty-First International FLAIRS Conference*, Coconut Grove, USA, May 2008. (Cited page 110.)

[54] A. Marzal and E. Vidal. Computation of normalized edit distance and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):926–932, 1993. (Cited page 26.)

[55] G. Mastorakis and D. Makris. Fall detection system using kinects infrared sensor. *Journal of Real-Time Image Processing*, 9(4):635–646, 2014. (Cited page 12.)

[56] Microsoft Kinect. `http://www.microsoft.com/en-us/kinectforwindows`, 2013. (Cited pages 9, 13, and 14.)

[57] Microsoft Kinect Software Development Toolkit. `https://dev.windows.com/en-us/kinect/develop`, 2015. (Cited page 15.)

[58] B. Ni, Y. Pei, P. Moulin, and S. Yan. Multi-level depth and image fusion for human activity detection. *IEEE Transactions on Cybernetics*, 43(5):1383–1394, Oct 2013. (Cited page 29.)

[59] B. Ni, G. Wang, and P. Moulin. Rgbd-hudaact: A colordepth video database for human daily activity recognition. In *International Conference on Computer Vision Workshops*, Colorado Springs, USA, June 2011. (Cited page 28.)

[60] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Sequence of the most informative joints (smij): A new representation for human skeletal action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Providence, USA, June 2012. (Cited pages xi, 25, and 26.)

[61] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Sequence of the most informative joints (smij): A new representation for human skeletal action recognition. *Journal of Visual Communication and Image Representation*, 25(1):24–38, 2014. (Cited pages 25 and 26.)

[62] E. Ohn-Bar and M. M. Trivedi. Joint angles similarities and HOG^2 for action recognition. In *Proc. CVPR Work. on Human Activity Understanding from 3D Data*, pages 465–470, Portland, Oregon, USA, June 2013. (Cited pages xi, 28, 29, 52, 69, 76, 84, and 108.)

[63] OpenNI 2 Software Development Toolkit. `http://structure.io/openni`, 2015. (Cited page 15.)

[64] O. Oreifej and Z. Liu. HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition*, pages 716–723, Portland, Oregon, USA, June 2013. (Cited pages xi, 23, 24, 30, 69, 70, and 108.)

[65] J. R. Padilla-Lopez, A. A. Chaaraoui, and F. Florez-Revuelta. A discussion on the validation tests employed to compare human action recognition methods using the msr action3D dataset. In *arXiv preprint arXiv:1407.7390v3*, 2015. (Cited page 69.)

[66] A. Paiement, L. Tao, S. Hannuna, M. Camplani, D. Damen, and M. Mirmehdi. Online quality assessment of human movement from skeleton data. In *Proceedings of British Machine Vision Conference (BMVC)*. (Cited page 12.)

[67] J. Pinquier, S. Karaman, L. Letoupin, P. Guyot, R. Mégret, J. Benois-Pineau, Y. Gaëstel, and J. Dartigues. Strategies for multiple feature fusion with hierarchical hmm: application to activity recognition from wearable audiovisual sensors. In *International Conference on Pattern Recognition (ICPR)*, Tsukuba, Japan, November 2012. (Cited page 110.)

[68] R. Poppe. A survey on vision-based human action recognition. *Image Vision Comput.*, 28(6):976–990, June 2010. (Cited page 17.)

[69] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb 1989. (Cited page 110.)

[70] M. Abdur Rahman, A.M. Qamar, M.A. Ahmed, M. Ataur Rahman, and S. Basalamah. Multimedia interactive therapy environment for children having physical disabilities. In *International conference on multimedia retrieval (ICMR)*, Dallas, USA, April 2013. (Cited page 12.)

[71] H. Rahmani, A. Mahmood, D. Q. Huynh, and A. Mian. Hopc: Histogram of oriented principal components of 3d pointclouds for action recognition. In *European Conference on Computer Vision (ECCV)*, Zurich, Swiss, September 2014. (Cited pages 18 and 22.)

[72] Z. Ren, J. Yuan, and Z. Zhang. Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera. In *Proc. ACM Int. Conf. on Multimedia*, pages 1093–1096, Scottsdale, Arizona, USA, Nov. 2011. (Cited page 14.)

[73] L. Rybok, B. Schauerte, Z. Al-Halah, and R. Stiefelhagen. Important stuff, everywhere! activity recognition with salient proto-objects as context. In *IEEE Winter Conf. on Applications of Computer Vision*

(WACV), pages 646–651, Steamboat Springs, CO, March 2014. (Cited page 118.)

[74] S. Saini, D. Rambli, S. Sulaiman, M. Zakaria, and S. Shukri. A low-cost game framework for a home-based stroke rehabilitation system. In *International Conference on Computer Information Science (ICCIS)*, Kuala Lumpur, Malaysia, June 2012. (Cited page 12.)

[75] A. A. Salah, T. Gevers, N. Sebe, and A. Vinciarelli. Challenges of human behavior understanding. In *International Workshop on Human Behavior Understanding*, volume 6219, pages 1–12. (Cited page 9.)

[76] L. Seidenari, V. Varano, S. Berretti, A. Del Bimbo, and P. Pala. Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses. In *Proc. CVPR Work. on Human Activity Understanding from 3D Data*, pages 479–485, Portland, Oregon, USA, June 2013. (Cited pages xi, 25, 26, 73, and 76.)

[77] L. Shao, X. Zhen, D. Tao, and X. Li. Spatio-temporal laplacian pyramid coding for action recognition. *IEEE Transactions on Cybernetics*, 44(6):817–827, 2014. (Cited page 9.)

[78] S. Shirazi, M. T. Har, C. Sanderson, A. Alavi, and B. C. Lovell. Clustering on Grassmann manifolds via kernel embedding with application to action analysis. In *Proc. Int. Conf. on Image Processing (ICIP*, pages 781–784, Orlando, USA, Spetember 2012. (Cited pages 38 and 55.)

[79] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 1–8, Colorado Springs, Colorado, USA, June 2011. (Cited pages 14, 15, 24, 52, and 56.)

[80] R. Slama, H. Wannous, M. Daoudi, and A. Srivastava. Accurate 3d action recognition using learning on the grassmann manifold. *Pattern Recognition*, 48(2):556–567, February 2015. (Cited page 27.)

[81] A. Srivastava, E. Klassen, S. H. Joshi, and I. Jermyn. Shape analysis of elastic curves in euclidean spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1415–1428, 2011. (Cited page 48.)

[82] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk and wearable computer-based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1371–1375, 1998. (Cited page 110.)

[83] J. Sung, C. Ponce, B. Selman, and A. Saxena. Human activity detection from rgbd images. In *In AAAI workshop on Plan, Activity and Intent Recognition (PAIR)*, San Francisco,USA, August 2011. (Cited page 18.)

[84] Y. Tian, L. Cao, Z. Liu, and Z. Zhang. Hierarchical filtered motion for action recognition in crowded videos. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 42(3):313–323, May 2012. (Cited page 9.)

[85] X. Tong, P. Xu, and X. Yan. Research on skeleton animation motion data based on kinect. *International Symposium on Computational Intelligence and Design (ISCID)*, 2. (Cited page 11.)

[86] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, November 2008. (Cited page 17.)

[87] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2273–2286, 2011. (Cited pages 38 and 55.)

[88] A. Veeraraghavan, A.K. Roy-Chowdhury, and R. Chellappa. Matching shape sequences in video with applications in human movement

analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(12):1896–1909, 2005. (Cited pages 53 and 79.)

[89] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Columbus, USA, June 2014. (Cited pages xi and 27.)

[90] Vicon Motion Systems. `http://www.vicon.com/`. (Cited page 13.)

[91] A. W. Vieira, E. R. Nascimento, G. L. Oliveira, Z. Liu, and M. F. M. Campos. STOP: Space-time occupancy patterns for 3D action recognition from depth map sequences. In *Iberoamerican Congress on Pattern Recognition*, pages 252–259, Buenos Airies, Argentina, Sept. 2012. (Cited pages xi, 23, 24, 30, 53, 69, and 100.)

[92] Virtual Sensei. `http://http://www.virtualsensei.it/`, 2012. (Cited page 11.)

[93] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu. Robust 3D action recognition with random occupancy patterns. In *Proc. Europ. Conf. on Computer Vision*, pages 1–8, Florence, Italy, Oct. 2012. (Cited pages 18, 23, 30, 53, 69, and 98.)

[94] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8, Providence, Rhode Island, USA, June 2012. (Cited pages xi, 30, 52, 67, 69, 89, 108, 123, and 124.)

[95] P. Wei, Y. Zhao, N. Zheng, and S. Zhu. Modeling 4d human-object interactions for event and object recognition. In *International Conference on Computer Vision (ICCV)*, Sydney, Australia, December 2013. (Cited pages xi, 32, and 33.)

[96] P. Wei, N. Zheng, Y. Zhao, and S. Zhu. Concurrent action detection with structural prediction. In *International Conference on Computer*

Vision (ICCV), Sydney, Australia, December 2013. (Cited pages 18 and 33.)

[97] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224–241, February 2011. (Cited page 17.)

[98] L. Xia and J. K. Aggarwal. Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *Proc. CVPR Work. on Human Activity Understanding from 3D Data*, pages 2834–2841, Portland, Oregon, USA, June 2013. (Cited pages xi, 21, 23, 53, 69, 108, 123, 124, and 128.)

[99] L. Xia, C. Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3D joints. In *Proc. Work. on Human Activity Understanding from 3D Data*, pages 20–27, Providence, Rhode Island, USA, June 2012. (Cited pages xi, 17, 18, 25, 26, 52, 53, 73, 75, and 110.)

[100] L. Xia, I. Gori, J. K. Aggarwal, and M. S. Ryoo. Robot-centric activity recognition from first-person rgb-d videos. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa Beach, USA, January 2015. (Cited page 18.)

[101] D. Xu and S.-F. Chang. Visual event recognition in news video using kernel methods with multi-level temporal alignment. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Minneapolis, USA, June 2007. (Cited page 27.)

[102] X. Yang and Y. Tian. Eigenjoints-based action recognition using naive-bayes-nearest-neighbor. In *Proc. Work. on Human Activity Understanding from 3D Data*, pages 14–19, Providence, Rhode Island, June 2012. (Cited pages xi, 24, 25, 27, 52, 69, 76, 123, and 128.)

[103] X. Yang and Y. Tian. Super normal vector for activity recognition using depth sequences. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Colombus, USA, june 2014. (Cited page 24.)

[104] X. Yang, C. Zhang, and Y. Tian. Recognizing actions using depth motion maps-based histograms of oriented gradients. In *Proc. ACM Int. Conf. on Multimedia*, pages 1057–1060, Nara, Japan, Oct. 2012. (Cited pages xi, 21, 22, 53, and 69.)

[105] M. Ye, Q. Zhang, L. Wang, J. Zhu, R. Yang, and J. Gall. A survey on human motion analysis from depth data. In *CVPR Tutorial on RGBD Cameras*, Portland, USA, June 2013. (Cited pages 17 and 89.)

[106] G. Yu, Z. Liu, and J. Yuan. Discriminative orderlet mining for real-time recognition of human-object interaction. In *Asian Conference on Computer Vision (ACCV)*, Singapore, November 2014. (Cited pages 17, 18, 30, 32, 117, 121, 123, 124, 125, 127, 128, and 129.)

[107] K. Yun, J. Honorio, D. Chattopadhyay, T. L. Berg, and D. Samaras. Two-person interaction detection using body-pose features and multiple instance learning. In *The 2nd International Workshop on Human Activity Understanding from 3D Data at Conference on Computer Vision and Pattern Recognition (HAU3D-CVPRW)*, Providence, USA, June 2012. (Cited page 18.)

[108] M. Zanfir, M. Leordeanu, and C. Sminchisescu. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pages 2752–2759, Sydney, AUstralia, December 2013. IEEE. (Cited pages 27, 28, 108, 123, 124, and 128.)

[109] Q. Zhang and B. Li. Discriminative k-svd for dictionary learning in face recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, USA, June 2010. (Cited page 27.)

[110] Y. Zhao, H. Cheng, and Lu Yang. 3d sparse quantization for feature learning in action recognition. In *IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, Chengdu, China, July 2015. (Cited page 22.)

[111] F. Zhou, F. De la Torre, and J. K Hodgins. Hierarchical aligned cluster analysis for temporal clustering of human motion. *IEEE Transac-*

tions on Pattern Analysis and Machine Intelligence, 35(3):582–596, Mar 2014. (Cited pages 106, 107, and 108.)

[112] L. Zhou, W. Li, Y. Zhang, P. Ogunbona, D. T. Nguyen, and Hanling Zhang. Discriminative key pose extraction using extended lc-ksvd for action recognition. In *International Conference on Digital lmage Computing: Techniques and Applications (DlCTA)*, Wollongong, Australia, November 2014. (Cited page 26.)