Geometric Approaches for 3D Human Motion Analysis: Application to Action Recognition and Retrieval

Rim Slama

Année 2014 Num d'ordre: 41442

UNIVERSITÉ LILLE1

THÈSE

pour obtenir le grade de
DOCTEUR,
SPÉCIALITÉ INFORMATIQUE

présentée et soutenue publiquement par

Rim SLAMA

le 06/10/2014

Geometric Approaches for 3D Human Motion Analysis: Application to Action Recognition and Retrieval

préparée au sein du laboratoire LIFL

COMPOSITION DU JURY

M. Edmond Boyer	Directeur de recherche, INRIA Grenoble Rhône-Alpes	Rapporteur
Mme. Rita Cucchiara	Professeur, University of Modena and Reggio Emilia	Rapporteur
Mme Saida Bouakaz	Professeur, Université Claude Bernard Lyon 1	Examinateur
M. Hubert Cardot	Professeur, Université François Rabelais de Tours	Examinateur
M. Olivier Colot	Professeur, Université Lille 1	Examinateur
M. Alain Trouvé	Professeur, Ecole Normale Superieure, Cachan	Examinateur
M. Mohamed Daoudi	Professeur, Institut Mines-Télécom	Directeur
M. Hazem Wannous	Maître de conférences, Université Lille 1	Encadrant

Acknowledgments

I would like to express my gratitude to my advisor, Pr. Mohamed Daoudi for guiding me through my research with expertise, understanding, and patience. His guidance helped me in all time of research and writing of this thesis.

I also thank my co-advisor Dr. Hazem Wannous for his outstanding human qualities, his unconditional encouragements and his time in beneficial scientific discussions. I would also like to thank Pr. Anuj Srivastava for the fruitful collaboration and for sharing his knowledge throw helpful discussions.

Special thanks are due 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: Pr. Edmond Boyer and Pr. Rita Cucchiara. I also thank the committee Pr. Saida Bouakaz, Pr. Hubert Cardot, Pr. Olivier Colot and Pr. Alain Trouvé. All these people made me the honor to be present for this special day despite their highly charged agendas and I am sincerely grateful for that.

A special thank goes to our research buddies Boulbaba Ben Amor, Hassen Drira and Jean-Philippe Vandeborre. I especially thank our research engineer Sebastien Poulmane for his help and nice collaboration.

I would also thank my freinds and collegues, Xia Baiqiang, Maxime Devanne, Taleb Alashkar, Meng Meng, Vincent Leon and Paul Audain Desrosiers for their assistance, friendship and support.

I also gratefully acknowledge the region of Nord-Pas-de-Calais for my thesis funding. I thank my friends, people at TELECOM Lille and LIFL for their warm encouragements. I want to thank all my family and my in-laws for their love, support and encouragements.

Finally, I would like to thank my loving husband, Moncef, for sticking

with me through all times, for his continuous support which make me very positive along my thesis.

Lille, the October 23, 2014.

Publications of the author

International journals

- **Rim Slama**, Hazem Wannous, Mohamed Daoudi, 3D Human Motion Analysis Framework for Shape Similarity and Retrieval. In *Image and Vision Computing Journal (IVC)*, volume 32, pages 131-154 ,2014.

- **Rim Slama**, Hazem Wannous, Mohamed Daoudi. Accurate 3D action recognition using smart learning on Grassmanian manifold. In *Pattern Recognition (PR)*, 2014. (Accepted)

- Hassen Drira, Boulbaba Benamor, Mohamed Daoudi, Anuj Strivastava, **Rim Slama**, 3D Face Recognition under Expressions, Occlusions, and Pose Variations. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, volume 35, pages 2270-2283, 2013.

International conferences

- **Rim Slama**, Hazem Wannous, Mohamed Daoudi, Extremal Human Curves: a New Human Body Shape and Pose Descriptor. In *IEEE Automatic Face and Gesture Recognition (FG)*, pages 1-6 , China, 2013. (Oral presentation)

- **Rim Slama**, Hazem Wannous, Mohamed Daoudi, 3D Human Video Retrieval: from Pose to Motion Matching. *In Eurographics Workshop on 3D Object Retrieval (3DOR)*, pages 33-40, Spain, 2013. (Oral presentation)

- **Rim Slama**, Hazem Wannous, Mohamed Daoudi, Grassmannian Representation of Motion Depth for 3D Human Gesture and Action Recognition. In *International Conference on Pattern Recognition (ICPR)*, Stockholm, Sweden, 2014. (Oral presentation)

National conferences

- **Rim Slama**, Hazem Wannous, Mohamed Daoudi, Indexation et recherche d'actions humaines 3D basées sur l'analyse des courbes surfaciques. In *COmpression et REprésentation des Signaux Audiovisuels (CORESA)*, Le Creusot, France, 2013. (Oral presentation)

- Hassen Drira, **Rim Slama**, Boulbaba Ben Amor, Mohamed Daoudi, Anuj Srivastava, Une nouvelle approche de reconnaissance de visages 3D partiellement occultés. In *Reconnaissance des Formes et l Intelligence Artificielle (RFIA)*, Lyon, France, 2012. (Oral presentation)

Contents

List of Figures

Introduction

1

Sommaire

The work presented in this thesis is motivated by the recent rise in 3D human video data, and the need for efficient algorithms, in order to fully exploit this data within classic computer vision tasks.

While human body analysis in 2D image and video have received a great interest during the last two decades, analysis of 3D videos of human body is still a little explored field. Parallel to this, 3D video sequences of human motion are more and more available. Their acquisition is possible using multiple view reconstruction systems which give a stream of 3D models of subjects in motion. In such videos, each frame is a mesh approximation of the body surface shape often generated independently regardless of its neighboring frames. Researches about 3D video have been mainly focused on performance, quality improvements and compression methods. Consequently, 3D videos are yet mainly used for display. However, the acquisition of long sequences produces massive amounts of data which necessitates efficient schemes for navigating, browsing, searching, and viewing video data.

Hence, we need to develop efficient and effective methods of retrieval to accelerate and facilitate browsing this data. There are two interesting retrieval scenarios: (1) Retrieving frames containing human in same poses, which helps to analyze repetitions in the sequence, to take decisions about motion transition and to concatenate 3D video sequences while producing a novel character animation. (2) Retrieving subsequences which represent human in same motion. Several applications arises from this such as video understanding, summarizarion and video synthesis. These potential applications subsequently require solving the problem of pose/motion retrieval in 3D human videos. This retrieval system is based on the definition of pose or motion descriptors and similarity measure to compare them.

More recently, effective and inexpensive depth video cameras are increasingly emerged. These range sensors provide 3D structural information of the scene, which offers more discerning information to recover human postures. Often compared to 2D cameras, this device are more robust to common low-level difficulties in RGB imagery like background

subtraction, light variation and can work even in total darkness. In addition to depth images, a real time 3D skeleton estimation is possible. The availability, the real time acquisition and the advantages of depth data encourage its use in several applications that need to understand and to recognize human actions using such a data stream instead of 2D videos.

Recognizing human actions have many potential applications including video surveillance, human computer interfaces, sport video analysis and health care. Each application has its own constraints, sometimes conflicting, often linked. However, main requirements in action recognition systems remain: accuracy and speed. Each solution must find its own balance between its constraints, depending on its application context. Despite the researchers efforts in the past decade there are still related issues to consider in human action recognition. The first one is the modelling of the human actions that are dynamic, ambiguous and interactive with objects. The second one is the response time which should be as speed as possible with accurate decision.

In this thesis a particular focus is firstly given to fully reconstructed human bodies in 3D videos in the purpose of pose and motion retrieval. Then, the work is oriented toward motion modelling and action learning for the task of human action and gesture recognition using RGB-D sensors.

Whatever using 3D data given by dynamic meshes or using depth images and skeletons, human video motion can be studied from mainly two perspectives, the feature space and the model space. These spaces can be described mathematically as manifolds. In fact, significant advancement have been recently made in the analytic and geometric understanding of these spaces. Therefore, an important development is marked by moving away from data-driven approaches to geometry driven approaches for characterizing videos.

In the literature, several examples of various analytical manifolds are found in pose and motion modeling. Far as we know, most of them are proposing solutions by extracting features from 2D videos. In this thesis, we propose geometric frameworks for analysing human motion in 3D videos. These frameworks are proposing solutions for the retrieval and

recognition task while modelling either feature space or model space on different manifolds adapted to each type of features extracted from 3D data.

1.1 Thesis contributions

This PhD thesis brings two main contributions; the first one is related to 3D human pose and motion modelling in 3D videos and the second one is directed toward 3D action learning and recognition from depth data.

- *3D human shape similarity for pose/motion retrieval in 3D videos* Interested with this special task, we propose a unified Riemannian framework to model both static and temporal shape descriptors and perform their comparisons. This framework relies on a novel 3D human pose descriptor called Extremal Human Curves (EHC), extracted from both the spatial and the topological dimensions of the body surface. The EHC is an extremal descriptor of the surface deformation which is composed of a collection of local open 3D curves. Its extraction is based on extremal features and geodesics between each pair of them. Once human body poses are represented by EHC, we propose to compare them in the Riemannian manifold of open curve shape space. Invariant to affine transformations, our EHC descriptor and its defined metric allow pose comparison of subjects regardless to translation, rotation and scaling. The first evaluation of this descriptor is performed on pose retrieval either on static datasets or on 3D videos. Then, we propose to extend this descriptor in the temporal domain in order to compare sequences and retrieve similar motions. The key idea is to represent the sequence as a succession of EHC representations and thus model the human motion as a trajectory on the shape space. To compare two sequences of motion, we propose the use of dynamic time warping to align correspondent trajectories and to give a similarity score between them.

- *Human action recognition from depth sensors* Here we propose a second Riemannian framewok for modelling and recognizing human

motion acquired by depth cameras. In this framework, we model sequence features temporally as subspaces lying in Grassmann manifold. We propose to test two kind of features in this framework: (1) 3D human joints given by the skeleton extracted in real time from depth maps and (2) local oriented displacement features extracted from boxes around each subject in depth images. Working with locale displacement features, this framework allows accurately recognizing actions which involve human object interaction and also recognizing hand gestures with high accuracies. In order to improve learning process, we proposed a new learning algorithm on Grassmann manifold which embeds each action, presented as a point on this manifold, in higher dimensional representation. This latter is using the notion of tangent spaces on specific classes providing a natural separation of action classes. Using this framework with joint features and the proposed new algorithm, we offered the possibility of recognizing actions involving human computer interaction with high accuracy and speed.

1.2 Organization of the thesis

We have divided the rest of the manuscript into two parts. The first part presents solutions for the retrieval task in 3D video sequences of people. Under this part, chapter 2 discusses related works in the area of static and temporal shape similarity and video retrieval. In chapter 3, we propose a new descriptor for 3D human shape modelling and pose comparison in a Riemannian framework. In chapter 4, using the same framework, a solution for human motion representation and comparison is presented and tested in several scenarios including motion retrieval, video summarization and hierarchical retrieval.

The second part presents solutions for action recognition using video sequences from depth sensors. Under this part, chapter 5 reviews the existing solutions suggested in the literature. In chapter 6, we propose a new framework for modelling and classifying actions which are represented either in joint space or in depth-map space.

Finally, we conclude this manuscript by summarizing the contributions of this thesis, enumerating remaining open problems and proposing directions for future research.

Part I

3D human motion analysis framework using open curve shape space

State-of-the-art

2

Sommaire

2.1 Introduction

As human actions are done in real 3D environments, naturally the use of 3D data describing these actions allows a more efficient analysis. 3D representation of human motion has been introduced through the use of multiple camera systems, in which the surface structure of the human body can be reconstructed, and thereby a more descriptive representation for human posture and motion can be captured. As the amount of dynamic 3D mesh data increases, the development of efficient and effective retrieval systems is being desired.

In this chapter, we first motivate 3D pose/motion retrieval and present main challenges to overcome. Second, 3D human datasets containing static or dynamic meshes are presented. Finally, existent 3D human body descriptors in 3D video sequences are reviewed, and to conclude we discuss their limitations.

2.2 Motivation and challenges

3D Human body shape similarity is itself an important area, recently attracted much attention in the field of human-computer interface (HCI) and computer graphics, with many related research studies. Among these, researches started with 3D features have been applied for body pose estimation and 3D video analysis. More than that, 3D video sequences of human motion is more and more available. In fact, their acquisition with a multiple view reconstruction systems or animation and synthesis approaches [22] [27] received a considerable interest over the past decade following the pioneering work of Kanade et al. [59].

Several potential applications arisen from this, such as content based pose retrieval in a basis of human models, transition decision and 3D video concatenation for character animation, 3D video summarization and compression and 3D mesh video retrieval. These potential applications subsequently require solving the problem of identifying frames with similar poses.

Most of the research topics on these 3D video focus mainly on perfor-

mance, quality improvements and compression methods [120] [27]. However, the acquisition of long sequences produces massive amounts of data which make the datasets difficult to handle: hence the need to develop efficient and effective segmentation and retrieval systems for managing the database and searching for relevant information quickly.

The main challenges in 3D human pose/motion descriptor modelling and comparison are :

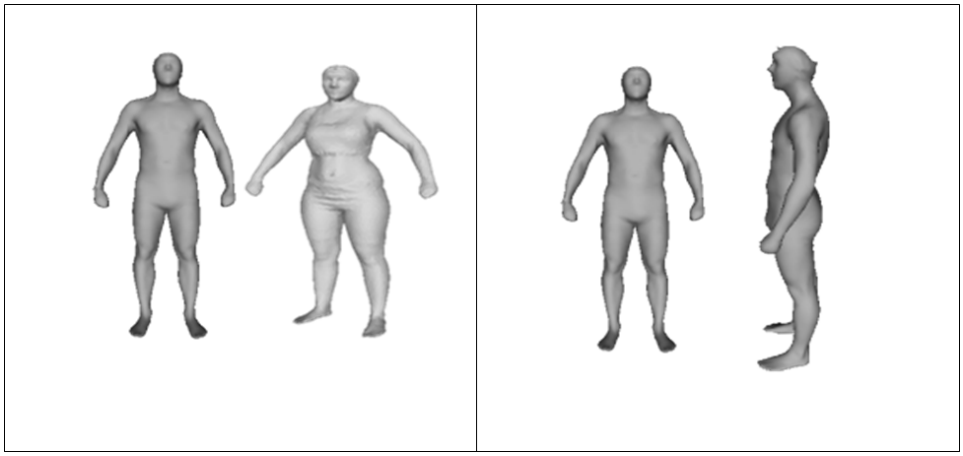
- Invariance to rotation, translation and anthropometry of the actors: the same pose can be done in different location in the scene and human body is different from one person to an other.
- Robustness to topology change: the descriptor should be robust to topological changes which can occur in the 3D video sequence because of a noisy reconstruction or loose clothes.
- Robustness to mesh resolution and robustness to noise: depending on the acquisition system the resolution of the 3D human mesh can be different. Thus, the developed descriptor should be robust to noise and should not depend on mesh resolutions.
- Complexity: the descriptor should not be complex or costly in time to be effective for the retrieval task.
- Invariance to speed variation : two motions can be similar but performed with differ speed.

Some of these challenges are illustrated in figure 2.1.

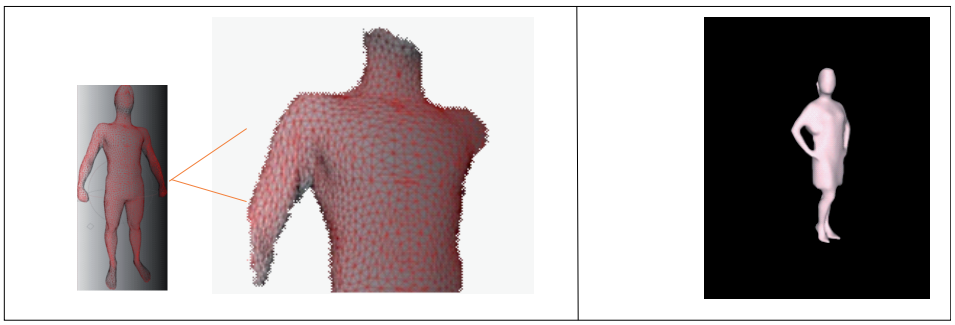
2.3 Acquisition systems

2.3.1 Static 3D human body acquisition

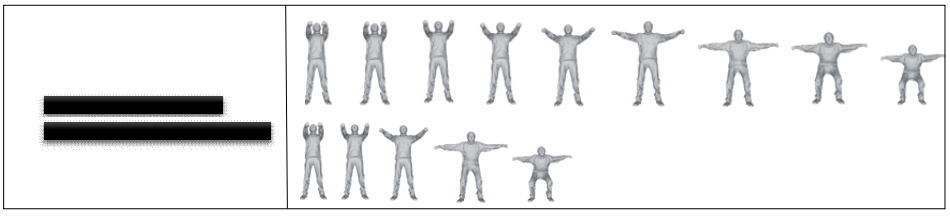
3D scanners are generally used to acquire real 3D human models [3, 5]. They are easy to use and offer various softwares to model the result measurements, but they are quite expensive. They work according to different technologies (laser beam, structured light, ...) and provide million of points with often related color information.



(a)



(b)



(c)

Figure 2.1 – *Different challenges in 3D human pose/motion retrieval: (a) human body shape change and variation in rotation, translation (b) connectivity change and topology change (c) variation in frame sequence number or in execution rate.*

Other techniques are based on silhouette extraction [157] or multi-image photogrammetry [26].

Recently, it is increasingly popular to scan the 3D human body using single or multiple depth sensors like kinect as introduced in works of [24, 117]. The acquired models using these technologies are noisy and have lower resolution than scanned models.

Moreover, synthetic 3D human bodies can be generated artificially. These synthetic models are created by graphic designer using specialized software (like 3D studio max [2]).

Examples of setup systems and 3D human bodies from both real and synthetic datasets are shown in Figure 2.2.

2.3.2 Dynamic 3D human body acquisition

3D human video is composed of a consecutive sequence of frames. Each frame is represented as a polygon mesh of a human in a certain pose. Namely, each frame is expressed by coordinates of vertices, their normals, their connection (topology), and sometimes color, and others information corresponding to the representation format.

Such kind of data can be generated using a multi-camera environment as shown in Figure 2.3. Such environment consists on a fixed zone of interest surrounded by various cameras facing it at different angles. These cameras are calibrated and the internal and external parameters of calibration of each camera are estimated beforehand. This system allows capturing synchronized multi-view images, taken at several instants over time. Then, images are used to build a sequence of textured meshes describing the captured dynamic scene [44, 120]. The most significant characteristic in 3D video generated from multi-camera system is that each frame is generated regardless to its neighboring frames. Therefore, the connectivity and topology differ from one frame to an other. Many recent approaches have been proposed to improve multi-reconstruction systems [35, 82, 156, 16].

Another approach which allows to capture 3D videos is the mesh animation. In fact, it is possible to scan a 3D human body statically and then animate it using Motion capture system. A recent work in this area

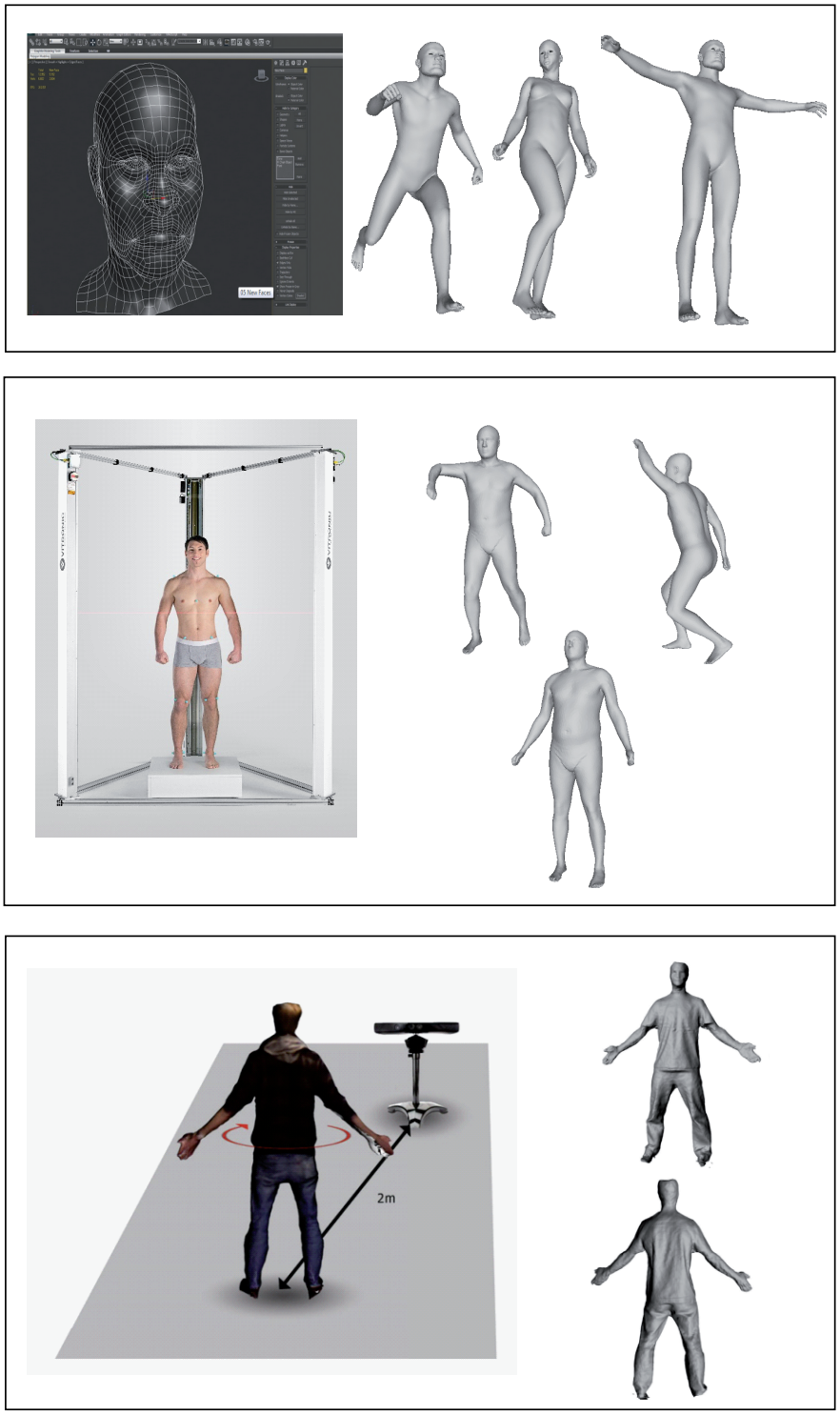


Figure 2.2 – *Examples of 3D human models. (top) Synthetic models from the dataset presented in[91]. (middle) Vitronics Vitus scanner [5] and examples of scans from CAESAR dataset [1]. (Bottom) The setup system of human body scanning using a single kinect [24] .*

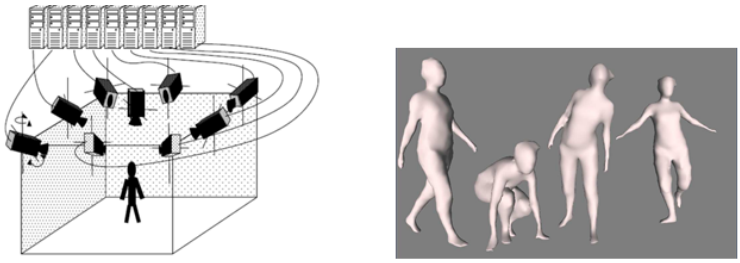


Figure 2.3 – *Multi-camera system for 3D video acquisition [38].*

of articulated mesh animation from multi-view silhouettes is presented in the work of Vlasic et al. [130]. As shown in Figure 2.4, the whole details of the subject clothers are well captured using the proposed approach.

Figure 2.4 – *3D shapes for three subjects in five sequences from the dataset presented in [130].*

2.4 Datasets

We provide in this section a summary of the most known static and dynamic datasets of 3d human body.

Civilian American and European Surface Anthropometry Resource (CAESAR) [1]: is an extensive database product which includes measurements from the European population sample (2,000 male and female subjects, aged 18-65). This database is the first to include 3-D model scans using camera views from the 3-D scan to accurately provide complete 3-D models in different poses. Recently, Pickup et al. [91] presents a real dataset composed of 400 meshes selected from CAESAR [1]. It is made up of 40 human subjects (half male, half female), each in 10 different poses. They also present a synthetic dataset made up of synthetic data created using DAZ Studio. This synthetic dataset consists of 15 different human

models, each with its own unique body shape. Five of these are male, five female, and five child body shapes. Each of these models exist in 20 different poses, resulting in a dataset of 300 models. The size of the triangles in both datasets is not uniform.

Hasler Dataset [45]: Dense full body 3D scans are captured using a Vitronic laser scanner. This dataset presents 114 subjects aged between 17 and 61, where 59 are males and 55 are females. In addition to one standard pose that allows the creation of a shape-only model, all subjects are scanned in at least 9 poses selected randomly from a set of 34 poses,

3D Sequences proposed by Starck et al. [107]: It contains people in motion acquired using multi-view system and reconstructed using Starck et al. approach [107]. 3D videos of dancers wearing loose clothes and performing different dancing styles are reconstructed. Besides, 3D videos of an actor running, walking and boxing while wearing different clothes are given from this work. Figure 2.5 shows some examples from these videos.

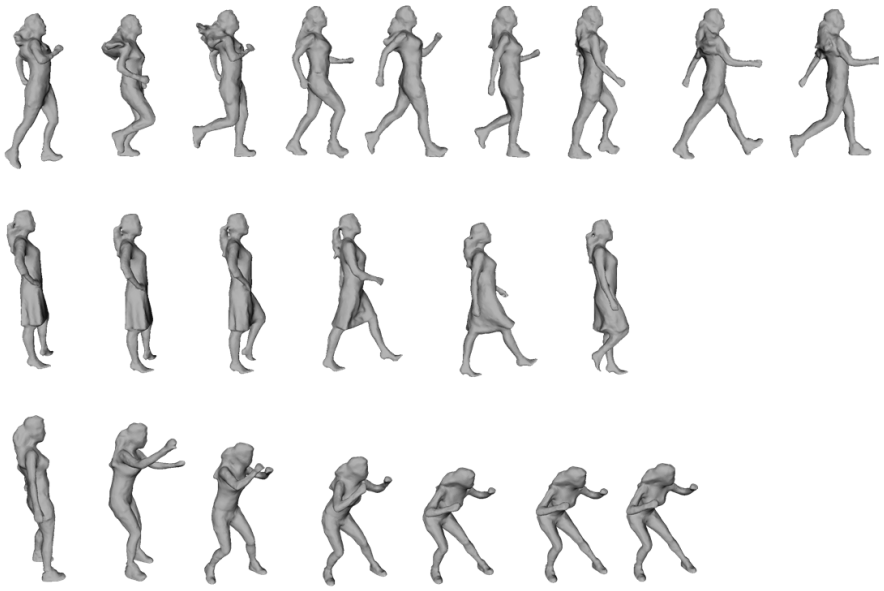


Figure 2.5 – *Dynamic 3D meshes for a human body wearing different costums and performing different actions.*

i3DPost Multi-View Human Action [38] : This dataset consists of 8 actors performing 10 different actions (walk, run, jump, bend, hand-wave...)

3D Sequences presented by Huang et al. [52]: It consists of a simulated

dataset created by animating an articulated character model for 14 people (10 men and 4 women) using motion capture sequences. 3D models of people with different body shape and clothing were reconstructed from multiple view images. Models are animated using 28 motion capture sequences from the Santa Monica Mocap archive for motions (like: sneak, walk, slow run, fast run). Each sequence comprises 100 frames giving a total of 39200 frames of synthetic 3D video with known ground-truth correspondence.

3D Sequences proposed by Vlasic et al.[130]: 3 people (2 men and 1 woman) in 6 motions (cran, marche, squat, handstand, samba, swing) giving a total of 1582 frames.

More dynamic datasets are also available such as: 4D repository [4], where many real dynamic sequences are proposed such as *man dance* sequence and also *flashkick*. Other 3D videos which are used for particular purposes are presented in [148] of Japanese traditional dances called *bon-odori*. In these videos, body surface shape sometimes contain temporal correspondence [52, 130] and sometimes this information is missed and mesh connectivity and geometry is changing from one frame to another [38, 107, 4].

Data	Static/Dynamic	Real/Synthetic
Ceasar [1]	static	real
Pickup et al.[91]	static	Synthetic and real
Haster et al. [45]	static	real
Liu et al. [16]	static	real
Gkalelis et al. [38]	dynamic	real
Huang et al. [52]	dynamic	synthetic
Vlasic et al. [130]	dynamic	real
Starck et al. [107]	dynamic	real
4dr [4]	dynamic	real

Table 2.1 – *Summary of datasets containing 3D human body in static poses and also in motion.*

2.5 3D HUMAN BODY SHAPE SIMILARITY

The problem of shape similarity has been widely studied in the 3D retrieval literature. Shape descriptors developed for this purpose aim to discriminate rigid shapes from different object classes (chair, table, human

...) and existing methods [114] achieve extremely high accuracy when evaluated on the most recent benchmarks. In this review we focus on shape descriptors which able to discriminate between instances from sequences of the same moving non-rigid object, a human body, which differ in both shape and motion. The temporal shape descriptor generally extends approaches used for measuring static shape similarity to temporal one in 3D video sequences. In this section we review static shape representations techniques followed by similarity metrics developed to compare these descriptions both in static and video sequences.

2.5.1 3D human body descriptors

Global descriptors Some of widely used 3D object representation approaches include: spin images, spherical harmonics, shape context and shape distribution. These histogram based representations are illustrated in Figure 2.6.

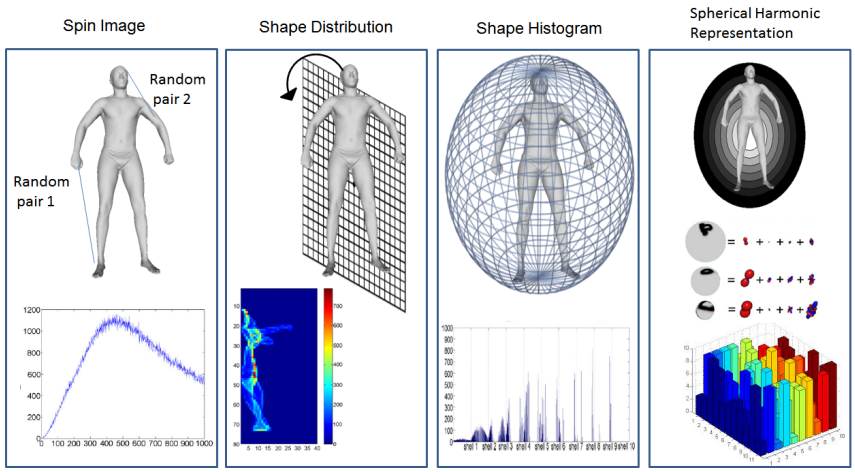


Figure 2.6 – *Illustration of the 3D human model representations using histograms. From left to right descriptors are: spine image, shape distribution, shape histogram and spherical harmonic.*

Johnson et al. [56] propose a spin image descriptor, encoding the density of mesh vertices into 2D histogram. Osada et al. [88] use a Shape Distribution, by computing the distance between random points on the surface. Ankerst et al. [9] represent the shape as a volume sampling spherical histogram by partitioning the space containing an object into disjoint cells corresponding to the bins of the histogram. This later is extended

with color information by Huang et al. [50]. A similar representation to the shape histogram is presented by Kortgen et al. [67] as a 3D extended shape context. Kazhdan et al. [61] apply spherical harmonics to describe an object by a set of spherical basis functions representing the shape histogram in a rotation-invariant manner.

These approaches use global features to characterize the overall shape and provide a coarse description, that is insufficient to distinguish similarity in 3D video sequence of an object having the same global properties in the time. A comparison of these shape descriptors combined with self-similarities is made by Huang et al. [52].

Local descriptors Another work using histograms is presented by Yamasaki et al. [148] who propose a modified version of the shape distribution histogram. The original shape distribution histogram as shown in Figure 2.7(a) does not remain the same even for exactly the same model. However the descriptor is required to clarify a slight shape difference among frames in 3D video. Therefore, Yamasaki et al. have modified the original shape distribution algorithm for more stability. Since vertices are mostly uniform on the surface in the served 3D models, they are firstly clustered into 1024 groups based on their 3D spatial distribution employing vector quantization as shown in Figure 2.7(b). The centers of mass of the clusters are used as representative points for distance histogram generation. Although this solution allows better frame retrieval, it remains computationally expensive because of the clustering process.

The above approaches represent shape descriptors which are often fast to compute and invariant to topology and rigid transformations, but they usually do not capture any geometrical information about the 3D human body pose and joint positions/orientations. This prevents its use in certain applications that require accurate estimation of the pose of the body parts.

The shape similarity in 3D video has also been addressed in the case of skeletal shape representation. Huang et al. [53] present a comparative evaluation of skeleton-based shape descriptors against spatial descriptors. They demonstrate that skeleton-based Reeb-Graph have good performances in the task of finding similar poses of the same person in

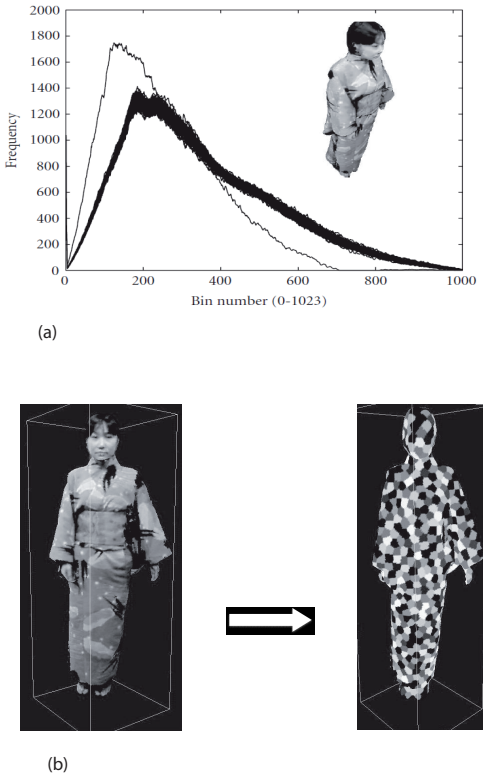


Figure 2.7 – *Concept of modified shape distribution as presented in [148]. (a) Thirty histograms for the same 3D model using the original shape distribution algorithm. (b) Vertices of 3D model are firstly clustered into groups by vector quantization in order to scatter representative vertices uniformly on 3D model surface.*

a 3D video. The Reeb-Graph descriptor used in these experiment is augmented Multiresolution Reeb Graph (aMRG) which is proposed by Tung et al. [121]. An illustration of this latter is shown in Figure 2.8. The main advantages of this descriptor is its robustness to topology change and noise. Its multiresolution property with a hierarchical node matching strategy allow go pass the NP-complete complexity of the graph matching, but the computational time remain slow.

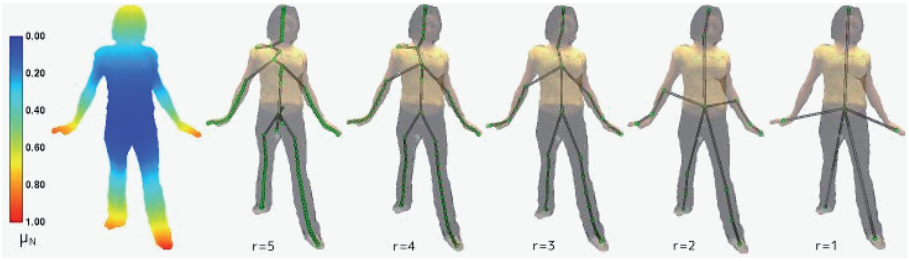


Figure 2.8 – *Reeb-graph descriptor as proposed by Tung et al. [119]. Enhanced Reeb graphs are extracted at different levels of resolution.*

Structure extraction from arbitrary shape is usually performed by fitting a 3D skeleton to the shape surface model, such as in [12]. When successful, this kind of approach is powerful because the kinematic structure of the object can be extracted, and the structure joints can be tracked while the object is in motion. However, it has not the advantage offered by Reeb-graph which overcome the topology changes and object model orientation. An example when skeleton fitting can fail is shown in Figure 2.9.

Figure 2.9 – *An example where skeleton fitting can fail.*

Skeleton fitting is also proposed by Huang et al. [49] by introducing

a learning based method that partition the point cloud observations into different rigid body parts to avoid the need for complex inverse kinematic parametrizations.

Other approaches, such as curve-skeleton [100, 18], can extract a graph with homotopy preservation property as illustrated in Figure 2.10. However, it is not suitable because numerous graph matching computation in huge data can quickly become intractable.

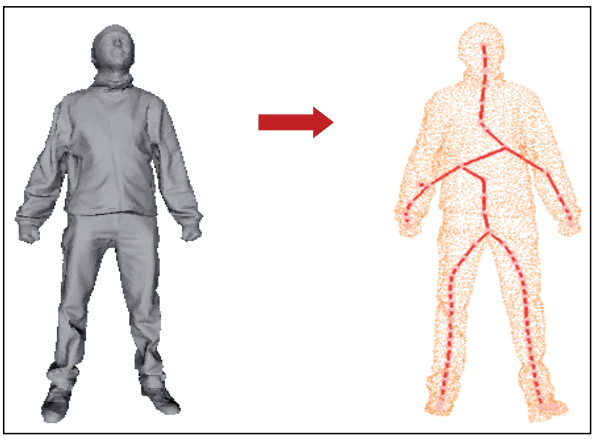


Figure 2.10 – *Curve skeleton extraction via Laplacian-based contraction using the algorithm proposed in [18].*

Other works can be found in the literature, where surface-based descriptors are often used with a step of features detection. The advantage of these features is that their detection is invariant to pose change. The extremities can be considered as the one among the most important features for the 3D objects. They can be used for extracting a topology description of the object like Reeb-graph descriptor [119].

Similarly, closed surface-based curves use specific features on the 3D mesh [111, 31, 80]. The extraction and the matching of these features have been widely investigated using different scalar functions from geodesic distances to heat-kernel [109, 76, 89]. Tabia et al. [111] propose to extract arbitrarily closed curves amounting from feature points and use a geodesic distance between curves for 3D object classification. Elkhoury et al. [31] extract the same closed curves but using a heat-kernel distance in the 3D object retrieval process. Figure 2.11 illustrates closed curves which are used for 3D body representation by Elkhoury et al. [31].

One of the earliest methods for multi-view 3D human pose tracking

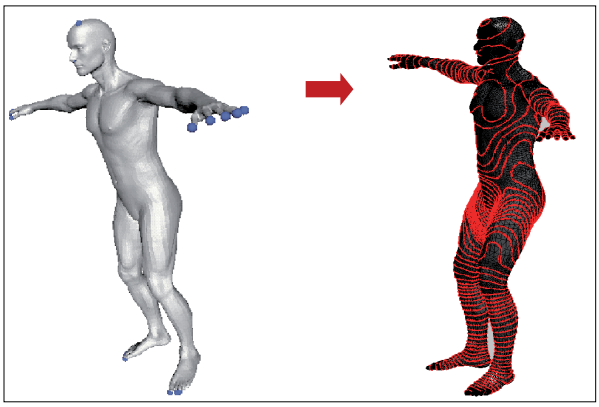


Figure 2.11 – *Illustration of the closed curves proposed by [31].*

using volumetric data was proposed by Mikic et al. [84], in which they use a hierarchical procedure starting by locating the head using its specific shape and size, and then growing to other body parts forming a cylindric human body model. Although this representation has good visual results, shown for several complex motion sequences, it is quite computationally expensive. Volumetric data have been also applied for body pose estimation and tracking where many human body models were presented like skeletal and super-quadratic models [118] (see Figure 2.12).

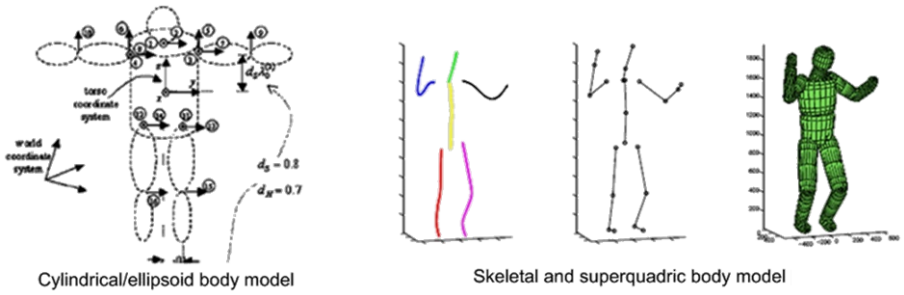


Figure 2.12 – *Cylindric [84] and skeletal model representations [118].*

2.5.2 Similarity metric

To measure the similarity between two shape models, an efficient similarity metric must be defined. Given two individual frames A, B of 3D video sequences and their descriptors N and M, frame-to-frame similarity is defined as SIM(M,N).

While representing the 3D human bodies in a certain pose by his-

togram as in [52], $SIM(M,N)$ can be computed as a simple L_2 distance between histograms M and N. We notice that other distances between histograms can also be used such as Kullback-Leibler divergence, Mahalanobis distance and Bhattacharyya distance.

Assuming two Reeb graphs M and N, a similarity is obtained by computing the following SIM function: $SIM(M,N) = \frac{1}{1+R}\sum_{r=0}^{R}\sum sim(m,n)$ where m and n are pair of consistent nodes at the resolution level $r \in [0,R]$, and sim: $M \times N \longmapsto [0,1]$. The global similarity SIM is obtained by summing similarity scores. Here each node embeds topological attributes such as relative surface area and graph connectivity information, and geometrical attributes, such as surface normal orientation histogram. Using this similarity measure Tung et al. [119] prove that Reeb graph is performant in the task of finding similar poses of the same person in 3D video.

In order to compare skeletons, joint angles can be computed and compared or direct joint 3D position can be hierarchically compared as skeleton is a kinematic model.

While comparing two frames in term of pose, static shape descriptor and similarity metric could be insufficient. Thus, Huang et al. [53, 52] propose an extension of static shape similarity to a temporal one to remove ambiguities inherent in static shape descriptors while comparing 3D video sequences of same shape. They propose temporal filtering in order to extend the static descriptor to the time domain. This solution has proven its effectiveness and was, therefore taken by Tong et al. [119] to solve the problem of the static descriptor.

2.6 3D human motion similarity

In 3D human motion retrieval system, a motion similarity measure is used to retrieve sequences with similar motion performed by different persons. The query is a 3D video of a human performing a specific movement. Such a system can help to identify repetitions in long sequence where the frame number is big and the amount of data is massive. It also allows retrieving sequences sharing the same motion from existing datasets to reuse them. Since 3D videos can contain several movements or repetitions,

a first step in a motion retrieval system could start by segmenting the sequence into atomic actions. A motion descriptor modelling can then be performed, and finally similarity measure between those descriptors can be computed. In the following, a review of each step is presented and a discussion is launched in order to highlight limits and advantages of each approach.

2.6.1 Motion segmentation

Video segmentation has been studied for various applications, such as gesture recognition, motion synthesis and indexing, browsing and retrieval. A vast amount of works in video segmentation has been performed for 2D video [66], where usually the object segmentation is firstly performed before the movement analysis. In Rui et al. [94], an optical flow of moving objects is used and motion discontinuities in trajectories of basis coefficient over time are detected. However, in Wang et al. [137], break points were considered as local minima in motion and local maxima in direction change.

Motion segmentation is strongly applied in several algorithms using 3D motion capture feature points trackable within the whole sequence, to segment the video. Detected local minima in motion (Shiratori et al. [102]) or extrema (Kahol et al. [58]) are used in motion segmentation for kinematic parameters.

Most of works on the 3D video segmentation use the motion capture data, and very few of them were applied to the dynamic 3D mesh. One of them is presented by Xu et al. [146], where a histogram of distance among vertexes on 3D mesh is generated to perform the segmentation through thresholding step defined empirically. In Yamasaki et al. [147], the motion segmentation is automatically conducted by analyzing the degree of motion using modified shape distribution for mainly japanese dances. These sequences of motion are paused for a moment and then they are considered as segmentation points. Weinland et al. [140] propose segmenting actions into primitives and classifying them into a hierarchy of action classes. Segmentation and clustering of action classes is based on a motion

descriptor which can be extracted from reconstructed volume sequences. Huang et al. [51] propose an automatic key-frame extraction method for 3D video summarization. To do so, they compute the self similarity matrix using volume-sampling spherical shape histogram descriptor. Then, they construct a graph based on this self similarity matrix and define a set of key frames as the shortest path of this graph.

2.6.2 3D human motion descriptors

Shape similarity is used for solving the problem of motion retrieval by matching frames and comparing correspondent ones using a specified metric. In Yamasaki et al. [148], the modified shape distribution histogram is employed as feature representation of 3D models. The sequence to sequence similarity is computed by Dynamic Programming matching using the feature vectors and Euclidean distance.

Recently, Tung et al. [119] propose a topology dictionary for video understanding and summarizing. Using the Multi-resolution Reeb Graph as a relevant descriptor for the shape in video stream for clustering. In this approach, they perform a clustering of the video frames into pose clusters and then they represent the whole sequence with a Markov motion graph in order to model the topology change states.

In [62], a body surface of a model is isotropically scaled so that it lies within the unit sphere located at the origin and re-oriented per frame such that the direction of motion of its centroid is always along the z axis. Then, this surface is represented by an implicit function and its shape histogram is obtained. Kullback Leibler divergence combined with an HMM, allow shape matching.

Some other works have trends to accumulate static human shape descriptors over time constructing a motion history volumes [141] for each sequence or they capture the involvement of shape changes in the sequence in order to add temporal information [139, 149, 23]. In [48], different representations of the body tracked in time are listed and compared. These various representations are: motion history volume (MHV), 3D optical flow, cylinder ellipsoid body model, skeletal and quadratic body

model. All these representations are used to track body in time and deduce a motion vector in order to perform motion retrieval. However the main problem of these approaches is the inability to recognizing actions which cannot be spatially segmented.

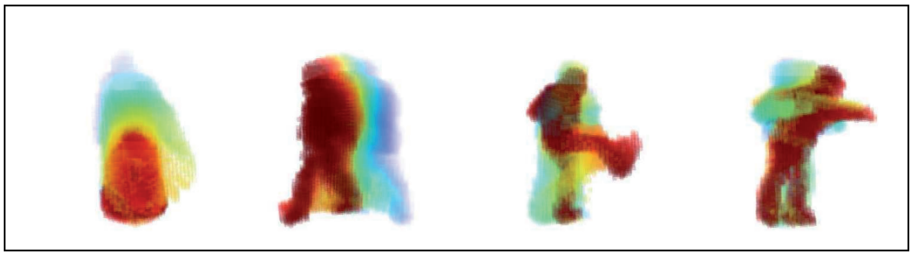


Figure 2.13 – *Motion history volume examples as presented in [141] : From left to right: sit down; walk; kick; punch. Color values encode time of last occupancy.*

Pehlivan et al. [90] present a view-independent representation based on human poses. The volume of the human body is first divided into a sequence of horizontal layers, and then the intersections of the body segments with each layer are coded with enclosing circles. The circular features in all layers are then used to generate one feature vector for each pose. Then, pose vectors are used to encode human actions as motion matrices formed by concatenating pose descriptors in all action frames. Additional motion features are added to this matrix to measure variations in spatial and temporal domains.

Holte et al. [47] propose to detect the motion of actors by combining optical flow into enhanced 3D motion vector fields. The motion of actors is captured by 3D optical flow which is first captured on each camera view and then extended to 3D using reconstructed 3D model and pixel to vertex correspondences. Finally, 3D Motion Context and Harmonic Motion Context are used to represent the extracted 3D motion vector fields in a view-invariant manner.

2.6.3 Similarity metric

Approaches based on static descriptors, like in [148], propose to measure the similarity between sequences by comparing correspondent frames. However, since sequences does not have the same number of frame, they should pass by an alignment process. The Dynamic Time Warping algo-

rithm (DTW), based on Dynamic Programming and some restrictions, was widely used to resolve this problem of temporal alignment. Given two time series with different size, DTW finds an optimal match measuring the similarity between these sequences which may vary in time or speed. Thereby, using a static shape descriptor computed from each frame and the temporal alignment using DTW, many authors succeed to perform action recognition or sequence matching for motion indexing using this temporal alignment [135, 136, 112].

There are many distance metrics allowing comparing sequences represented by a matrix, such in [90]. These distance metrics can be for example: L_p norms, Earth Mover´ s Distance (EMD), Diffusion Distance.

2.7 Discussion and conclusion

From the above review we can identify certain issues in order to consider to better improve existing approaches in pose/motion retrieval.

Most of existing works have attempted to use global descriptions of the model ignoring the local details, especially histograms based descriptions. Local 3D shape descriptors perform better than global features. As local descriptors, mainly Reeb-Graph and skeleton representations are used. These latter present limitations related to computational cost and fitting problem respectively.

Concerning motion retrieval, the sequence can be represented as a motion vector which model the dynamic of the pose over time. Then, using a defined similarity metric we can measure a certain distance between motion descriptors. However, if we hold an efficient pose description and discriminant distance metric to compare these descriptors, a dynamic time warping embedded by this metric could be a good solution to compare sequences regardless to their speed variation.

Thus, to perform accurate pose/motion comparison someone can focus on defining a compact and efficient shape descriptor and its appropriate metric. This representation should satisfy certain constraints: (1) invariance to rotation/translation/anthropometry changes, (2) taking into account the non regularity of the connectivity, (3) fast to compute, (4) of-

fer an efficient temporal correspondence. Once this descriptor is designed and validated we can extend it to a motion description by comparing sequences using dynamic time warping.

3 Static and temporal shape retrieval

Sommaire

3.1 Introduction

Automatic estimation of 3D shape similarity from video is a very important factor for human motion analysis, but also a challenging task due to variations in body topology and the high dimensionality of the pose configuration space. We consider the problem of 3D shape similarity in 3D video sequences for different actors and motions. Most current approaches use conventional global features as a shape descriptor and define the shape similarity using L_2 distance. However, such methods are limited to coarse representation and do not sufficiently reflect the pose similarity of human perception. Besides, they are not allowing doing statistics on human body pose representations.

Thus we are interested in pose descriptors which represent and compare the pose information, in high dimentionality, using special geometric frameworks.

3.1.1 Existing geometric approaches

Modelling human shapes is a well studied problem in the literature, especially using 2D videos and static 3D models. Here we are interested in variety of techniques based on manifold analysis to represent and compare human poses.

Veeraraghavan et al. [125] propose the use of human silhouettes extracted from 2D video images as a representation of the pose. Silhouettes are then characterized as points on the shape space manifold. In another manifold shape space, Abdelkader et al. [6] represent each pose silhouette as a point on the shape space of closed curves. Other approaches use 2D visual tracker to extract skeleton representation from each frame. Indeed, Gong et al. [39] propose a Spatio-Temporal Manifold (STM) model to analyze non-linear multivariate time series with latent spatial structure of skeleton representations in a view invariant human action recognition system. This work is extended in [40], where a Kernelized Temporal Cut (KTC) is proposed, by incorporating Hilbert space embedding of distributions to handle the non-parametric and high dimensionality issues.

Other works can be found in the literature on the 3D shape similarity

for 3D object retrieval where surface-based descriptors are often extracted after a step of features detection. The advantage of these features is that their detection is invariant to non-rigid transformations. Tabia et al. [111] propose to extract arbitrarily closed curves amounting from feature points and compare these curves in the Riemannian closed curve shape space. Elkhoury et al. [31] extract the same closed curves but using heat-kernel distance in the curves extraction process.

3D Closed curves amouting 3D object extremities have been proven to be robust descriptors agains pose variation allowing retrieving similar shapes. Besides, the comparison of these curves within a Riemannian framework allowing their shape comparison regardless to other rigid transformations (translation, scaling, rotation), noise addition and elastic variations. This representation in the 3D domain in more appropriate to represent a 3D object in a compact way, unlike 2D silhouettes which represents only projections in a certain view of the object.

Thus, in the following we are proposing an approach based on 3D curves extracted from the mesh surface. However, these curves should capture the pose information in order to find similar poses of the same person in a 3D video. What are the best curves, to choose, for this task? How to compare these curve representations in term of shape?

We are trying to respond to all these issues in our proposed approach as the following.

3.1.2 Overview of our approach

In this chapter, we present a novel 3D human shape descriptor called Extremal Human Curve (EHC), extracted from body surface, robust to topology changes and invariant to rotation and scale. It is based on extremal features and geodesics between each pair of them. Every 3D frame will be represented by a collection of open curves whose comparison will be performed in a Riemannien Shape Space using an appropriate elastic metric. Our ultimate goal is to be able to perform reliable reduced representation based-geodesic curves for shape and pose similarity metric, which can be employed in several potential applications like video annotation and con-

catenation, activity analysis and behavior understanding. The overview of the proposed framework is shown in Figure 3.1.

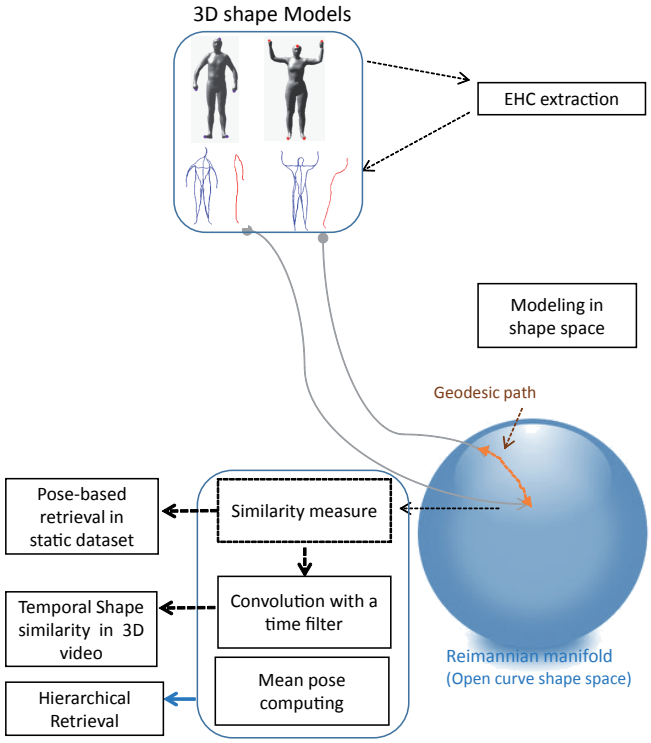


Figure 3.1 – *Overview of our proposed approach for static and temporal shape retrieval framework.*

In the rest of the chapter, we explain each step of our approach and present a quantitative analysis of the effectiveness of our descriptor for both 3D shape similarity in video and content-based pose retrieval for static shapes.

3.2 Extremal Human Curve

We aim to describe the body shape as a skeleton based shape representation. This skeleton will be extracted on the surface of the mesh by connecting features located on the extremities of the body. The main idea behind the use of this representation is to analyze pose variation with elastic deformation of the body, using representative curves on the surface.

3.2.1 Feature point detection

Feature points refer to the points of the surface located at the extremity of its prominent components. They are successfully used in many applications, including deformation transfer, mesh retrieval, texture mapping and segmentation. In our approach, feature points are used to represent a new pose descriptor based on curves connecting each two extremities. Several approaches have been proposed in the literature to extract feature points; Mortara et al. [85] select as feature points the vertices, where Gaussian curvature exceeds a given threshold. Unfortunately, this method can miss some feature points because of the threshold parameter and cannot resolve extraction on constant curvature areas. Katz et al. [60] develop an algorithm based on multidimensional scaling, in quadratic execution complexity. Another approach more robust, is proposed by Tierny et al. [116] to detect extremal points, based on geodesic distance evaluation. This approach is used successfully to detect the body extremities, since it is stable and invariant to geometrical transformations and model pose. The extraction process can be summarized as the following:

Let v_1 and v_2 be the most geodesic distant vertices on a connected triangulated surface M of a human body. These two vertices are the farthest on M, and can be computed using Tree Diameter algorithm (Lazarus et al. [71]). Now, let f_1 and f_2 be two scalar functions defined on each vertex v of the surface M as follows:

$$f_1(v) = g(v, v_1) \setminus f_2(v) = g(v, v_2) \tag{3.1}$$

where $g(x, y)$ is the geodesic distance between points x and y on the surface. Let E_1 and E_2 be respectively the sets of extrema vertices (minima and maxima) of f_1 and f_2 on M (calculated in a predefined neighborhood). We define the set of feature points of the surface of human body M as the intersection of E_1 and E_2. Concretely, we perform a crossed analysis in order to purge non-isolated extrema, as illustrated in Figure 3.2. The f_1 local extrema are displayed in blue color, f_2 local extrema are displayed in red color and feature points resulting from their intersection are displayed

in mallow color. Figure 3.3 shows different persons from three different datasets where feature extraction is stable despite change in shape, pose and clothing for each actor.

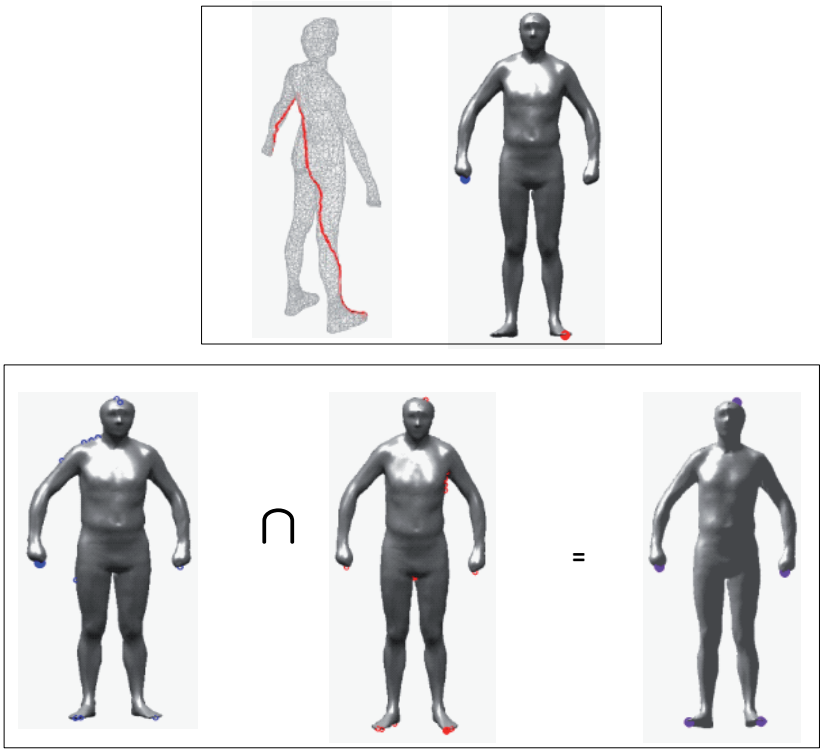


Figure 3.2 – *Extraction process of extremity points on the 3D human body. (top) The two distant vertices on the surface of the human body. (bottom) The set of local extrema and the result of their intersection.*

3.2.2 Body curves extraction

Let M be a body surface and $E = \{e_1, e_2, e_3, e_4, e_5\}$ a set of feature points on the body representing the output of the extraction process. Let β denotes the open curve on M which joints two feature points of M $\{e_i, e_j\}$. To obtain β, we seek for the geodesic path P_{ij}, whose length is shortest while passing through the surface of the mesh, between e_i and e_j. We repeat this step to extract extremal curves from the body surface ten times so that we do all possible paths between elements of E. As illustrated in the top of Figure 3.4, the body posture is approximated by using these extremal curves $M \sim \bigcup \beta_{ij}$, and we can categorize these curves into 5 categories (Figure 3.4 bottom):

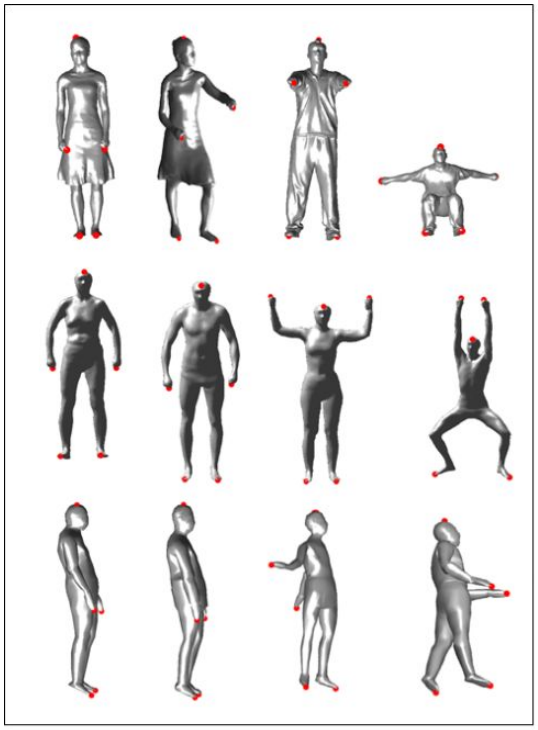


Figure 3.3 – *Extremity points extracted on different human body subjects in different poses.*

- Curves connecting hand and foot on the same side: for controlling the movement of the left/right half of the body.
- Curves between hands and between feet: for controlling the movement of the upper/lower body.
- Curves connecting crossed hand and foot: for controlling the movement of the crossed limbs.
- Curves between head and feet: for controlling the movement of right/left foot.
- Curves between head and hands: for controlling the movement of right/left hands.

Note that modeling objects with curves is recently carried out for several applications; Abdelkader et al. [6] use closed curves extracted from human silhouettes to characterize human poses in 2D videos for action recognition. Drira et al. [30] use open curves extracted from nose tip and face surface as a surface parametrization for 3D face recognition.

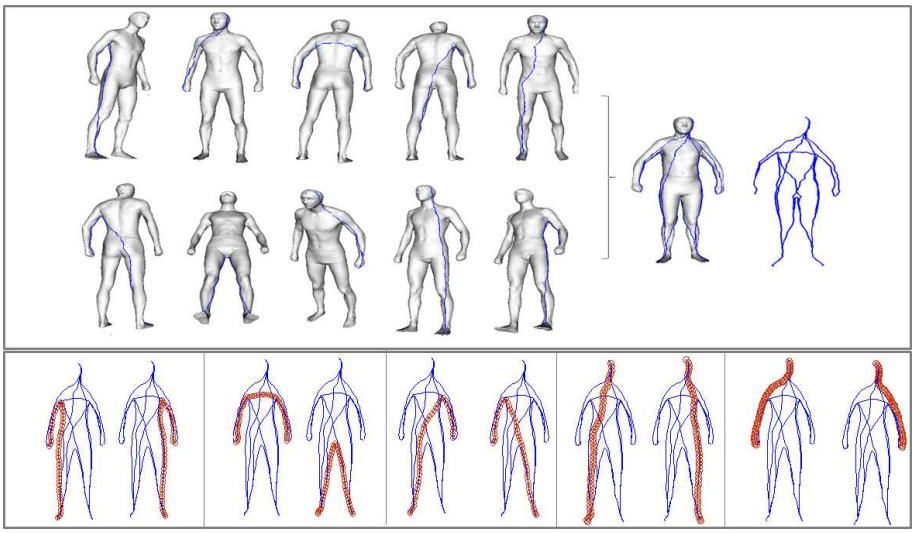


Figure 3.4 – *Body representation as a collection of extremal curves.*

In our approach, we have chosen to represent the body pose by a collection of curves for two reasons. Firstly, these curves connect limbs and give obviously a good representation of the body shape and pose, using a reduced representation of the mesh surface. Secondly, this representation allows studying the shape variation using Riemannian geometry by projecting these curves in the shape space of curves and using its elastic metric introduced by Joshi et al. [57].

3.3 Pose modeling in shape space

In order to compare the similarity between two human body postures, we must quantify the change of shape between correspondent curves. To do this, the metric used to compare shape of curves can be computed inside an open curve shape space.

In the last few years, many approaches have been developed to analyze shapes of 2-D curves. We can cite approaches based on Fourier descriptors, moments or the median axis. More recent works in this area consider a formal definition of shape spaces as a Riemannian manifold of infinite dimension on which they can use the classic tools for statistical analysis. The recent results of Michor et al. [83], Klassen et al. [64] and Yezzi et al. [153] show the efficiency of this approach for 2-D curves. Joshi et al. [57] have recently proposed a generalization of this work to the case of curves

defined in $\mathbb{R}^n$. We adopt this work to our problem since our 3-D curves are defined in $\mathbb{R}^3$.

3.3.1 Elastic distance

While human body is an elastic shape, its surface can be simply affected by a stretch (raising hand) or a shrinking (squatting). In order to analyze human curves independently to this elasticity, an elastic metric is needed within a shape space framework.

Let $\beta : I \to \mathbb{R}^3$, for $I = [0, 1]$, represents an extremal curve obtained as described above. To analyze its shape, we shall represent it mathematically using a *square-root velocity function* (SRVF), denoted by $q(t)$, according to:

$$q(t) = \frac{\dot{\beta}(t)}{\sqrt{\|\dot{\beta}(t)\|}} \,. \tag{3.2}$$

$q(t)$ is a special function introduced by Joshi et al.[57] that captures the shape of β and is particularly convenient for shape analysis.

Actually, the classical elastic metric for comparing shapes of curves becomes the $\mathbb{L}^2$-metric under the SRVF representation. This point is very important as it simplifies the calculus of elastic metric to the well-known calculus of functional analysis under the $\mathbb{L}^2$-metric. Hence, the SRV representation finds its potential for its ability for elastic matching. Actually, under $\mathbb{L}^2$-metric, the re-parametrization group acts by isometry on the manifold of q function (or SRV representation). This is not valid in the case of β. More formally, let β_1 and β_2 represent two open curves and $\Gamma = \{\gamma : [0,1] \to [0,1] / \gamma \text{ is a diffeomorphism } \}$ is the set of all re-parametrizations.

$$\|\beta_1 - \beta_2\| \neq \|\beta_1 \circ \gamma - \beta_2 \circ \gamma\|. \tag{3.3}$$

The use of SRV representation allows the re-parametrization group to act by isometry on the manifold of SRV representations. This point is very important as the curve matching could be done after re-parametrization. The change of parametrization before the matching is able to reduce the effect of stretching and/or biding of the curve.

We define the set (pres-shape space):

$$\mathcal{C} = \{q : I \to \mathbb{R}^3, \|q\| = 1\} \subset \mathbb{L}^2(I, \mathbb{R}^3) \,. \tag{3.4}$$

where using $\mathbb{L}^2$-metric on its tangent spaces, $\mathcal{C}$ becomes a Riemannian manifold.

Since the elements of $\mathcal{C}$ have a unit $\mathbb{L}^2$ norm, $\mathcal{C}$ is a hypersphere in the Hilbert space $\mathbb{L}^2(I, \mathbb{R}^3)$. In order to compare the shapes of two extremal curves, we can compute the distance between them in $\mathcal{C}$ under the chosen metric. This distance is defined to be the length of a geodesic connecting the two points in $\mathcal{C}$. Since $\mathcal{C}$ is a sphere, the geodesic length between any two points $q_1, q_2 \in \mathcal{C}$ is given by:

$$d_c(q_1, q_2) = cos^{-1}(\langle q_1, q_2 \rangle) \,, \tag{3.5}$$

and the geodesic path $\psi : [0, 1] \to \mathcal{C}$, is given by:

$$\psi(\tau) = \frac{1}{\sin(\theta)} \left(\sin((1-\tau)\theta) q_1 + \sin(\theta\tau) q_2\right) \,,$$

where $\theta = d_c(q_1, q_2)$.

We define the equivalent class containing q as:

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

to be equivalent from the perspective of shape analysis. The set of such equivalence classes, denoted by $\mathcal{S} \doteq \mathcal{C}/(SO(3) \times \Gamma)$ is called the *shape space* of open curves in $\mathbb{R}^3$. $\mathcal{S}$ inherits a Riemannian metric from the larger space $\mathcal{C}$ due to the quotient structure [106].

Thanks to SRV representation, the groups $\Gamma \times SO(3)$ act by isometries. This is a necessary condition to let the quotient space $\mathcal{S}$ inherit the metric from the pre-shape space $\mathcal{C}$.

To obtain geodesics and geodesic distances between elements of $\mathcal{S}$, one needs to solve the optimization problem:

$$(O^*, \gamma^*) = \arg \min_{(O,\gamma) \in SO(3) \times \Gamma} d_c(q_1, \sqrt{\dot{\gamma}} O(q_2 \circ \gamma)) \,.$$

For a fixed O in $SO(3)$, the optimization over Γ is done using Dynamic Programming. Similarly, for a fixed $\gamma \in \Gamma$, the optimization over $SO(3)$ is performed using Singular Value Decomposition method.

By iterating between these two, we can reach a solution for the joint optimization problem. Let $q_2^*(t) = \sqrt{\dot{\gamma^*}(t)} O^* q_2(\gamma^*(t)))$ be the optimal element of $[q_2]$, associated with the optimal rotation O^* and reparameterization γ^* of the second curve, then

$$d_s([q_1],[q_2]) \doteq d_c(q_1, q_2^*) \ , \tag{3.6}$$

and the shortest geodesic between $[q_1]$ and $[q_2]$ in $\mathcal{S}$ is given by:

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

,

where θ is now $d_s([q_1],[q_2])$.

In Figure 3.5, the geodesic path on the open curve shape space is illustrated between two given extremal curves.

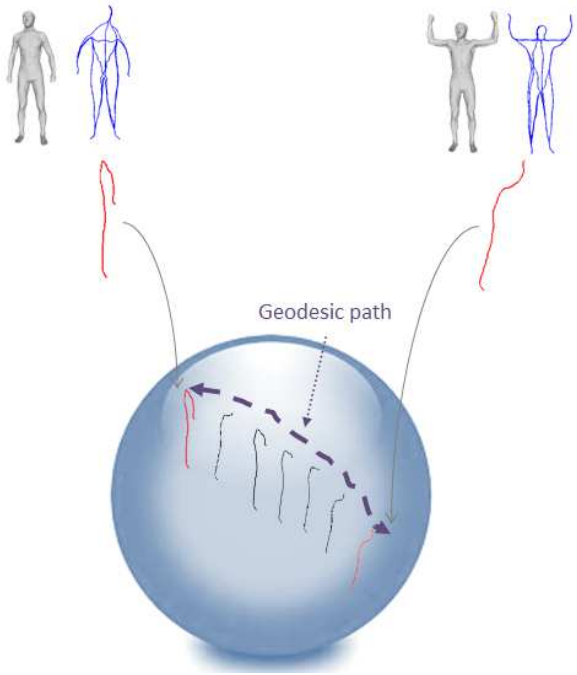


Figure 3.5 – *Geodesic path between extremal human curves of neutral pose with raised hands.*

Figure 3.6 shows examples of geodesic paths between each correspond-

ing two extremal curves, taken from two human bodies doing different poses. For the left model, the person's arm is down and for the right model it is raised. The geodesic path between each two curves is shown in the shape space. This evolution looks very natural under the elastic matching.

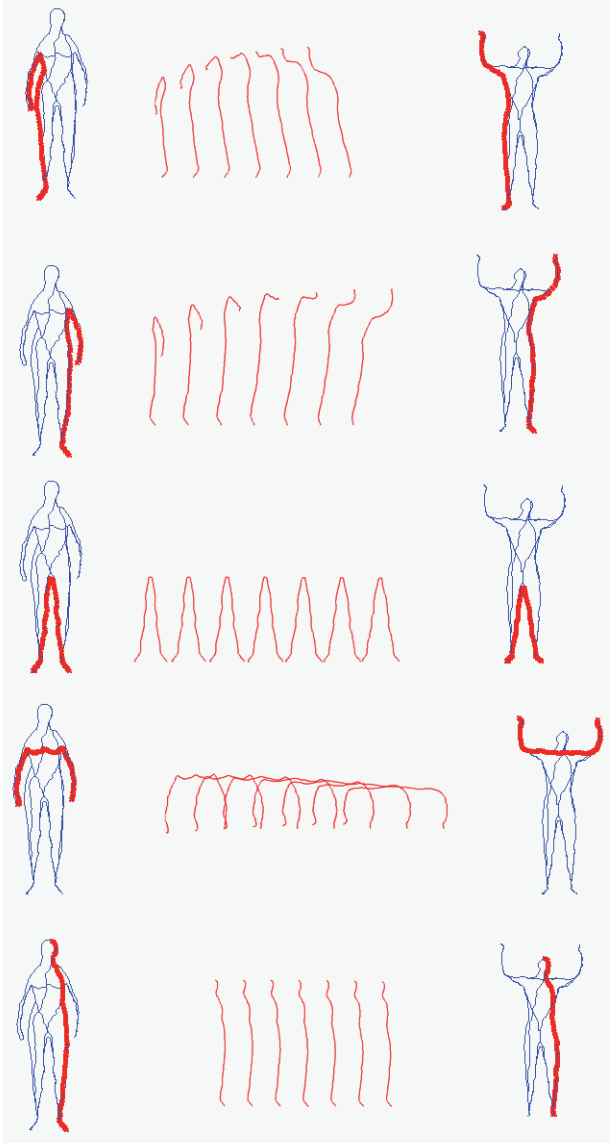


Figure 3.6 – *Examples of geodesic paths between different extremal curves.*

3.3.2 Static shape similarity

The elastic metric applied on extremal curve-based descriptors can be used to define a similarity measure. Given two 3D meshes x, y and their descriptors $x' = \{q_1^x, q_2^x, q_3^x, ..., q_N^x\}$ and $y' = \{q_1^y, q_2^y, q_3^y, ..., q_N^y\}$, the mesh-to-

mesh similarity can be represented by the curve pairwise distances and can be defined as follows:

$$s(x, y) = d(x', y') \,, \tag{3.7}$$

$$d(x', y') = \frac{\sum_{i=1}^{N} d(\beta_i^x, \beta_i^y)}{N} = \frac{\sum_{i=1}^{N} d_s(q_i^x, q_i^y)}{N}. \tag{3.8}$$

where N is the number of curves used to describe the mesh and d_s is the distance defined in Equation 3.6. The mean of curve distances between two descriptors captures the similarity between their mesh poses. In case of shape change in even one curve, the global distance is affected and it increases indicating that the poses are different. In order to have a global distance, an arithmetic distance can be computed in order to compare human poses.

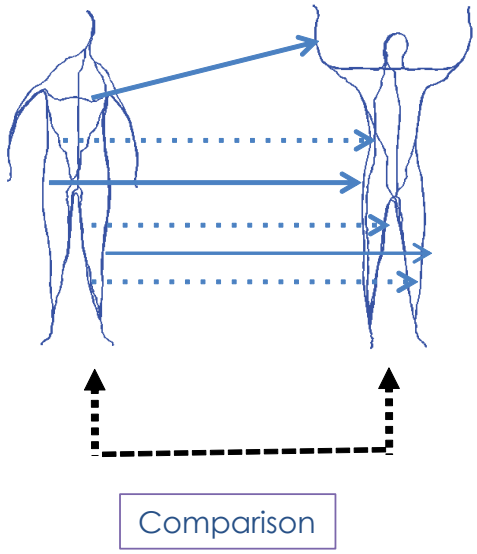


Figure 3.7 – *Shape similarity measure by pairwise curves comparisons.*

3.3.3 Average poses using statistics on the manifold

The use of EHC descriptor to represent the human pose by a collection of 3D open curves allows analyzing the human shape using the geometrical framework. It also allows computing some related statistics like "average" of several extremal human curves. Such an average, called Karcher mean, is introduced by Srivastava et al. [106]. It can be computed between dif-

ferent poses to represent the intermediate pose, or between similar poses done by several actors to represent a template of similar poses.

To compute the average of EHC representation, we only need to know how to compute an average for a collection of 3D open curves. The Riemannian structure defined on the shape space $\mathcal{S}$ enables us to perform such a statistical analysis for computing average and variance for each 3D open curve on body surface. The intrinsic average or the Karcher mean utilizes the intrinsic geometry of the manifold to define and compute a mean on that manifold.

For a given collection of extremal curves $\{\beta_1, \beta_2, ..., \beta_n\}$, with shape representations, $\{q_1, q_2, ..., q_n\}$, the Karcher mean $\overline{\mu}$ is defined as:

$$\overline{\mu} = \underset{[q]\in\mathcal{S}}{argmin} \sum_{i=1}^{n} d_s([q],[q_i])^2 \tag{3.9}$$

The principle of karcher mean computation is given by Algorithm 1. This computation is based on an iterative calculation which converges to the optimal solution which is the mean.

Algorithm 1: Karcher mean algorithm on shape space manifold

Input: $\{q_1, q_2 \cdots q_N\}$: shape representations of 3D open curves, $\epsilon = 0.5$, τ: threshold which is a very small number

Output: μ_j : mean of $\{q_i\}_{i=1:N}$

1- μ_0: initial estimate of Karcher mean, for example one could just take $\mu_0 = q_1$, j=0

repeat

  for $i \leftarrow 1$ **to** N **do**

    2- Compute $v_i = \frac{\theta_i}{sin(\theta_i)}(q_i^* - cos(\theta_i)\mu_j)$, where $cos(\theta_i) = \langle \mu_j, q_i^* \rangle$

    3- Compute the average direction $\overline{v} = \frac{1}{n}\sum_{i=1}^{n} v_i$

    4- Move μ_j in the direction of $\overline{v}$ by ϵ: $\mu_{j+1} = cos(\epsilon\|\overline{v}\|)\mu_j + sin(\epsilon\|\overline{v}\|)\frac{\overline{v}}{\|\overline{v}\|}$

  5- j=j+1

until $\|\overline{v}\| < \tau$;

An example of using the Karcher mean to compute average curve for

6 extremal human curves connecting hand and foot from the same side is shown in the top of Figure 3.8. In the bottom of this figure, we show the average EHC representation computed using the Karcher mean.

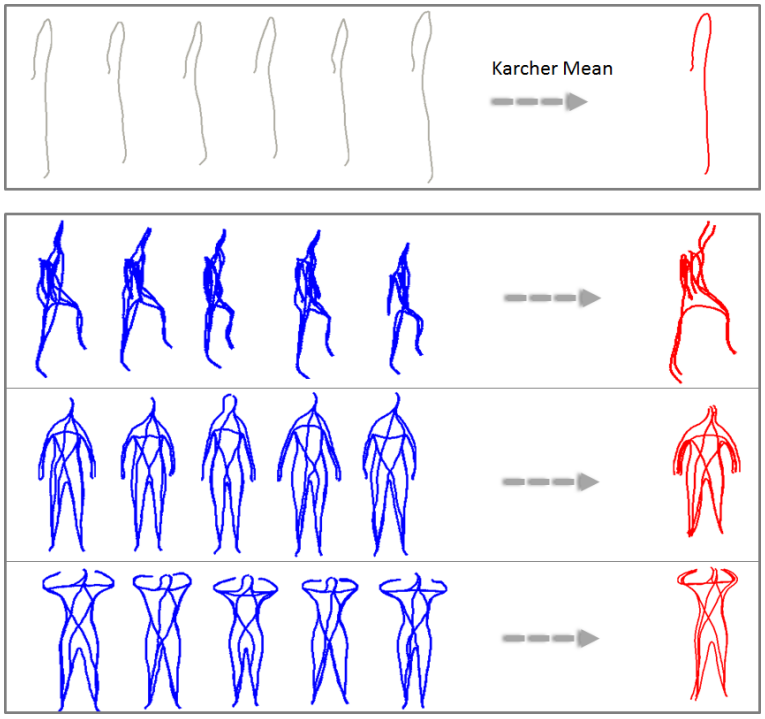


Figure 3.8 – *Examples of Karcher mean computation. (top) Mean curve for six extremal human curves: curve connecting hand and foot from the same side. (bottom) Examples of average poses computed using Karcher mean.*

3.4 Pose retrieval in 3D videos

As in a classical retrieval procedure, in response to a given query, an ordered list of responses that the algorithm found nearest to the query is given. Then to evaluate the algorithm, this ranked list is analyzed. Whatever the given query pose, the crucial point in the retrieval system is the notion of "similarity" employed to compare different objects.

Static shape similarity We are able to compare human poses using their extremal human curve descriptors and decide if two poses are similar or not. In this scenario, the query consists of a 3D human shape model in a given pose and the response is 3D human bodies which are more similar in pose to the query. We advocate the usage of the EHC to represent the 3D human shape model in a given pose and then comparison between each

pair of models using the elastic metric defined in 3.3.2. This system can find a number of utilities like pose-based searching and facilitate retrieval of efficient information as subjects in same poses in the database of 3D models scanned in different poses [8, 45].

Temporal shape similarity Pose retrieval in 3D videos is also useful in different applications. In fact, identifying frames with similar shape and pose can be used potentially for concatenative human motion synthesis. Concatenate existing 3D video sequences allows the construction of a novel character animation. A good descriptor that match correctly correspondent frames allows the synthesis of videos with smooth transitions and finding best frames to summarize the video. However, extension of static shape descriptor to include temporal motion information is required to remove the ambiguities inherent in static shape descriptor for comparing 3D frames in video sequences. Therefore, the static shape descriptor can be extended to the time domain by applying a simple time filter. This time filter is a way of incorporating motion in the similarity measure, as so-called temporal similarity, also used by Huang et al. [52]. The temporal similarity is presented in the following equation:

$$S_{ij}^{t} = S \otimes T(N_t) = \frac{1}{2N_t + 1} \sum_{k=-N_t}^{N_t} s(i+k, j+j) \tag{3.10}$$

where S is the frame-to-frame similarity matrix and $T(N_t)$ is a time filter having a window size $2N_t + 1$.

Time-filtering emphasizes the diagonal structure of the similarity matrix and reduces minima in the anti-diagonal direction resulting from motion and mirror ambiguities in the static shape descriptor.

3.5 Experimental evaluation

To show the practical relevance of our method, we perform an experimental evaluation on several databases (summarized in Table 3.1) and compare it to the most efficient descriptors of the state-of-the-art methods. We first evaluate our descriptor for shape similarity application over public static

shape database [45] and evaluate the results against Spherical Harmonic descriptor [61]. Secondly, we measure the efficiency of our descriptor to capture the shape similarity in 3D video sequences of different actors and motions from other public 3D synthetic [52] and real [130, 107] video databases. We evaluate this later against Temporal Shape Histogram [52], Multi-resolution Reeb-graph [53] and other classic shape descriptors, using provided Ground Truth.

Dataset	Motions/Poses	Number of frames
Dataset (1) [45]: 144 subjects (59 men/55 women)	18 static poses (1 neutral done by all subjects and 17 other different poses)	⊘
Dataset (2) [52]: 14 people (10 men and 4 women)	28 motions: sneak, walk (slow, fast, turn left/right, circle left/right, cool, cowboy, elderly, tired, macho, march, mickey, sexy,dainty), run (slow, fast,turn right/left, circle left/right), sprint, vogue, faint, rockn'roll, shoot.	392 seq, 39200 f (100 f per seq.)
Dataset (3) [130]: 3 people (2 men and 1 woman)	6 motions: 2×cran, 2×marche, 2×squat, 1×handstand, 1×samba, 1×swing.	1582 f (on average 226 ± 48 per seq.)
Dataset (4) [107]: Roxanne	Game character motion: walk	32 f

Table 3.1 – *Summarization of data used for all experimental tests.*

3.5.1 Extremal feature matching

The extraction and comparison of our curves require the identification of feature end-points as head, right/left hand and right/left foot, which is not affordable in practice. This requirement is important to perform the curve matching separately between models. In order to overcome this issue, our method is based on two benefits from the morphology of the human body. First, we deduce that geodesic path connecting each one of the hand end-points and the head end-point is shortest among all possible geodesics between the five end-points. Second, the geodesic path connecting right hand to left foot end-points or left hand to right foot end-points is the longest. The first observation allows to identify precisely the end-point corresponding to the head, the two end-points connected to this later corresponding to the hands without distinguishing between right and left. The second one allows the identification of the

couple of hand/foot as corresponding to same side of the body without distinguishing between right and left. A prior knowledge on the direction of the posture of the human body in the starting frame for video sequence has allowed to distinguish between left and right. Once the end-points are correctly detected from the starting frame in the video sequence, a simple algorithm of end-point tracking over time is performed.

3.5.2 Static shape similarity

The protocol and the dataset used to validate the experiments are firstly presented and then, the results following this protocol are analyzed and compared to those obtained by other approaches.

Evaluation methodology

To assess the performance of the EHC for static shape similarity, several experiments were performed on a statistical shape database [45]. This database, summarized in Table 3.1 (1^{st} row), is challenging for human body shape and pose retrieval as it is realistic shape database captured with a 3D laser scanner. It contains more than hundred subjects doing more than thirty different poses. We perform our descriptor on a subset of 338 shape models obtained from 144 subjects composed of 59 males and 55 females aged between 17 and 61 years. There are 18 consistent poses (p0, p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, p13, p16, p28, p29, p32). Some poses are illustrated in Figure 3.9. Each pose represents a class where at least 4 different subjects do the same pose.

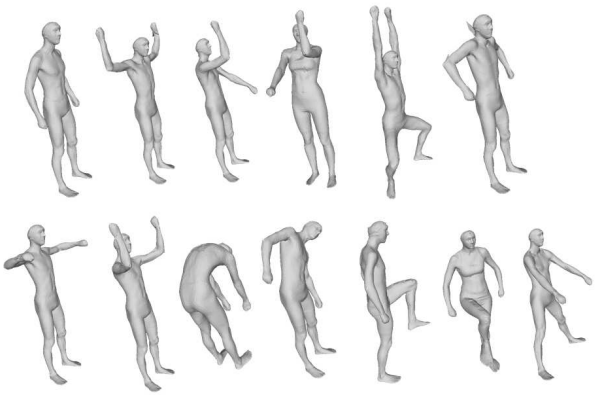


Figure 3.9 – *Example of body poses in the static human dataset [45].*

For evaluation, we use Recall/Precision plot in addition to the three statistics which indicate the percentage of the top K matches that belong to the same pose class as the query pose:

- The nearest neighbor statistic (NN): it provides an indication to how well a nearest neighbor classifier would perform (here $K = 1$).
- The first tier statistic (FT): it indicates the recall for the smallest K that could possibly include 100% of the models in the query class.
- The second tier statistic (ST): it provides the same type of result, but it is a little less stringent (i.e., K is twice as big).
- E-Measures: it is a composite measure of precision and recall for a fixed number of retrieved results.

We note here that these statistics will be used for static and video retrieval evaluations.

Curve selection

From five feature endpoints, we have extracted ten extremal curves representing the human body shape model. According to the human poses, extremal curves exhibit different shapes and some curves are more efficient to capture the shape similarity between two poses. The similarity between two shape models, doing two different poses, is represented by a vector of ten elastic distance values. Before all tests, we analyze the performance of all possible combinations of curves on the shape similarity measurements. A Sequential Forward Selection method, applied on elastic distance values and coupled with ST statistic, has been used to select the best combination of curves among all possible ones (1013 combinations according to Eq. 3.11):

$$\sum_{k=2}^{n} C_n^k = \sum_{k=2}^{n} \frac{n!}{k!(n-k)!} \tag{3.11}$$

where n represents the number of curves and equals to 10.
Experiment of pose-based retrieval on the dataset (1) [45] shows that the best combination is obtained by the five curves: right hand to right foot,

left hand to left foot, left hand to right hand, left foot to right foot, and head to the right foot (Figure 3.10).

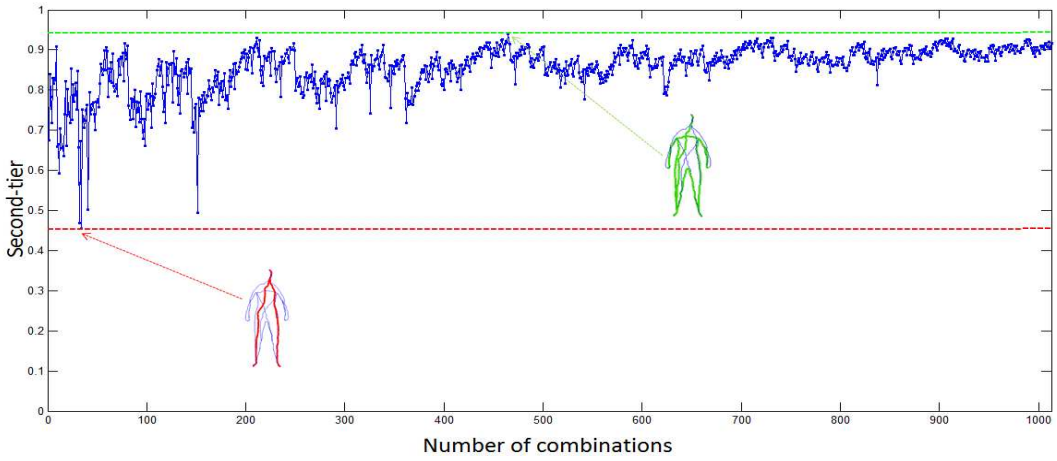


Figure 3.10 – *Second-Tier statistic for all combinations of curves. The best combination is obtained by 5 curves (green) and the worst combination is obtained by 2 curves (red).*

The selected five curves seem to be the most stable ones and they are sufficient to represent at best the body like a skeleton on the surface. Besides, they are the most shape independent curves from the 10 initial ones. Therefore, the elimination of five curves allows to eliminate the ambiguity due to the redundancy of some curves on the body parts.

Result analysis

The self similarity matrix obtained from the mean elastic distance of the five selected curves is shown in the Figure 3.11.

This matrix demonstrates that similar poses have a small distance (cold color) and that this distance increases with the degree of the change between poses (hot color). This allows pose classification or pose retrieval by comparing models using their extremal curve representation and the elastic metric.

From a quantitative point of view, we present the Recall/Precision plot obtained by EHC compared to the popular Spherical Harmonic (SH) descriptor with optimal parameter setting ($N_s = 32$ and $N_b = 16$) [9]. This plot and accuracy rates (NN, FT and ST) reported in Table 3.2 show that our approach provides better retrieval precision. EHC using only the five

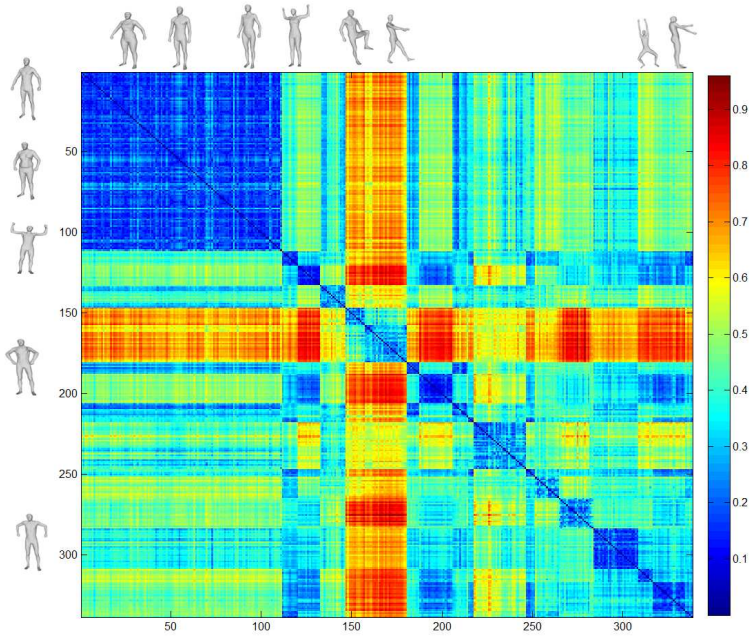


Figure 3.11 – *Confusion similarity matrix. The matrix contains pose similarity computation between models of a 3D humans in different poses. More the color is cold more the two poses are similar.*

selected curves outperforms SH and EHC using the 10 curves to retrieve models with the same pose.

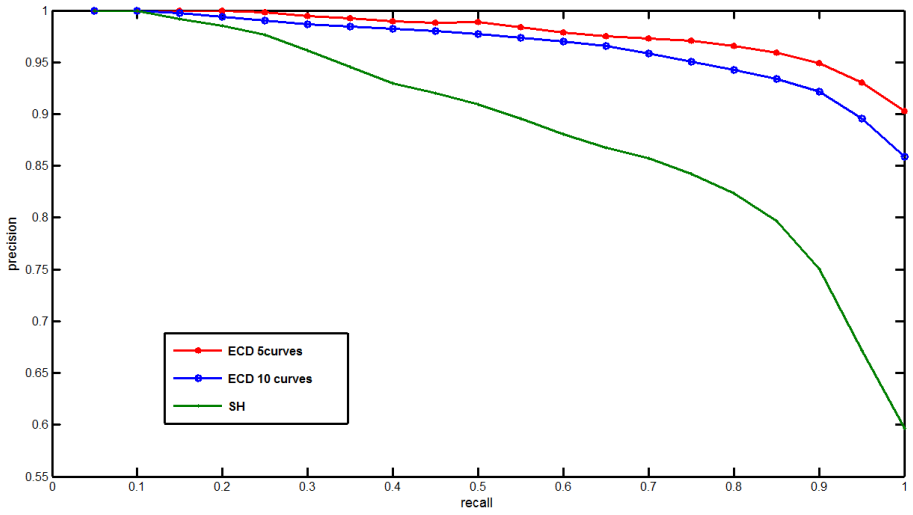


Figure 3.12 – *Precision-recall plot for pose-based retrieval.*

Approach	NN(%)	FT(%)	ST(%)	E-Measure(%)
SH	71.0	57.9	75.5	41.3
EHC 10 curves	80.3	75.5	85.2	42.5
EHC 5 curves	**84.8**	**77.2**	**89.1**	**43.0**

Table 3.2 – *Retrieval statistics for pose based retrieval experiment*

Note finally that the accuracies of retrieval ranks for some poses are relatively low. Such ambiguities can be noticed in the case of comparison between neutral pose and a pose where subjects just twist their body to the left, or twist their torso to look around.

3.5.3 Temporal shape similarity for 3D video sequences

We firstly present the protocol and the dataset used in these experiments and then, the results following this protocol are analyzed and compared to the most relevant state-of-the-art approaches.

Evaluation methodology

The recognition performance of the our descriptor using temporal filter is evaluated using a ground-truth dataset from a synthetic 3D video sequences proposed by Huang et al. [52] and a real captured 3D video sequences of people [130]. As described in Table 3.1 (2^{nd} raw), the synthetic data is obtained by 14 people (10 men and 4 women) performing 28 motions. Each sequence is composed of 100 frames and the whole dataset contains a total of 39200 frames.

Given the known correspondences, a temporal ground-truth similarity is computed between each two surfaces. The known correspondence is only used to compute this ground truth similarity. Having two meshes X and Y with N vertices $x_i \in X$ and $y_i \in Y$, a temporal-ground truth C_T is computed by combining a shape similarity C_p and a temporal similarity C_v as follows:

$$\begin{aligned} C_T(X,Y) &= (1-\alpha)C_p(x_i,y_j) + \alpha C_v(x_i,y_j) \\ C_p(X,Y) &= \tfrac{1}{N}\textstyle\sum_{k=1}^{N} d(x_i,y_j) \\ C_v(X,Y) &= \tfrac{1}{N}\textstyle\sum_{k=1}^{N} d(\dot{x}_i,\dot{y}_j) \end{aligned} \tag{3.12}$$

where $\dot{x}_i$ and $\dot{y}_j$ are the derivation of x and y between next and current frame and d is an Euclidean distance. the parameter α is used to balance the equation and it is set to 0.5 . In order to identify frames as similar or dissimilar, the temporal ground truth similarity matrix is binarized using a threshold set to 0.3 similarly to Huang et al. [52].

Finally, similarity performances are evaluated using the Receiver-Operator-Characteristic (ROC) curves, created by plotting the fraction of true-positive rate (TPR) against the fraction of false-positive rate (FPR), at various threshold settings. The true and false dissimilarity compare the predicted similarity between two frames, against the ground-truth similarity.

An example of self-similarity matrix computed using temporal ground-truth similarity, static and temporal descriptors are shown in Figure 3.13. This figure illustrates also the effect of time filtering with increasing temporal window size for EHC descriptors on a periodic walking motion.

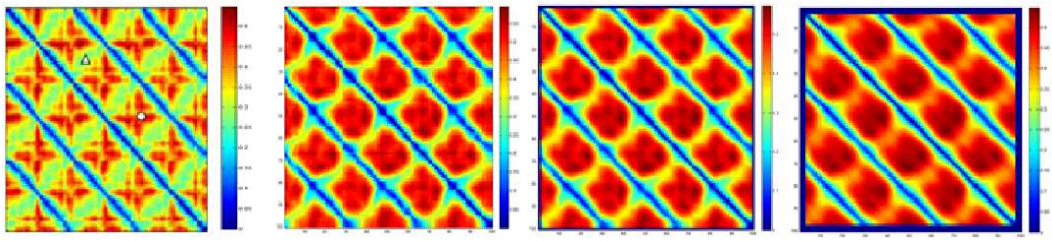


Figure 3.13 – *Similarity measure for "Fast Walk" motion in a straight line compared with itself. Coldest colors indicate most similar frames.* 1^{st} *matrix: temporal Ground-Truth (TGT).* 2^{nd}, 3^{rd} *and* 4^{th} *matrix: self-similarity matrix computed with EHC using temporal similarity with a window size 3, 5 and 7 respectively.*

Result analysis

A comparison is made between our Extremal Human Curve using time filter (noted *EHCT*) and several descriptors from the state-of-the-art: Shape Distribution (SD) , Spin Image (SI) , Spherical Harmonics Representation (SHR), two Shape-flow descriptors, the global / local frame alignment Shape Histograms (SHvrG / SHvrS) (Huang et al. [52]) and Reeb-Graph as skeleton based shape descriptors (aMRG) (Tung et al. [121]). Huang et al. [52] evaluated the performances of all these descriptors for the purpose of shape similarity.

The effectiveness of our descriptor have been evaluated by varying temporal window and comparing it to the most relevant state-of-the-art descriptors [52] as shown in the plot of ROC curves in Figure 3.14.

Several observations can be made on the obtained results: (i) Our

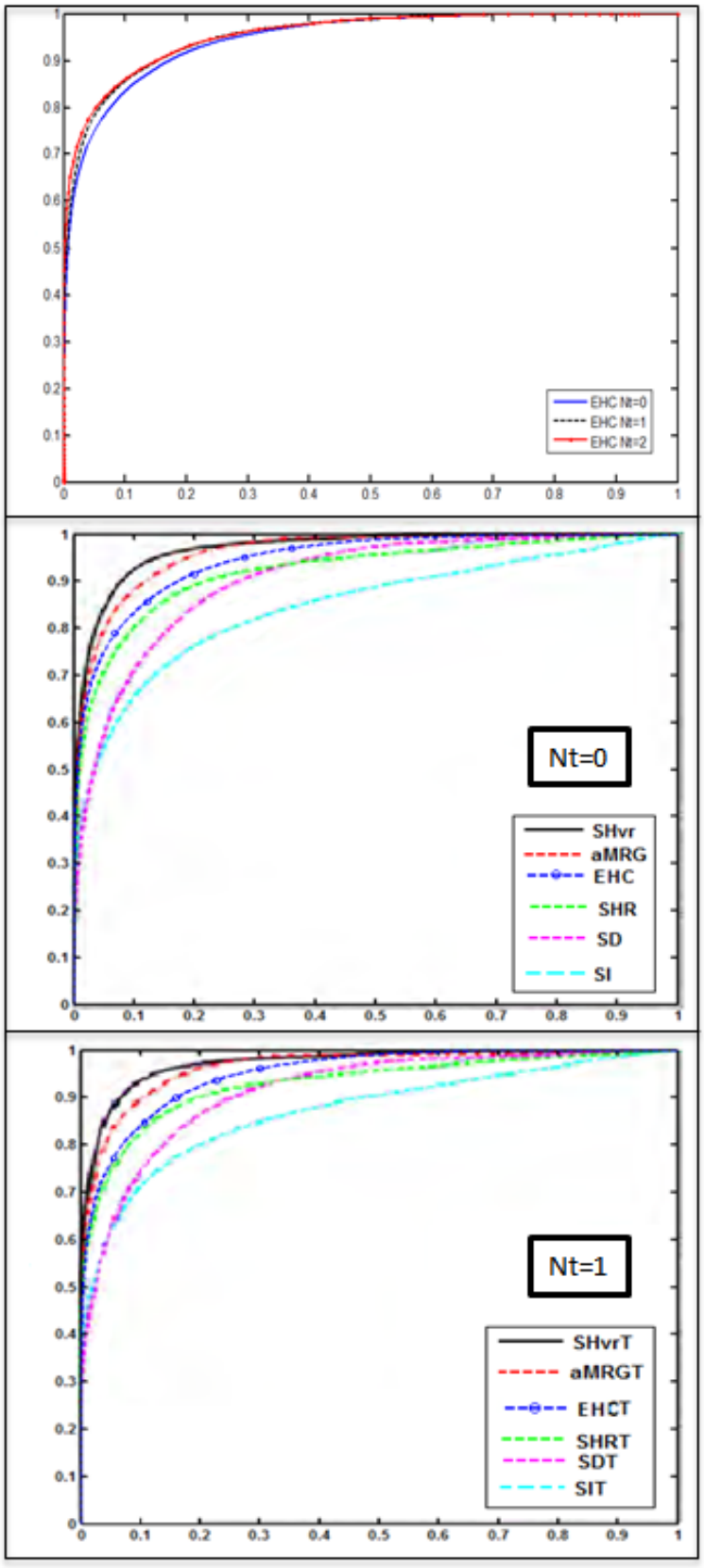


Figure 3.14 – *Evaluation of ROC curve for static and time-filtered descriptors on self-similarity across 14 people doing 28 motions. From top to bottom: ROC curves obtained by our EHCT descriptor with three different values of windows size Nt, ROC curve obtained by our EHC descriptor compared to different algorithms and ROC curves obtained with Nt= 1.*

descriptor outperforms classic shape descriptors (SI, SHR, SD) and shows competitive results with SHvrS and aMRG. We also notice that recognition performance of EHCT increases with the increase of the window size of time-filter like any other descriptor. In fact, time-filter reduces the minima in the anti-diagonal direction, resulting from motion in the static descriptor (Figure 3.14). Multiframe shape-flow matching required in SHvrS allows the descriptor to be more robust but the computational cost will increase by the size of selected time window.
(ii) EHC descriptor by its simple representation, demonstrates a comparable recognition performance to aMRG. It is efficient as the curve extraction is instantaneous and robust as the curve representation is invariant to elastic and geometric changes thanks to the use of the elastic metric.
(iii) The result analysis for each motion shows that EHC gives a smooth rates that are stable and not affected by the complexity of the motion. Such complex motions are rockn'roll, vogue dance, faint, shot arm (Figure 3.15). However, this is not the case for SHvrS where performance recognition falls suddenly with complex motions as illustrated in Figure 3.16.

We also applied the time filtering on similarities, obtained by EHC descriptors, on two real captured 3D video sequences of people. The first sequence is extracted from the dataset of Valsic et al. [130] described in Table 3.1 (3^{rd} row). The second one is extracted from real data reconstructed by multiple camera video which is presented by Starck et al. [107] and described in Table 3.1 (4^{th} row).

Inter-person similarity across two people in a walking motion with an example similarity curve are shown in Figure 3.17 (a). Our temporal similarity measure identifies correctly similar frames across different people. These similar frames are located in the minima of the similarity curve. In addition, despite the topology change and the reconstruction noise, as shown in Figure 3.17 (b), our algorithm succeed to identify correctly frames which are similar to the query.

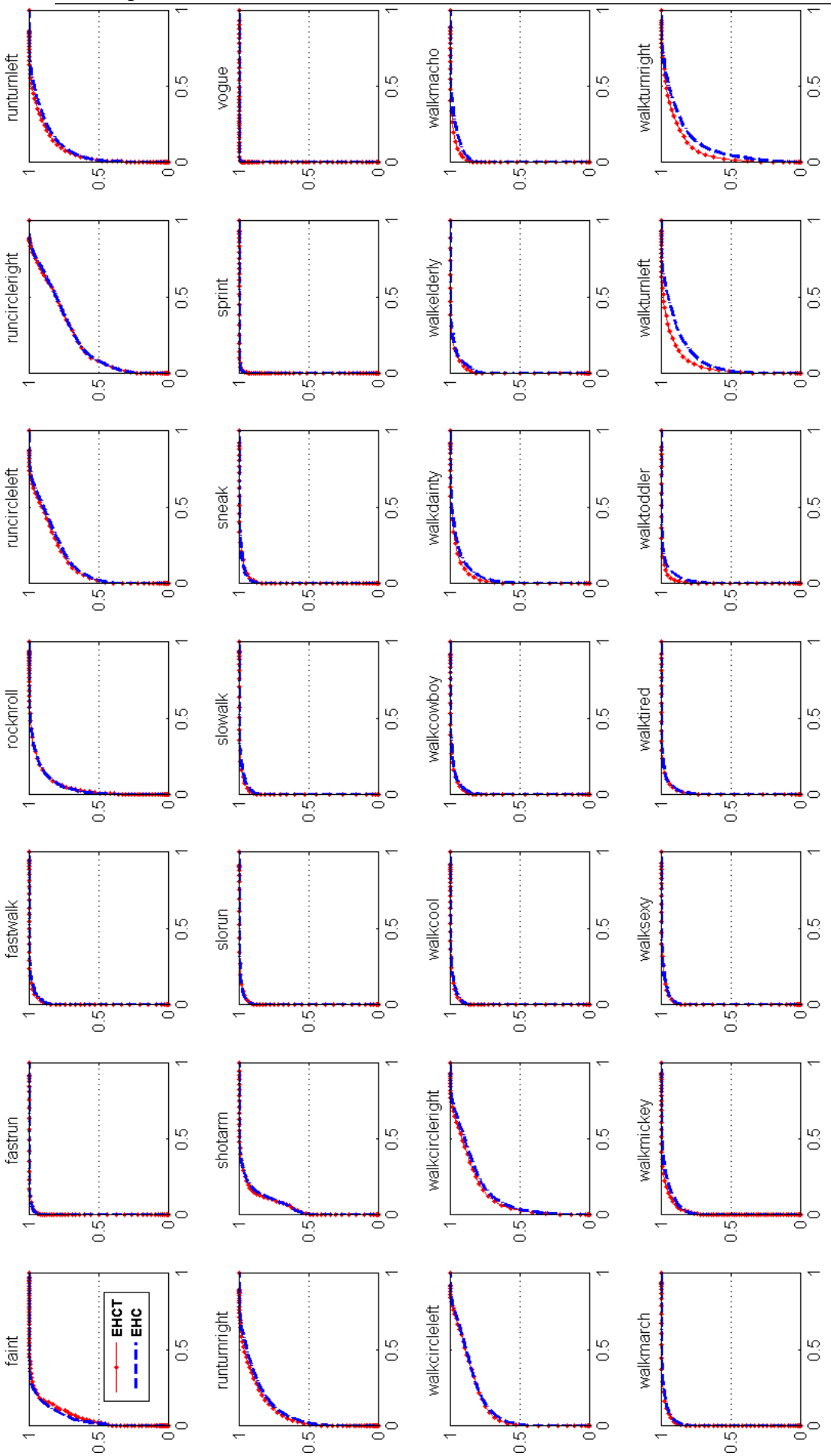
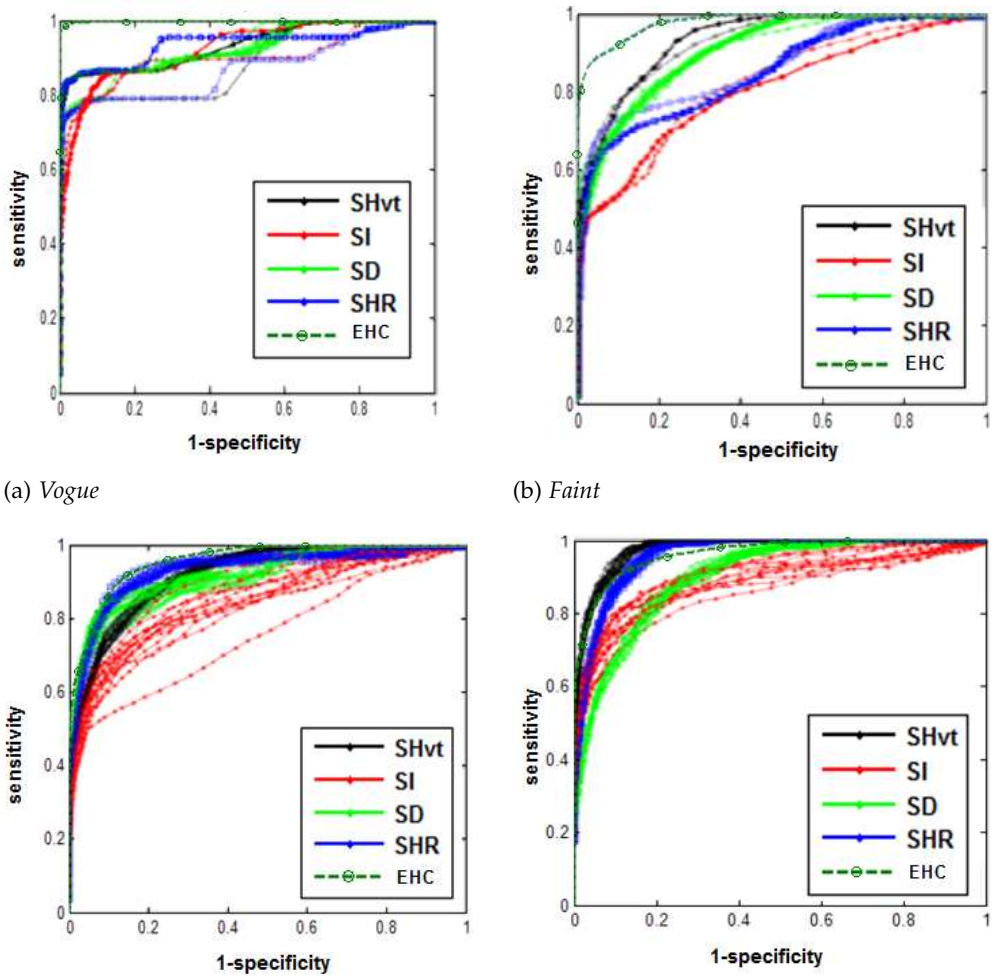


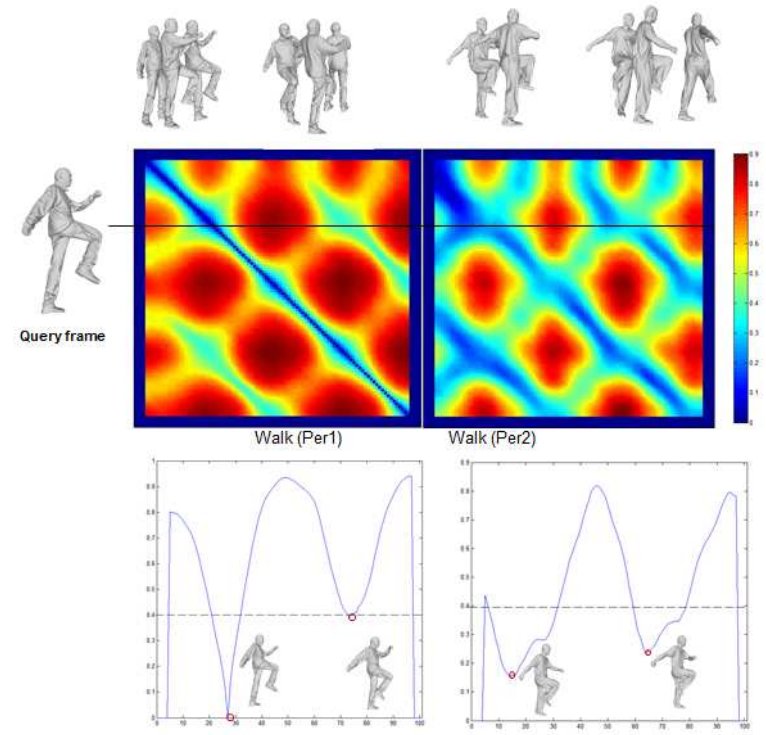
Figure 3.15 – *Evaluation of EHCT for Nt=0 and Nt=1. ROC performance for 28 motions across 14 people.*



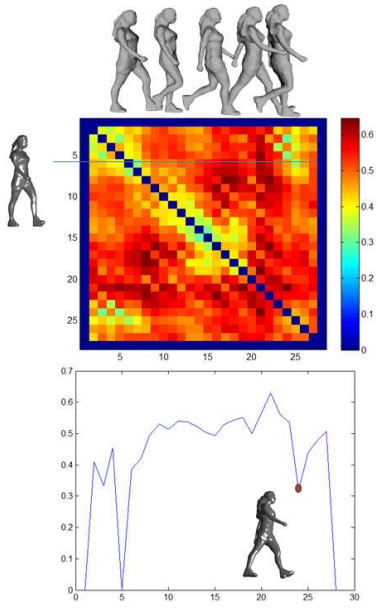
(a) *Vogue* (b) *Faint*

(c) *Shot arm* (d) *Rockn'roll*

Figure 3.16 – *Evaluation of ROC curves for complex motions with Nt=3.*



(a)



(b)

Figure 3.17 – *Inter-person similarity measure for real sequences. Similarity matrix, curve and example frames for (a) walk motion across two actors [130] (b) walk motion for Roxanne Game Character Walk [107].*

3.5.4 Hierarchical data retrieval

For a mesh model of 1 MB size, the size of the 3D video sequence grows linearly of 1 MB per frame. Hence, the video retrieval becomes very difficult in long sequences.

Within our framework, we propose to combine the data clustering approach with the content-based retrieval in order to perform an hierarchical retrieval.

The clustering approach gathers models with similar poses in clusters. If we consider the element of cluster as a pose, clusters are firstly performed over the entire sequence in order to gather frames with similar poses and then a template model is obtained for each cluster by computing its average using Karcher mean algorithm 1 as described in section 3.3.3. The retrieval system can then be described as an hierarchical structure composed of two levels, the first one containing templates and the second one containing all models of the dataset. In view of this structure, a natural way is to start at the top, compare the query with the template of each cluster and proceed down the branch that leads to the closest shape.

We reconsider the same experiments for pose based retrieval in section 3.5.2 by applying the hierarchical approach to the dataset summarized in Table 3.1 (1^{st} raw) . Each query model is compared to each one of the template models representing the clusters. The elastic measure values are used to generate a confusion matrix for all classes of poses. The matrix of comparison in the first level (model-template comparison), is shown in Figure 3.18.

If we compare this matrix to that already obtained for the same dataset without the use of hierarchical clustering (Figure 3.11), we can easily notice the effectiveness of our approach. The main advantage of this approach is the reduction of computation time which complexity pass from n to $log(n)$ while keeping relevant information. Retrieval performances obtained from this matrix for FT, ST and E-Measure are respectively 84.5%, 88.2% and 43.6%. Comparing these results to those in Table 3.2, a small improvement is achieved for classic retrieval scenario in term of second tier.

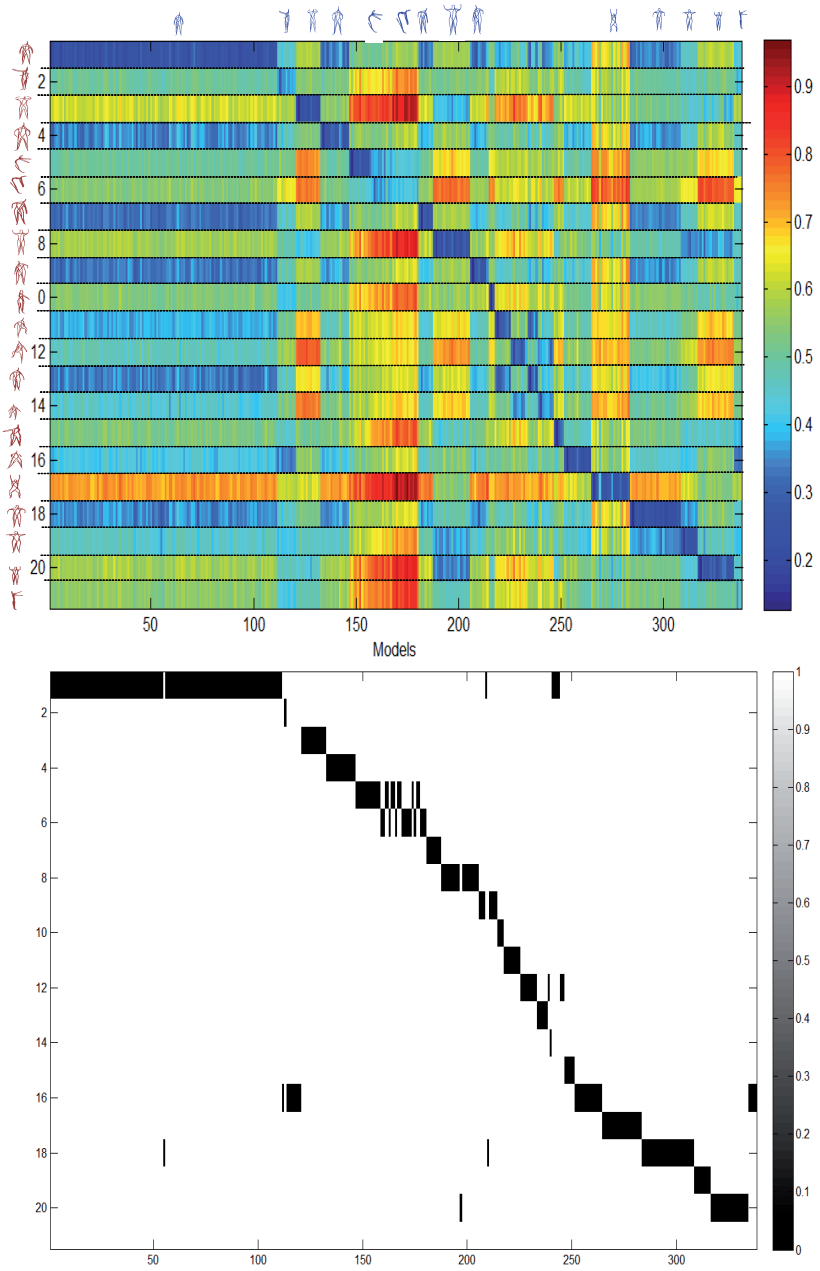


Figure 3.18 – *Similarity matrix and its binarization for template pose of each class against all models in the dataset.*

In term of a template based pose classification, the obtained accuracy is about 90.24%. Models of the class ♯2 are the most ones affected by misclassification and are assigned to the class ♯16. Looking at these two classes, we perceive that their poses are close to each other, both represent people with hands outstretched. The only difference is that one does with open legs and the other with closed ones.

3.6 Discussion

The advantages of using EHC to represent human pose and motion in our approach include: (1) invariance to affine transformation (2) possibility to compute mean poses (3) the use of well defined measure for pose comparison in Reimannian manifold and (4) possibility of retrieving frames in 3D videos by adding time filtering.

However, this representation has some limitations. Firstly, EHC depends on the accuracy of extremities (head and limbs) extraction and on the definition of the path connecting end-points. In fact, the extraction of end-points and extremal curves is based on the definition of geodesic distance between each pair of curves. Thus, geodesic distances play an important role in our geometric representation of the human body shape. However, they are sensitive to significant topology changes as shown in Figure 3.19. In this figure, only 4 extremities are successfully detected and the left hand extremity is missed. Thus, information about position of this hand is lost. Other strategies could be investigated for the extremities extraction step and shortest path detection on the mesh by using diffusion or commute time distances as presented by Elkhoury et al. [31] and Sun et al. [109].

Secondly, we note that our curve extraction can be sensitive to loose clothes. For example, the mesh represented in Figure 3.19 shows a girl wearing a skirt and the shape of the curve connecting her feet is different from the same curve extracted on her mesh when she is wearing a trouser. This problem will be even more critical if she wears a long skirt.

Thirdly, a prior knowledge on the direction of the posture of the hu-

man body for the starting frame in video sequence is used to distinguish between left/right hand and foot. Other feature matching algorithms, as proposed by [158], could be used in future work to correctly identify the right from the left side.

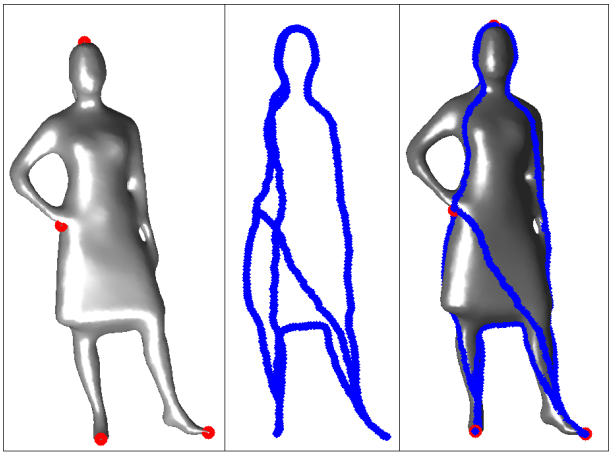


Figure 3.19 – *Example of failed extraction of EHC in presence of a topological change.*

3.7 Conclusion

In this chapter, a novel 3D shape descriptor for the purpose of 3D human shape similarity has been proposed. Some general rules for the extraction of extremal curves as geometric invariant descriptors of body shape within Riemannian Shape Space framework have been discussed. Body shape in a given pose is firstly represented as a set of geodesic curves extracted from shape surface using extremal feature points. Then, an elastic metric is calculated as a pairwise descriptor distance in the Shape Space, allowing the comparison between two shape models in order to estimate their similarity. The quality of our descriptor regarding the recognition performance of pose retrieval in 3D video was analyzed and verified also with respect to another related recent techniques. Results obtained from extensive experiments have clearly shown the promising performance of the proposed descriptor and also the advantages of using such reduced representation of the shape model.

Since the proposed descriptor showed good performances in human body pose retrieval in 3D video, we investigate its usage for further related

applications such as motion retrieval and video summarization. In the next chapter, these issues are discusses and experimented.

3D human motion retrieval

4

Sommaire

4.1 Introduction

Large motion databases of 3D videos have emerged in the past few years. In order to better reuse the recorded data, an efficient approach of motion retrieval in large motion databases is still a challenging problem. Some of these data have long sequences which capture natural behavior over extended periods of time. Thus, segmentation is an important preprocessing to divide the whole 3D video data into small sub-sequences which are meaningful and manageable. It is the first step towards automatic retrieval system.

In this chapter, we are proposing a new approach for the task of video segmentation and comparison between motion segments for video retrieval. The overview of the proposed approach is sketched in Figure 4.1. After extracting EHC from each frame, the sequence is segmented into clips. These clips are then represented in the shape space manifold by trajectories. Finally motion comparison is performed by comparing two trajectories in this space using Dynamic Time Warping (DTW) algorithm. We also introduce the notion of mean clip computation which allows performing video summarization and hierarchical motion retrieval.

4.2 Motion segmentation and matching

Based on our EHC representation, presented in the previous chapter, it is possible to compare two video sequences by matching all pairwise correspondent extremal curves using the geodesic distance in the shape space. However, a sequence of human action can be composed of several distinct actions, and each one can be repeated several times. Therefore, the motion segmentation using EHC can play an important role in the dynamic matching by dividing continuous sequences into clips.

4.2.1 Motion segmentation

We propose an approach fully automatic to segment a 3D video efficiently without making neither thresholding step nor assumption on the motion's nature. In motion segmentation, the purpose is to split automatically the

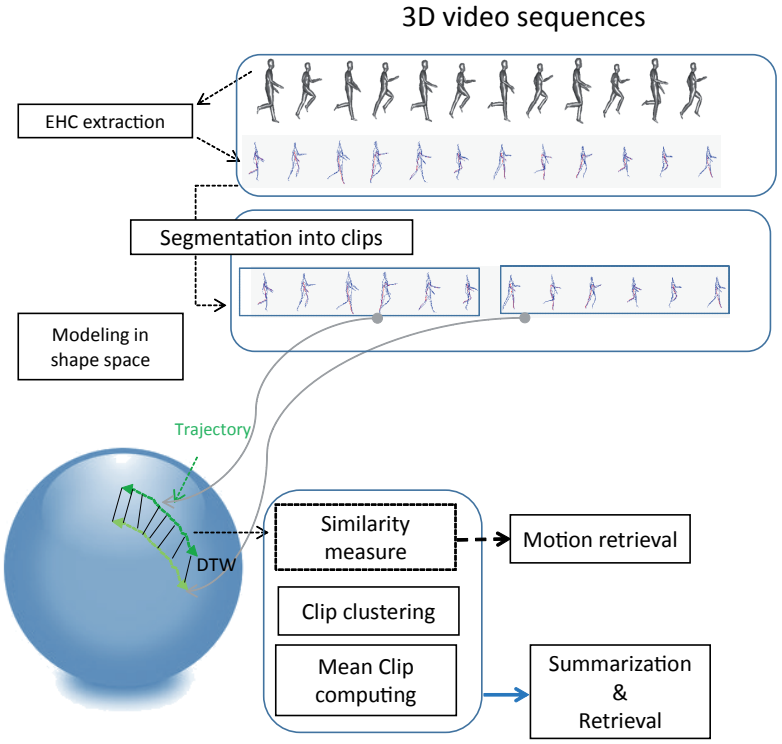


Figure 4.1 – *Overview of our proposed approach for 3D human motion retrieval framework.*

continuous sequence into segments which exhibit basic movements, called clips. As we need to extract meaningful clips, the segmentation should be overly fine and can be considered as finding the alphabet of the motion. For a meaningful segmentation, motion speed is an important factor [25]. In fact, when human changes motion type or direction, the motion speed becomes small and this results in dips in velocity. We exploit this latter by finding the local minima for the change in type of motion and local maxima for the change in direction. The extrema detected on velocity curve should be selected as segment points (see Figure 4.2). We show frames detected as maxima (the actor changes the foot's direction) on the top of the plot, and frames detected as minima (the actor raise the other foot) on the bottom. In our approach, we consider only the change in type of motion as a meaningful clip. Thus, clips with slight variations and a small number of frames are avoided.

Note that optimum local minimum, that detect precise break points where the motion changes, should be selected in a predefined neighbour-

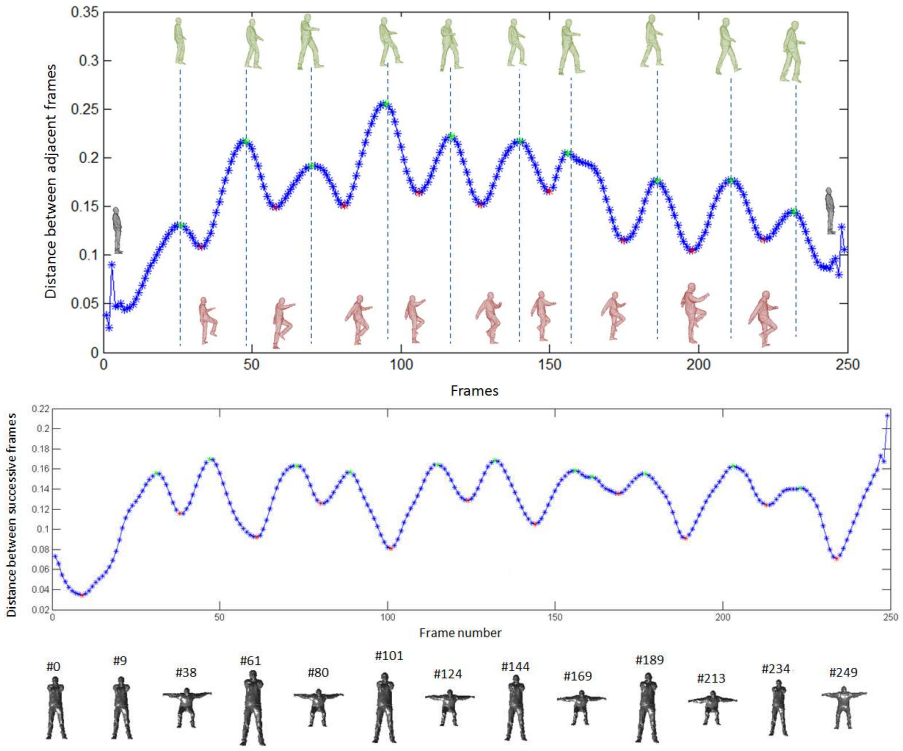


Figure 4.2 – *Segmentation of a 3D sequence into motion clips. Feature vector and detected frames as local extrema are presented at the top of the figure and detected frames as minima are at the bottom.*

hood. For this raison, we fix a size of window to test the efficiency of the local minimum in this condition. To calculate the speed variation, distance between each two successive EHC in the sequence is computed. The variations of the sequence are represented in a vector of speed and a further smoothing filter is applied to obtain the final degree of motion vector.

4.2.2 Clip matching

To seek for similar clips, we need to encode motions in a specific representation that we can compare regardless to certain variations. In fact, two motions are considered similar even if there are changes in the shape of the actor and the speed of the action execution. This problem is similar to time-series retrieval where a distance metric is used to look for, in a database, the sequences whose distance to the query is below a threshold value. Each clip is represented as a temporal sequence of human poses, characterized by EHC representation associated to shape model. Then, extremal curves are tracked in each sequence to characterize a trajectory of

each curve in the shape space as illustrated in Figure 4.3 (top). Finally, the trajectories of each curve are matched and a similarity score is obtained. However, due to the variation in execution rates while doing the same motion, two trajectories do not necessarily have the same length. Therefore, a temporal alignment of these trajectories is crucial before computing the global similarity measure, as shown in Figure 4.3 (bottom).

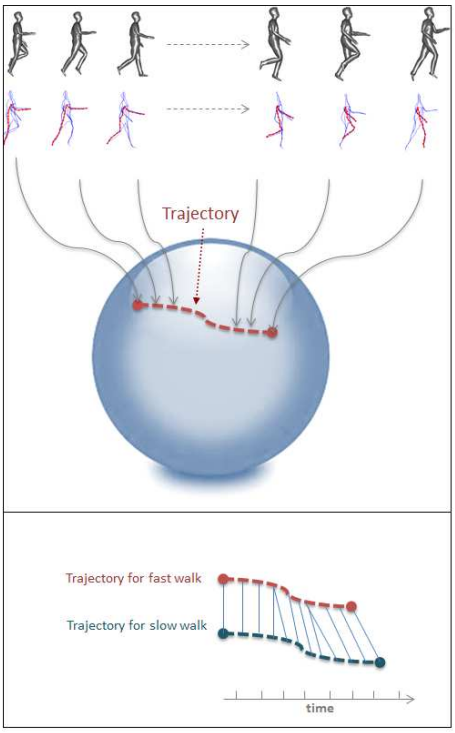


Figure 4.3 – *Graphical illustration of a sequence, obtained during a walking action, as trajectory on shape space manifold (top). Alignment process between trajectories of same curve index using DTW (bottom).*

In order to solve the temporal variation problem, we use DTW algorithm (Giorgino et al. [37]). This algorithm is used to find optimal non-linear warping function to match a given time-series with another one, while adhering to certain restrictions such as the monotonicity of the warping in the time domain. The optimization process is usually performed using dynamic programming approaches given a measure of similarity between the features of the two sequences at different time instants. Since DTW can operate with any measure of similarity between different temporal features, we adapt it to features that reside on Riemannian manifolds. The global accumulated costs along the path define a global

distance between the query clip and the motion segments found in the database.

Assume two clips A and B are denoted as follows:

$$\begin{aligned} A &= a^1, \cdots, a^K \\ B &= b^1, \cdots, b^L \end{aligned} \tag{4.1}$$

where K and L represent the number of frames in A and B. a_i and b_i are EHC representations at each frame number i. The distance between two frames a_i and b_j is given by :

$$d(i,j) = d_s(a_i, b_j) \tag{4.2}$$

where d_s is the geodesic distance defined to compute similarity measure between two EHC descriptors.

Then, the cost function cost(i,j) is defined as follows:

$$cost(i,j) = \begin{cases} d(1,1), & \text{if i=j=1} \\ d(i,j) + & \\ min(cost(i,j-1), cost(i-1,j), cost(i-1,j-1)) & \text{otherwise} \end{cases} \tag{4.3}$$

The final similarity measure between clip A and B is given by :

$$D(A,B) = \frac{cost(L,K)}{\sqrt{L^2+K^2}} \tag{4.4}$$

Since the cost is a function of the sequence lengths, this distance is normalized by $\sqrt{L^2+K^2}$. The lower the D is, the more similar the sequences are.

4.2.3 Average clip

Based on the two algorithms, Karcher mean and DTW, we can extend the notion of mean of a set of human poses to the mean of trajectories of poses in order to compute an "average" of several clips.

Let N be the number of clips represented by N trajectories $T_1, T_2 \cdots T_N$. For a specific human curve index, we look for the mean trajectory that has

the minimum distance to the all N trajectories. As shown in Algorithm 2, the mean trajectory is given by computing the non-linear warping functions and setting iteratively the template as the Karcher mean of the N warped trajectories represented in the Riemannian shape space.

Algorithm 2: Computing trajectory template

Require: N trajectories from N clips $T_1, T_2 \cdots T_N$
Initialization: chose randomly one of the N input trajectories as an initial guess of the mean trajectory T_{mean}
repeat
  for i=1 : N **do**
    find optimal path p^* using DTW to warp T_i to T_{mean}
  end for
  Update T_{mean} as the Karcher mean of all N warped trajectories
until Convergence

4.3 Video summarization and retrieval

In order to represent compactly a video sequence, we need to know how to exploit the redundancy of information over time. However, when this information should be extracted from motion and not from frames separately, the challenge is then about complex matching processes required to find geometric relations between consecutive data stream elements. We therefore propose to use EHC to represent a pose and a trajectory as key descriptors characterizing geometric data stream. Based on EHC representation, we develop several processing modules as clustering, summarization and retrieval.

4.3.1 Data clustering

Let V denotes a video stream of human sequence containing elements $\{e_i\}_{i=1...k}$, where e can be a frame or a clip. To cluster V, the data set is recursively split into subsets C_t and R_t as described in the following recursive algorithm 3.

The result of clustering is contained in $C_{t=1..k}$ where C_t is a subset of V representing a cluster containing similar elements to e_t. For each iteration of clustering steps, $t = 1 \cdots K$, the closest matches to e_t are retrieved and indexed with the same cluster reference as e_t. Any visited element e_t

Algorithm 3: Data clustering

Require: $V\{e_i\}_{i=1...k}$;
Ensure: $C_0 = \varnothing$; $R_0 = \{e_1, \ldots, e_k\}$;
  if $(R_t \neq \varnothing)\&\&(t <= k)$ **then**
    $C_t = \{f \in R_{t-1} : dist(e_t, f) < Th\}$;
    $R_t = R_{t-1} \setminus C_t$;
  end if

already assigned to a cluster in C during iteration step is considered as already classified and is not processed subsequently. We regroup nonempty sub sets C_t in l clusters $\{c_1, ..., c_l\}$ (with $l \leqslant k$). Similarities between elements of V are evaluated using a similarity distance *dist* allowing to compare the elements of V. The threshold Th is defined experimentally .

If we consider the video V as a long stream of 3D meshes, the clusters that should be obtained must gather models with similar poses. In this case, the EHC feature vector is used as an abstraction for every mesh and the similarity distance is the elastic metric computed between each pair of human poses. Motion can be incorporated in this similarity by applying a simple time filter on static similarity measure with a window size chosen experimentally [104]. The use of temporal filter integrates consecutive frames in a fixed time window, thus allowing the detection of individual poses while taking into account smooth transitions.

The video V can also be considered as a stream of clips resulting from the video segmentation approach and clusters here gather clips with similar repeated atomic actions. In this case: (1) the feature vector used as abstraction for each clip is a trajectory on shape space of extremal human curves; and (2) the similarity distance, used to compare clips, is based on the DTW algorithm.

4.3.2 Content-based summarization

Our approach for video summarization is based on four steps: First, the whole video is segmented and clustered into several clusters of clips. Second, only the most significant clip (the nearest one to all cluster elements) of each cluster is kept. Third, we construct a subsequence, from the starting video, where this representative clips of each cluster are concatenated.

Finally, This new subsequence is clustered into clusters of poses, and only most representative poses are kept to describe the dataset.

This summarization allows a reduction of dimension for the original dataset where we can display only main clips if we stop on third step, or to display key frames if we continue summarization process until pose clustering.

4.3.3 Motion Retrieval

For content-based motion retrieval, we advocate the usage of the EHC representation, where a query consists of a trajectories representing a clip on the shape space. As response to this specific query, our approach looks in the sequence for most similar trajectories and returns an ordered list of similar ones using the process of clip matching explained in section 4.2.2.

4.4 Experimental evaluation

4.4.1 Motion segmentation and retrieval

In this section, we evaluate our descriptor with temporal shape similarity. Details about the computation of the ground truth descriptor are given in addition to the description of the different datasets used for evaluation. The results obtained by our approach, compared to those of different state-of-the-art descriptors, are then discussed.

Evaluation methodology

The two datsets (2) and (3) presented in Table 3.1 in previous chapter are used in these experiments. From the synthetic dataset [52], we have chosen 14 different motions: walk (slow, fast, circle left/right, cowboy, march, mickey), run (slow, fast, circle left/right), sprint, and rockn'roll. These motions are performed by two actors (a woman and a man) making a total of 28 motions (2800 frames). They are chosen for their interesting challenges as: (i) change in execution rate (slow/fast motions) (ii) change in direction while moving (walking in straight line, moving in circle and

turning left and right) (iii) change in shape (a woman and a man). We used these motion sequences for both segmentation and retrieval experiment.

To validate the segmentation step, we segment all these 3D video sequences with the proposed approach and then compare results to manual segmented ground-truth. In the retrieval process, each query clip is compared to all other clips obtained by the segmentation of sequences. Finally, statistic measures (NN, FT, ST and E-measure) are used for the evaluation.

Analysis of motion segmentation result

Plotting the distance between EHC representation of successive frames gives a very noisy curve. The break points from this curve do not define semantic clips and the extracting of minima leads to an over-segmentation of the sequence (see Figure 4.4 top). To obtain more significant local minima, we convolve the curve with a time-filter allowing to take into account the motion variation, not only between two successive frames but also in a time window. The motion degree after convolution is shown in Figure 4.4 (bottom). Break points are more precise and delimit significant clips corresponding to step changes in the video sequence. In order to evaluate its efficiency, we apply our segmentation method on the whole dataset (3) described in Table 3.1 (3^{rd} raw) and then compare the results to a manual segmentation of the base done carefully.

We performed the clip segmentation for all window size values from 1 to 11 over a representative set of clips extracted from the dataset (3) [130]. Compared to manual ground truth, the best segmentation is obtained using a window size of 5. This value is then fixed for the rest of the tests. The segmentation of the dataset (3) gives 83 segmented clips (78 correct clips and 5 incorrect clips). This can be explained by the fact that the 5 failing clips are short. They contain about 6 frames at most and do not describe atomic significant actions. Otherwise, a total of 144 clips have been obtained by the segmentation of the 14 motions taken from the dataset (2) described in Table 3.1 (2^{nd} raw) performed by two actors.

Figure 4.5 shows some results of motion segmentation on a "slow walk" and a "fast walk" motions. Although the walk speed increases, the motion

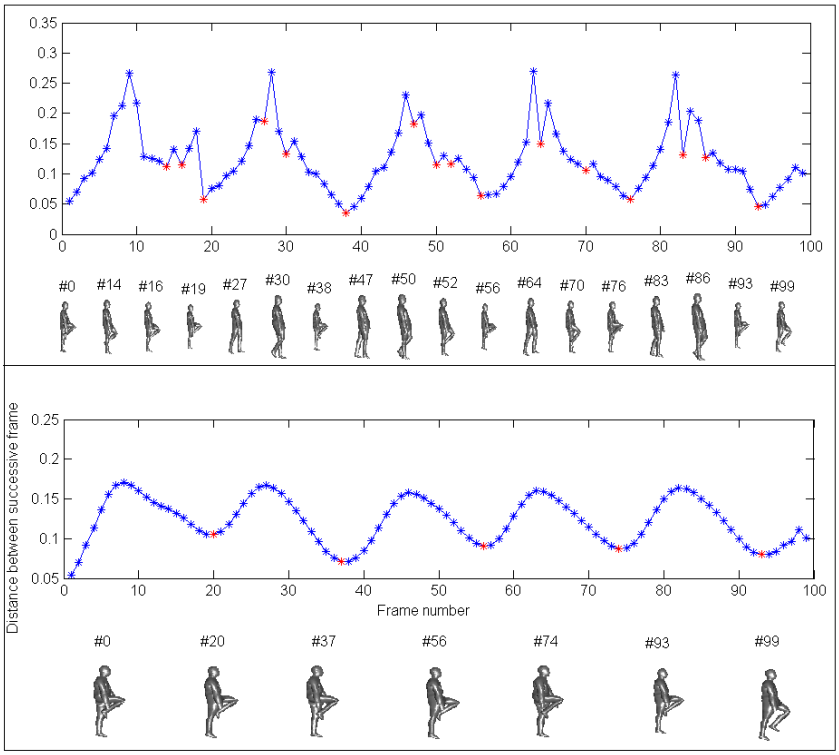


Figure 4.4 – *Speed curve smoothing: (top) speed curve before smoothing, (bottom) speed curve after smoothing.*

segmentation remains significant and does not change and corresponds to the step change of the actor. The Rockn'roll dance motion segmentation is also illustrated in Figure 4.5 (bottom). Thanks to the selection of local minima in a precise neighborhood, only significant break points are detected.

Analysis of motion retrieval result

The motion segmentation method, applied on 14 motion sequences from the dataset (2) and performed by a man and a woman, gives a total of 144 clips. These clips, with an average number of frames per clip equal to 15, are categorized into 14 classes. The motion sequences consist mainly of different styles of walking, running and some dancing sequences. Classes grouped together represent different styles of walking, running and dancing steps. For example, a step change in a walk, may represent a class and groups similar clips done with different speed and in different trajectories. We notice that right to left change step is grouped in a different class than left to right change step.

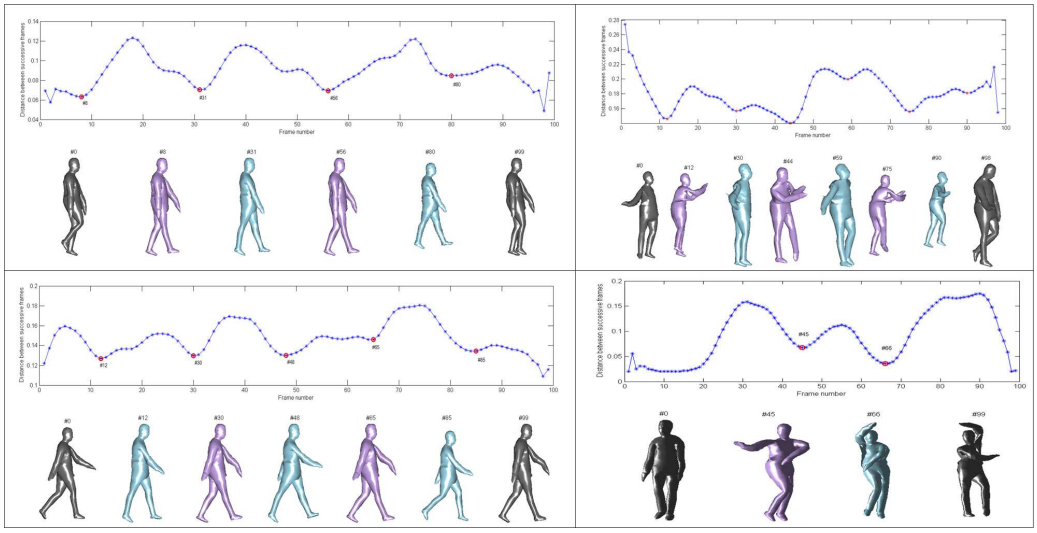


Figure 4.5 – *Various examples of motion segmentation result. From right top to left bottom, motions are: slow walk, Rockn'roll dance, fast walk, vogue dance.*

The similarity metric represented by elastic measure values between each pair of clips allows us to generate a confusion matrix for all classes of clips, in order to evaluate the recognition performance by computing dynamic retrieval measures thanks to a manually annotated ground truth. An example of the matrix representing the similarity evaluation score among clips in sequences performed by a female actress against the clips of sequences of motions performed by a male actor is shown in Figure 4.6. More the color is cold more the clips are similar.

Thanks to the use of DTW, it is noticed that similarity score between same clips done in different speeds is small (see Figure 4.6). Also, the similarity score between the clip representing change in step in a slow walk motion composed of 25 frames and a fast walk motion, composed of 18 frames, is small.

Besides, our approach succeed to retrieve clips within motions done in different ways. For example, the walk circle clips can be matched to the clips of slow walk motion done in a straight line (see Figure 4.6). This explains why the use of an elastic metric, to compare and match trajectories, makes the process independent to rotation. Although the actors performing the motions are different, it is observed that similar clips yield smaller similarity score. Like it is shown in "Rockn'roll" dance motion, steps of the dance performed by different actors are correctly retrieved.

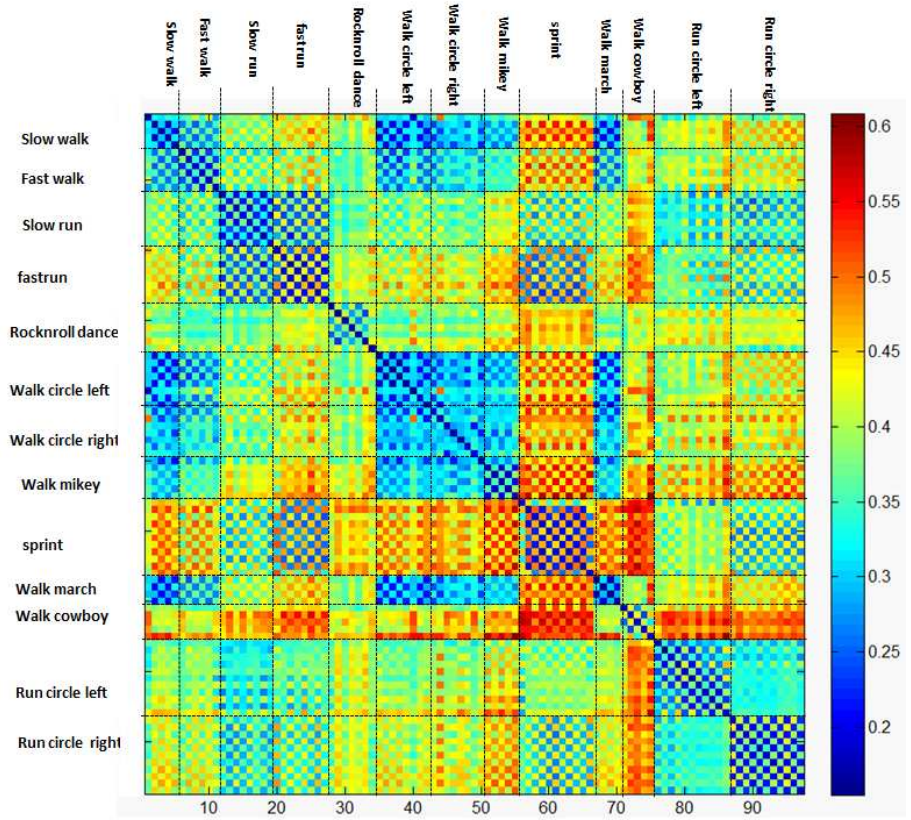


Figure 4.6 – *Similarity matrix evaluation between clips. More the color is cold more the two clips are similar.*

It is demonstrated that 79.26% of similar motion clips are included in the first tier and 93% of clips are correctly retrieved in the second tier. It is a rather good performance considering that only such low-level feature as the EHC is utilized in the matching. This can be explained by the fact that geodesics are not completely invariant to the topology changes. Thereby, the extracted sequential curves that represent the trajectory tend to change the path on the models for certain motions and therefore mislead the matching performed by DTW.

We also apply our retrieval approach to a real captured 3D video sequence from the real dataset (3) described in Table 3.1 (3^{rd} raw). Self similarity example with an actor in a walking motion (walking in circular way) and its similarity curve are shown in Figure 4.7. For the query clip presented at the left of the figure, retrieved clips are found correctly in the sequence when the actor is turning.

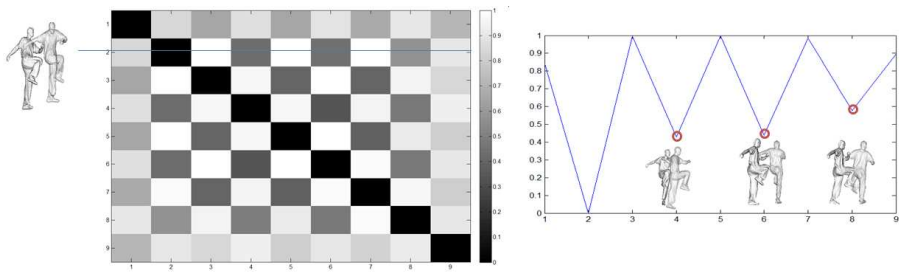


Figure 4.7 – *Experimental results for 3D video retrieval using motion of "walk in circle".*

4.4.2 Data summarization and content-based retrieval

In this section, we firstly conducted multiple experimental trials by analyzing the video clustering method on two aspects: the pose-based clustering and the clip-based clustering. Secondly, we evaluate the impact of the summarization process on the retrieval system by comparing the results with and without using clustering.

Content-based summarization

The performance of the content-based summarization approach is evaluated for pose and clip data. To validate the pose-based summarization, we use a composed long sequence of a subject performing walk and squat

motions from the dataset (3). For clip-based summarization experiment, the same 28 motions used for video segmentation and the retrieval have been used.

The effectiveness of clustering process is evaluated by the number of clusters found which should allow the identification of eventual redundant patterns. The threshold Th in the Algorithm 3 is set accordingly to the values of the similarity function. The distances computed between descriptors (EHC for pose and trajectory of EHC for clip) are normalized to return values in the range [0 1], and Th is then defined experimentally. An optimal setting of Th should return a set of clusters similar to what a "hand-made" ground-truth classification would perform. The Figure 4.8 shows the clustering result obtained from the composed long sequence. The number of clusters decreases with the increase of the threshold Th. We obtain the best result for $Th = 0.5$ with 51 clusters partitioned as the bar diagram shown in the right of the Figure 4.8.

Pose-based clustering process can be improved by increasing the window size of the time filter as shown in Figure 4.9.

We notice from this figure that for a $Th = 0.2$, the number of clusters varies from 330 to 440 and a good compromise is obtained for $Nt = 3$.

Furthermore, clustering is applied on 14 motions extracted from the dataset (3) and performed by two actors (a man and a woman) in order to evaluate the efficiency of the clip-based clustering. By decreasing the threshold Th of the clustering algorithm, we obtain more clusters. Experimentally, we set Th to 0.43 and obtain 23 clusters from initially 110 clips for the first actor and 26 clusters for the second one (see Figure 4.10). We notice that clips representing sprint or running steps are clustered together.

The video summarization process can be used efficiently in hierarchical structure, starting by video segmentation into clips, followed by clip-based clustering and then a pose-based clustering performed on the frames of all represented clusters of the clips resulting from the last step. The effectiveness of our summarization process is shown in Figure 4.11 for the sequence of a real actor performing walking and squatting mo-

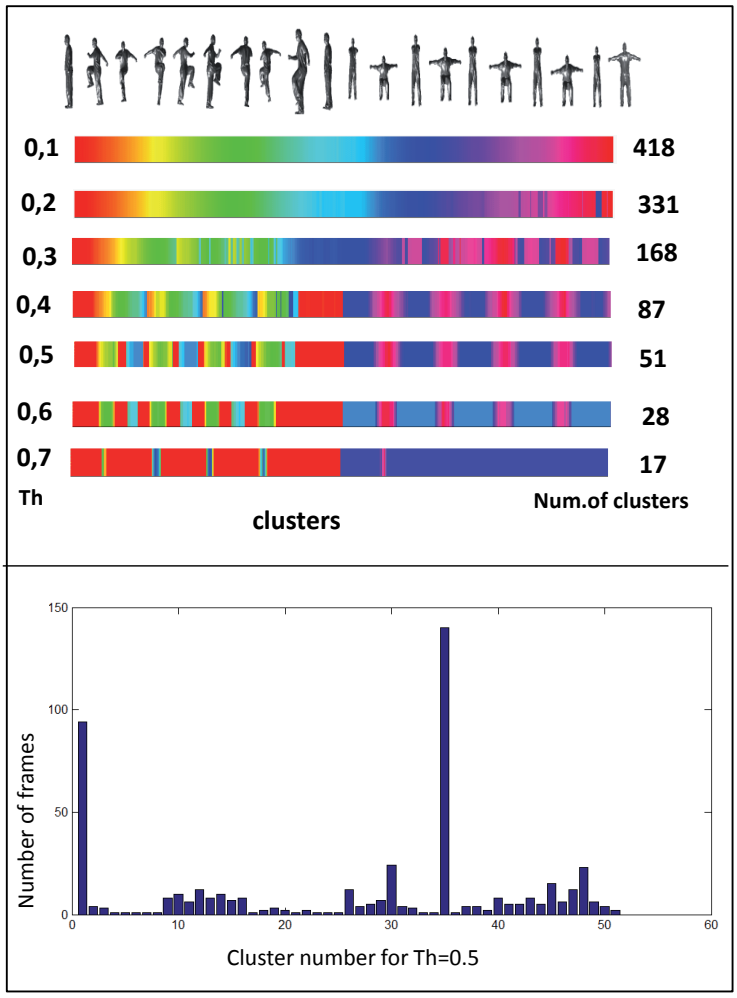


Figure 4.8 – *Frame clustering process with respect to different values of the threshold Th. (top) Variation of number of clusters regarding threshold values. (bottom) Distribution of the number of frames in clusters while* $Th = 0.5$.

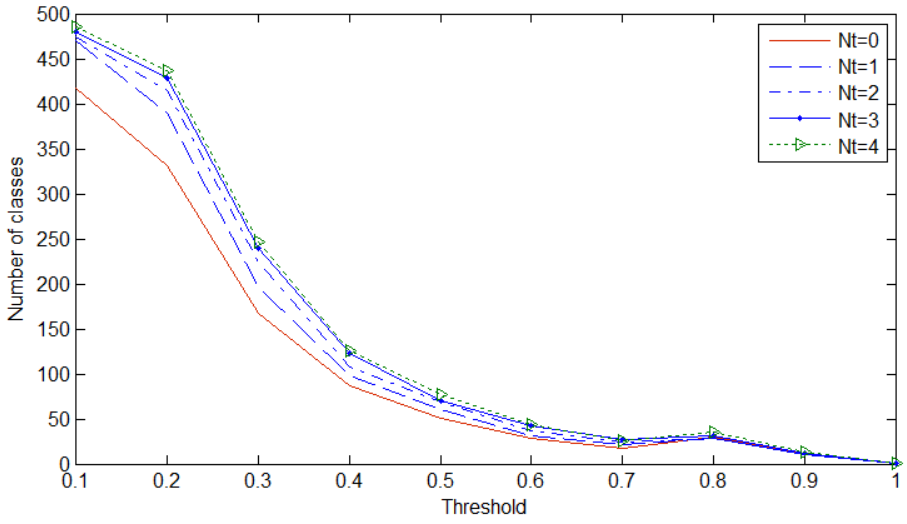


Figure 4.9 – *Frame clustering with respect to a threshold and with different window size varying from 0 to 4.*

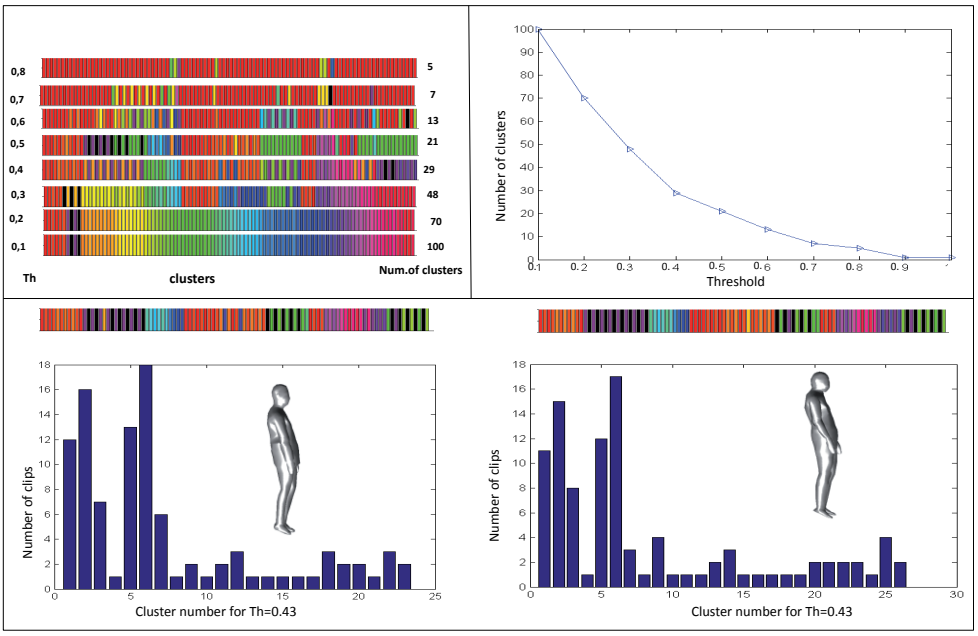


Figure 4.10 – *Clustering clips from a sequence of two actors performing 14 motions from the dataset (3) for a total of 1400 frames, with respect to Th. In second raw, the variation of clip number in each cluster is presented.*

tion. From 500 frames segmented into 18 clips, the clustering process gives 6 clusters. The new subsequence containing 6 clips (most representative clip in each cluster) and 180 frames is then clustered into 41 clusters where each one represent a class of pose.

Hierarchical data retrieval

Let consider clips as the elements of clusters. In this case, the template model is a "mean clip" representing a cluster of clips and is computed using the Algorithm 2. The retrieval system can then be performed hierarchically. In experimental tests, we performed a similar experimentation to motion retrieval on the 14 motions performed by two actors as already evaluated in the section 7.3. In this experimentation, each query is a clip compared to each one of the template models representing the clusters of clips. The similarity measure values obtained by DTW algorithm between clips are used to generate a confusion matrix for all classes of clips, in order to evaluate the recognition performance by computing statistic retrieval measures thanks to a provided ground truth. The matrix of com-

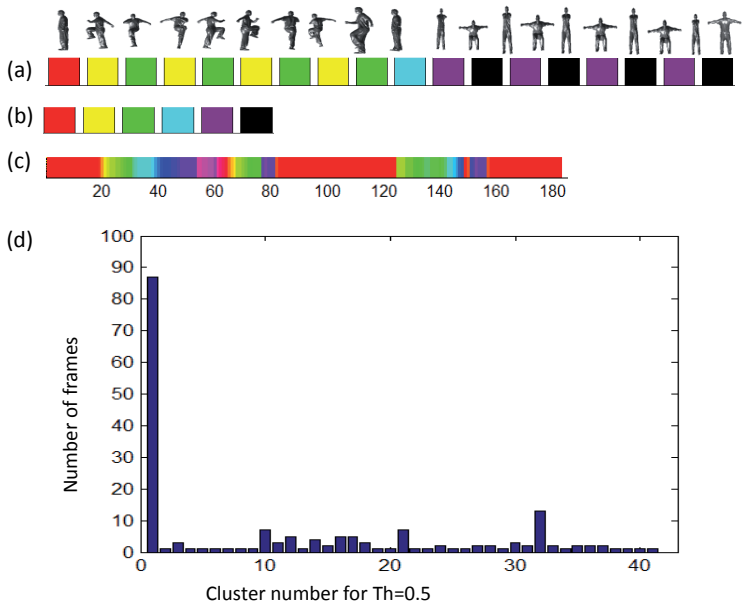


Figure 4.11 – *Summarization process: (a) for a sequence of 500 frames segmented into 18 clips, the clustering process returns 6 clusters of clips using* $Th = 0.38$ *(b) subsequence of clustered clips (180 frames) where each cluster is represented by only one clip chosen as the Karcher mean clip of the cluster, (c) clustering of subsequence into* 41 *clusters of frames using* $Th = 0.5$*, (d) distribution of the number of frames in clusters.*

parison in the first level (model-template comparison) is shown in Figure 4.12.

Retrieval performances obtained from this matrix for FT, ST and E-Measure are respectively 84.09%, 95.83% and 55.26%. In term of clip classification using nearest neighbor template, obtained accuracy is about 93.75%. The analysis of the result given by the binarized matrix shows that the most misclassified clips are those of "fast run" class. In fact, they are assigned to class template representing "sprint" motion class.

4.5 Discussion

Our approach of motion segmentation and clip matching could be used for more semantic tasks like human motion classification for action and gesture recognition. However, the lake of fully reconstructed 3D human videos dedicated to action recognition and the difficulty of the applicability of dynamic meshes acquisition systems in real scenarios make this task inappropriate. In the other side, the emergence of novel RGB-D sen-

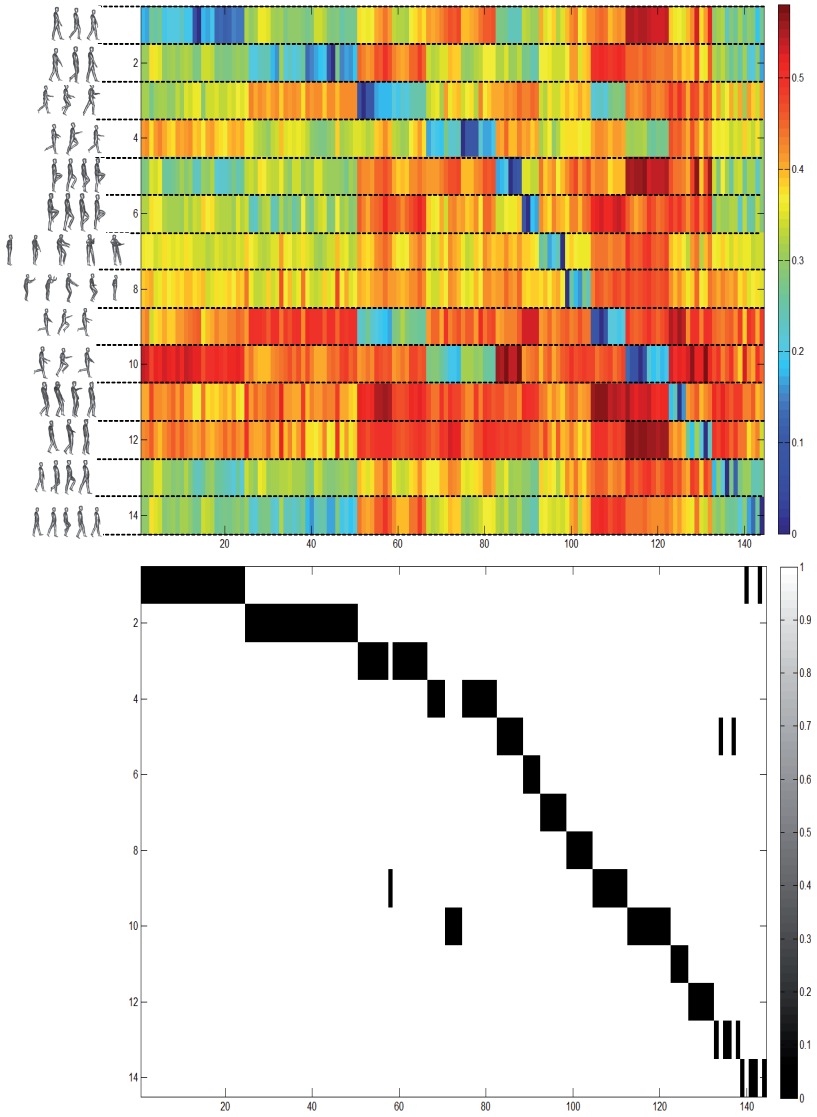


Figure 4.12 – *Similarity matrix and its binarization for template clip of each class against all clips in the dataset.*

sors with high efficiency in real time processing, make their stream more adapted for action recognition and human motion understanding.

4.6 Conclusion

In this chapter, we extended our EHC descriptor to 3D video retrieval, where a motion segmentation is performed on continuous a sequence to split it into elementary action segments called clips. These later are represented by a temporal trajectories of 5 selected human curves on the open curve shape space. Video retrieval is then performed by matching the trajectories using DTW algorithmin on the features that reside on Rienmanian manifolds. Finally, based on statistical tools offered by our geometric framework, we propose efficient solutions for data summarization and hierarchical motion retrieval.

Part II

3D Human action recognition framework using Grasmann manifold

State-of-the-art

5

Sommaire

5.1 Introduction

With the developmental of depth sensors and algorithms for pose estimation, new opportunities have emerged in the field of human motion analysis. Especially, in action recognition domain, a large amount of research has been conducted to achieve a high level understanding of human activities. The problem of action recognition can be defined as follows: given a collection of annotated action videos, how to recognize an unknown action of a query video. An example is illustrated in Figure 5.1 where we would like to recognize the action 'jog' based on the prior knowledge of several actions.

In this section, we present main applications where human action recognition can be involved. Besides, we present a taxonomy of action recognition levels. Depth sensor technologies which able acquirement of depth images are explained and datasets collected for the purpose of testing action recognition systems are enumerated. Finally, a review of existing approaches is presented and discussed.

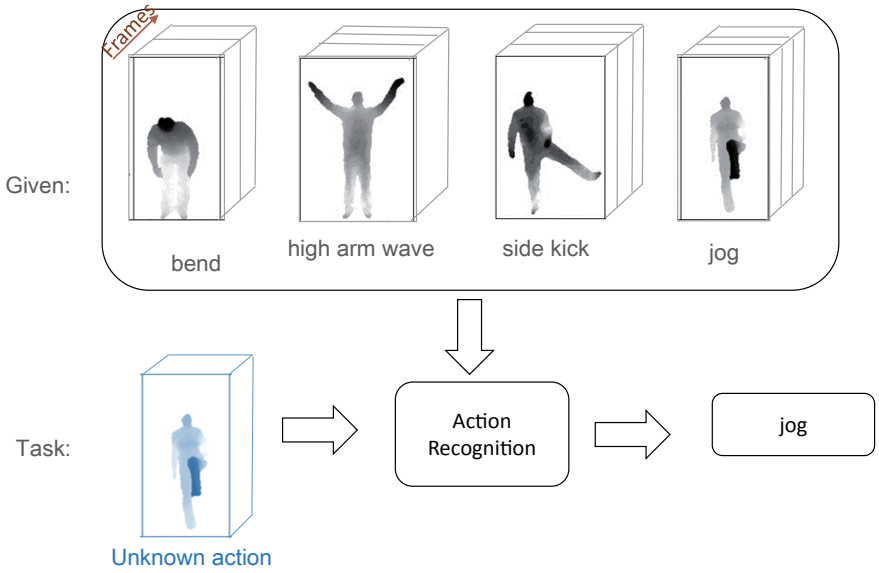


Figure 5.1 – *Illustration of action recognition process: the input is a query video with an unknown action, the output is an action label of this query video.*

5.2 Motivation and challenges

The motivation behind the great interest for action recognition is the large number of possible applications in: consumer interactive entertainment and gaming [34], surveillance systems [70], smart home and life-care systems [55].

In the past, research has mainly focused on learning and recognizing actions from image sequences taken by RGB cameras [127, 128, 17]. In fact, human action modelling from 2D video is a well studied problem in the literature. Main works are summarized in recent surveys of Aggarwal et al. [7], Weinland et al. [142] and Poppe [92].

However, there are several limitations coming from 2D cameras. In fact, they are sensitive to color and illumination changes, background clutters, and occlusions. Although several works exist, recognizing actions accurately remains a challenging task. With the recent advent of cost-effective depth cameras, researchers give much attention to data produced by such kind of cameras. The reason is that the depth sensor has several advantages over visible light camera. First, they provide a 3D structural information of the scene, offering more discerning information to recover postures and recognize actions. They also allow significantly alleviate low-level difficulties in RGB imagery like background subtraction and light variation. Second, the depth camera can work in total darkness which is a benefit for several applications which run day and night such as patient/animal monitoring systems. Third, thanks to these advantages many interesting research have emerged allowing the estimation of human skeletons in 3D coordinate system from a single depth image. These skeletons which are estimated from depth images give additional possibilities to investigate action recognition. The possible skeleton estimation obtained by such a low-cost acquisition depth sensor has provided new opportunities for human-computer-interaction applications, where popular gaming consoles involve the player directly in interaction with the computer. Besides, hand/arm movement are better studied using depth data which able a natural tracking of hands and arms in the scene.

Using sequences from depth cameras, we have 3 types of data: depth

images, skeletons and color images. Using these data simultaneously or separately to model the action appearance and dynamic and to perform classification is the new challenge. Whatever sequence length or specific applications they are used for, what is expected most of these systems is a high accuracy and a low latency.

5.2.1 Taxonomy of human activities

Human behavior analysis from lower to higher degree of abstraction consists of four levels as illustrated in Figure 5.2 and defined as follows :

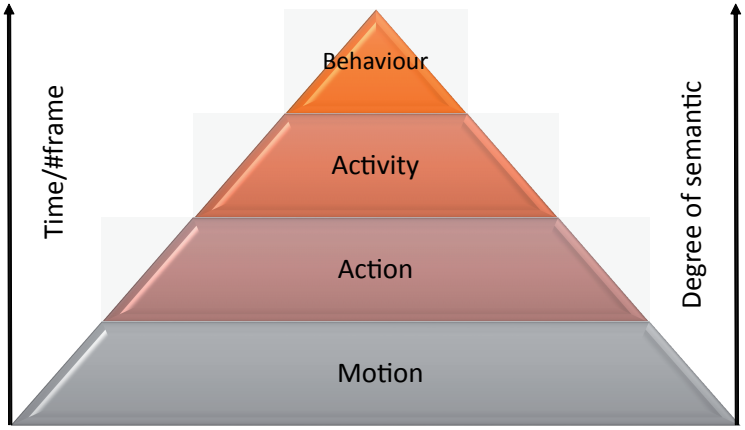


Figure 5.2 – *levels of Human activity analysis [7].*

- Basic motion: it is primitive which consists of entities out of which actions are built. There are atomic movements that can be described at the limb level. Their time laps do not exceed very few seconds. In this level, the movement is detected.

- Action: it is a set of repetitive different primitives. At this level, the body motion is recognized in order to know what a person is doing or the objects he is interacting with. The duration of an action is few seconds, during which a person can do simple activity, like standing, biding, walking, etc. We can consider gestures as a specific type of actions, usually specific to the motion of arms and hands.

- Activity: it consists of a set of multiple actions in order to understand human behaviour. It can last from tens of seconds to units of

minutes. Examples of activities are: taking a shower, making a bed, cooking, etc.

- Behaviour: it is the highly semantic comprehension of human motion. Frames acquired during hours or even days construct a sequence of behaviour. At this level, abnormal behaviour and anomalies can be detected.

Regarding the timescale of the motion, action and activity descriptions, there is a wide range of helpful distinctions. We distinguish between actions, activity and behaviours, corresponding to longer timescales and increasing complexity of representation. In this thesis, we study action recognition on short-timescale representations, like a forward-step or a hand-raise; and medium timescale movements, like walking, running, jumping, standing, waving.

5.2.2 Applications

To better understand what we expect from an action recognition system, it is important to understand exactly how to use it in practical applications and what requirements are needed for each application.

The goal in an activity-driven application is to analyze classified activities so that their semantic meaning can be understood in each specific domain. The application depends on the degree of the semantic we need to understand from the sequence.

Although human action recognition can be used in several domains, here we focus on three dominant applications including:

- Surveillance environments [75]: In surveillance systems, the goal is to automatically track individuals, so as to support security personnel to observe and understand activities, resulting in recognition of the criminal and detecting suspicious activities. Most security surveillance systems are equipped with several cameras and require laborious human monitoring on screens for video content understanding. However, by applying automatic human activity recognition techniques, it is possible to reduce the work staff. In fact, it will

be possible to systematically create an alert immediately when security events are detected in order to prevent potentially dangerous situations.

Besides, video-surveillance based human activity recognition systems can also be applied in marketing analysis for detecting customers interest while shopping and also ensure safety of swimmers in pools. In such application, video tracking and identification are among the challenges, in addition to a decision making quick and accurate.

- Entertainment environments [33]: Human activity recognition can also be used to recognize entertainment activities, such as sport, dance and gaming, in order to enrich lifestyles. In such cases, we often care about time response since to interact with games we need a quick response and even knowing the action before finishing. One of the most popular leisure activities is playing video games. A number of methods are recently developed for this purpose using depth cameras.

- Sign Language Recognition [97]: Gesture recognition, which is a sub-domain of action recognition that operates over the upper body parts, serves a lot for automatic understanding of sign language.

- Healthcare systems [155]: In healthcare systems, the applications based on activity recognition consists of analyzing and understanding of patients activities. The purpose is to facilitate health workers to diagnose, treat and care for patients, resulting in improving the reliability of diagnosis. Advantages of such system are: decreasing the working load for the medical personnel, shortening the hospital stay for patients, and improving patients quality of life.

5.3 RGB-D data acquisition

Naturally, the human eye registers x, y and z coordinates for everything seen, and the brain interprets those coordinates into a 3D image. Depth

information, which is represented by "z" coordinate enables capabilities well beyond the 3D scene reconstruction.

In the past, few approaches and techniques have been proposed for seeing the scene in 3D. However, recently there are several common technologies that can acquire 3D images, each with its own technique: stereoscopic vision, structured light pattern and time of flight (TOF). These technologies, has significantly lighten difficulties that reduce the action recognition performance in 2D video. These cameras provide in addition to the RGB image a depth stream allowing to discern changes in depth in certain viewpoints.

Most important technologies that can acquire 3D images with depth information are:

- **Stereoscopic vision** : It is the most common 3D acquisition system. It uses two cameras to obtain a left and right stereo image which are slightly offset on the same order as the human eyes are. As the computer compares the two images, it develops a disparity image that relates the displacement of objects in the images. Commonly used in 3D movies, stereoscopic vision systems enable exciting and low-cost entertainment. It is ideal for 3D movies and mobile devices, including smartphones and tablets.

- **Structured light pattern**: Structured light illuminates patterns to measure or scan 3D objects. Light patterns are created using either a projection of laser or LED light interference or a series of projected images. Structured-light-based technology basically exploits the same triangulation as a stereoscopic system does to acquire the 3D coordinates of the object. Single 2D camera systems with an IR- or RGB-based sensor can be used to measure the displacement of any single stripe of visible or IR light, and then the coordinates can be obtained through software analysis. These coordinates can then be used to create a digital 3D image of the shape.

- **Time of flight (TOF)** : Relatively new among depth information systems, time of flight (TOF) sensors are a type of light detection and ranging (LIDAR) system that transmit a light pulse from an emitter

to an object. A receiver determines the distance of the measured object by calculating the travel time of the light pulse from the emitter to the object and back to the receiver in a pixel format.

Depth cameras like kinect or PrimeSense uses a structured light technique to generate real-time depth maps containing discrete range measurements of the physical scene. Given the low-cost and real-time nature of these devices, the quality of these depth sensing, is compelling although it is still inherently noisy. The stream given in each frame by depth sensors consists of: RGB image, depth map and a human skeleton estimation as illustrated in Figure 5.3.

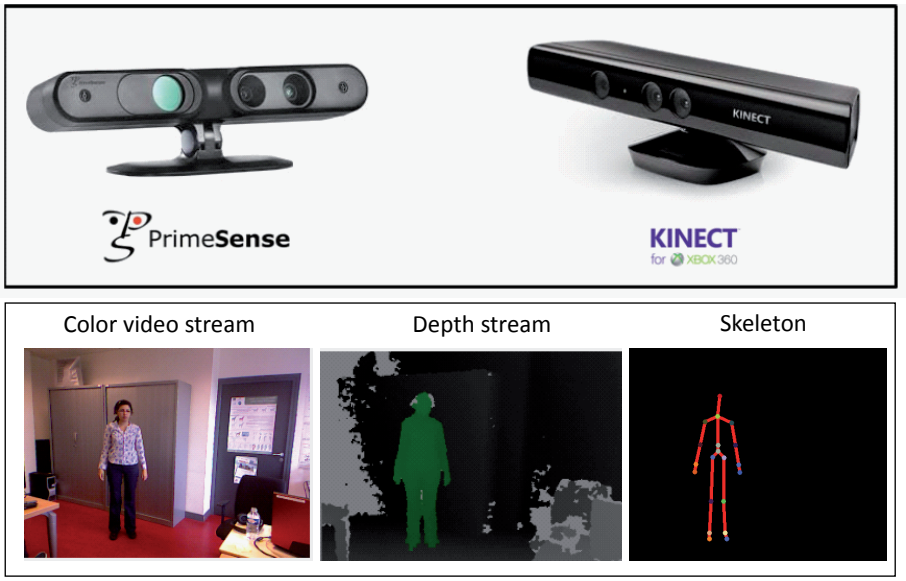
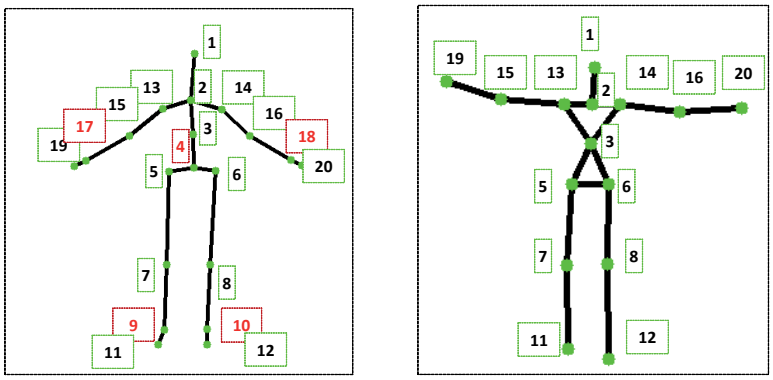


Figure 5.3 – *Video streams given by depth sensors. (top) Examples of depth sensors. (bottom) RGB image, depth image and skeleon given in a frame.*

It is Shotton et al. [103] works who have proposed a real-time approach for estimating 3D positions of body joints using extensive training on synthetic and real depth streams. The two best-known skeletons provided by the Microsoft Kinect sensor, are those obtained by official Microsoft SDK, which contains 20 joints, and PrimeSense NiTE which contains only 15 joints (see Figure 5.4).



(a) (b)

Figure 5.4 – *Skeleton joint locations captured by Microsoft Kinect sensor (a) using Microsoft SDK (b) using PrimeSense NiTE. Joint signification are: (1) head (2) shoulder center (3) spine (4) hip center (5/6) left/right hip (7/8) left/ right knee (9/10) left/right ankle (11/12) left/right foot (13/14) left/right shoulder (15/16) left/right elbow (17/19) left/right wrist (19/20) left/right hand.*

5.4 Benchmarks datasets

When developing a new recognition system or improving an existing one, the datasets to test need to be chosen carefully. Publicly available datasets are numerous, mostly collected by various authors for evaluation purpose. Each of the datasets includes various types of actions performed multiple times by different subjects, and each one of the benchmarks is designed to solve a specific challenge. Table 5.1 provides a summary of most popular datasets, while Figure 5.5 shows some examples.

Here, only datasets which are used to evaluate activity or gesture recognition from video sequences acquired by depth sensors are presented. It is possible to categorize these datasets into four main categories based on the activity level taxonomy or on the applications where they can be involved :

- Simple actions: there are elementary and intend to interact with computer or game consols [73, 32].
- Daily activities: indoor activities in different environment: bathroom, kitchen, bedroom, office [145, 145, 134, 110].
- Gestures with hands or upper body part: used for sign language in-

terpretation and recognition or also for recognizing cooking motions [68, 77, 101].

- Complex activities: involving human interaction and can be composed of several simple actions [154, 143].

Dataset	Size	Properties
MSR action 3D [73]	10 subjects/20 actions/3 tries	interaction with game consoles (examples of actions: draw x, draw tick, draw circle, hand clap, two hand wave..)
UT-kinect [145]	10 subjects/10 actions/2 tries	human actions in indoor settings (examples of actions: walk, push, carry...)
UCF-kinect [32]	16 subjects/16 actions/5 tries	long sequences to test latency (examples of actions: twist left, twist right, hop...)
MSR Daily activity [134]	10 subjects/16 classes/2 tries	indoor daily activities (examples of actions: drink, eat, read book..)
Cornell Activity [110]	4 subjects/12 activities/60 sequences	daily activities in different environments: office, kitchen, bedroom, bathroom, and living room (examples of actions: rinsing mouth, brushing teeth)
MSRGesture3D [68]	10 subjects/12 gestures/2-3 tries	dynamic American Sign Language (ASL) gestures (examples of gestures: ASL-Z, ASL-J, ASL-Where...)
Sheffield Kinect Gesture (SKIG) Dataset [77]	6 subjects/10 categories of hand gestures	hand gesture sequences (examples of actions: cyrcle, triangle, up-down...)
ChaLearn Gesture [42]	20 subjects/100 gestures	upper-body hand and arm gestures (interacting with a computer by performing gestures to: play a game, remotely control appliances or robots, learn to perform gestures from an educational software.)
Kitchen scene action [101]	9 activities	recognize cooking motions (examples of actions: cooking eggs, turning, ...). Mainly arms and hands gestures.
LIRIS human activity [143]	10 actions/828 sequences	discussion of two or several people (examples of actions: a person gives an item to a second person, an item is picked up or put down, a person enters or leaves an office, a person tries to enter an office unsuccessfully, ...)
SBU Kinect interaction [154]	7 subjects/8 interaction/300 interactions	two-person Interaction (examples of actions: slap, hug...)

Table 5.1 – *Summary of the most popular publicly available RGB-D datasets for evaluating activity and gesture recognition performance.*

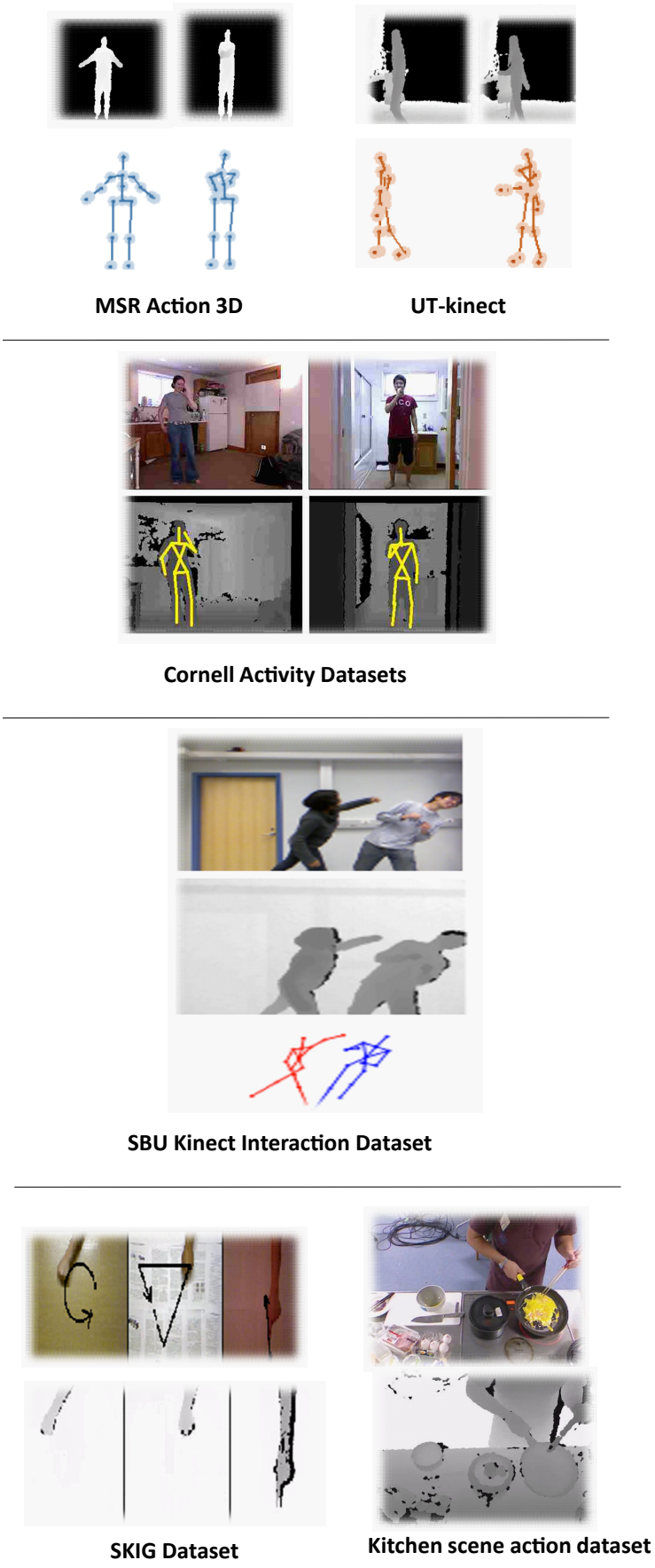


Figure 5.5 – *Examples of frames from different datasets.*

5.5 Action recognition related work

The pipeline of activity recognition approaches generally involves three steps: feature extraction, quantization/dimension reduction and classification. The classification approaches using depth images are generally inspired by approaches already used in 2D videos . However, approaches dealing with human action recognition in depth sequences have received growing attention, as reported in recent surveys [21, 152]. Depth camera output consists of a stream of color, depth and skeleton. Here we differentiate methods that rely on depth maps or features therein, methods that take skeleton and those who take both as inputs. In the following, all motion descriptors extracted from these data are discussed. Thereafter, most popular classification algorithms which are used for action recognition are introduced. Finally, in discussion section a conclusion of all these approaches is presented.

5.5.1 Depth maps approaches

Maps obtained by depth sensors are able to provide additional body shape information to differentiate actions that have similar 2D projections from a single view. It has therefore motivated recent research works, to investigate action recognition using the 3D information. First methods used for activity recognition from depth sequences have tendency to extrapolate techniques already developed for 2D video sequences. These approaches use points in depth map sequences as a gray pixels in images to extract meaningful spatiotemporal descriptors.

Local feature extraction approaches like spatiotemporal interest points (STIP) are for example employed for action recognition on depth videos. Bingbing et al. [86] use depth maps to extract STIP and encode Motion History Image (MHI) in a framework combining color and depth information. Xia et al [144] propose a method to extract STIP from depth videos (DSTIP). Then, around these points of interest they build a depth cuboid similarity feature as descriptor for each action.

In Wanqing et al. [73], depth maps are projected onto the three orthogonal Cartesian planes ($X-Y$, $Z-X$, and $Z-Y$ planes) and the contours

of the projections are sampled for each frame. Figure 5.6 illustrates the 3D silhouettes extracted using this approach. The sampled points are used as bag-of-points to characterize a set of salient postures that correspond to the nodes of an action graph used to model explicitly the dynamics of the actions. One limitation of this approach [73] is due to noise and occlusions in the depth maps, the silhouettes viewed from the side and from the top may not be reliable. This makes it very difficult to robustly sample the interest points given the geometry and motion variations across different persons.

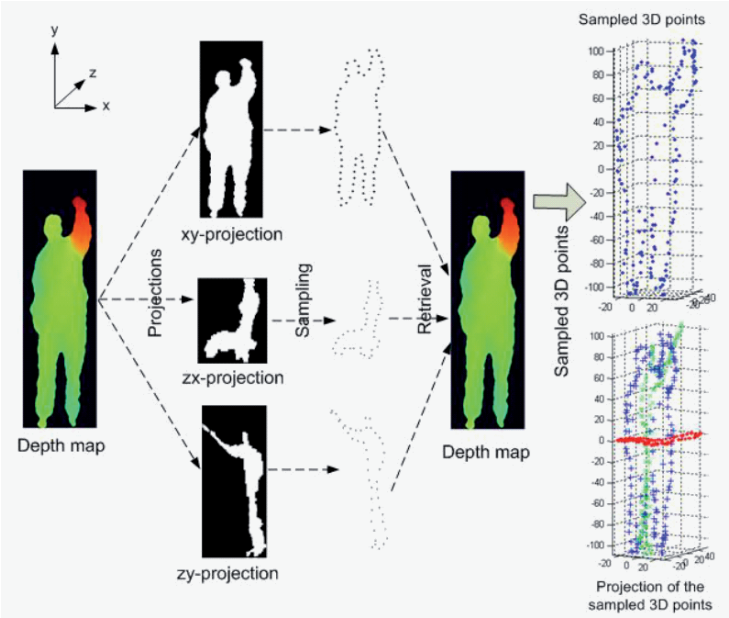


Figure 5.6 – *Projection of the depth map into three axes to represent 3D silhouette as proposed by [73].*

To overcome this limitation, Vieira et al. [129] represent each depth map sequence as a 4D grid by dividing the space and time axes into multiple segments in order to extract SpatioTemporal Occupancy Pattern (STOP) features. Figure 5.7 illustrates an example of space-time cells extracted along a depth sequence.

Similarly, in order to address the noise and occlusion issues, Wang et al. [133] consider the sequence as a 4D shape and extracted 4D subvolumes randomly with different sizes and at different locations. This feature, called Random Occupancy pattern (ROP), are less sensitive to occlusion.

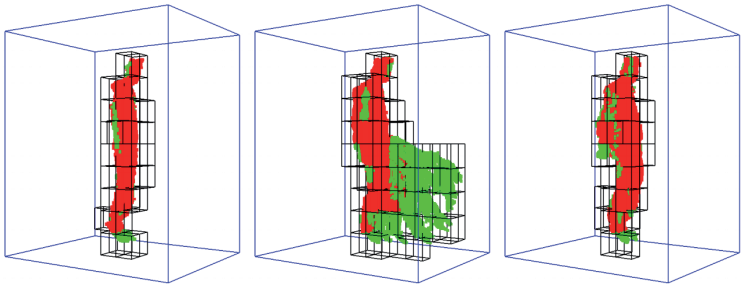


Figure 5.7 – *Examples of space-time cells of a depth sequence of the action forward kick as proposed by [129].*

Yang et al. [151] employ Histograms of Oriented Gradients features (HOG) computed from Depth Motion Maps (DMM), as the representation of an action sequence. They project each depth map onto three pre-defined orthogonal Cartesian planes. Each projected map is normalized and a binary map is generated by computing and thresholding the difference of two consecutive frames. The binary maps are then summed up to obtain the DMM for each projective view. Histogram of Oriented Gradients (HOG) is then applied to DMM map to extract features from each view. The concatenation of HOG from the three views together form the DMM-HOG descriptors. An illustration of the steps of HOG extraction from DMM is presented in Figure 5.8.

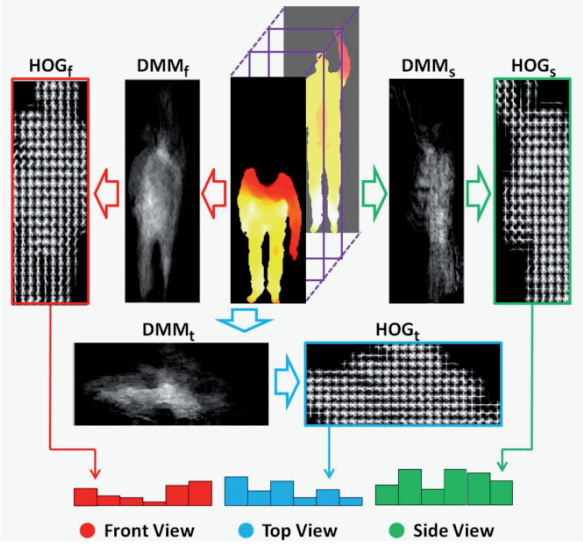


Figure 5.8 – *Histograms of Oriented Gradients descriptor on Depth Motion Map [151].*

An SVM classifier is trained on these descriptors. Although high ac-

curacies and low complexity of this approach, the hand-crafted projection planes raise problems related to view-dependency.

Another histogram descriptor is presented by Oreifej et al. [87]. This descriptor is a 4D histogram computed over depth, time, and spatial coordinates capturing the distribution of the surface normal orientation. As illustrated in Figure 5.9, they first compute 4D normals over 4D surface then they partition the depth sequence into a fixed number of spatiotemporal cells.

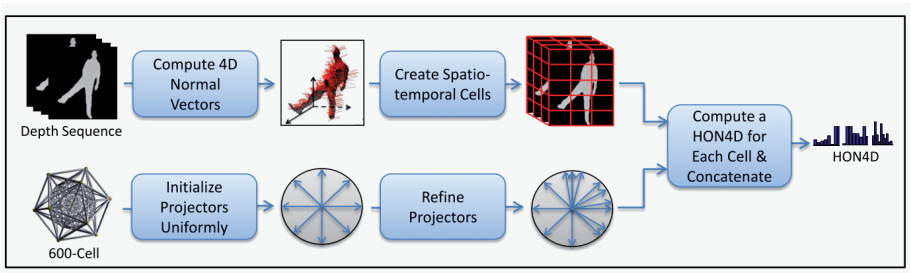


Figure 5.9 – *The 4D normals and their quantization as proposed by Oreifej et al. [87].*

For each cell, the normal occurrence are quantified using 4D projectors. The limitation of this descriptor is that it assumes coarse spatial and temporal correspondence between the spatiotemporal cells across the sequences. This assumption is not valid when actors significantly change their spatial locations, and when the temporal extent of the activities vary significantly.

5.5.2 Skeleton approaches

The availability of 3D sensors has recently made possible to estimate 3D positions of body joints. Especially thanks to the work of Shotton et al. [103], where a real-time method is proposed to accurately predict 3D positions of body joints in individual depth map without using any temporal information. Thanks to this work, skeleton based methods have become popular and many approaches in the literature, either space time volume or sequential, propose to model the dynamic of the action using these features.

As a space time volume approach, Yang et al. [150] extract three features, as pair-wise differences of joint positions, for each skeleton joint.

These features include posture (f_{cc}) and motion (f_{cp}) features which encode spatial and temporal aspect. It include also offset features (f_{ci}) which represent the difference of a pose with the initial pose. Then, after normalization of these features, principal component analysis (PCA) is applied in order to reduce redundancy and noise and thus obtain a compact *Eigen Joints* descriptor for each frame. Finally, a naïve-Bayes nearest-neighbour classifier is used for multi-class action classification. Figure 5.10 illustrates the process used by Yang et al. [150] to obtain *Eigen Joints* descriptor. The major limitation of this approach is the offset feature computation which rely on an assumption assuming that the initial pose is neutral which is not always the case.

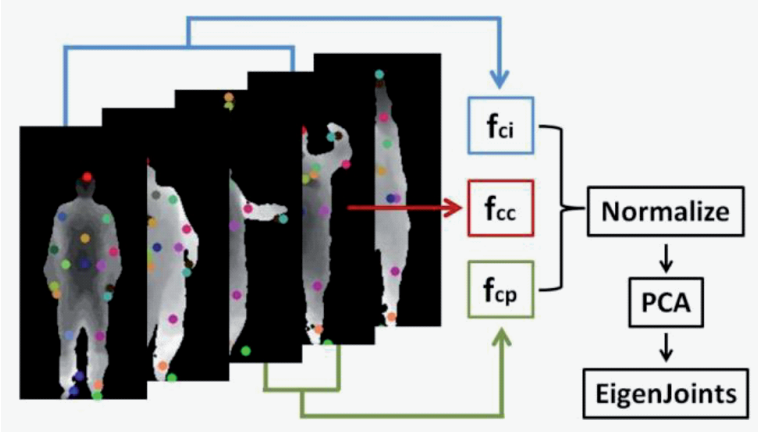


Figure 5.10 – *EigenJoint features developed by Yang et al. [150] .*

The rest of skeleton-based approaches in this review are sequential ones. In fact, the reason behind the popularity of temporal dynamic modelling explicitly is the natural correspondence of skeletons across time.

Devanne et al. [28], propose a spatiotemporal motion representation to characterize the action as a trajectory which corresponds to a point on Riemannian manifold of open curves shape space. These motion trajectories are extracted from 3D joints, and the action recognition is performed by K-Nearest-Neighbor method applied on geodesic distances computed between curves on the shape space.

Wang et al. [131] propose a method to improve the estimation of human joint locations and classify these joints into five body parts. Data mining techniques is then applied on these parts to obtain a representa-

tion of spatiotemporal structure of human actions. Some recent studies are made to find optimal subset of skeleton joints, taking into account the topological structure of the skeleton, in order to improve the accuracy [19].

The popular Dynamic Time Warping (DTW) technique [37], well-known in speech recognition area, is also used for gesture and action recognition using depth data. Reyes et al. [93] perform DTW on a feature vector defined by 15 joints on a 3D human skeleton obtained using PrimeSense NiTE. Similarly, Sempena et al. [99], compute quaternions from the 3D human skeleton model to form a 60-element feature vector. Although DTW have shown good results on clean 3D skeletons given by Motion capture and demonstrated that it is a good way to compare two sequences regardless to their execution rate variation, in the case of 3D joints estimated from depth images, recognition rates are not good enough because of the noisy nature of skeleton joint location which lead to an inappropriate wrapping of skeletons.

Xia et al. [145] compute histograms of the locations of 12 3D joints (HOJ3D) as a compact representation of postures and use them to construct posture visual words of actions. Towards this end, they define a modified spherical coordinate system on the hip center and partition the 3D space into bins, as shown in Figure 5.11 (a) and (b) respectively. A probabilistic voting is established to determine the fractional occupancy as demonstrated in Figure 5.11 (c). Then, the HOJ3D are projected using LDA and clustered into k posture visual words which represent the prototypical poses if actions. Finally, the temporal evolution of those visual words are modeled by a discrete HMM. The major problem of this approach is its reliance on the hip joint location which might potentially compromise the recognition accuracy, due to the noise embedded in the estimation of this joint location.

Bag-of-words approaches originated from text retrieval research is being adopted to action recognition using skeletons, as proposed by Seidenari et al. [98]. The key idea of this approach is to use joint positions to align multiple-parts of the human body using a bag-of-poses solution applied in a nearest-neighbor framework.

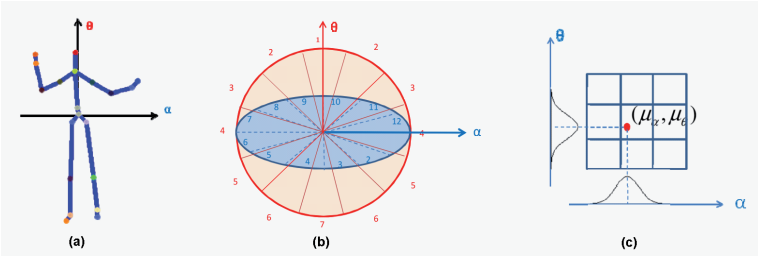


Figure 5.11 – *HOJ3D descriptor as proposed by Xia et al. [145] (a) Cartesian coordinate system for joint location. (b) Spherical coordinate system for joint location. (c) The probabilistic voting for spatial occupancy via a Gaussian weighting function.*

Recent research has carried on more complex challenges of on-line recognition systems for different applications, in which a trade-off between accuracy and latency can be highlighted. Especially gaming and technologies involving human computer interaction, need to be highly accurate and also fast in taking decisions.

Recently, Ellis et al. [32] study this trade-off and employe a Latency Aware Learning (LAL) method, reducing latency when recognizing actions. They learn a logistic regression-based classifier on 3D joint position sequences captured by kinect camera, to search a single canonical posture for recognition. Another work is presented by Barnachon et al. [13], where a histogram-based formulation is introduced for recognizing streams of poses. In this representation, classical histogram is extended to integral one to overcome the lack of temporal information. They also prove the possibility of recognizing actions even before they are completed using the integral histogram approach. Tests are made on both 3D MoCap from TUM kitchen dataset [115] and RGB-D data from MSR-Action3D dataset [73].

5.5.3 Hybrid approaches

Some hybrid approaches are combining both skeleton data features and depth information in order to improve recognition performances.

Azary et al. [10] propose a spatiotemporal descriptor combining image features extracted using radial distance measures and 3D joint tracking to formulate time-invariant action surfaces. Manifold learning is then used to reduce the dimensionality of the data surfaces and obtain a representa-

tion which can be compared against other actions for classification. This approach is computationally inexpensive because of the simplicity of the feature extraction algorithms and manifold learning approach, however the reported recognition rates are low.

Wang et al. [134] utilizes both skeleton and point cloud information. The key idea is that some actions differ mainly due to the objects in interactions, while only skeleton information is not sufficient in such cases. Towards this end, they discretize the local space of each joint using spatial grid and compute the 3D point cloud located around, which construct the Local Occupancy Patterns (LOP) features. Figure 5.12 shows examples of LOP features illustrated on skeleton human body representation. 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.

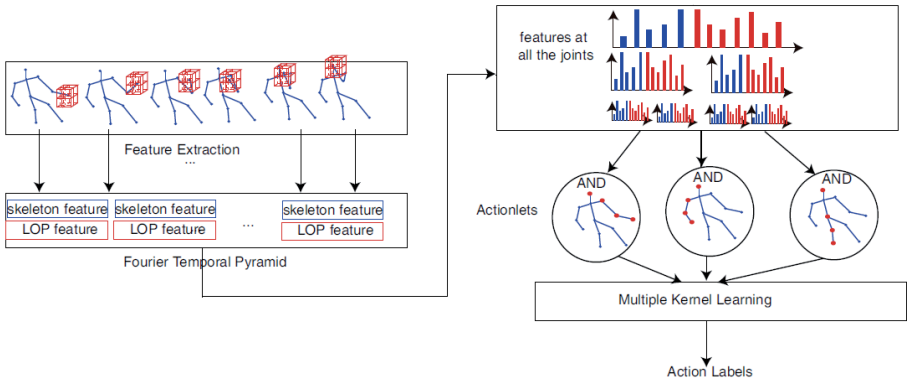


Figure 5.12 – *The actionlet framework proposed by Wang et al. [134].*

In Oreifej et al. [87], a spatiotemporal histogram (HON4D) computed over depth, time, and spatial joint coordinates is used to encode the distribution of the surface normal orientations. Similarly to Wang et al. [134], HON4D histograms [87] are computed around each joint to provide the input of an SVM classifier.

Althloothi et al. [95] propose 3D shape features based on spherical harmonics representation and 3D motion features using kinematic structure of the skeleton. Both features are then merged using a multi kernel learning method.

Koppula et al. [65] explicitly consider human-object interactions. They consider the problem of jointly labeling the object affordances and human activity which is composed of several sub-activities (actions). They also define a Markov Random Field (MRF) over the spatiotemporal sequence. In this Markov model nodes represent objects and sub-activities, and edges represent the relationships between object affordances, their relations with sub-activities, and their evolution over time.

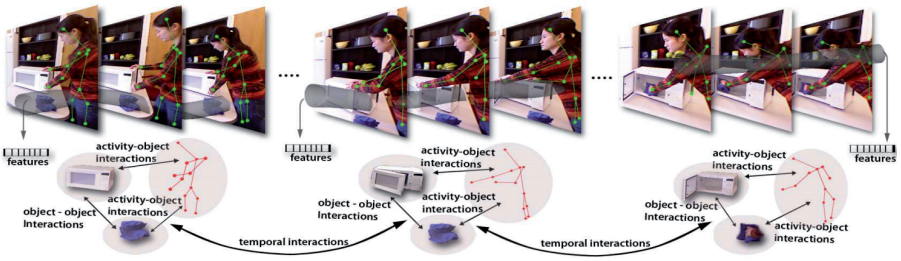


Figure 5.13 – *Pictorial representation presented by [65] of the different types of nodes and relationships modeled in part of the cleaning objects activity comprising three sub-activities: reaching, opening and scrubbing.*

5.6 Gesture recognition related work

Advanced gaming interfaces have renewed interest in hand gesture recognition as an ideal interface for human computer interaction. Capturing the motion of hands shares many similarities with full body pose estimation. However, hands impose some additional challenges like very large pose variations and severe occlusions. Also, hand interacts with other hand or objects, thus capturing hand motion is still a very challenging task. Using depth information, approaches are more performant than those using color information. In fact, depth map offers a natural segmentation of the hand from the scene background.

Depth based approaches proposed initially for action recognition are tested in performing hand gesture recognition, especially on sign language datasets (see Figure 5.14).

Several works [87, 133, 151], which encode depth sequences in spatiotemporal descriptors, have proven their efficiency to perform sign language recognition.

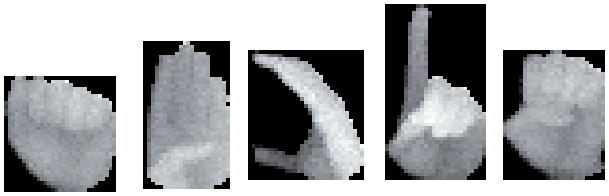


Figure 5.14 – *Alphabet (A-E) of the American sign language captured with a ToF camera.*

In fact, the depth map of the hand is represented by a histogram of 4D normals (HON4D) [87], histogram of oriented gradient (HOG) [151] or also by random occupancy pattern [133]. Similarly to the action recognition process, the recognition system pass by a learning step where machine learning algorithms are used to learn each gesture representation.

Guan et al. [46] propose a system that can interpret a user's gestures in real time to manipulate windows and games. The system uses a 3D depth camera to extracts hand and fingertips. It recognizes the movement and click gesture by analyzing the location and shape change of fingertips. Results of fingertips extraction are shown in Figure 5.16.

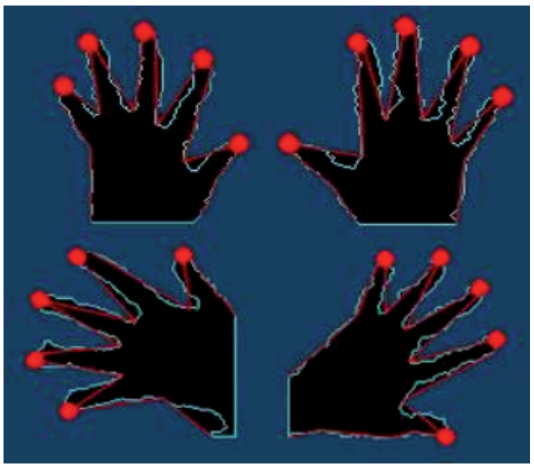


Figure 5.15 – *Fingertips detection results as proposed by Guan et al. [46] works.*

Marnik et al [81] propose an approach to classify Polish finger alphabet symbols. The input for each of the considered 23 gestures consists of a gray-scale image at a relatively high resolution and depth data acquired by a stereo setup. Uebersax et al. [124] propose a method based on average neighborhood margin maximation that recognizes the ASL finger alphabet from low-resolution depth data in real-time.

Jaemin et al. [54] propose a hand gesture recognition system using depth data, which is robust for environmental changing. This approach involves an extraction of hand shape features based on gradient value instead of conventional 2D shape features (see Figure 5.16), and arm movement features based on angles between joints.

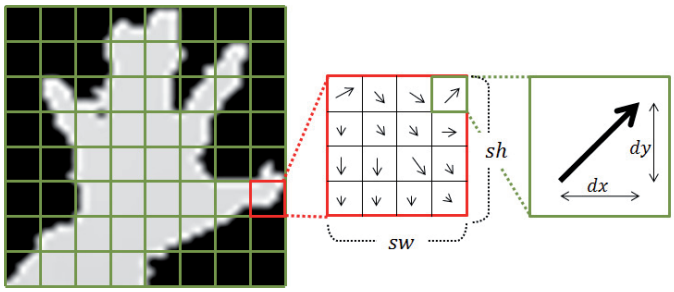


Figure 5.16 – *Description of hand shape feature proposed by Jaemin et al. [54].*

Using depth images, Liang et al. [74] present an approach capable to recognizing the gesture from only one example of each class. In this work, background removal and denoising are firstly performed on depth images. Motion Energy Information (MEI) images are then obtained through calculating the differences between consecutive frames (see Figure 5.17). Within each MEI image, successive movements are represented by time series using Histograms of Oriented Gradients (HOG) descriptor. A PCA reconstruction approach is applied on the descriptor to find a set of discriminately informative principle components (PCs) from the corresponding training gesture.

Figure 5.17 – *MEI image and corresponding HOG descriptors presented in [74].*

Guan et al. [46] segment hand regions from the depth images and convert them into 3D point clouds. 3D moment invariants are then computed as feature descriptors. However, this features encoded only the shape information of human hand.

A more complete review [108] presents a literature review on the use of depth for hand tracking and gesture recognition. This survey examines 37 papers describing depth-based gesture recognition systems in terms of (1) hand localization and gesture classification methods developed and used, (2) the applications where gesture recognition has been tested, and (3) the effects of the low-cost Kinect and OpenNI software libraries on gesture recognition research.

5.7 Discussion and conclusion

With the advantages provided by the low-cost depth sensors for activity recognition, recent research works investigate on several approaches using either depth or skeleton stream for this task.

The depth map-based methods rely mainly on features, either local or global, extracted from the space time volume. In fact, depth images provide natural surfaces which can be exploited to capture the geometrical features of the observed scene in a rich descriptor.

Some holistic approaches, use global feature to describe the entire sequence. In these approaches, the whole sequence is represented in one unique description giving the advantage to be robust to noise and occlusions.

The skeletons estimated from depth images are quite accurate under experimental settings and bring benefits to action modelling and recognition. However, joint location estimation is limited at the same time. In fact, it fails when the human body is partly in view, and when the action involves human object interaction. On the other hand, features extracted from depth images can be efficient in describing actions when skeleton fail to do it. This observation leads us to believe that approaches using skeleton data can be efficient and sufficient in certain applications, such as gaming or human computer interaction. However, in other cases, when human is in interaction with objects or when sequences contain hand gesture, the depth data is more efficient and also sufficient.

Most of state-of-the-art approaches are presenting solutions which are based on the nature of the data, where the whole process is changing with

the given descriptors. A better solution would be to present an unified framework that can work with either skeleton or features extracted from depth images.

Thus, in the following chapter, we investigate this issue and propose a unified framework, which can work independently of the input features. We have seen that the requirement of a system varies with the application needs. Thus, we focus on specific scenarios, as in action recognition systems of single actions. By single actions, we refer to the action sequences where the human in motion is engaged with one action only, through the whole sequence and can or not interact with an object. The fundamental characteristics of the needed system for single action recognition in human computer interaction systems are: (1) high accuracy, where the system must be reliable and thus accurate at recognizing actions, (2) low latency i.e a system with fast response which can even recognize the action before the end on the sequence.

6 Human Gesture and Action Recognition using depth cameras

Sommaire

6.1 Introduction

The recent release of consumer depth cameras, like Microsoft Kinect, has significantly lighten certain difficulties that reduce the action recognition performance in 2D video. These cameras provide in addition to the RGB image a depth stream allowing to discern changes in depth in certain viewpoints. In addition to their invariance to illumination changes, these cameras have eased the task of object segmentation and background subtraction. However, the major problem in an action recognition system is how to model the spatiotemporal sequences ? Once the dynamic of actions is modelled how to learn actions while decreasing intra-classe variability and increasing inter-classes variability ?

Many researchers have recently proposed a variety of techniques for action recognition using depth data, where most of them inspired by existing methods in 2D video. However, although geometric approaches and specially subspaces form non-Euclidean and curved Riemannian manifolds are allowing a video to be conveniently represented as a point on a Grassmann manifold, these approches are still very little explored and investigated using the 3D data.

Variety of works mainly on 2D video show that better performances can be achieved when the geometry of Grassmann manifold is explicitly considered in action modelling. Thus, interested by geometric approaches which are only tested on 2D videos using image color information or 2D silhouettes, we propose to investigate the action recognition task in 3D video using such approach.

6.1.1 Grassmann manifold

The Grassmann manifold has long been known for its interesting mathematical properties, and as an example of homogeneous spaces of Lie groups [138]. However, its applications in computer science and engineering have appeared rather recently in signal processing and control, numerical optimization and machine learning in computer vision.

In our case, we are interested in the representation of the video sequence in a space where each element of this space is a sequence of or-

dered elements. In such a space, we have to be able to compute distance between elements, and also to perform some statistical operations needed for temporal sequence classification task.

Let us define a video as an ordered collection of feature vectors with time-stamps (temporal information). This sequence can be modelled as linear subspaces through linear dynamic systems that take into account the temporal information.

These subspaces represented in Grassmann manifold allow encoding a matrix information as a point on this manifold. Besides, studies show that better performance can be achieved when the geometry of Riemannian spaces is explicitly considered [126, 41]. Especially, Grassmann manifold provides a natural way to deal with the problem of sequence representation, matching and clustering.

In fact, this manifold offers tools to compare and to perform statistics. The recognition problem of a sequence represented by a collection of features can be transformed to point classification problem on the Grassmann manifold as illustrated in the cartoon Figure 6.1.

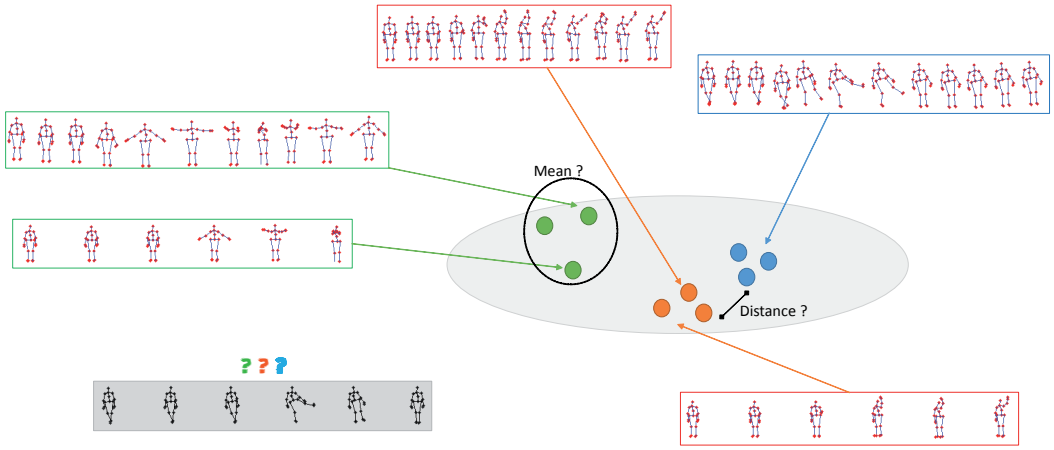


Figure 6.1 – *Structural illustration for a sequence classification task, where query and gallery sequences possess multiple instances of data. In this figure, each sequence (presented here by a set of skeletons) can be represented as a point on Grassmann manifold and thus it is possible to compute distances between sequence elements and statistics.*

6.1.2 Existing approaches

Beside classical methods performed in Euclidean space, a variety of techniques based on manifold analysis are recently proposed. These geometric

methods explore the characteristics of Grassmann manifold and perform classification based on intrinsic geometry of data space.

Turaga et al. [123] involve a study of the geometric properties of the Grassmann and Stiefel manifolds and give appropriate definitions of Riemannian metrics and geodesics for the purpose of video indexing and action recognition. In another work, Turaga et al. [122] use the same approach to represent complex actions by a collection of subsequences. These sub-sequences correspond to a trajectory on the Grassmann manifold. Both DTW and HMM are used for action modelling and comparison.

Lui et al. [79] introduce the notion of tangent bundle to represent each action sequence on the Grassmann manifold. Videos are expressed as a third-order data tensor of raw pixel from action images, which are then factorized on the Grassmann manifold. As each point on the manifold has an associated tangent space, tangent vectors are computed between elements on the manifold and obtained distances are used for action classification in a nearest neighbour fashion. In the same way, Lui et al. [78] factorize raw pixel from images by high-order singular value decomposition in order to represent the actions on Stiefel and Grassmann manifolds. However, in these works, no dynamic modelling of the sequence, where the raw pixels are directly factorized as manifold points. In addition, no training process on data and only distances obtained between all actions are used for action classification.

Kernels [96, 43] are also used in order to transform the subspaces of a manifold onto a space where Euclidean metric can be applied. Shirazi et al. [96] embed Grassmann manifolds upon a Hilbert space to minimize clustering distortions and then apply a locally discriminant analysis using a graph. Video action classification is then obtained by a Nearest-Neighbour classifier applied on Euclidean distances computed on the graph-embedded kernel. Similarly, Harandi et al. [43] propose to represent the spatio temporal aspect of the action by subspaces as elements of the Grassmann manifold. Figure 6.2 illustrates an example of action presented by subspaces. They embed this manifold into reproducing ker-

nel Hilbert spaces in order to tackle the problem of action classification on such manifolds.

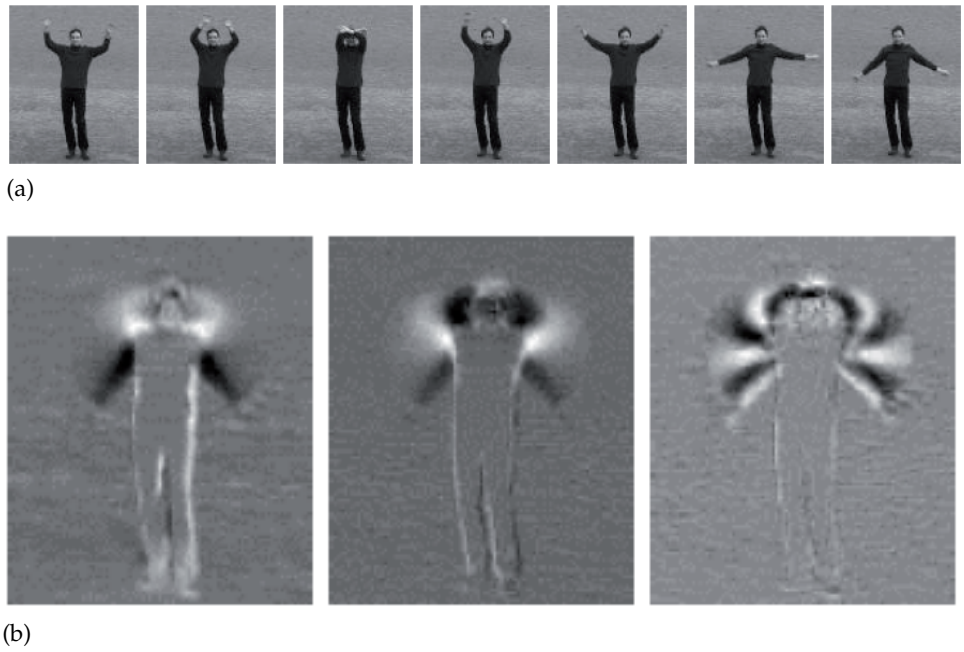


Figure 6.2 – *Example of modelling an action sequence by a subspace of order three [43]. (a) Examples of a hand-waving action in a 2D video. (b) Basis vectors for a subspace of order three, modelling the entire action; the subspace is a point on a Grassmann manifold.*

It is important to note that, to date, few works have recently proposed to use Grassmann manifold analysis for 3D action recognition. Indeed, Azary et al. [11] use a Grassmannian representation as an interpretation of Depth Motion Image (DMI) computed from depth pixel values. All DMI in the sequence are combined to create a motion depth surface representing the action as a spatiotemporal descriptor.

From the above state of the art, we can conclude that the geometrical modelling of the action sequence from 2D images on the Grassmann manifold is significant and it allows discriminating between different classes of actions. This has been shown by the work of [96, 43, 79] who proposed to compare sequences using a metric defined on the Grassmann manifold. This metric is sometimes complex and is based on the notion of tangent Bundle. Recently, Harandi et al. [43] have checked the performance of Riemannian manifolds, in representing human activity, against several state-of-the-art methods. Conducting several experiments, including ges-

ture recognition and person identification, Grassmann manifold has been demonstrated as the one that gives the best performance.

Besides, Linear Dynamic Systems (LDS) [132] show more and more promising results on the motion modelling since they exhibit the stationary properties in time, so they fit for action representation. Thus, the problem of action recognition using 3D images from depth stream can be investigated using the LDS and Grassmann manifold geometry.

6.1.3 Overview of our approach

Motivated by the above issues, we propose in this chapter a novel method to recognize human actions in 3D video sequences, using a geometric structure inherent in the Grassmann manifold. Action recognition is performed by introducing a learning algorithm on the manifold in conjunction with dynamic modelling process.

First, we construct time series as a sequence of consecutive feature vectors with temporal order. Second, to capture the temporal deformation and the dynamic of the motion, we propose to capture spatiotemporal information by linear dynamic systems. Then, the observability matrix of this model is characterized as an element of a Grassmann manifold.

To formulate our learning algorithm, we propose two distinct process: (1) In the first one, we perform classification using a Truncated Wrapped Gaussian model using features computed from depth map information, one for each class in its own tangent space. (2) In the second one, we propose an original learning method using a vector representation formed by 3D skeleton coordinates in tangent spaces associated with different classes in order to train a linear SVM. The overview of the proposed approach is sketched in Figure 6.3.

6.2 Mathematical notations and definitions

To model, learn and compare sequences on the Grassmann manifold, we need to understand (1) the representation of points, (2) distance metrics and (3) statistical models on the manifold.

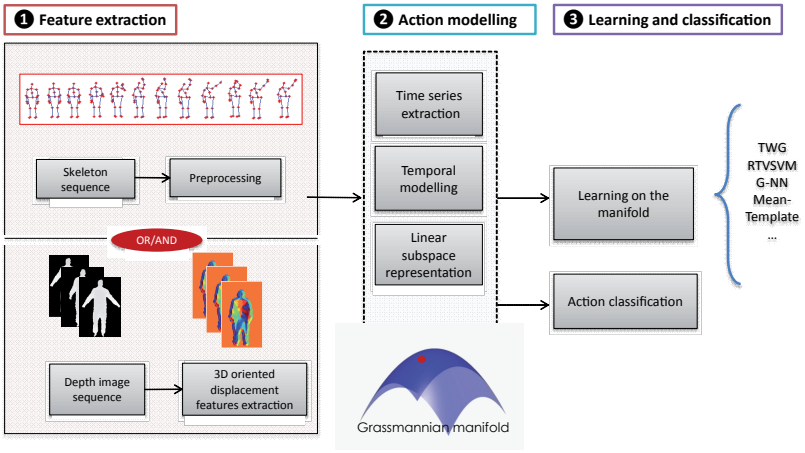


Figure 6.3 – *Overview of the approach using both joint and depth information. The global overview have the following main steps : (1) feature extraction from input video stream (2) Spatiotemporal modelling of the features on the Grassmann manifold (3) Inference on the manifold in order to perform the learning step.*

A manifold is a topological space locally similar to Euclidean space and a Riemannian manifold is provided with a metric which allows measuring the similarity between two points. In this work, we are interested in Grassmann manifolds which definition is as below.

Definition 6.2.1 *The Grassmann manifold $G_{n,d}$ is the set of all d-dimensional linear subspaces of $\mathbb{R}^n$.*

Several textbooks describe the Grassmann manifold structure and its geometry and calculus. In this thesis we focus on the algorithms proposed by Gallivan et al. [36]. Here, the Grassmann manifold is viewed as the quotient space : $SO(n)/SO(d) \times SO(n-d)$ where $SO(n)$ is the special orthogonal group of orthogonal matrix with determinant +1.

6.2.1 Special orthogonal group $SO(n)$

Let $GL(n)$ be the *generalized linear group* of $n \times n$ nonsingular matrices. The set $GL(n)$ is a differentiable manifold, therefore although it is not a vector space, it can be locally approximated as a vector space using smoothly varying Euclidean coordinates. This property is essential to understanding the task of modifying tools from standard Euclidean statistics to nonlinear manifolds. By being a group and a differentiable manifold $GL(n)$ is

a Lie group. The subset of all orthogonal matrices with determinant $+1$, form a subgroup $SO(n)$, called the *special orthogonal group*. This latter is a submanifold of $GL(n)$ and, therefore, also possesses a Lie group structure.

To perform differential calculus on a manifold, one needs to specify its tangent spaces.

For the $n \times n$ identity matrix I, the tangent space $T_I(SO(n))$ is the set of all $n \times n$ skew-symmetric matrices given by [15]:

$$T_I(SO(n)) = X \in R^{n \times n} : X + X^T = 0 \tag{6.1}$$

Proposition 6.2.1 *The **tangent space** at an arbitrary point $O \in SO(n)$ is obtained by a simple rotation of $T_I(SO(n))$:*

$$T_O(SO(n)) = \{OX | X \in T_I(SO(n))\} \tag{6.2}$$

Define an inner product for any $Y, Z \in T_O(SO(n))$ by $\langle Y, Z \rangle = trace(YZ^T)$, where *trace* denotes the sum of diagonal elements. With this metric $SO(n)$ becomes a Riemannian manifold. Using the bi-invariant Riemannian structure, it becomes possible to define lengths of paths on a manifold. Let $\alpha : [0,1] \longrightarrow SO(n)$ be a parameterized path on $SO(n)$ that is differentiable everywhere on [0,1]. Then $\frac{d\alpha}{dt}$, the velocity vector at t, is an element of the tangent space $T_{\alpha(t)}(SO(n))$.

For any two points $O_1, O_2 \in SO(n)$, a distance between them can be defined as the infimum of the lengths of all smooth paths on $SO(n)$ which start at O_1 ($\alpha(0) = O_1$) and end at O_2 ($\alpha(1) = O_2$):

$$d(O_1, O_2) = \inf_{\{\alpha:[0,1] \longrightarrow SO(n)\}} \int_0^1 \sqrt{(\langle \frac{d\alpha}{dt}, \frac{d\alpha}{dt} \rangle)} dt. \tag{6.3}$$

A path $\hat{\alpha}$ which achieves the above minimum is a geodesic between O_1 and O_2 on $SO(n)$.

Geodesics on $SO(n)$ can be written explicitly using the matrix exponential.

Definition 6.2.2 *For an $n \times n$ matrix A, define its matrix exponential* $\exp(A)$ *by:* $\exp(A) = I + \frac{A}{1!} + \frac{A^2}{2!} + \frac{A^3}{3!} + ...$

We can see that given any skew-symmetric matric X, $\exp(X) \in SO(n)$. Now we can define geodesics on $SO(n)$ as follows: for any $O \in SO(n)$ and any skew-symmetric matrix X, $\alpha(t) = O\exp(tX)$ is the unique geodesic in $SO(n)$ passing through O with velocity vector OX at $t = 0$.

The **exponential map** is an important tool in statistics on the manifold. If M is a Riemannian manifold and $p \in M$, the exponential map $\exp_p : T_p(M) \longrightarrow M$, is defined by $\exp_p(v) = \alpha_v(1)$ where α_v is a constant speed geodesic starting at p. In case of $SO(n)$, the exponential map $\exp_O : T_O(SO(n)) \longrightarrow SO(n)$ is given by

$$\exp_O(X) = Oexp(X) \tag{6.4}$$

where the exponential on the right side is actually the matrix exponential.

6.2.2 $G_{n,d}$ as a quotient space

A quotient of a space defines equivalence relations between points in the space. If one wants to identify certain elements of a set, using an equivalence relation, then the set of such equivalent classes forms a quotient space. This framework is very useful in understanding the geometry of $G_{n,d}$ by viewing it as a quotient space, using different equivalence relations, of $SO(n)$.

In order to obtain a quotient space structure for $G_{n,d}$, let $SO(d) \times SO(n-d)$ be a subgroup of $SO(n)$ using the embedding $\phi : (SO(d) \times SO(n-d)) \longmapsto SO(n)$:

$$\phi(V1, V2) = \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix} \in SO(n) \tag{6.5}$$

Definition 6.2.3 *Define an **equivalent relation** on $SO(n)$ according to $O_1 \sim O_2$ if $O_1 = O_2\phi(V_1, V_2)$ for some $V_1 \in SO(d)$ and $V_2 \in SO(n-d)$. In other words, O_1 and O_2 are equivalent if the first d columns of O_1 are rotations of the first d columns of O_2 and the last $(n-d)$ columns of O_1 are rotations of the last $(n-d)$ columns of O_2.*

The set of all equivalence classes is $G_{n,d}$, where an **equivalence class** is

given by :

$$[O] = \{O\phi(V_1, V_2) \setminus V_1 \in SO(d), V_2 \in SO(n-d)\} \tag{6.6}$$

An other notation of $G_{n,d}$ could be as fellow: $SO(n)/(SO(d) \times SO(n-d))$. For efficiency, we denote the set of $[O]$ by the set:

$$[U] = \{UO \in \mathbb{R}^{n \times d} | O \in SO(d)\} \tag{6.7}$$

where U denotes the first d columns of O.

The main advantage of studying the Grassmann manifold as quotient spaces of $SO(n)$ is that it allow using systematically the well-known results about geodesics and tangent planes of $SO(n)$.

6.2.3 Tangent space of $G_{n,d}$

Let M/H is a quotient space of M under the action of a group $H \subset M$ (assuming H acts on M). Then, for any point $p \in M$, a vector $v \in T_p(M)$ can be identifies as tangent to M/H as long as it is perpendicular to the tangent space $T_p(pH)$. Here, $T_p(pH)$ is considered as a subspace of $T_p(M)$.

Following the same principle, we define a tangent space on $G_{n,d}$, while $M = SO(n)$ and $H = \phi(SO(d) \times SO(n-d))$, with ϕ as given in Equation 6.5. A tangent space $T_I(H)$ is considered as a subspace of $T_I(SO(n))$ under the embedding $d\phi$

$$d\phi(A1, A2) = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \in T_I(SO(n)) \tag{6.8}$$

The vectors tangent to $SO(n)$ and perpendicular to the space $(T_{I_d}(SO(d)) \times T_{I_{n-d}}(SO(n-d)))$, can be identified to the tangent to $G_{n,d}$ after multiplication on right by J (where $J \in \mathbb{R}^n \times d$ is first d columns of I_n). The resulting tangent space at $[J] \in G_{n,d}$ is:

$$T_{[J]}(G_{n,d}) = \{ \begin{bmatrix} 0 \\ B^T \end{bmatrix} \setminus B \in \mathbb{R}^{d \times (n-d)} \} \tag{6.9}$$

For any other point $[U] \in G_{n,d}$, let $O \in SO(n)$ be a matrix such that

$U = O^T J$. Then, the **tangent space** at $[U]$ is given by

$$T_{[U]}(G_{n,d}) = \{O^T R \setminus R \in T_{[J]}(G_{n,d})\} \tag{6.10}$$

The geodesic flow starting from a point $[U] \in G_{n,d}$ in a direction $O^T AJ \in T_{[}U](G_{n,d})$, is given by:

$$\Psi_U(O^T AJ, .) : t \longmapsto O^T \exp(tA)J \tag{6.11}$$

where A is of the type $\begin{bmatrix} 0 & -B \\ B^T & 0 \end{bmatrix}$

6.2.4 Exponential map and logarithm map computation

Exponential map and logarithm map operators are interesting tools allowing going from the manifold to the tangent space and vice versa from the tangent space to the manifold. They are specially used to take benefit from the fact that the tangent space is a vector space. Besides, these tools will be used in statistical computation step, for example to compute intrinsic mean. Also the action modelling and classification is using these operators in the learning algorithms presented thereafter.

Computing velocity matrix (log) [36] Given two points on the manifold U_1 and U_2 with orthonormal basis Y_1 and Y_2, we need an efficient way to compute the velocity parameter V such that traveling in this direction from S_0 leads to S_1 in unit time. Given two subspaces $S0$ and $S1$ and corresponding $n \times d$ orthonormal basis vectors Y_1 and Y_2:

1. Compute the $n \times n$ orthogonal completion Q of Y_1.
2. Compute the thin decomposition of $Q^T Y_2$ given by $Q^T Y_1 = \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix} \begin{bmatrix} \Gamma(1) \\ \Sigma(1) \end{bmatrix} V_1^T$
3. Compute $\{\theta_i\}$ which are given by the *arcsine* and *arcos* of the diagonal elements of Γ and Σ respectively. Form the diagonal matrix Θ containing θ s on its diagonal.

4. Compute $V = M_2 \Theta M_1$.

Moving along the geodesic (exp) [36] Given a point on the Grassmann manifold U_1 represented by orthonormal basis Y_1, and a direction matrix B, the geodesic path emanating from Y_1 in this direction is given by $Y(t) = Q \exp(tA) J$, where, $Q \in SO(n)$ and $Q^T Y_1 = J$ and $J = [I_d; 0_{n-d,d}]$. Given Y_1 and A the following are the steps involved in sampling $Y(t)$ for various values of t:

1. Compute the $n \times n$ orthogonal completion Q of Y_1. This can be achieved by the QR decomposition of Y_1.
2. Compute the compact SVD of the direction matrix $B = M_2 \Theta M_1$.
3. Compute the diagonal matrices $\Gamma(t)$ and $\Sigma(t)$ such that $\gamma_i(t) = cos(t\theta_i)$ and $\sigma_i(t) = sin(t\theta_i)$, where θ are the diagonal elements of Θ.
4. Compute $Y(t) = \begin{bmatrix} M_1 \Gamma(t) \\ -M_2 \Sigma(t) \end{bmatrix}$ for various values of $t \in [0, 1]$.

Let now μ denotes an element of $G_{n,d}$, the tangent space to this element is noted T_μ, it is the tangent plane to the surface of the manifold at μ. It is possible to map a point U_1, of the Grassmann manifold, to a vector V_1 in the tangent space T_μ using the logarithm map as defined by Gallivan et al. [36]. This operation will be noted in this thesis by log where $log_\mu : G_{n,d} \longmapsto T_\mu(G_{n,d})$. An other important tool in statistics is the exponential map, $\exp_\mu : T_\mu(G_{n,d}) \rightarrow G_{n,d}$ which allows to move on the manifold in certain direction. An illustration of these concepts is presented in Figure 6.4.

6.2.5 Angles and distance

Between two points U_1 and U_2 on $G_{n,d}$ there are d principal angles of $\mathbb{R}^n$: $0 \leq \theta_1 \leq \theta_2 \leq \cdots \leq \theta_d \leq \frac{\pi}{2}$. The principal angles may be computed as the inverse cosine of the singular values of $U_1^T U_2$. The minimum length curve connecting these two points is the geodesic between them computed as:

$$dG(U_1, U_2) = \| [\theta_1, \theta_2, \cdots, \theta_i, \cdots, \theta_d] \|_2 \tag{6.12}$$

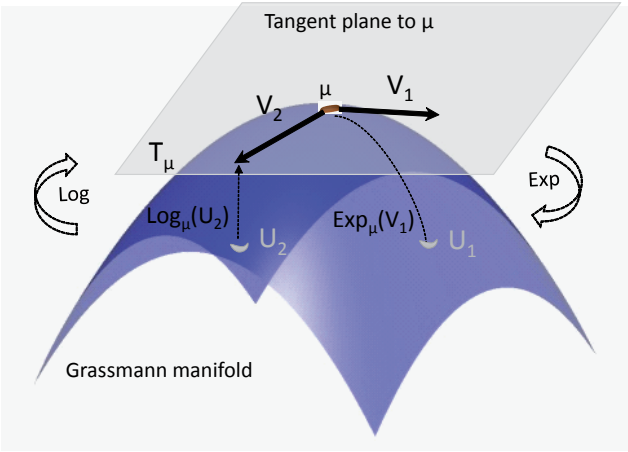


Figure 6.4 – *Illustration of tangent spaces, tangent vectors, and geodesics on Grassmann manifold. μ is a point on the manifold. T_μ is the tangent space at μ. Tangent vector corresponds to the velocity of the curve on the manifold. Geodesic path is constant velocity curves on the manifold. The exponential map is a pullback map which takes a point on the tangent space and pulls it onto the manifold in a manner that preserves distances. An example of one point V_1 on the tangent space at pole μ.*

This is known as the arc length metric, commonly used to compute distances on the Grassmann manifold. The geometric framework for this description is presented with more details in [36].

6.3 Statistics on Grassmann manifold

Any statistical inference problem on $G_{n,d}$ requires computation of sample statistics. Since $G_{n,d}$ is a non linear space, it is not straightforward to define and to compute even basic statistics such as covariances and means. There are two types of statistics popularly used on non linear spaces :

(1) **Extrinsic statistics** [105]: The manifold $G_{n,d}$ is embedded in a larger Euclidean space, statistics are computed in this larger space and then projected back to $G_{n,d}$. Here the computation is relatively simple, however the main limitation is related to the non uniqueness of the embedding which leads to a non-uniqueness of statistics.

(2) **Intrinsic statistics** [14]: They are completely restricted to the manifolds themselves and do not rely on any euclidean embedding. The computation of such statistics requires an iterative procedure where both exponentiation and logarithm are used iteratively in each step. Despite the

complex aspect of this computation, here the Riemannian structure of $G_{n,d}$ is used to define uniquely statistics of interest.

In view of the efficient nature of intrinsic models, we opt for the use of intrinsic statistics in our work. In the following, mean computation of a set of Grassmann point cloud is performed via an intrinsic mean called Karcher mean. Then, an unsupervised clustering algorithm, which allows to obtain homogeneous subsets and their centers, is explained.

6.3.1 Karcher mean on Grassmann manifold

Given a set of data points $\{U_1, U_2 \cdots U_N\}$ on a Grassmann manifold sufficiently close to each others, one way to define their geometric mean is via the minimization of a certain cost function. If one chooses the cost as the sum of squared geodesic distances between a given point and all the data points, we end up with the definition of the Karcher mean. The Figure 6.5 illustrates a Karcher mean of a sample of elements.

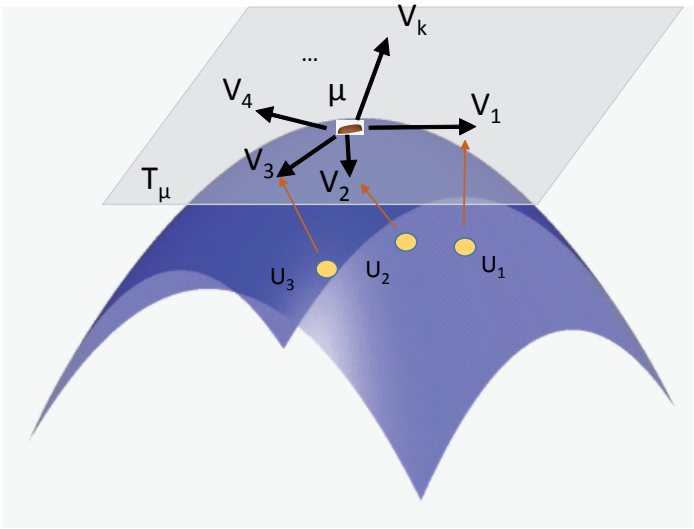


Figure 6.5 – *Grassmann points, their Karcher mean and their projection onto the tangent space of μ.*

The algorithm exploits *log* and *exp* maps (6.2.4) in a predictor/corrector loop until convergence to an expected point. The pseudocode for computing a sample karcher mean on Grassmann manifold is summarized in Algorithm 4.

Karcher mean in our geometric framework for action recognition is useful in various situations, including: computation of mean of each

class of actions to use it as a template, computation of mean of all action observations to construct a vocabulary of actions.

Algorithm 4: Karcher mean computation on a Grassmann manifold

Input: $\{U_1, U_2 \cdots U_N\}$: points belonging to $G_{n,d}$,

$\epsilon = 0.5$, τ: threshold which is a very small number

Output: μ_j : mean of $\{U_i\}_{i=1:N}$

1- μ_0: initial estimate of Karcher mean, for example one could just take $\mu_0 = U_1$

repeat

  for $i \leftarrow 1$ **to** N **do**

    2- Compute $v_i = log_{\mu}(U_i)$

    3- Compute the average direction $\overline{v} = \frac{1}{n}\sum_{i=1}^{n} v_i$

    4- Move μ_j in the direction of $\overline{v}$ by ϵ: $\mu_{j+1} = exp_{\mu_j}(\epsilon\overline{v})$

  5- j=j+1

until $\|\overline{v}\| < \tau$;

6.3.2 K-means on Grassmann manifold

The Karcher mean is a statistical tool which can also be used for unsupervised learning tasks such as data clustering. In fact, it is possible to estimate clusters of elements on Grassmann manifold in an intrinsic manner. Let us assume that we have a set of points $\{U_1, U_2 \cdots U_N\}$ on the Grassmann manifold. We seek to estimate k clusters with cluster centers $(\mu_1, \mu_2, \cdots, \mu_k)$ so that the sum of geodesic distance squares, is minimized. Like standard k-means, this problem is solved using an EM-based approach [123]. First, we initialize the algorithm with a random selection of k points as the cluster centers. In the E-step, each of the points is assigned to the nearest cluster center. Then in the M-step, the cluster centers are computed using the Karcher mean algorithm as described in Algorithm 4. The intrinsic k-means computation algorithm is summarized in Algorithm 5. The intrinsic k-means [123] computation on Grassmann manifold, allows unsupervised clustering on actions which can be usefull for several applications such as Hierarchical clustering or unsupervised

clustering and learning.

Algorithm 5: K-means clustering algorithm

Input: $\{U_1, U_2 \cdots U_N\}$: points belonging to $G_{n,d}$, k number of clusters, N_{max} maximum iteration

Output: $\{\mu_i\}_{i=1:k}$ k cluster centers

while $(j < N_{max})$ **do**

1- Initialize cluster centers randomly $(\mu_1^0, \mu_2^0, \cdots, \mu_k^0)$

2- Compute distances from each U_i to all μ_k: $d(U_i, \mu_k)$

3- Assign each U_i to nearest cluster center μ_k

4- Recompute new cluster centers,$(\mu_1^j, \mu_2^j, \cdots, \mu_k^j)$, using Karcher mean algorithm 4.

5- j=j+1

6.4 Action and gesture recognition using depth information

In this section, we model the human motion in the depth map space. Particularly, our intent is to represent the motion from depth images in a geometric and efficient way, leading to an accurate action-recognition system. In this representation, we propose to consider data information from depth images and represent each sequence by a time series from its local displacement features. The overview of the proposed approach is sketched in Figure 6.6.

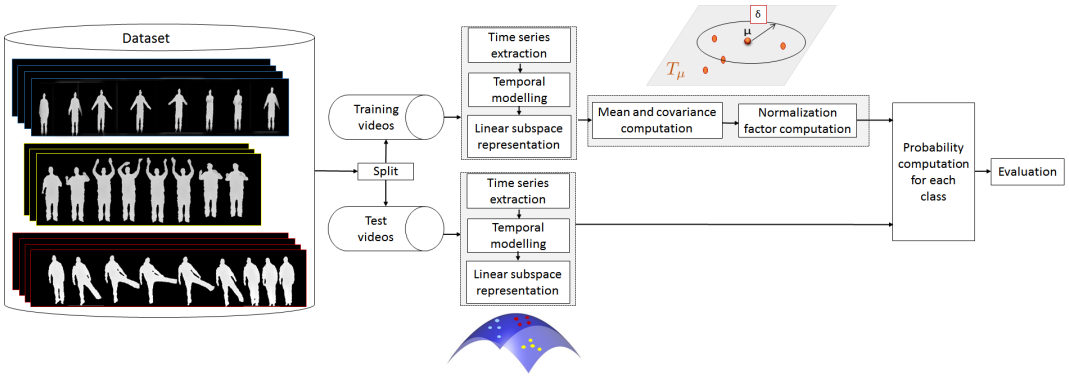


Figure 6.6 – *Overview of the approach. The illustrated pipeline is composed of two main modules: (1) temporal modelling of time series data and manifold representation (2) learning approach using probability density function on tangent class-specific.*

6.4.1 Time series of 3D oriented displacement features

In 3D action sequence produced by depth sensors, each frame is represented of either a skeleton or a depth image. In both cases, it is possible to extract from each frame some descriptors that we represent in a data vector. A motion sequence can then be seen as a matrix collecting all time-series from p features in each frame, i.e., $M = [f^1 f^2 \cdots f^T]$, $f \in \mathbb{R}^p$ where f^i is the vectorised representation of features of frame i.

The depth information captured by a depth sensor is usually called the depth image. We denote each pixel in the depth image as $P = (x; y; z)$. Let $I = [I(1), I(2), ..., I(t), I(\tau)]$ denotes the depth sequence. This sequence can be seen as a 4D surface S in the 4D space if we consider a function [87] :

$$\begin{aligned} &\mathbb{R}^3 \longrightarrow \mathbb{R}^1 \\ &(x, y, t) \longmapsto z = f(x, y, t) \end{aligned} \tag{6.13}$$

Since the orientation of a normal vector, at every surface point, can describe the surface of an object, the local 4D geometry characteristics (Depth + motion) can be represented as a local displacement of the normal vector orientation. The normals of this surface are given by a derivation of $S(x, y, z, t)$ where $S(x, y, z, t) = f(x, y, t) - z = 0$.

Thus, the result of the derivation, following the same demonstration of Tang et al. [113] is given by:

$$n = \bigtriangledown S = (\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}, \frac{\partial z}{\partial t}, -1)^T = (n_x, n_y, n_t, -1)^T \tag{6.14}$$

Experimentally $\frac{\partial z}{\partial x}$, $\frac{\partial z}{\partial y}$ and $\frac{\partial z}{\partial t}$ are calculated using the finite difference approximation respectively:

$$\begin{aligned} n_x &= \tfrac{\partial z}{\partial x} \simeq I(x - Diff, y, t) - I(x + Diff, y, t) \\ n_y &= \tfrac{\partial z}{\partial y} \simeq I(x, y - Diff, t) - I(x, y + Diff, t) \\ n_t &= \tfrac{\partial z}{\partial t} \simeq I(x, y, t) - I(x, y, t + 1) \end{aligned} \tag{6.15}$$

where $Diff$ is a positif value of displacement on image matrix. Encoding the orientation information of this normal is more meaningful for describing the surface than (x,y,z,t) coordinates. Thus, these local ori-

ented displacements can be parametrized using spherical coordinates represented as 3 angles Θ, Φ and Ψ descibing respectively zenith angle, azimuth angle and inclination angle. These angles, which are illustrated in Figure 6.7, are computed as follows:

$$\begin{aligned} \Theta &= \tan^{-1}\left(\sqrt{n_x^2 + n_y^2 + n_t^2}\right) \\ \Phi &= \tan^{-1}\left(\frac{n_y}{n_x}\right) \\ \Psi &= \tan^{-1}\left(\frac{n_t}{\sqrt{(n_x^2 + n_y^2)}}\right) \end{aligned} \tag{6.16}$$

The local oriented displacements describe the motion of an object, indicating how much distance it moves in each one of the three directions. If the movement in all directions are saved accurately, the movement can be repeated from the initial position to the final destination regardless of the displacements order. However, a temporal modeling of these descriptors along the action sequence is needed to capture the dynamic of the action, as shown in next section.

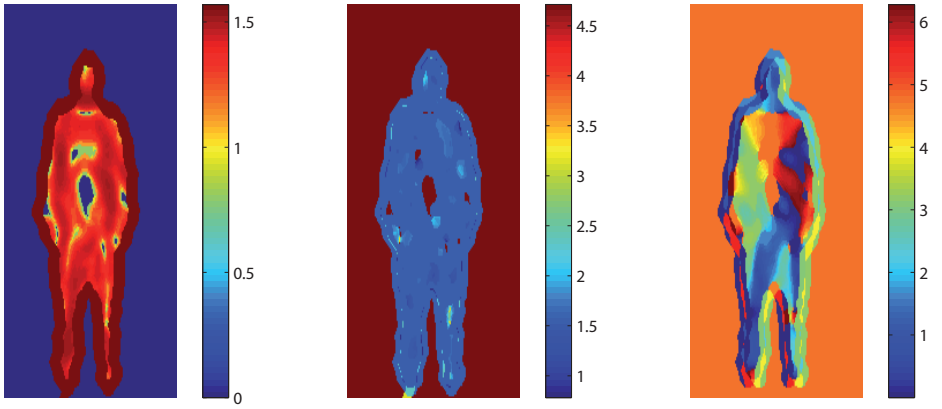


Figure 6.7 – *3D angles illustration. Each pixel of these images is given by the value of* Θ, Φ *and* Ψ *respectively from left to right.*

6.4.2 Spatiotemporal modelling of action

Modelling the sequence as a feature vector set $M = \{p^1 p^2 \cdots p^T\}$, $p \in \mathbb{R}^p$ by linear subspaces has been shown to deliver improved performance in the presence of practical issues such as misalignment as well as variations in pose and presence of noise. Thus, this matrix can be represented as a subspace (and hence as a point on a Grassmann manifold) through any

orthogonalisation procedure like Singular Value Decomposition (SVD). However, modelling of actions by frame sets can be sufficient provided that the order in which the action is performed is not very relevant to decision making. This assumption is restrictive, and a recent study shows that an extended type of frame set, obtained through a block Hankel matrix formalism, can capture the temporal information [72].

Let assume now that the sequence is represented by time series corresponding to ordered feature vectors extracted at time t from each frame.

Considering time series, we could use DTW algorithm [37] to find optimal non-linear warping function to match these given time-series as proposed by [93, 99, 39]. However, we opted for a system combining a linear dynamic modelling with statistical analysis on a manifold, avoiding the boundary and the monotonicity constraints presented by classical DTW algorithm. Such a system is also less sensitive to noise due to the poor estimation of depth data (joint locations or depth maps) in addition to its reduced computational complexity.

Dynamical systems are a powerful tool to work with temporally ordered data in time series. They have been used in several applications in computer vision, including tracking, human recognition from gait, activity recognition and dynamic texture. The main idea is to use a dynamical system to model the temporal evolution of a measurement vector $p(t) \in \mathbb{R}^n$ as a function of a relatively low dimensional state vector $z(t) \in \mathbb{R}^d$ that changes over time. The measurement vector $p(t)$ can represent the pixel values or the feature values of an image captured at time t. The simple dynamical model is an Auto-Regressive and Moving Average (ARMA) model. Its main advantage is that it decouples the appearance of the spatiotemporal data from the dynamics of the motion. The ARMA model equations are given by:

$$p(t) = Cz(t) + w(t), \quad w(t) \sim N(0, R), \tag{6.17}$$

$$z(t+1) = Az(t) + v(t), \quad v(t) \sim N(0, Q) \tag{6.18}$$

where $z \in \mathbb{R}^d$ is a hidden state vector, $A \in \mathbb{R}^{d\times d}$ is the transition matrix and $C \in \mathbb{R}^{p\times d}$ is the measurement matrix. w and v are noise components modeled as normal with mean equal to zero and covariance matrix $R \in \mathbb{R}^{p\times p}$ and $Q \in \mathbb{R}^{d\times d}$ respectively. The goal is to estimate parameters of the model (A, C) given by these equations. Let $U\sum V^T$ be the singular value decomposition of the matrix M. Then, the estimated model parameters $\hat{A}$ and $\hat{C}$ are given by [29]:

$$\begin{aligned} \hat{C} &= U \\ \hat{A} &= \sum V^T D_1 V (V^T D_2 V)^{-1} \sum^{-1} \end{aligned} \tag{6.19}$$

where $D_1 = \begin{pmatrix} 0 & 0 \\ I_{\tau-1} & 0 \end{pmatrix}$, $D_2 = \begin{pmatrix} I_{\tau-1} & 0 \\ 0 & 0 \end{pmatrix}$ and $I_{\tau-1}$ is the identity matrix of size $\tau - 1$.

Comparing two ARMA models can be done by simply comparing their observability matrices. Starting from an initial condition z(0), it can be shown that the expected observation sequence is given by :

$$E\begin{bmatrix} p(0) \\ p(1) \\ p(2) \\ . \\ . \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ . \\ . \end{bmatrix} z(0) = \theta_\infty(M) z(0) \tag{6.20}$$

Thus, the expected observation sequence generated by an ARMA model $\mathcal{M} = (A, C)$ lies in the column space of the extended observability matrix given by

$$\theta_\infty^T = [C^T, (CA)^T, (CA^2)^T, ...] \tag{6.21}$$

This can be approximated by the finite observability matrix [123]:

$$\theta_m^T = [C^T, (CA)^T, (CA^2)^T, ..., (CA^{m-1})^T] \tag{6.22}$$

The subspace spanned by columns of this finite observability matrix (obtained by any orthogonalisation procedure) corresponds to a point on a Grassmann manifold. In the rest of this chapter, each video sequence is

modelleded as described above to become an element of the Grassmann manifold and the action learning and recognition problem is brought back to a classification problem on this manifold.

6.4.3 Learning on the Grassmann manifold by Truncated Wrapped Gaussian

Let $\mathcal{L} = \{(U_1, l_1), (U_2, l_2), \cdots, (U_i, l_j), \cdots, (U_N, l_k)\} = \{(U_i, l_j)\}_{i=1:N, j=1:k}$ be a dataset of labeled actions, where (U_i, l_j) is a couple of an action represented by a point U_i on the Grassmann manifold and its label l_j, N is the total number of actions in $\mathcal{L}$ and k is the number of classes. Each class with a certain label l_j will contain n_j actions.

A common learning approach on manifolds is based on the use of only one-tangent space, which usually can be obtained as the tangent space to the mean (μ) of the entire data points $\{U_i\}_{i=1:N}$ without regard to class labels. All data points on the manifold are then projected on this tangent space to provide the input of a classifier. This assumption provide an accommodated solution to use a classical supervised learning on the manifold. However, this flattening of the manifold through tangent space is not without drawbacks. In fact, the tangent space on the global mean can be far from other points, and the distance on this tangent space between two arbitrary points is generally not equal to the true geodesic distance, which may lead to inaccurate modelling.

It is more intuitive to use several tangent spaces, each obtained on a class of the learning dataset than using only one tangent space computed on the whole data. However, the question here is how to learn a classifier in this case?

The first possibility that is offered to us, is to learn a template for each class by computing Karcher mean for each sample. Then, to recognize an unknown action we compute its distance to all templates and affect it to the class represented by the nearest template.

However, an other possibility more efficient consists on learning a probability law on each class sample having the same label. Indeed, in addition to the mean μ, it is possible to compute the standard deviation σ

between all actions belonging to the same class. The σ value can be computed on $\{V_i\}_{i=1:N}$ where $V = exp_{\mu}^{-1}(U_i)$ are the projections of actions from the Grassmann manifold into the tangent space defined on the mean μ.

Thus, we can estimate the parameters of a probability density function such as a Gaussian and then use the exponential map to wrap these parameters back onto the manifold using exponential map operator [123]. However, the exponential map is not a bijection for the Grassmann manifold. In fact, a line on tangent space with infinite length, can be wrapped around the manifold many times. Thus, some points of this line are going to have more than one image on $G_{n,d}$. It becomes a bijection only if the domain is restricted. Therefore, we can restrict the tangent space by a truncation beyond a radius of π in $T_{\mu}(G_{n,d})$ as illustrated in 6.8.

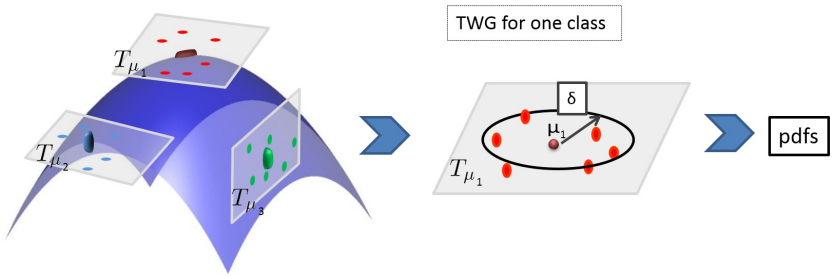


Figure 6.8 – *Conceptual TWG learning method on the Grassmann manifold. Actions belonging to the same class, illustrated with same color, are projected to the tangent space presented with its mean and then Gaussian function is computed on each truncated tangent space.*

By truncation, the normalization constant changes for multivariate density in $T_{\mu}(G_{n,d})$. In fact, it gets scaled down depending on how much of the probability mass is left out of the truncation region.

Let $f(x)$ denotes the probability density function (pdf) defined on $T_{\mu}(G_{n,d})$ by :

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-x^2}{2\sigma^2}} \tag{6.23}$$

After truncation, an approximation of f gives:

$$\hat{f}(x) = \frac{f(x) \times \mathbb{1}_{|x|<\pi}}{z} \tag{6.24}$$

where z is the normalization factor :

$$z = \int_{-\pi}^{\pi} f(x) \times \mathbb{1}_{|x|<\pi} dx \tag{6.25}$$

Using Monte Carlo estimation, it can proved that the estimation of z is given by:

$$\hat{z} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{|x_i|<\pi} \tag{6.26}$$

As illustrated in Figure 6.8, we employ wrapped Gaussians in each class-specific tangent space. Separate tangent spaces is considered for each class at its mean computed by Karcher mean algorithm. Predicted class of an observation point is estimated in these individual tangent spaces. In the training step, the mean, the standard deviation and the normalization factor in each class of actions are computed. The predicted label of unknown class action is estimated as a function of probability density in class-specific tangent spaces.

Algithm 6 is summarizing the whole procedure for the pdf classification by TWG. In this algorithm we highlight the training and testing steps.

6.5 Experimental results in depth spaces

We experimented our proposed approach on three public 3D action and gesture datasets containing various challenges, including MSR-action 3D [73], UT-kinect [145] and MSR-Gesture3D [133] which is a dataset of hand gestures. All details about these datasets: different types and number of motions, number of subjects executing these motions and the experimental protocol used for evaluation are summarized in Table 6.1. Examples of actions from these datasets are shown in Figure 6.9.

6.5.1 Evaluation metric

Activity recognition methods are evaluated mainly by their accuracy which is the percentage of correctly recognizing actions. Several validation techniques are used for this evaluation:

Algorithm 6: pdf classification by TWG on class-specific tangent space

**** *Training* ****

Input: N training actions as points on $G_{n,d}$, belonging to k classes: $\mathcal{L} = \{(U_i, l_j)\}_{i=1:N, j=1:k}$

Output: Estimated multiplication factor $\{\hat{z}_j\}_{j=1:k}$ and standard deviation $\{\sigma_j\}_{j=1:k}$ for each class

for *j=1 : k* **do**

- **1-** Compute the Karcher mean μ_j of the j^{th} class using algorithm 4
- **2- for** *i=1 : n_j* **do**
  - Compute $v_i = log_{\mu_j}(U_i)$
- **3-** Compute the standard deviation σ_j of $\{v_i\}$
- **4-** Sample a large number of points from the Gaussian, $N(0, \sigma_j)$, estimated by the fitted Gaussian to the set of points $\{v_i\}$.
- **5-** Count N_π points from N generated ones that lie within a distance π from the origin of $T_{\mu_j}(G_{n,d})$
- **6-** Compute multiplication factor $\hat{z}_j = N_\pi / N$ using Equation 6.26
- **7-** Adjust normalization factor for $\hat{f}$, which is the j^{th} class conditional density, using Equation 6.24

**** *Testing* ****

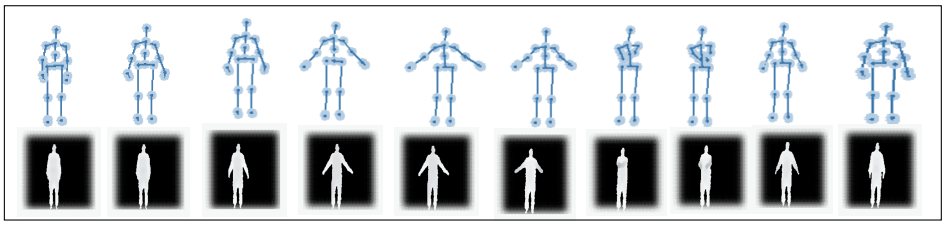
Input: U: unknown action, $\{\hat{z}_j\}_{j=1:k}$, $\{\sigma_j\}_{j=1:k}$

Output: l: class label

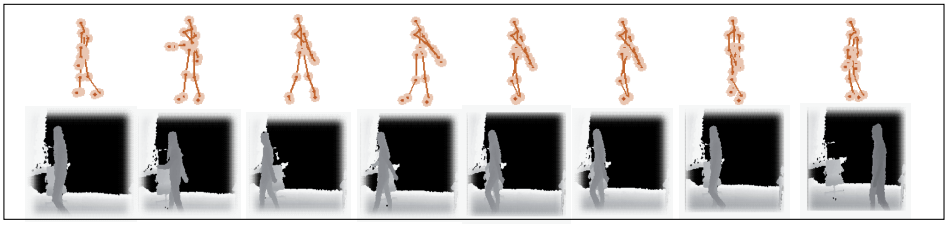
for *j=1 : k* **do**

- **1-** Compute $v_j = log_{\mu_j}(U)$
- **2-** Compute the probability of belonging to class j: $\hat{f}_j = f(v_j)/\hat{z}_i$

3- Predict the class label l of the action U which belong to the class with maximum probability $\hat{f}_j$.



(a) *MSR-action 3D*



(b) *UT-Kinect*

(c) *MSR-Gesture*

Figure 6.9 – *Examples of human actions from datsets used in our experiments: (a) 'hand clap' from MSR-action 3D , (b) 'walk' from UT kinect and (c) Hand frames from MSR-Gesture datset.*

Dataset	Motions	Total number of actions	Experimental protocol
MSR-action 3D [73]	RGB + depth (320*240) + 20 joints: 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 and throw	10 subjects | 20 actions | 3 tries $\Rightarrow$ Total of 520 actions	50% Learning / 50% Testing
UT-kinect [145]	RGB + depth (320*240) + 20 joints: walk, sit down, stand up, pick up, carry, throw, push, pull, wave and clap hands	10 subjects | 10 actions | 2 tries $\Rightarrow$ Total of 200 actions	leave-one-out cross-validation
MSR Gesture 3D [68]	depth sequences: bathroom, blue, finish, green, hungry, milk, past, pig, store, where, j, z	10 subjects | 12 hand gestures | 2-3 tries $\Rightarrow$ Total of 336 gestures	Leave-one-subject-out-cross-validation

Table 6.1 – *Overview of the datasets used in the experiments.*

- Leave One Out Cross Validation (LOOCV): Each time one sequence is taken for prediction and the rest of sequences for training.
- LOOSCV: Each time all sequences concerning one subject are taken for testing and the rest of the sequences for training.
- Cross subject: The dataset is split into 2 subsamples. The first subsample contains sequences of half of subjects, which are used for training. The second subsample contains the remaining sequences of the other half of subjects which are used for testing.
- $N - fold$ cross validation: The whole dataset is randomly partitioned into N equal size subsamples. Of the N subsamples, a single subsample is retained for testing the model, and the remaining $N - 1$ subsamples are used for training. The cross-validation process is then repeated N times (the folds), with each of the N subsamples

used exactly once as the validation data. The N results from the folds are then averaged (or combined) to produce a single estimation.

The rest of the section summarizes our results and provides an analysis of the performances of our proposed approach on these datasets compared to the state-of-the-art approaches for action and gesture recognition.

6.5.2 Action recognition

MSR-Action 3D dataset

MSR-Action 3D [73] is a public dataset of 3D action captured by a depth camera. It consists of a set of temporally segmented actions where subjects are facing the camera and they are advised to use their right arm or leg if an action is performed by a single limb. The background is pre-processed clearing discontinuities and there is no interaction with objects in performed actions. Despite of all of these facilities, it is also a challenging dataset since many activities appear very similar due to small inter-class variation.

For each image of the the action sequences in this dataset, angle normal computation is performed on cropped area around the subject (actor). For each frame normal, angles features computed on cropped area gives 3800 features. To reduce this feature dimension, we learnt a low dimension features using PCA. This dimension reduction allows working with features with lower size and also avoid the manipulation of long vectors, whose computation is costly, containing redundant information. The feature vectors initially contains 3800 features. This feature dimension can be reduced to 500 while kiping 100% of informations. In our experiments, we chose to reduce the feature vector to 200 by kipping 87% of the information.

This final feature vector is computed on each frame allowing to build the time series that characterize the action. Then, we fit an ARMA model and we compute observability matrix and its basis which represents the action as a point on $G_{n,d}$ with $n = 200 \times m$ and $d = m = 16$. The recognition rates obtained by state-of-the-art methods and our approach are

summarized in Table 6.2. To evaluate our approach, we followed the same experimental setup as in Oreifej et al. [87] and Jiang et al. [134], where first five actors are used for training and the rest for testing.

Method	Accuracy %
Grassmannian Sparse Representations [11]	78.48
DMM-HOG [151]	85.52
Random Occupancy patterns [133]	86.50
Histograms of 3D Joints [145]	78.97
Eigen Joints [150]	82.33
HON4D [87]	85.80
HOH4D + D_{disc} [87]	88.89
θ angle	79.02
Φ angle	84.14
$\theta + \Psi + \Phi$ angles	85.19
Ψ angle	**86.21**

Table 6.2 – *Recognition accuracy (in %) for the MSR-Action 3D dataset obtained using our approach and the most known state-of-the-art approaches.*

We firstly choose to test the efficiency of normal angles separately, then we use the 3 angles as features for each image.

We note that our method, by using Ψ angles as features to model the time series, gives the best recognition rate comparing to Θ, Φ or even the three angles together as illustrated in Table 6.2. Using angle Ψ,our approach achieves its highest performance by an accuracy of 86.21%. It is just below the best method from the state-of-the-art proposed by Oreifej et al. [87].

All results in the rest of experiments are obtained using only Ψ angle as feature to represent the time series.

Figure 6.10 gives more details about recognition per class. The first observation is that using our approach about 10 actions are 100% correctly classified. The second observation is on the misclassified actions which are mainly 3 actions: 'Hammer' confused with 'draw X', 'hand catch' confused with 'draw tick' and 'hight serve' with 'hight throw'.

UT-Kinect dataset

Sequences of this dataset are taken using one depth camera (kinect) in indoor settings and their length vary from 5 to 120 frames. We use this dataset because it contains several challenges:

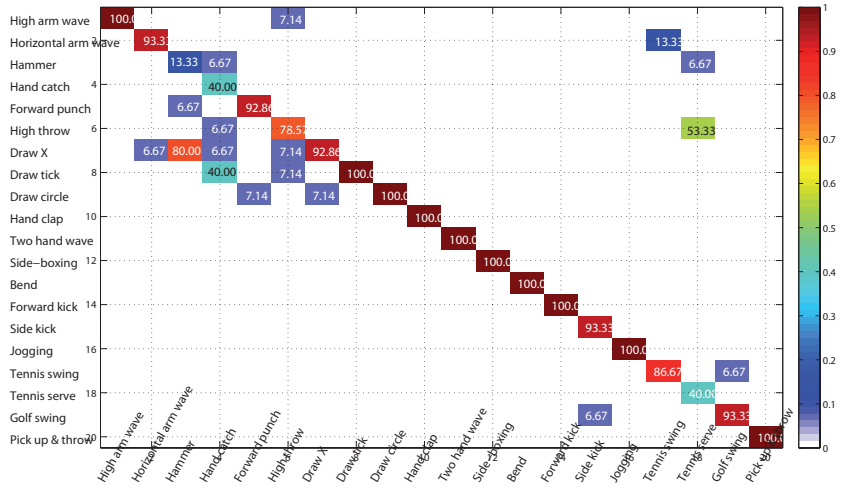


Figure 6.10 – *Confusion matrix for the proposed approach on MSR-Action 3D dataset.*

- View change, where actions are taken from different views: right view, frontal view or back view.
- Significant variation in the realization of the same action: same action is done with one hand or two hands can be used to describe the 'pick up' action.
- Variation in duration of actions: the mean and standard-deviation are respectively for the whole actions 31.1 and 11.61 frames at 30 fps.

From this dataset, we use only depth sequences which resolution is 320×240. We remember that this dataset contain the challenge of human-object interaction (see Table 5.1). To compare our results with state of the art approaches, we follow experiment protocol proposed by Xia et al. [145]. The protocol is leave-one-out cross-validation.

Table 6.3 compared the recognition accuracy produced using our approach and previous systems. As shown, our approach outperforms the tow methods proposed in literature. Indeed, all the actions are correctly classified with a score more than 90%. Some actions in this dataset include human-object interaction (pick-up, carry, throw), which Devanne et al. [28] fail to correctly classify these actions since their approach rely totally on skeleton features. Thus, actions like throw (action with ob-

ject interation) and push (action without object iteration) are classified the same.

However, our approach, since it is based on features computed on depth images, overcomes this problem.

Method	Accuracy %
Histogram of 3D joints [145]	90.92
Space-time Pose Representation [28]	91.5
Our approach	**95.25**

Table 6.3 – *Recognition accuracy (in %) for the UT-kinect dataset using our approach compared to the previous approaches.*

6.5.3 Gesture recognition

MSR Gesture 3D dataset

The MSR Gesture 3D dataset [68] contains 336 depth sequences of 12 hand gesture defined by American sign language (ASL). These gestures are: bathroom, blue, finish, green, hungry, milk, past, pig, store, where, j, z. Following experiment setup used by Kurakin et al. [68], the protocol used for evaluation is Leave-one-subject-out-cross-validation. We note that the resolution of depth maps is different from one sequence to an other. In order to ensure the consistency of the scale, each depth sequence is resized to the same size given images with resolution 50×50. Accuracies obtained with our approach and using state-of-the-art approaches are summarized in table 6.4. The precision given by the proposed approach is better than HON4D method which is presented by Oreifej et al. [87]. This can be explained by the fact that HON4D computes histograms of 4D normals while we are using directly the normal information. Besides, he is segmenting the sequence into fixed number of cells which is very sensitive to change in execution rate. Finally, using subspaces allows being robust to noise and missing data and in this dataset, several frames are either empty or with noise.

6.5.4 Limitations of depth-based approach

Our proposed method based on 3D oriented displacement features extracted from depth maps, shows good performances when actions contain

Method	accuracy
Oreifej et al. [87]	92.45
Jiang et al. [151]	88.50
Yang et al. [133]	89.20
Klaser et al. [63]	85.23
Our approach	**98.21**

Table 6.4 – *The performance on MSR Hand gesture 3D dataset compared to previous approaches.*

object-subject interaction as results obtaines on UT-kinect dataset. Besides, while only depth data is available such as the case in 3D gesture dataset high accuracy is achieved. However, when actors are facing the camera in interaction with the computer as in gaming or sport action scenarios [73], our approach gives performances equal or less than approaches using only skeleton information. In the same time, the computation cost in our approach is expensive because of the use of the entire set of points around each model which give long features extracted on each frame. Although, we are using PCA to reduce feature dimension, the Grassmann manifold dimension remains high ($n = 200 \times m$). In order to reduce computational time and latency effect, and motivated by the robust joints extraction of RGB-D, we propose to compute time-series using 3D joint coordinates and investigate action recognition in the joint space.

6.6 Action recognition using 3D joint coordinates

In this section, we model human motion in the 3D human joint space. Particularly, our intent is to represent skeletal motion in a geometric and efficient way, leading to an accurate action-recognition system. This representation avoids an overly complex design of feature extraction and is able to recognize actions performed by different actors in different contexts.

Our overall approach, using 3D joint coordinates, is sketched in Figure 6.11 and which has the following modules:

First, each action is represented by a dynamical system whose observability matrix is characterized as an element of a Grassmann manifold. The dynamic system of a motion is obtained via an autoregressive and moving average model (ARMA) from its time series.

Second, using the Riemannian geometry of this manifold, we present a solution for solving the classification problem. We studied statistical modelling of inter-classes and intra-class variations in conjunction with appropriate tangent vectors on this manifold.

To formulate our learning algorithm, we propose here a new learning algorithm to work with data points which are geometrically lying to the Grassmann manifold as in the learning process performed using TGW. However, this novel algorithm uses a vector representation formed by concatenating local coordinates in tangent spaces associated with different classes in order to train a classical classifier like linear SVM. The learning algorithm of our approach will be discussed below.

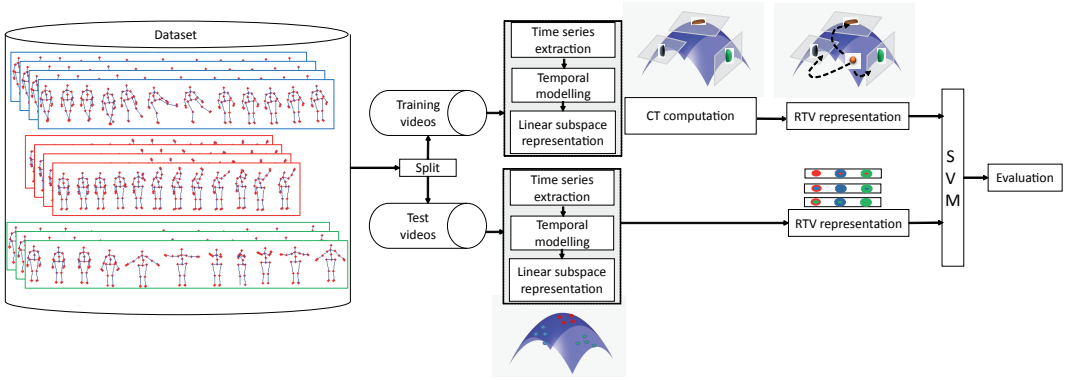


Figure 6.11 – *Overview of the approach. The illustrated pipeline is composed of two main modules: (1) temporal modelling of time series data and manifold representation (2) learning approach using vector representations formed by concatenating local coordinates in tangent spaces associated with different action classes.*

6.6.1 Time series of 3D Joints

The skeletal data provides 3D joint positions of the whole body. The 3D joint coordinates of these skeleton are, however, not invariant to the position and the size of actors.Therefore to be invariant to human location in the scene, the hip joint of each skeleton is placed at the origin of the coordinates system. Besides, to be scale invariant, each skeleton is normalized such that all skeletons parts lengths are equal.

Let p_t^j denote the 3D position of a joint j at a given frame t i.e., $p^j = [x^j, y^j, z^j]_{j=1:J}$, with J is the number of joints. The joint position time-series of joint j is $p_t^j = \{x_t^j, y_t^j, z_t^j\}_{j=1:J}^{t=1:\tau}$, with T the number of frames. A motion

sequence can then be seen as a matrix collecting all time-series from J joints, i.e., $M = [p^1 p^2 \cdots p^\tau]$, $p \in \mathbb{R}^{3*J}$. Figure 6.12 illustrates the matrix construction process.

Each 3D joint sequence is represented as time series matrix of size $p \times \tau$ with τ the number of frames in the sequence and p the number of features per frame. The number of features p depends on the number of estimated joints (60 values for Microsoft SDK skeleton and 45 for PrimeSense NiTE skeleton).

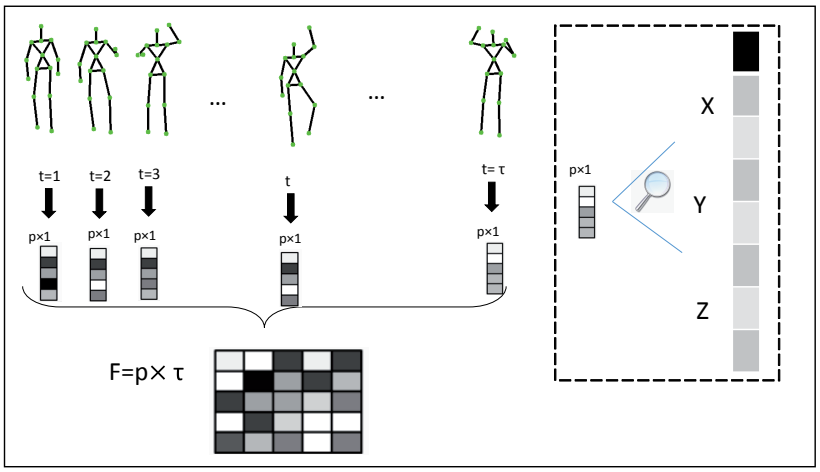


Figure 6.12 – *Time-series matrix construction using 3D joint coordinates.*

6.6.2 Learning on the Grassmann manifold using Representative Tangent Vectors

As mentioned above, using mean computation offered by Karcher mean algorithm, we can perform a classification process by learning a template for each class. However, using multiple class-specific tangent spaces is decidedly more relevant than single one. However, restricting the learning to only the mean and the standard-deviation in each tangent space, as in TGW method, is probably insufficient to classify actions with small inter-class variation.

Besides, limiting the learning process to distances computed locally on the tangent spaces as in [78] is also insufficient.

Our idea is to consider an embedding of data points in higher dimensional representation providing a natural and implicit separation of

directions. In fact, we use the notion of tangent bundle on the manifold to formulate our learning algorithm.

For every point on a manifold there is an associated tangent space. The tangent bundle of a manifold comprises the set of all tangent vectors at all manifold points. These tangent vectors are equipped with mappings from the manifold points to the tangent vectors and from tangent vectors to the manifold points. Differently than [78] who define an intrinsic distance for the tangent bundle, here we propose to use the notion of tangent bundle to define a set of tangent planes which are equipped with a local Euclidean coordinate system.

Thus, we generate Control Tangents (CT) on the manifold, which represent all class-specific tangent vectors. Each CT can be seen as the tangent space of the mean, mean of all points belonging to the same class of actions taken only from training data. Karcher mean algorithm can be employed here for mean computation.

We introduce then the Representative Tangent Vectors (RTV) representation in which proximities are required between each point on the manifold and all CTs. The RTV can be viewed as a parameterization of a point on the manifold which incorporates implicitly release properties in relation to all class clusters, by mapping this point to all CTs using logarithm map. The RTV is then the map of a certain point to all CTs which then constitutes an ordered list of mapped tangent vectors representing the proximity of an action to all existing classes. The final RTV representation is obtained by concatenating local coordinates in tangent spaces associated with different classes and this representation can provide the input of a classifier, like the linear SVM classifier as in our case.

In doing so, the learning model of the classifier is constructed using RTV instead of classifying as function of the local distances (mean and standard-deviation of each class seperately) as in TWG method.

We finally notice that training a linear SVM classifier on our representation of points provided by RTV is more appropriate than the use of SVM with classical Kernel, like RBF, on original points on the manifold.

In experiments, we compare our learning approach RTVSVM to the

classical one denoted as One-Tangent SVM (TSVM), in which the mean is computed on the entire training dataset regardless to class labels. Then, all points on the manifold are projected on this later to provide the inputs of a linear SVM. A graphical illustration of the RTV construction can be shown in Figure 6.13. Algorithm 7 presents each given step of training and testing stages using this proposed learning method by RTVSVM.

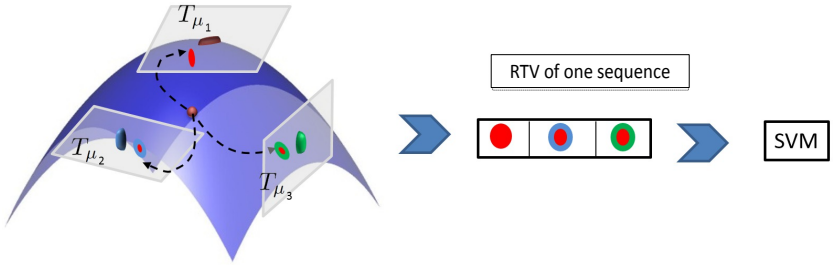


Figure 6.13 – *Conceptual RTV learning methods on the Grassmann manifold. An action presented by a point on the manifold is projected on all CTs, and thus construct a new observation which is the input of the SVM classifier.*

Algorithm 7: SVM classification by Representative Tangent Vectors learning.

**** ***Training*** ****

Input: N training actions as points on $G_{n,d}$, belonging to k classes:

$$\mathcal{L} = \{(U_i, l_j)\}_{i=1:N, j=1:k}$$

Output: $\{CT_j\}_{j=1:k}$ and a training model ϕ

for *j=1 : k* **do**

  1- Compute the Karcher mean μ_j of the j^{th} class using Algorithm 4

  2- Generate CT_j from μ_j as its tangent space

for *i=1 : N* **do**

  3- $V_i = \oslash$

  for *j=1 : k* **do**

    4- Map each point to class-specific CT_j: $v_j = log_{\mu_j}(U_i)$

  5- Construct RTV_i as concatenation of v_j: $V_i \leftarrow [v_1 v_2 ... v_k]$

6- Train a Linear-SVM on $\{RTV_i\}_{i=1:N}$ and obtain a training model ϕ

**** ***Testing*** ****

Input: An unknown action U, $\{CT_j\}_{i=1:k}$ and ϕ

Output: l: class label

1- $V = \oslash$

for *j=1 : k* **do**

  2- Compute the *log map* on the class-specific CT_j: $v_j = log_{\mu_j}(U)$

3- Construct RTV as combination of v_j: $V \leftarrow [v_1 v_2 ... v_k]$

4- Predict label of U as $l =$ Linear-SVM (RTV, ϕ)

6.7 Experimental results in 3D joint space

We extensively experimented our proposed approach on three public 3D action datasets containing various challenges, including MSR-action 3D [73], UT-kinect [145] and UCF-kinect [32], where all details about these datasets are summarized in Table 6.1.

First of all, each action from all datasets is interpreted as an element of the Grassmann manifold $G_{n\times d}$ with $n = m \times J$ where J represents the

number of joints and d is subspace dimension learnt on the training data. We set $m = d$, while m represents the truncation parameter of observation.

In our RTVSVM approach, we train a linear SVM on our RTV representations of points on the Grassmann manifold. We use a multi-class SVM classifier from LibSVM library [20], where the penalty parameter C is tuned using a 5-fold cross-validation on the training dataset.

We evaluate the performance of our approach for action recognition and explore the latency on recognition by evaluating the trade-off between accuracy and latency over varying number of actions. To allow a better evaluation of our approach, we conducted experiments respecting those made in the state-of-the-art approaches.

6.7.1 Evaluation of action recognition

MSR-Action 3D dataset

We test again on MSR-action 3D dataset since it is the benshmark dataset where all appraoches valid their approaches. The first experiment is presented in Figure 6.14, where each class is represented by a template. This latter is computed as the mean of the class sample using Karcher mean, then we compute distances from the test sample to these templates and show distance matrix and its binarization.

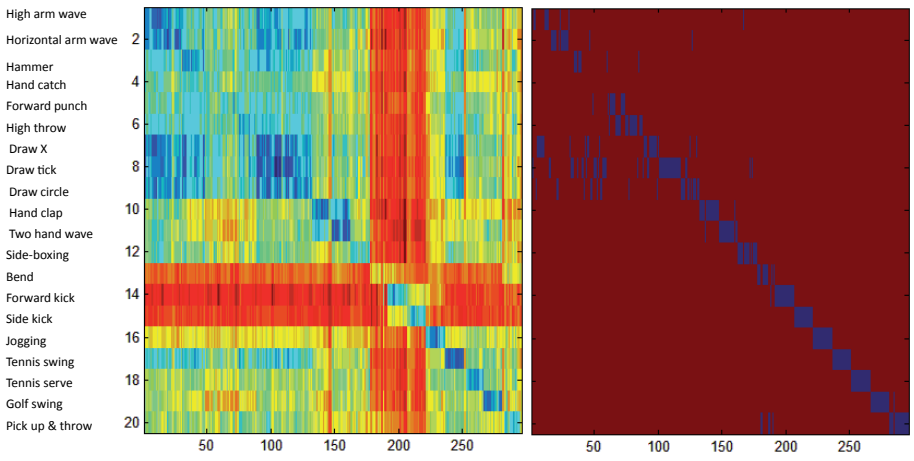


Figure 6.14 – *Results of using the template based method for classification on the MSR 3D action dataset (a) The* 20×260 *similarity matrix between the 260 test sequences on the 20 action models learnt (better viewed in color)(b) The same matrix binarized.*

Several works have already been conducted on this dataset. Table

6.5 shows the accuracy of our approach compared to the state-of-the-art methods. We followed the same experimental setup as in Oreifej et al. [87] and Jiang et al. [134], where first five actors are used for training and the rest for testing.

Method	accuracy %
Histograms of 3D Joints [145]	78.97
Eigen Joints [150]	82.33
DMM-HOG [151]	85.52
HON4D [87]	85.80
Random Occupancy patterns [133]	86.50
Actionlet Ensemble [134]	88.20
HOH4D + D_{disc} [87]	88.89
TSVM on one tangent space	**74.32**
KM	**77.02**
TWG	**84.45**
RTVSVM	**91.21**

Table 6.5 – *Recognition accuracy (in %) for the MSR-Action 3D dataset using our approach compared to the previous approaches.*

Our results obtained in this table correspond to four learning methods: simple Karcher Mean (KM), One tangent SVM (TSVM), Truncated Wrapped Gaussian (TWG) and Representative Tangent Vectors SVM (RTVSVM). Our approach using RTVSVM achieves an accuracy of 91.21%, exceeding the best method from the state-of-the-art proposed by Oreifej et al. [87]. Knowing that our approach is based on only skeletal joint coordinates as motion features, compared to other approaches, such as Oreifej et al. [87] and Wang et al. [133] which use the depth map or depth information around joint locations.

To evaluate the effect of the changing of the subspace dimensions, we conduct several tests on MSR-Action 3D dataset with different dimensions of subspace. Figure 6.15 shows the variation of recognition performances with the change of the subspace dimension. We remark that until dimension 12, the recognition rate generally increase with the increase of the size of the subspaces dimensions. This is expected, since a small dimension causes a lack of information but also a big dimension of the subspace keeps noise and brings confusion between inter-classes. We also compare in this figure, our new introduced learning algorithm RTVSVM to TWG and KM.

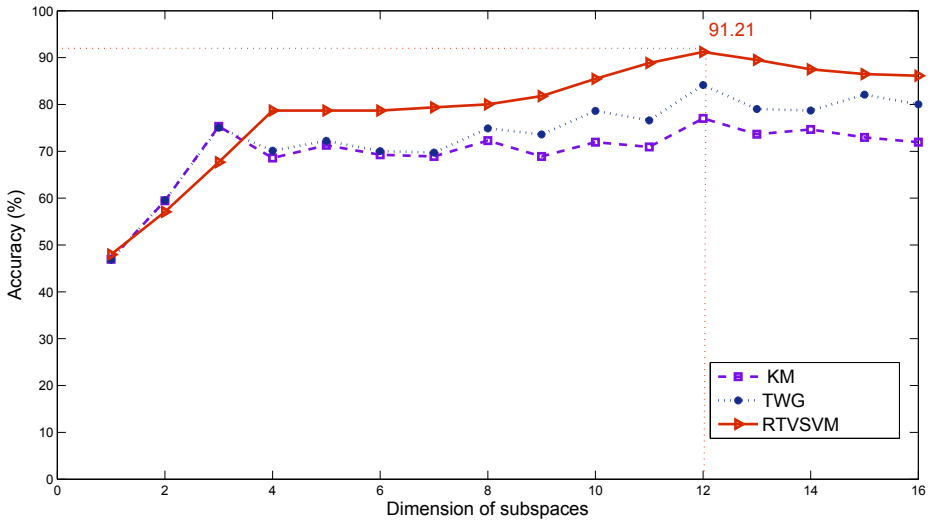


Figure 6.15 – *Recognition rate variation according to different subspace dimensions.*

To better understand the behavior of our approach according to the action type, the confusion matrix is illustrated in Figure 6.16. For most of the actions, about 11 classes of actions, video sequences are 100% correctly classified.

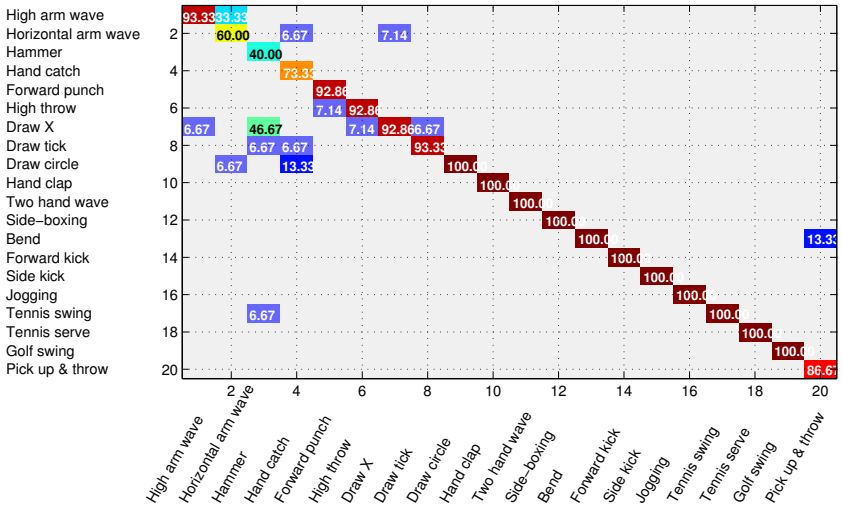


Figure 6.16 – *The confusion matrix for the proposed approach on MSR-Action 3D dataset. It is recommended to view the Figure on the screen.*

The classification error occurs if two actions are very similar, such as 'horizontal arm wave' and 'high arm wave'. Besides, one of most problematic action to classify is 'hammer' action which frequently is confused with 'draw X'. The particularity of these two actions is that they start in the same way but one finish before the other. If we show only the first

part of 'draw X' action and the whole sequence of 'hammer' action we can see that they are very similar. The same for 'hand catch' action which is confused with 'draw circle'. It is important to note that 'hammer' action was completely misclassified with the approach presented by Oreifej et al. [87] which present second better recognition rate after our approach.

While our focus in these experiments is mainly on action recognition and also partially on reducing latency when recognizing actions, some applications need to train with a very reduced number of data. To study the effect of the amount of training dataset, we measured how the accuracy changed as we iteratively reduced the number of actions per class in the training dataset. Table 6.6 shows obtained accuracy results with different size of training dataset.

Actions per class	Training dataset %	Accuracy %
5	37.17	73.36
6	44.23	77.64
7	51.13	83.10
8	58.36	84.79
9	65.54	88.51
10	72.49	89.18
11	79.95	87.83
12	86.24	88.85
13	91.07	90.20
14	95.91	90.54
15	100	91.21

Table 6.6 – *Recognition accuracy, obtained by our approach using RTVSVM on MSR-Action 3D dataset, with different size of training dataset.*

These results show that, in contrast to the approaches that use HMM who require a large number of training dataset, our approach reveals a robustness and efficiency. This robustness due to the fact that the Control Tangents, which play an important role in learning process, can be computed efficiency using small number of action points per class on the manifold.

UT-Kinect dataset

To compare our results on this dataset with state of the art approaches, we follow experiment protocol proposed by Xia et al. [145]. The protocol is leave-one-out cross-validation. In Table 6.7, we show comparison between

the recognition accuracy produced by our approach and the approach presented by Xia et al. [145].

Action	Acc % Xia et al. [145]	Acc % RTVSVM
Walk	96.5	**100**
Stand up	91.5	**100**
Pick up	97.5	**100**
Carry	97.5	**100**
Wave	100	**100**
Throw	59	60
Push	81.5	65
Sit down	91.5	80
Pull	92.5	85
Clap hands	100	95
Overall	90.92	88.5

Table 6.7 – *Recognition accuracy (per action) for the UT-kinect dataset obtained by our approach using RTVSVM compared to Xia et al.*

This table shows the accuracy of the five least-recognized actions in UT-kinect dataset and the five best-recognized actions. Our system performs the worst when the action represents an interaction with an object: 'throw', 'push', 'sit down' and 'pick up'. However, for the best five recognized actions, our approach improves the recognition rate reaching 100%. These actions contain variations in view point and realization of the same action. This means that our approach is view-invariant and it is robust to change in action types thanks to the used learning approach. The overall accuracy of Xia et al. [145] is better than our recognition rate. However on MSR Action3D database, the recognition rate obtained by this approach gives only 78.97%. This can be explained by the fact that this approach requires a large training dataset. Especially for complex actions which affect adversely the HMM classification in case of small samples of training.

6.7.2 Evaluation of Latency

In this experiment, our approach is evaluated in terms of latency, i.e. the ability for a rapid (low-latency) action recognition. The goal here is to automatically determine when enough of a video sequence has been observed to permit a reliable recognition of the occurring action. For many applications, a real challenge is to define a good compromise between "making forced decision" on partial available frames (but potentially unreliable) and "waiting" for the entire video sequence.

To evaluate the performance of our approach in reducing the latency, we experimented on UCF-kinect dataset [32].

UCF-kinect dataset

All details about existing motions, number of actions and experimental protocol are reported in Table Table 6.8. The skeletal joint locations (15 joints) over sequences of this dataset are estimated using Microsoft Kinect sensor and the PrimeSense NiTE. Examples of actions from this dataset are shown in Figure 6.17.The same experimental setup as in Ellis et al. [32] is followed. For a total of 1280 action samples contained in this dataset, a 70% and 30% split is used for respectively training and testing datasets. From the original dataset, new subsequences were created by varying a parameter corresponding to the K first frames. Each new subsequence was created by selecting only the first K frames from the video. For videos shorter than K frames, the entire video is used. We compare the result obtained by our approach to those obtained by Latency Aware Learning (LAL) method proposed by Ellis et al. [32] and other baseline algorithms: Bag-of-Words (BoW) and Linear Chain Conditional Random Field (CRF), also reported by Ellis et al. [32].

Dataset	Motions	Total number of actions	Experimental protocol
UCF-kinect [32]	15 joints: balance, climb up, climb ladder, duck, hop, vault, leap, run, kick, punch, twist left, twist right , step forward, step back, step left, step right	16 subjects | 16 actions | 5 tries $\Rightarrow$ Total of 1280 actions	70% Learning / 30% Testing

Table 6.8 – *UCF dataset properties.*

As shown in Figure 6.18, our approach using RTVSVM clearly achieves improved latency performance compared to all other baseline approaches. Analysis of these curves shows that, accuracy rates for all other approaches are close when using small number of frames (less than 10) or a large number of frames (more than 40). However, the difference increases significantly in the middle range. The table joint to Figure 6.18 shows numerical results at several points along the curves in the figure. Thus,

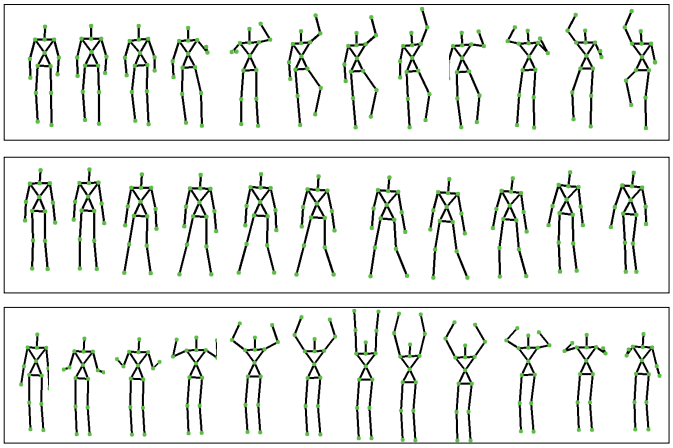
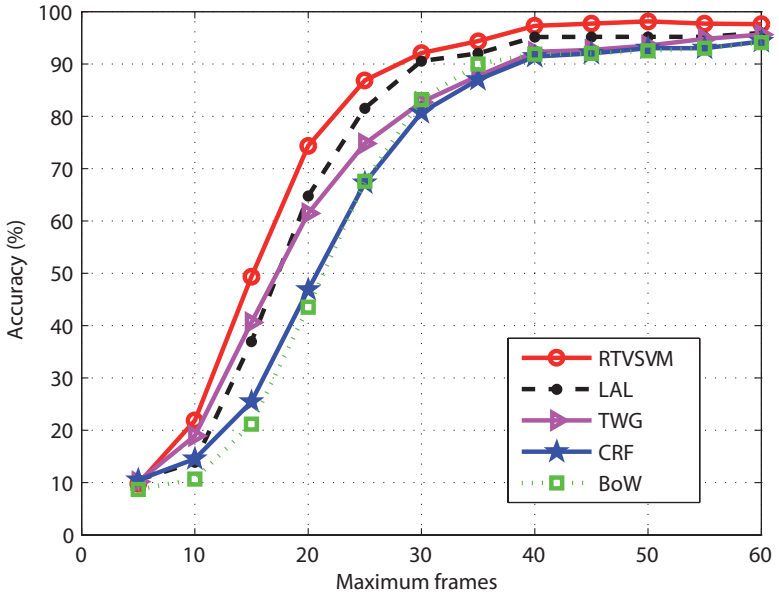


Figure 6.17 – *Examples of human actions from UCF-kinect dataset. From top to bottom actions are: 'climb ladder', 'step left' and 'climb up'.*

given only 20 frames of input, our system achieves 74.37%, while BOW, CRF recognition rate below 50% and LAL achieves 61.45%.



Approach/frames	10	15	20	25	30	40	60
RTVSVM	**21.87**	**49.37**	**74.37**	**86.87**	**92.08**	**97.29**	**97.91**
TWG	18.95	40.62	61.45	74.79	82.7	92.29	95.62
LAL [32]	13.91	36.95	64.77	81.56	90.55	95.16	95.94
CRF [32]	14.53	25.46	46.88	67.27	80.70	91.41	94.06
BOW [32]	10.7	21.17	43.52	67.58	83.20	91.88	94.06

Figure 6.18 – *Accuracies obtained by our approach vs. state-of-the-art approaches over videos truncated at varying maximum lengths. Each point of this curve shows the accuracy achieved by the classifier given only the number of frames shown in the x-axis.*

It is also interesting to notice the improvement of accuracy of 92.08% obtained by RTVSVM compared to 82.7% obtained by TWG, with maximum frame number equal to 30. For a large number of frames, all of the

methods perform globally a good accuracy, with an improvement of the ours (97.91% comparing to 95.94% obtained by LAL proposed in Ellis et al. [32]). These results show that our approach can recognize actions at the desired accuracy with reducing latency.

Finally, the detail of recognition rates, when using the totality of frames in the sequence, are shown through the confusion matrix in Figure 6.19. Unlike what gives LAL, we can observe that the 'twist left', 'twist right' actions are not confused with each others. All classes of actions are classified with a rate more than 93.33% which gives a lot of confidence to our proposed learning approach.

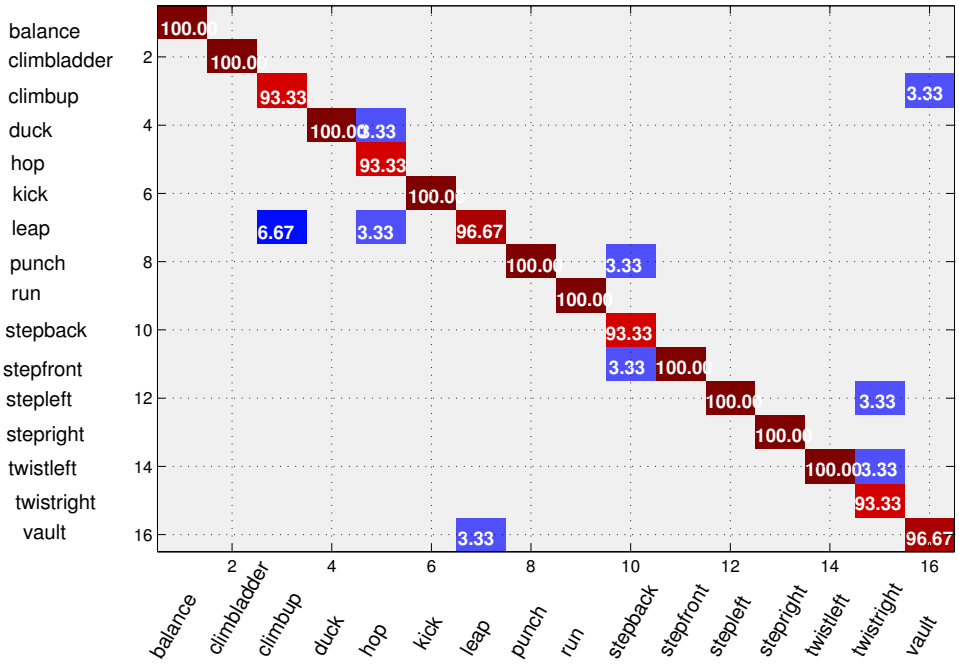


Figure 6.19 – *The confusion matrix for the proposed method on UCF-kinect dataset. Overall accuracy achieved 97.91%. It is recommended to view the figure on the screen.*

6.7.3 Discussion

Representation and learning

Data representation is one of the most important factors in the recognition approach, on which we must take a lot of consideration. Our data representation, like many state-of-the-art manifold techniques [122, 125, 79], consider the geometric space and incorporates the intrinsic nature of the

data. In our framework, which is 3D joint-based, both geometric appearance and dynamic of human body are captured simultaneously. Furthermore, unlike the manifold approaches using silhouettes [125, 6, 123], or directly raw pixels [78, 122], our approach use informative geometric features, which capture useful knowledge to understand the intrinsic motion structure. Thanks to recent release of depth sensor, these features are extracted and tracked along the action sequence, while classical pixel-based manifold approaches relying on a good action localization, or on tedious feature extraction from 2D videos like silhouettes.

In terms of learning method, we generalized a learning algorithm to work with data points which are geometrically lying to a Grassmann manifold. Other approaches are tested in the learning process on the manifold: one tangent space (TSVM) and class-specific tangent spaces (TWG). In the first one, recognition rate is low. In fact, the computation of the mean of all actions from all classes can be inaccurate. Besides, projections on this plane can lead to big deformations. A better solution is to operate on each class by computing its proper tangent space, as in TWG [69] which improve TSVM results (see Table 6.5). In our approach (RTVSVM), both Control Tangent and statistics on the manifold are used. The purpose was to formulate our learning algorithm using a discriminative parametrization which incorporate class separation properties. The particularity of our learning model is the incorporation of proximities relative to all Control Tangent representing class clusters, instead of classifying using a function of local distances. The results in Table 6.5 demonstrate that the proposed algorithm is more efficient in action recognition scenario when inter-variation classes is present as a challenge.

Furthermore, the analysis of the impact of reducing the number of actions in the training set on the accuracy of the classifier shows its robustness. Even with a small number of actions in the training data recognition rates remain good as demonstrated in Table 6.6. However it is a limitation especially for approaches using an HMM learning because they require a large number of training dataset. Such as Xia et al. approach [145], which

gives only 78.97% of recognition rate while performing cross subject test on MSR dataset.

We also analysed the dispersion of actions in each dataset while representing actions by Grassmann representation and using the appropriate metric defined on. In Figure 6.20, we display the resulting multidimensional scaling (MDS) for the three datasets used in this experimental section. The MDS plot gives an impression on where the actions are located in action space. It allows to display the information contained in a distance matrix. Here, the distance matrix is computed using distance defined in equation 6.12 between each two actions presented as points on Grassmann manifold. We note that our modelisation via Grassmann manifold allows a good separation of classes especially for UCF and UT kinect datasets. In MSR-action dataset some overlapoing between classes can be seen. These classes are mainly 'Hammer' and 'Draw X' actions.

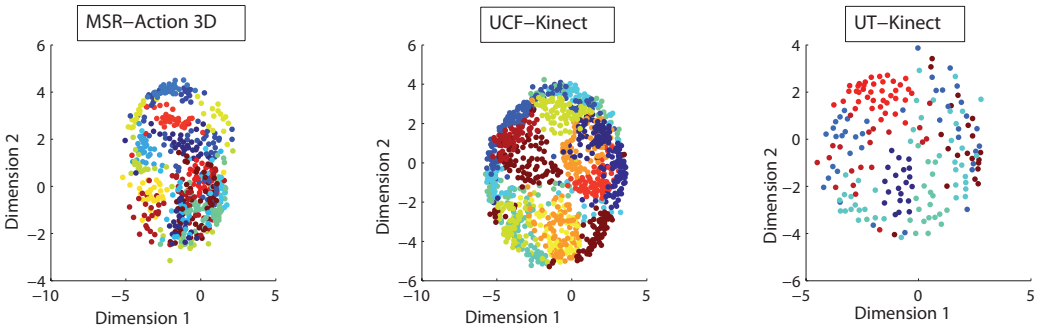


Figure 6.20 – *MDS plots for actions from three datasets using our proposed geometric framework. In this plot, each point is an action and each color represents a class.*

Latency and Time computation

The evaluations in terms of latency have clearly revealed the efficiency of our approach for a rapid recognition. It is possible to recognize actions up to 95% using only 40 frames which is a good performance comparing to state-of-the-art approaches presented in [32]. Thus, our approach can be used for interactive systems. Particularly, in entertainment applications to resolve the problem of lag and improve some motion-based games.

Since the proposed approach is based on only skeletal joint coordinates, it is simple to calculate and it needs only a small computation time. In fact, with our current implementation written in C++, the whole recog-

nition time takes 0.26 sec to recognize a sequence of 60 frames. The joint extraction and normalisation take 0.0001 sec, the Grassmann and the RTV representation take 0.0108 sec and the prediction on SVM takes 0.251 sec. These computation time are reported on UCF-Kinect dataset, with Grassmann manifold dimension $n = 540$ and $d = 12$. We also reported the computation time needed to recognize actions while incorporating latency on UCF-Kinect dataset. Figure 6.21 illustrates inline time recognition with time progression. After only 40 frames, the recognition is given at the 0.94 sec within 97.29% of correctness rate. After 60 frames, in 1.3 sec the algorithm recognize correctly the action with 97.91%. All the computation time experiments are launched on a computer having Intel Core i5-3350P (3,1 GHz) CPU, 4GB RAM and a PrimeSense camera for skeleton extraction giving about 60 skeleton/sec.

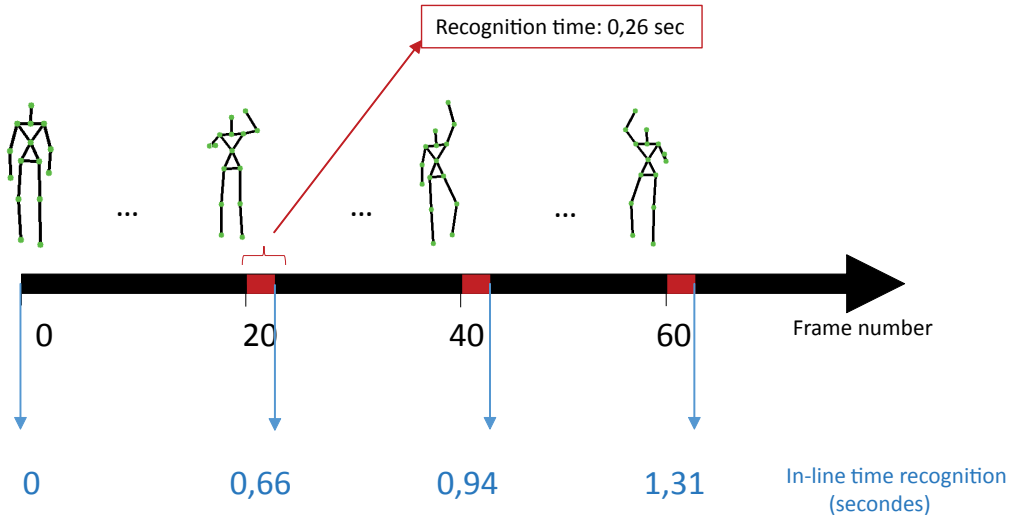


Figure 6.21 – *Time computation details.*

Limitations of 3D joints-based approach

Our proposed approach is a 3D joint-based framework designed for human action recognition from skeletal joint sequences. In the case of presence of object interaction in human actions, our approach do not provides any relevant information about objects and thus, action with and without objects are confused. This limitation can be leveraged the use of additional features, which can be extracted from depth or color images associated.

The proposed approach works with atomic actions which are not complex and continuous. To be operational in all action recognition scenarios, specially in real-time scenarios and while actions are more complex, the present framework should be increased by modules for: (1) Identification of the beginning and the end of each atomic action, (2) Identification of each skeleton for sequences containing more than one person in the scene.

6.8 Depth Vs 3D joint features

After testing our framework using both depth and skeleton data and using appropriate learning algorithms we summarize results of our appraoch on each dataset in Table 6.9.

Dataset	RTVSVM-3D joint (%)	TWG-depth (%)
MSR-Action	91.21	86.21
UT-Kinect	88.5	95.25
UCF-Kinect	97.91	-
MSR-Gesture	-	98.21

Table 6.9 – *Comparison between depth and skeleton approaches.*

In databases where only depth images can be acquired, our approach using descriptors computed on depth images can be used. As is the case for the MSR-gestures dataset. However, when it is possible to extract the skeleton, 3D joint descriptor can be used as an input in our proposed geometrical framwork. Our approach using the skeleton could also be used for databases where only skeleton exists as the datasets of Mocap. The prediction is fast and accurate using skeleton, since the representation is simple and only 20 joints are used. However, using the depth images the size of each descriptor is much larger and depends on the size of the image thus the computation time increase. According to the needs, our framework could be used to solve the problem of gesture and action recognition either with depth or skeleton descriptors. Merging the two approaches, by considering the disjoint probability, provides more accuracy giving a rate of 93 % while testing cross subject protocol on MSR-3D Action dataset. This is explained by the fact that the methods are complementary.

6.9 Conclusion

In this chapter, we introduced Grassmann manifold mathematical definition and tools which are then used in a geometric framework for sequence representation and action learning.

The proposed framework allows modelling and recognizing human motion in both 3D skeletal joint space and depth images. In this framework, sequence features are modeled temporally as subspaces lying to a Grassman manifold. A new learning algorithm on this manifold is introduced to improve action recognition performances. Our approach in terms of accuracy/latency reveals an important ability for a low-latency action recognition system. Obtained results show that with minimum number of frames, it provides the highest recognition rate.

Conclusion

7

Sommaire

7.1 Summary

In this thesis, we proposed different frameworks which are proving the usefulness and effectiveness of statistical analysis on manifolds to specific applications in 3D video analysis. Typical video analysis is usually composed of a feature extraction stage and then a model building stage. We highlight different applications using manifold analysis, one for each of the two stages in a typical video analysis framework.

First, we have proposed a unified framework able to represent human body shape with a pose descriptor, as well as a sequence of frames with a specific representation. This framework relies on an Extremal Human Curve descriptor (EHC), based on extremal features and geodesics between each pair of them. This descriptor has the advantage of being a skeletal representation, which is trackable over time. It describes the surface deformation which is composed of a collection of local 3D open curves. The representation of these curves and the comparison between them are performed in the Riemannian shape space of open curves. By this way, we have chosen to represent the human pose represented by its mesh, regardless to its rotation, translation and scale. Convoluted with a time filter to incorporate the motion, it becomes a temporal descriptor for pose retrieval in 3D video sequences. The degree of motion using feature vector, extracted from this descriptor, is used for splitting continuous sequences into elementary motion segments called clips. Each clip describing an atomic movement is characterized by EHC representation. The open curves in 3D space, which are the elements of EHC representation, are viewed as a point in the shape space of open curves and hence each clip is represented by a trajectory on this space. Dynamic time warping is used to align different trajectories and to give a similarity score between each two clips. The quality of our descriptor regarding the performance of shape similarity in 3D video is analyzed and verified by comparison with other related recent techniques. This comparison shows that our approach gives very competitive results compared with state of the art approaches.

Second, we addressed the problem of human gesture and action recognition in depth image sequences. We introduced a novel framework, in

which sequence of local oriented displacement features are modeled temporally as subspaces lying on the Grassmann manifold. We then formulated our learning algorithm using the notion of the class-specific tangent space on this manifold. Thanks to statistical tools applied on this Riemaniann manifold, the classification process is performed as a function of probability density by Truncated Wrapped Gaussian on specific-class tangent spaces. The evaluation of our approach in terms of human action recognition even in presence of object interaction and hand gesture recognition reveals a remarkable efficiency exceeding existing approaches on datasets containing these challenges.

Third, the same manifold modelling via Grassmann is tested in an effective framework in the 3D skeletal joint space. In this framework, temporal modelling and geometric representation are included. Besides, a new learning algorithm on this manifold is introduced. It embeds each action, presented as a point on this manifold, in higher dimensional representation providing natural separation directions. Experimental results of our proposed approach are promising and show high accuracies either equal or even outperform existing approaches. The evaluation of our approach in terms of accuracy/latency reveals an important ability for a low-latency action recognition system. Obtained results show that with minimum number of frames, it provides the highest recognition rate comparing to the state of the art approaches.

Each of these frameworks suffer from limitations and can be improved and extended for further efficiency. Thus in the next section, we present some open issues that could be addressed in future.

7.2 Limitations, Future Work, and Open Issues

This section briefly describes a few directions that could extend our work.

Robustness to topology changes EHC descriptor depends on the accuracy of extremities (head and limbs) extraction and on the definition of the path connecting end-points. However, this process is based on geodesic distances which are sensitive to significant topology changes. Thus, other

strategies can be investigated for the extremity extraction step and shortest path detection on the mesh by using diffusion or commute time distances as presented by Elkhoury et al. [31] and Sun et al. [109].

Depth data-driven fusion In this thesis, we introduced two examples of features based either on 3D skeleton stream or on local displacement features. However, we used each of them independently. We propose in future works to adapt our framework to work with both features in the same time, passing by a feature fusion step [159].

Recognizing group actions and complex activities In our framework for action recognition using depth images, we considered only sequences with a single subject in the scene doing a single action. However, recognizing subjects interaction or recognizing actions of multiple subjects in scene is a challenging problem. For interactions which can be described as one action, such as 'hand shake' or 'hag' it is possible to learn a global model. However, for more complex interaction scenarios we need to understand relation between individual subject actions. Besides, In our system, only simple actions are taken into account. In future works, we would investigate more in recognizing complex actions as human activities in natural environments. For this purpose, sequence segmentation into simple actions could be taken in consideration in order to simplify the recognition of long sequences that contain higher degree of semantic.

Using Time-Invariant Models In the future, the problem of modeling and recognizing complex activities which exhibit time-varying dynamics can be addressed within the same proposed geometric framework.

Complex human activities are characterized by non-linear dynamics that make learning, inference and recognition hard. Linear Dynamic Systems (LDS) are not adapted to model complex activities, thus to extend our approach to this task, time-varying LDS model can be considered. Particularly, this model can be described as a trajectory on the space of LDS models. Thus, under local stationary assumptions, we could perform

learning and classification problems as trajectory modeling on the Grassmann manifold and exploit the geometry of this manifold for this task.

Bibliography

[1] In *CAESAR. http://store.sae.org/caesar/*.

[2] 3d studio max, http://www.3dmax.com/ [october 2002].

[3] Cyberware: http://www.cyberware.com [october 2002].

[4] http://4drepository.inrialpes.fr, september 2010.

[5] Vitus: http://www.vitus.de/english/ [october 2002].

[6] M.F. Abdelkader, W. Abd-Almageed, A. Srivastava, and R. Chellappa. Silhouette-based gesture and action recognition via modeling trajectories on riemannian shape manifolds. In *Computer Vision and Image Understanding*, volume 115, pages 439 – 455, 2011.

[7] J.K. Aggarwal and M.S. Ryoo. Human activity analysis: A review. In *ACM Computing Surveys*, volume 43, pages 1–43, 2011.

[8] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: reconstruction and parameterization from range scans. In *ACM SIGGRAPH*, volume 22, pages 587–594, New York, NY, USA, 2003.

[9] M. Ankerst, G. Kastenmüller, H-P. Kriegel, and T. Seidl. 3d shape histograms for similarity search and classification in spatial databases. In *Proceedings of the 6th International Symposium on Advances in Spatial Databases*, pages 207–226, 1999.

[10] S. Azary and A. Savakis. A spatiotemporal descriptor based on radial distances and 3D joint tracking for action classification. In *IEEE International Conference on Image Processing*, pages 769–772, 2012.

[11] S. Azary and A. Savakis. Grassmannian sparse representations and motion depth surfaces for 3D action recognition. In *CVPR Workshop on Human Activity Understanding from 3D Data*, pages 492–499, 2013.

[12] I. Baran and J. Popovic. Automatic rigging and animation of 3D characters. In *ACM Trans. Graph.*, volume 26, New York, NY, USA, 2007.

[13] M. Barnachon, S. Bouakaz, B. Boufama, and E. Guillou. Ongoing human action recognition with motion capture. In *Pattern Recognition*, volume 47, pages 238 – 247, 2014.

[14] R. Bhattacharya and V. Patrangenaru. Nonparametic estimation of location and dispersion on riemannian manifolds. In *Journal of Statistical Planning and Inference*, volume 108, pages 23 – 35, 2002.

[15] W.M. Boothby. An introduction to differentiable manifolds and riemannian geometry. In *Academic Press*, 1975.

[16] Q. Dai C. Wu, Yebin Liu and B. Wilburn. Fusing multiview and photometric stereo for 3d reconstruction under uncalibrated illumination. In *IEEE Transactions on Visualization and Computer Graphics*, volume 17, pages 1082–1095, 2011.

[17] S. Calderara, A. Prati, and R. Cucchiara. Markerless body part tracking for action recognition. *IJMIS*, 1:76–89, 2010.

[18] J. Cao, A. Tagliasacchi, M. Olson, H. Zhang, and Z. Su. Point cloud skeletons via laplacian-based contraction. In *Shape Modeling International Conference*, pages 187–197, June 2010.

[19] A.A. Chaaraoui, J. R. Padilla-López, P. Climent-Pérez, and F. Flórez-Revuelta. Evolutionary joint selection to improve human action recognition with RGB-D devices. In *Expert Systems with Applications*, volume 41, pages 786–794, 2014.

[20] C-C. Chang and C-J. Lin. LIBSVM: A library for support vector machines. volume 2, pages 1–27, 2011.

[21] L. Chen, H. Wei, and J. Ferryman. A survey of human motion analysis using depth imagery. In *Pattern Recognition Letters*, volume 34, pages 1995 – 2006, 2013.

[22] K.M. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette across time part I: Theory and algorithms. In *International Journal of Computer Vision*, volume 62, pages 221 – 247, 2005.

[23] I. Cohen and H. Li. Inference of human postures by classification of 3d human body shape. In *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, pages 74–81, Washington, USA, 2003.

[24] Y. Cui, Will C., T. Noll, and D. Stricker. Kinectavatar: Fully automatic body capture using a single kinect. In *Asian Conference on Computer Vision Workshops*, volume 7729, pages 133–147, 2013.

[25] L. Vaina D. Marr. Representation and recognition of the movements of shapes. In *Proe. R. Soc. Lond. B*, pages 501–524, 1982.

[26] N. D'Apuzzo. Modeling human faces with multi-image photogrammetry. In *Proceedings of SPIE*, volume 4661, pages 191–197, 2002.

[27] E. De Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. In *ACM SIGGRAPH*, volume 27, pages 1–10, 2008.

[28] M. Devanne, H. Wannous, . Berretti, P. Pala, M. Daoudi, and A. Del Bimbo. Space-time pose representation for 3D human action recognition. In *Workshop on Social Behaviour Analysis ICIAP*, volume 8158, pages 456–464, 2013.

[29] G. Doretto, A. Chiuso, Y.N. Wu, and S. Soatto. Dynamic textures. In *International Journal of Computer Vision*, volume 51, pages 91–109, 2003.

[30] H. Drira, B. Ben Amor, A. Srivastava, M. Daoudi, and R. Slama. 3D face recognition under expressions, occlusions, and pose variations.

In *IEEE transactions on Pattern Analysis and Machine Intelligence*, volume 35, pages 2270–2283, 2013.

[31] R. El Khoury, J-P. Vandeborre, and M. Daoudi. Indexed heat curves for 3d-model retrieval. In *International Conference on Pattern Recognition*, pages 1964–1967, 2012.

[32] C. Ellis, S.Z. Masood, M.F. Tappen, J. Laviola, and R. Sukthankar. Exploring the trade-off between accuracy and observational latency in action recognition. In *International Journal of Computer Vision*, volume 101, pages 420–436, 2013.

[33] S. R. Fanello, I. Gori, G. Metta, and F. Odone. Keep it simple and sparse: Real-time action recognition. In *Journal of Machine Learning Research*, volume 14, pages 2617–2640, 2013.

[34] S. Fothergill, H. Mentis, P. Kohli, and S. Nowozin. Instructing people for training gestural interactive systems. In *CHI Conference on Human Factors in Computing Systems*, pages 1737–1746, 2012.

[35] R. Furukawa, R. Sagawa, A. Delaunoy, and H. Kawasaki. Multiview projectors cameras system for 3D reconstruction of dynamic scenes. In *IEEE International Conference on Computer Vision Workshops*, pages 1602–1609, 2011.

[36] K.A. Gallivan, A. Srivastava, L. Xiuwen, and P. Van Dooren. Efficient algorithms for inferences on grassmann manifolds. In *IEEE Workshop on Statistical Signal Processing*, pages 315–318, Sept 2003.

[37] T. Giorgino. Computing and visualizing dynamic time warping alignments in R: The DTW package. In *Journal of Statistical Softwar*, volume 31, page 1–24, 2009.

[38] N. Gkalelis, H. Kim, A. Hilton, N. Nikolaidis, and I. Pitas. The i3Dpost multi-view and 3D human action/interaction database. In *Proceedings of the Conference for Visual Media Production*, pages 159–168, 2009.

[39] D. Gong and G. Medioni. Dynamic manifold warping for view invariant action recognition. In *IEEE International Conference on Computer Vision*, pages 571–578, Barcelona, Spain, 2011.

[40] D. Gong, G. Medioni, and X. Zhao. Structured time series analysis for human action segmentation and recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 36, pages 1414–1427, 2014.

[41] K. Guo, P. Ishwar, and J. Konrad. Action recognition from video using feature covariance matrices. In *IEEE Transactions on Image Processing*, volume 22, pages 2479–2494, June 2013.

[42] I. Guyon, V. Athitsos, P. Jangyodsuk, B. Hamner, and H.J. Escalante. Chalearn gesture challenge: Design and first results. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–6, June 2012.

[43] M. T. Harandi, C. Sanderson, S. Shirazi, and B. C. Lovell. Kernel analysis on grassmann manifolds for action recognition. In *Pattern Recognition Letters*, volume 34, pages 1906 – 1915, 2013.

[44] J.M. Hasenfratz, M. Lapierre, J-D. Gascuel, and E. Boyer. Real-time capture, reconstruction and insertion into virtual world of human actors. In *Vision, Video and Graphics*, pages 49–56, 2003.

[45] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H-P. Seidel. A statistical model of human pose and body shape. In *Computer Graphics Forum*, volume 2, pages 337–346, 2009.

[46] G-F. He, S-K. Kang, W-C. Song, and S-T. Jung. Real-time gesture recognition using 3D depth camera. In *International Conference on Software Engineering and Service Science (ICSESS)*, pages 187–190, 2011.

[47] M.B. Holte, T.B. Moeslund, N. Nikolaidis, and I. Pitas. 3D human action recognition for multi-view camera systems. In *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 342–349, May 2011.

[48] M.B. Holte, C. Tran, M.M. Trivedi, and T.B. Moeslund. Human action recognition using multiple views: a comparative perspective on recent developments. In *Proceedings of the joint ACM workshop on Human gesture and behavior understanding*, pages 47–52, NY, USA, 2011.

[49] C-H. Huang, E. Boyer, and S. Ilic. Robust human body shape and pose tracking. In *International Conference on 3D Vision*, pages 287–294, 2013.

[50] P. Huang and A. Hilton. Shape-colour histograms for matching 3D video sequences. In *IEEE International Conference on Computer Vision Workshops*, pages 1510 –1517, 2009.

[51] P. Huang, A. Hilton, and J. Starck. Automatic 3D video summarization: Key frame extraction from self-similarity. In *International Symposium on 3D Data Processing, Visualization and Transmission*, pages 1 – 8, 2008.

[52] P. Huang, A. Hilton, and J. Starck. Shape similarity for 3D video sequences of people. In *International Journal of Computer Vision*, volume 89, pages 362–381, 2010.

[53] P. Huang, T. Tung, S. Nobuhara, H. Hilton, and T. Matsuyama. Comparison of skeleton and non-skeleton shape descriptors for 3D video. In *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT'10)*, Pairs, France, 2010.

[54] L. Jaemin, H. Takimoto, H. Yamauchi, A. Kanazawa, and Y. Mitsukura. A robust gesture recognition based on depth data. In *Korea-Japan Joint Workshop on Frontiers of Computer Vision*, pages 127–132, Jan 2013.

[55] A. Jalal, M.Z. Uddin, and T. S Kim. Depth video-based human activity recognition system using translation and scaling invariant features for life logging at smart home. In *IEEE Transactions on Consumer Electronics*, volume 58, pages 863–871, 2012.

[56] A.E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 21, pages 433–449, 1999.

[57] S.H. Joshi, E. Klassen, A. Srivastava, and I. Jermyn. A novel representation for riemannian analysis of elastic curves in Rn. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1 –7, 2007.

[58] K. Kahol, P. Tripathi, and S. Panchanathan. Automated gesture segmentation from dance sequences. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 883 – 888, 2004.

[59] T. Kanade, P. Rander, and P. J. Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. In *IEEE MultiMedia*, volume 4, pages 34–47, 1997.

[60] S. Katz, G. Leifman, and A. Tal. Mesh segmentation using feature point and core extraction. In *The Visual Computer*, volume 21, pages 649–658, 2005.

[61] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *ACM SIGGRAPH Symposium on Geometry Processing*, pages 156–164, 2003.

[62] J. Kilner, J. Y Guillemaut, and A. Hilton. 3D action matching with key-pose detection. In *International Conference on Computer Vision Workshops*, pages 1–8, Sept 2009.

[63] A. Klaser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3D-gradients. In *British Machine Vision Conference*, pages 1–10, 2008.

[64] E. Klassen, A. Srivastava, W. Mio, and S. Joshi. Analysis of planar shapes using geodesic paths on shape spaces. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 26, pages 372–383, 2004.

[65] H.S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from RGB-D videos. In *International Journal of Robotics Research*, volume 32, pages 951–970, 2013.

[66] I. Koprinska and S. Carrato. Temporal video segmentation: A survey. In *Signal Processing: Image Communication*, pages 477–500, 2001.

[67] M. Körtgen, G-J. Park, M. Novotni, and R. Klein. 3D shape matching with 3D shape contexts. In *The 7th Central European Seminar on Computer Graphics*, 2003.

[68] 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)*, pages 1975–1979, 2012.

[69] S. Kurtek, A. Srivastava, E. Klassen, and Z. Ding. Statistical modeling of curves using shapes and related features. In *Journal of the American Statistical Association*, volume 107, pages 1152–1165, 2012.

[70] W. Lao, J. Han, and P.H.N. de With. Automatic video-based human motion analyzer for consumer surveillance system. In *IEEE Transactions on Consumer Electronics*, volume 55, pages 591–598, 2009.

[71] F. Lazarus and A. Verroust. Level set diagrams of polyhedral objects. In *ACM symposium on Solid modeling and applications*, pages 130–140, New York, NY, USA, 1999.

[72] B. Li, M. Ayazoglu, T. Mao, O.I. Camps, and M. Sznaier. Activity recognition using dynamic subspace angles. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3193–3200, June 2011.

[73] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3D points. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 9–14, 2010.

[74] B. Liang and L. Zheng. Gesture recognition from one example using depth images. In *Lecture Notes on Software Engineering*, volume 1, pages 339–343, 2013.

[75] W. Lin, M-T. Sun, R. Poovandran, and Z. Zhang. Human activity recognition for video surveillance. In *IEEE International Symposium on Circuits and Systems*, pages 2737–2740, 2008.

[76] Y. Lipman and T. Funkhouser. Mobius voting for surface correspondence. In *ACM Transactions on Graphics*, volume 28, pages 1–12, 2009.

[77] L. Liu and L. Shao. Learning discriminative representations from RGB-D video data. In *International Joint Conference on Artificial Intelligence*, 2013.

[78] Y. M. Lui. Tangent bundles on special manifolds for action recognition. In *IEEE Transactions on Circuits and Systems for Video Technology*, volume 22, pages 930–942, 2012.

[79] Y. M. Lui and J. R. Beveridge. Tangent bundle for human action recognition. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 97–102, 2011.

[80] S. Mahmoudia and M. Daoudib. A probabilistic approach for 3d shape retrieval by characteristic views. In *Pattern Recognition Letters*, volume 28, pages 1705 – 1718, 2007.

[81] J. Marnik. The polish finger alphabet hand postures recognition using elastic graph matching. In *Computer Recognition Systems*, volume 45, 2007.

[82] T. Matsuyama, S. Nobuhara, T. Takai, and T. Tung. Multi-camera systems for 3d video production. In *3D Video and Its Applications*, pages 17–44, 2012.

[83] P. W. Michor and D. Mumford. Riemannian geometries on spaces of plane curves. In *Journal of the European Mathematical Society*, volume 8, pages 1–48, 2006.

[84] I. Mikic, M. Trivedi, E. Hunter, and P. Cosman. Human body model acquisition and tracking using voxel data. In *International Journal of Computer Vision*, volume 53, pages 199–223, 2003.

[85] M. Mortara and G. Patane. Affine-invariant skeleton of 3d shapes. In *Proceedings of the Shape Modeling International*, pages 245 – 252, Washington, DC, USA, 2002.

[86] Bingbing Ni, Gang Wang, and P. Moulin. RGBD-HuDaAct: A color-depth video database for human daily activity recognition. In *International Conference on Computer Vision Workshops*, pages 1147–1153, 2011.

[87] O. Oreifej and Z. Liu. HON4D: Histogram of oriented 4d normals for activity recognition from depth sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–723, Washington, DC, USA, 2013.

[88] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. In *ACM Transactions on Graphics*, volume 21, pages 807–832, 2002.

[89] M. Ovsjanikov, Q. Mérigot, F. Mémoli, and L. J. Guibas. One point isometric matching with the heat kernel. In *Computer Graphics Forum*, volume 29, pages 1555–1564, 2010.

[90] S. Pehlivan and P. Duygulu. A new pose-based representation for recognizing actions from multiple cameras. In *Computer Vision and Image Understanding*, volume 115, pages 140 – 151, 2011.

[91] D. Pickup, X. Sun, P.L. Rosin, R.R. Martin, Z. Cheng, Z. Lian, M. Aono, A. Ben Hamza, A. Bronstein, M. Bronstein, S. Bu, U. Castellani, S. Cheng, V. Garro, A. Giachetti, A. Godil, J. Han, H. Johan, L. Lai, B. Li, C. Li, H. Li, R. Litman, X. Liu, Z. Liu, Y. Lu, A. Tatsuma, and J. Ye. SHREC'14 track: Shape retrieval of non-rigid 3d human models. In *Proceedings of the 7th Eurographics workshop on 3D Object Retrieval*, 2014.

[92] R. Poppe. A survey on vision-based human action recognition. In *Image and Vision Computing*, volume 28, pages 976–990, 2010.

[93] M. Reyes, G. Dominguez, and S. Escalera. Feature weighting in dynamic time warping for gesture recognition in depth data. In *IEEE*

International Conference on Computer Vision Workshops, pages 1182–1188, 2011.

[94] Y. Rui and P. Anandan. Segmenting visual actions based on spatio-temporal motion patterns. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 111 –118, 2000.

[95] X. Zhang S. Althloothi, M. H. Mahoor and R. M. Voyles. Human activity recognition using multi-features and multiple kernel learning. In *Pattern Recognition*, volume 47, pages 1800 – 1812, 2014.

[96] C. Sanderson A. Alavi S. Shirazi, M.T. Harandi and B.C. Lovell. Clustering on grassmann manifolds via kernel embedding with application to action analysis. In *International Conference on Image Processing*, pages 781–784, 2012.

[97] D. Sánchez, M. Tentori, and J. Favela. Activity recognition for the smart hospital. volume 23, pages 50–57, 2008.

[98] 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 *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 479–485, June 2013.

[99] S. Sempena, N.U. Maulidevi, and P.R. Aryan. Human action recognition using Dynamic Time Warping. In *International Conference on Electrical Engineering and Informatics*, pages 1–5, July 2011.

[100] A. Sharf, T. Lewiner, A. Shamir, and L. Kobbelt. On-the-fly curve-skeleton computation for 3D shapes. In *Eurographics 2007 (Computer Graphics Forum)*, volume 26, 2007.

[101] A. Shimada, K. Kondo, D. Deguchi, G. Morin, and H. Stern. Kitchen scene context based gesture recognition: A contest in ICPR2012. In *Advances in Depth Image Analysis and Applications*, volume 7854, pages 168–185, 2013.

[102] T. Shiratori, A. Nakazawa, and K. Ikeuchi. Rhythmic motion analysis using motion capture and musical information. In *Proceedings of*

IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, pages 89 – 94, 2003.

[103] 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 *Machine Learning for Computer Vision*, volume 411, pages 119–135, 2013.

[104] R. Slama, H. Wannous, and M. Daoudi. Extremal human curves: a new human body shape and pose descriptor. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 1–6, Shanghai, China, 2013.

[105] A. Srivastava and E. Klassen. Monte carlo extrinsic estimators of manifold-valued parameters. In *IEEE Transactions on Signal Processing,*, volume 50, pages 299–308, Feb 2002.

[106] A. Srivastava, E. Klassen, S.H. Joshi, and I.H. Jermyn. Shape analysis of elastic curves in euclidean spaces. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 33, pages 1415–1428, 2011.

[107] J. Starck and A. Hilton. Surface capture for performance-based animation. In *Computer Graphics and Applications*, volume 27, pages 21–31, 2007.

[108] J. Suarez and R.R. Murphy. Hand gesture recognition with depth images: A review. In *IEEE RO-MAN*, pages 411–417, 2012.

[109] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Proceedings of the Symposium on Geometry Processing*, pages 1383–1392, 2009.

[110] J. Sung, C. Ponce, B. Selman, and A. Saxena. Human activity detection from RGBD images. In *In AAAI workshop on Pattern, Activity and Intent Recognition (PAIR)*, 2011.

[111] H. Tabia, M. Daoudi, J-P. Vandeborre, and O. Colot. A new 3D-matching method of nonrigid and partially similar models using

curve analysis. In *IEEE Transactons on Pattern Analysis and Machine Intelligence*, volume 33, pages 852–858, 2011.

[112] Y-S. Tak, J. Kim, and E. Hwang. Hierarchical querying scheme of human motions for smart home environment. In *Engineering Applications of Artificial Intelligence*, volume 25, pages 1301–1312, 2012.

[113] S. Tang, X. Wang, X. Lv, T. X. Han, J. Keller, Z. He, M. Skubic, and S. Lao. Histogram of oriented normal vectors for object recognition with a depth sensor. In *Asian Conference of Computer Vision*, volume 7725, pages 525–538, 2013.

[114] J.W.H. Tangelder and R.C. Veltkamp. A survey of content based 3D shape retrieval methods. In *Multimedia Tools and Applications*, volume 39, pages 441–471, 2008.

[115] M. Tenorth, J. Bandouch, and M. Beetz. The TUM kitchen data set of everyday manipulation activities for motion tracking and action recognition. In *International Conference on Computer Vision Workshops*, pages 1089–1096, Sept 2009.

[116] J. Tierny, J-P. Vandeborre, and M. Daoudi. Invariant high level reeb graphs of 3D polygonal meshes. In *International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT)*, pages 105–112, Los Alamitos, CA, USA, 2006.

[117] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan. Scanning 3D full human bodies using kinects. In *IEEE Transactions on Visualization and Computer Graphics*, volume 18, pages 643–650, 2012.

[118] C. Tran and M.M. Trivedi. Human body modelling and tracking using volumetric representation: Selected recent studies and possibilities for extensions. In *IEEE International Conference on Distributed Smart Cameras*, pages 1–9, 2008.

[119] T. Tung and T. Matsuyama. Topology dictionary for 3d video understanding. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 34, pages 1645–1657, 2012.

[120] T. Tung, S. Nobuhara, and T. Matsuyama. Complete multi-view reconstruction of dynamic scenes from probabilistic fusion of narrow and wide baseline stereo. In *IEEE International Conference on Computer Vision*, pages 1709 –1716, 2009.

[121] T. Tung and F. Schmitt. The augmented multiresolution reeb graph approach for content-based retrieval of 3D shapes. In *International Journal of Shape Modeling*, volume 11, pages 91–120, 2005.

[122] P. Turaga and R. Chellappa. Locally time-invariant models of human activities using trajectories on the grassmannian. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2435–2441, 2009.

[123] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical computations on grassmann and stiefel manifolds for image and video-based recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 33, pages 2273–2286, 2011.

[124] Gall J. den Bergh M.V. Van Gool L Uebersax, D. Real-time sign language letter and word recognition from depth data. In *IEEE International Conference on Computer Vision Workshops*, pages 383–390, 2011.

[125] A. Veeraraghavan, A.K. Roy-Chowdhury, and R. Chellappa. Matching shape sequences in video with applications in human movement analysis. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 27, pages 1896–1909, 2005.

[126] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3D human skeletons as points in a lie group. In *IEEE conference on Computer Vision and Pattern Recognition*, 2014.

[127] R. Vezzani, D. Baltieri, and R. Cucchiara. HMM based action recognition with projection histogram features. In *Recognizing Patterns in Signals, Speech, Images and Videos*, volume 6388, pages 286–293, 2010.

[128] R. Vezzani, M. Piccardi, and R. Cucchiara. An efficient bayesian framework for on-line action recognition. In *IEEE International Conference on Image Processing*, pages 3553–3556, 2009.

[129] A. W. Vieira, Erickson R. N., Gabriel L. O., Zicheng L., and M. F.M. Campos. STOP: Space-time occupancy patterns for 3D action recognition from depth map sequences. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, volume 7441, pages 252–259, 2012.

[130] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. In *ACM Siggraph*, pages 1–97, 2008.

[131] C. Wang, Y. Wang, and A.L. Yuille. An approach to pose-based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 915–922, June 2013.

[132] H. Wang, C. Yuan, G. Luo, W. Hu, and C. Sun. Action recognition using linear dynamic systems. In *Pattern Recognition*, volume 46, pages 1710 – 1718, 2013.

[133] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu. Robust 3D action recognition with random occupancy patterns. In *European Conference on Computer Vision*, pages 872–885, 2012.

[134] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1290–1297, 2012.

[135] J. Wang and H. Zheng. View-robust action recognition based on temporal self-similarities and dynamic time warping. In *IEEE International Conference on Computer Science and Automation Engineering*, volume 2, pages 498–502, 2012.

[136] L. Wang, L. Cheng, and L. Wang. Elastic sequence correlation for human action analysis. In *IEEE Transactions on Image Processing,*, volume 20, pages 1725–1738, 2011.

[137] T-S. Wang, H-Y. Shum, Y-Q. Xu, and N-N. Zheng. Unsupervised analysis of human gestures. In *IEEE Pacific Rim Conference on Multimedia: Advances in Multimedia Information Processing*, pages 174–181, 2001.

[138] F. W. Warner. Foundations of differentiable manifolds and lie groups. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Graduate texts in mathematicals, New York, NY, 1983. Springer.

[139] D. Weinland, E. Boyer, and R. Ronfard. Action recognition from arbitrary views using 3d exemplars. In *International Conference on Computer Vision*, pages 1–7, Rio de Janeiro, Brésil, 2007.

[140] D. Weinland, R. Ronfard, and E. Boyer. Automatic discovery of action taxonomies from multiple views. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1639–1645, 2006.

[141] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. In *Journal of Computer Vision and Image Understanding*, volume 104, pages 249–257, 2006.

[142] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. In *Computer Vision and Image Understanding*, volume 115, pages 224–241, 2011.

[143] C. Wolf, J. Mille, E. Lombardi, O. Celiktutan, M. Jiu, M. Baccouche, C-E. Bichot E. Dellandrea, C. Garcia, and B. Sankur. The liris human activities dataset and the icpr 2012 human activities recognition and localization competition. In *Technical Report RR-LIRIS-2012-004, LIRIS Laboratory*, 2012.

[144] L. Xia and J.K. Aggarwal. Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2834–2841, June 2013.

[145] L. Xia, C-C. Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3D joints. In *Computer Vision and Pattern Recognition Workshops*, pages 20–27, 2012.

[146] J. Xu, T. Yamasaki, and K. Aizawa. 3D video segmentation using point distance histograms. In *IEEE International Conference on Image Processing*, volume 1, pages 701–704, 2005.

[147] T. Yamasaki and K. Aizawa. Motion segmentation of 3D video using modified shape distribution. In *IEEE International Conference on Multimedia and Expo*, pages 1909 –1912, 2006.

[148] T. Yamasaki and K. Aizawa. Motion segmentation and retrieval for 3D video based on modified shape distribution. In *Journal on Applied Signal Processing EURASIP*, volume 2007, pages 211–211, 2007.

[149] P. Yan, S. M. Khan, and M. Shah. Learning 4D action feature models for arbitrary view action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[150] X. Yang and Y. Tian. Eigenjoints based action recognition using naive bayes nearest neighbor. In *Computer Vision and Pattern Recognition Workshops*, pages 14–19, 2012.

[151] X. Yang, C. Zhang, and Y. Tian. Recognizing actions using depth motion maps-based histograms of oriented gradients. In *international conference on ACM Multimedia*, pages 1057–1060, 2012.

[152] M. Ye, Q. Zhang, L. Wang, J. Zhu, R. Yang, and J. Gall. A survey on human motion analysis from depth data. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, volume 8200, pages 149–187, 2013.

[153] A. Yezzi and A. Mennucci. Conformal metrics and true "gradient flows" for curves. In *IEEE International Conference on Computer Vision*, pages 913–919, 2005.

[154] K. Yun, Jean Honorio, Debaleena Chattopadhyay, Tamara L. Berg, and Dimitris Samaras. Two-person interaction detection using body-pose features and multiple instance learning. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 28–35, 2012.

[155] Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti. American sign language recognition with the kinect. In *International Conference on Multimodal Interfaces*, pages 279–286, 2011.

[156] A. Zaharescu, E. Boyer, and R. Horaud. Transformesh a topology-adaptive mesh-based approach to surface evolution. In *Asian Conference on Computer Vision*, volume 4844, pages 166–175, 2007.

[157] J. Y. Zheng. Acquiring 3-D models from sequences of contours. In *IEEE transactions on Pattern Analysis and Machine Intelligence*, volume 16, pages 163–178, 1994.

[158] Y. Zheng, C-L. Tai, E. Zhang, and P. Xu. Pairwise harmonics for shape analysis. In *IEEE Transactions on Visualization and Computer Graphics*, volume 19, pages 1172–1184, Los Alamitos, CA, USA, 2013.

[159] Y. Zhu, W. Chen, and G. Guo. Fusing spatiotemporal features and joints for 3d action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 486–491, 2013.

Abstract In this thesis, we focus on the development of adequate geometric frameworks in order to model and compare accurately human motion acquired from 3D sensors. In the first framework, we address the problem of pose/motion retrieval in full 3D reconstructed sequences. The human shape representation is formulated using *Extremal Human Curve* (EHC) descriptor extracted from the body surface. It allows efficient shape to shape comparison taking benefits from Riemannian geometry in the open curve shape space. As each human pose represented by this descriptor is viewed as a point in the shape space, we propose to model the motion sequence by a trajectory on this space. Dynamic Time Warping in the feature vector space is then used to compare different motions. In the second framework, we propose a solution for action and gesture recognition from both skeleton and depth data acquired by low cost cameras such as Microsoft Kinect. The action sequence is represented by a dynamical system whose observability matrix is characterized as an element of a Grassmann manifold. Thus, recognition problem is reformulated as a point classification on this manifold. Here, a new learning algorithm based on the notion of tangent spaces is proposed to improve recognition task. Performances of our approach on several benchmarks show high recognition accuracy with low latency.

Keywords Motion analysis, shape similarity, 3D video retrieval, depth images, skeleton, human action recognition, gesture recognition, Riemannian manifold, shape space, Grassmann manifold, observational latency, classification.

Résumé Dans le cadre de cette thèse, nous proposons des approches géométriques permettant d'analyser des mouvements humains à partir de données issues de capteurs 3D. Premièrement, nous abordons le problème de comparaison de poses et de mouvements dans des séquences contenant des modèles de corps humain en 3D. En introduisant un nouveau descripteur, appelé *Extremal Human Curve* (EHC), la forme du corps humain dans une pose donnée est décrite par une collection de courbes. Ces courbes extraites de la surface du maillage relient les points se situant aux extrémités du corps. Dans un formalisme Riemannien, chacune de ces courbes est considérée comme un point dans un espace de formes offrant la possibilité de les comparer. Par ailleurs, les actions sont modélisées par des trajectoires dans cet espace, où elles sont comparées en utilisant la déformation temporelle dynamique. Deuxièmement, nous proposons une approche de reconnaissance d'actions et de gestes à partir de vidéos produites par des capteurs de profondeur. A travers une modélisation géométrique, une séquence d'action est représentée par un système dynamique dont la matrice d'observabilité est caractérisée par un élément de la variété de Grassmann. Par conséquent, la reconnaissance d'actions est reformulée en un problème de classification de points sur cette variété. Ensuite, un nouvel algorithme d'apprentissage basé sur la notion d'espaces tangents est proposé afin d'améliorer le système de reconnaissance. Les résultats de notre approche, testés sur plusieurs bases de données, donnent des taux de reconnaissance de haute précision et de faible latence.

Mots-clés Analyse du mouvement, comparaison de formes, la recherche dans les vidéos 3D, images de profondeur, squelette, la reconnaissance de l'action humaine, la reconnaissance des gestes, variété Riemannienne, varietés de Grassmann , la latence, classification