Towards Understanding Human Behavior by Time-Series Analysis of 3D Motion

Hazem Wannous

MANUSCRIT

présenté en vue d'obtenir le diplôme de

HABILITATION À DIRIGER DES RECHERCHES

Université de Lille
Spécialité: Informatique

Par

Hazem WANNOUS

Towards Understanding Human Behavior by Time-Series Analysis of 3D Motion

Soutenue le 5 Décembre 2018 devant le jury composé de :

Mme Jenny BENOIS-PINEAU	Professeur, Université de Bordeaux,	*Rapporteur*
M. Philippe-Henri GOSSELIN	Principal Scientist, Technicolor	*Rapporteur*
M. Christophe ROSENBERGER	Professeur, ENSICaen	*Rapporteur*
M. Pierre BOULET	Professeur, Université de Lille	*Examinateur*
M. Franck MULTON	Professeur, Université de Rennes 2	*Examinateur*
M. Jean-Philippe VANDEBORRE	Professeur, IMT Lille Douai	*Examinateur-Garant*

Univ. Lille, CNRS, Centrale Lille, IMT Lille Douai, UMR 9189 - CRIStAL, F-59000 Lille, France

Preface

THIS document constitutes the manuscript submitted to obtain the *"Habilitation à Diriger des Recherches"* of the University of Lille. It describes the professional activities that I have carried out since my recruitment as associate professor at University of Lille and IMT Lille Douai in 2010. This document is organized in two parts as follows:

- The first part contains a synthetic curriculum vitae which details my teaching activities at IMT Lille Douai as well as my research activities at CRIStAL laboratory UMR CNRS 9189.

- The second part presents my research activities on human motion analysis. This part is organized in 4 chapters dealing with different modalities of human behavior analysis: motion retrieval from 3D videos, human action recognition from 3D joint sequences, human activity recognition in depth video and hand gesture recognition from depth cameras.

Contents

Part I

Synthesis of Activities

CHAPTER 1

Curriculum-vitae

I am currently Associate Professor at University of Lille and IMT Lille Douai. I am a member of the Centre de Recherche en Informatique, Signal et Automatique de Lille (CRIStAL - UMR 9189). This chapter provides a summary of my activities since my nomination in my present position at IMT Lille Douai. It summarizes all my teaching and research activities. The exhaustive list of my publications is given at the end of the chapter.

1.1 Personal Information

Last and first names	WANNOUS Hazem
Date and place of birth	August 5, 1975 Bechraghi (Syria)
Nationalities	French and Syrian
Personal address	30Ter, Rue Jules Guesde, 59390 Lys Lez Lannoy
Current position	Maître de conférences, Université de Lille / IMT Lille Douai
Research Team	MINT - CRIStAL UMR CNRS 9189
Professional address	IMT Lille Douai - Rue Marconi, 59653 Villeneuve d'Ascq Cedex, France
Telephone number	+33 3 20 43 64 27
Email address	hazem.wannous@univ-lille.fr
Webpage	http://pagesperso.telecom-lille.fr/wannous/

1.2 Academic Positions

Since 12/2010	**Associate Professor** University of Lille / IMT Lille Douai, department of Computer Science, Lille, France CRIStAL Centre de Recherche en Informatique, Signal et Automatique de Lille (UMR CNRS 9189), France
09/2009 - 08/2010	**Research Engineer** (*Ingénieur de recherche*) Institut Polytechnique de Bordeaux, Universiteé de Bordeaux 1, Bordeaux, France
09/2008 - 08/2009	**Temporary Lecturer and Research Assistant** (*Attaché Temporaire d'Enseignement et de Recherche*) Ecole Polytechnique, Université d'Orléans, Laboratoire PRISME, Orléan, France

1.3 Academic Background

December 2008	**Ph.D. in Computer Science, University of Orleans, France**
Specialty	Image Processing & Computer Vision

Thesis	Multi view classification of color regions : application to the 3D assessment of chronic wounds
June 2005	**M.Sc. (Master Professionnel), University of Bourgogne, France**
Specialty	Image, vision and artificial intelligence
Thesis	Automatic adjustment of a road image processing chain
June 2003	**M.Sc. (DEA), University of Clermont-Ferrand, France**
Specialty	Composants et Systemes pour le Traitement de l'Information (CSTI) - Robotic vision
Thesis	Real-time object tracking by a particle filter
June 2000	**Engineer Degree, Tishreen University, Syria**
Specialty	Computer Science

1.4 Teaching activities

Since joining IMT Lille Douai (formerly Télécom Lille) as an Associate Professor in December 2010, I am a member of the department of Computer Science (*Informatique et Réseaux*). IMT Lille Douai is an engineering school resulting from the merger of Mines Douai and Télécom Lille on January 1, 2017. IMT Lille Douai, as one of the 11 engineering and management schools form IMT (*Institut Mines-Télécom*), was established under the supervision of the French Minister of the Economy and Industry, in partnership with University of Lille.

In the rest of this section, I quickly present my past teaching activities in other French institutions before my current position at IMT, then I describe my main current teaching activities and my administrative duties within the school.

My teaching activities started in September 2006, at the second year of my Ph.D. thesis, as temporary teacher at École Nationale Supérieure d'Ingénieurs (ENSI) and at Institut Universitaire de Technologie (IUT) in Bourges (France) for 2 years. Then, I was nominated full-time *"Attaché temporaire d'enseignement et de recherche"* (ATER) at École Polytechnique of University of Orléans for one year. In 2009, I had the opportunity to teach at ENSEIRB-MATMECA in Bordeaux (France). From 2006 to 2010, I was teaching to students in 2nd year IUT and 1st to 3rd year at ENSI and Polytech Orléans (393h ETD[1]). The lectures in which I am mostly involved during this periode are:

- Signal processing on DSP processors (laboratory courses for the development of digital filters on the DSK 5416 board): 16h ETD, 3rd year of engineering, ENSEIRB-MATMECA (2009).
- Object-oriented C++ language exercises: 78h ETD, 3rd year of engineering, Polytech Orléans (2008).
- Windows environment programming MFC laboratory courses: 80h ETD, 2nd year of engineering, Polytech Orléans (2008).
- Image processing lecturers, 26h ETD, 2nd year of engineering, Polytech Orléans (2018).
- Computer vision laboratory courses, microprocessor (assembler) laboratory courses, telecommunication lectures and exercises and industrial electrical engineering lectures and exercises, 80h ETD, 2nd and 3rd years of engineering ENSI Bourges (2007-2008).

[1] ETD (*"Equivalent Travaux Dirigés"*) is the common measure, in the French academic system, for calculating the number of hours taught - the following formula is used to obtain the ETD: (40m lecture)=(1h exercise course)=(1h lab. course).

- Computer tools laboratory courses and digital electronic circuits laboratory courses, 103h ETD, 1st and 2nd years IUT Bourges (2006).

In the remaining of this document, I will describe only my teaching activities since my nomination in December 2010 as Associate Professor at IMT Lille Douai.

1.4.1 Synthetic report of teaching experience

In Table 1.4, details of my teaching activities are reported in terms of lectures/exercise courses/lab. courses.

	2010-2011	2011-2012	2012-2013	2013-2014	2014-2015	2015-2016	2016-2017	2017-2018
Algo. and struct. prog.	24/21/30	23/6/3	22/3/22	27/6/23	33/-/21	38/-/-	38/-/-	30/-/-
MultiMedia indexing	-/-/-	18/-/24	17/-/24	18/-/17	21/-/37	-/-/-	-/-/-	-/-/-
Advanced prog. tech.	-/-/-	18/-/42	18/-/24	12/3/21	9/6/23	9/6/-	9/-/15	9/-/12
Intro. to telcom	-/-/-	3/6/3	-/6/-	-/6/-	-/6/-	-/-/-	-/-/-	-/-/-
Big multimedia data	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	21/-/12	21/-/12	21/-/12
Image proc. mobile	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	9/-/21	9/-/30	6/-/36
3D vision and IHM	-/-/-	6/-/30	24/-/27	21/-/30	21/-/27	24/-/24	24/-/24	24/-/24
Internship tutor	-/15/-	-/15/-	-/15/-	-/25/-	-/20/-	-/20/-	-/25/-	-/25/-
Project tutor	-/25/-	-/15/-	-/15/-	-/15/-	-/30/-	-/30/-	-/20/-	-/30/-
Number of hours (ETD)	115	212	217	224	254	214	227	229

Table 1.4: Teaching activities in hours (lectures/exercise courses/lab. courses) per field/specialty and year

1.4.2 Details on Main Courses

In this section, the courses in which I am involved the most at IMT Lille Douai are detailed.

Algorithmic and Structured Programming. It is the first course on programming for the 1st year students at IMT, I am coordinating it. It aims to present the basic requirements for the design of a program and its implementation on a computer.

- Architecture of computers and operating systems
- Introduction to algorithmic
- Programming in C language (expressions, variables, control flow, functions, arrays, structures, pointers, ...)
- Software development project: programming a game in C

I am responsible for lectures and have also participated in the past in exercise and laboratory courses.

Introduction to telecommunications It is a 1st year engineering course at IMT. I am coordinating it and have also participated in the past in exercise and laboratory courses. The course aims to provide a basic culture on computer and telephone networks for the 1st year students. The focus is on the principles and technologies implemented to transmit information between two elements of a network. The topics covered here include:

- transmission media
- network protocols,
- xDSL technologies, Wifi, RTC
- mobiles (1/2/3/4G)

Advanced programming technologies. It is a 2nd year engineering course. I am coordinating it. The objective of this course is to deepen the students' knowledge of computer science and programming through teamwork on a project. Each team is typically composed of 4 subgroups. Each subgroup handles one aspect of the project (lexical analyzer, synthetic parser, evaluation engine and graphics and HMI). The final integration of the project depends on the quality of the preliminary work concerning the study, the specification, the overall coherence of each of the parts. I am responsible for lectures and have also participated in the past in exercise and laboratory courses.

Computer vision and human-machine interaction. It is a 5th year engineering course. The main goal of this course is to introduce the basic technological components involved in the digital entertainment industry. An important part of this course is devoted to the study of 3D modeling techniques that are used in the creation of 3D content, 3D acquisition techniques that allow 3D detection in the real world and an introduction to the techniques of the human-machine interaction. The main subjects on this courses:

- digital representations of real objects (3D acquisition techniques, reconstruction);
- presentation of advanced technologies for 3D modeling;
- virtual reality, human-machine interaction.

Machine learning and multimedia data processing. It is a 5th year course at IMT. It is part of a bigger course (Big Multimedia Data), which we offer since 2016 with colleagues from the department as an evolution of a former course that I coordinated "Multimedia Retrieval and Networks". This course addresses a growing field, the big data, with a focus on application related to multimedia data (web pages, videos, ...). The topics covered in my course include:

- classical machine learning techniques;
- deep learning techniques for time-series;
- analysis and retrieval of multimedia content;
- retrieval for information on the web.

Image processing on mobile terminal It is a 4th year E-learning engineering course. In September 2015, I created a novel 4th year engineering course on data science, which is a quickly growing field, at the interface of computer vision, data analysis and mobile applications. The topics covered here include:

- basic image processing, filtering;
- visual keypoint detection and feature description;

- convolutional neural networks;
- development of applications for mobile terminals: the Android platform;
- web services and HTML;
- analysis and retrieval of multimedia content.

1.4.3 Educational responsibilities

At IMT Lille Douai, the coordination of a *Unité de Valeur (UV)* or a course is a complete task which involves the creation of the course in terms of lectures, exercises, and laboratory courses, the design of the teaching materials, including room reservations/management. The coordination of teaching staff, the design and grading of exams, the participation in committees, etc. The teaching times given in this paragraph are in-class hours, that is excluding coordination, grading, etc. The table below summarizes my activities in my main UV/courses. It indicates, for each UV, the average number of students enrolled each year (which varies from year to another), the number of hours spent in class by students, the number of these hours (lecture / exercise course / lab. course) that I personally performed in this UV, and the number of teachers involved and managed in the overall course of the UV.

UV/courses	Nb. of students	Nb. of hours	Nb. hours of interventions	Nb. of speakers to manege
algorithmic and structured programming	130	90	25	4
advanced programming technologies	80	30	15	3
multimedia indexing and retrieval	25	120	40	6
big multimedia data	30	120	30	1
3D digital entertainment technologies	15	120	40	10
image processing on mobile terminal	20	80	40	3
Introduction to telecommunications	130	30	0	4

Table 1.5: A summary of my activities in my main UV/courses

My educational responsibilities are mainly the coordination of the following courses:

- algorithmic and structured programming (1st year of engineering),
- advanced programming technologies (2nd year of engineering),
- 3e entertainment technologies (5th year of engineering),
- image processing on mobile terminal (4th year E-learning engineering),
- introduction to telecommunications.

I am the designer and coordinator of the UV 3DETech. 3DETech is a scientific optional advanced course for 5th year engineering students, which presents methods and techniques for interacting with digital and virtual environments, revolving around 3D digital entertainment. An important part of this course is devoted to 3D vision and image synthesis techniques that will be at the heart of the services and uses of tomorrow. This UV represents 120 hours of classes for the student. My courses on 3D computer vision and IHM represent about 40 hours of teaching.

I am the co-designer and the coordinator of the UV "Image processing on mobile terminal". This UV is a scientific optional advanced course for 4th year E-learning engineering students, on data analysis and mobile applications. This UV represents 120 hours of teaching for the student, spread

over 4 months for E-learning setting in class with a software project (smartphone application). My courses (lecture and lab. course) on computer vision and machine learning represent about 60 hours of teaching.

I am also the coordinator of "Algorithmic and Structured Programming", "Advanced programming technologies" and "Introduction to telecommunications" courses.

Due to the creation of the new school IMT Lille Douai after the merging of Telecom Lille with Mines Douai, I have been also largely involved, since 2016, in the design of the syllabus of the new engineering program that began in September 2018. I am also in charge, with a colleague from the school, of the creation of a Specialized Master in "Data Science and Applications", which is scheduled to start in September 2019. Finally, I represent IMT at *Campus des métiers et des qualifications Image numérique et industries créatives*[2] and I am in the steering committee and the scientific committee.

[2]http://campus-inic.fr

1.5 Research activities

1.5.1 Supervision

Post-doctoral students

December 2015–August 2016	**Maxime Devanne**
Funding	FUI grant
Co-supervision	Mohamed Daoudi
Subject	Human Behavior Understanding by Body and Face Analysis
Publications	[C4]

Ph.D. students

October 2018–September 2021	**Théo Voillemin**
Funding	University of Lille and IMT Lille Douai
Co-supervision	Jean-Philippe Vandeborre
Subject	Personalized augmented reality assistance by hand gesture recognition using head-mounted displays
October 2014–December 2017	**Quentin De Smedt**
Funding	Bourse d'Excellence IMT Lille Douai
Co-supervision	Jean-Philippe Vandeborre
Subject	Dynamic Hand Gesture Recognition - from Traditional Hand-crafted to recent Deep Learning Approaches
Publications	[C1, P3, C3, C5]
February 2012–December 2014	**Maxime Devanne**
Funding	University of Lille 1 and University of Florence, Italy
Co-supervision	Mohamed Daoudi and Pietro Pala
Subject	3D Human Behavior Understanding by Shape Analysis of Human Motion and Pose
Publications	[J1, C6, J4, C10]
November 2011–October 2014	**Rim Slama**
Funding	University of Lille 1 and Region Nord-Pas-de-Calais
Co-supervision	Mohamed Daoudi
Subject	Geometrical Approach for 3D Human Motion Analysis: Application to Action Recognition and Retrieval
Publications	[J2, C7, J3, C8, C9]

Master students

April 2018–August 2018	**Denis Balschakov**
Master	Master DCISS – Université de Grenoble Alpes
Co-supervision	Esperanza Perdrix and Aude Bourin
Subject	Deep Learning Approach for Prediction of Atmospheric Pollutants
April 2017–September 2017	**Manel Rhif**

Master	Projet de Fin d'Etude ISAMM Tunis, Tunisie
Subject	Action Recognition from Skeleton Sequences using Convolutional Neural Network on Lie Group Manifold
Publications	[C2]
February 2016–August 2016	**Elliot Vanegue**
Master	Master Recherche IVI – Université de Lille1
Co-supervision	Jean-Philippe Vandeborre
Subject	An Interactive Approach to Semantic Segmentation of 3D Objects Retrieval
February 2012–August 2012	**Maxime Devanne**
Master	Projet de Fin d'Etude – Télćom Lille
Co-supervision	Olivier Losson
Subject	3D Human body modeling by Kinect camera

1.5.2 Research Projects

2013 – 2017	**CrABEx ANR-13-CORD-0013** (Participant)
Subject	Creating 3D Graphic Content Assisted by a Database of Examples
Academic partners	CRIStAL - UMR CNRS 9189, LIRIS UMR 5205, LTCI UMR 5141 (Télécom PariTech)
Industrial partners	3DDUO (Plaine Image, Tourcoing), ICOM / Gamagora (Université de Lyon 2)
Participation	Proposal co-writing and participation in the 3D interactive semantic segmentation part
2014 – 2016	**FUI MAGNUM - Fonds Unique Interministériel** (Participant)
Subject	Measurement, Analysis and Flow Management, Native Unified in Stores 17ème appel à projets des pôles de compétitivité
Academic partners	CRIStAL - UMR CNRS 9189 (Université Lille 1), LSIS UMR 7296 (Université Aix Marseille)
Industrial partners	Easycomptage, euroshaktiware, Robopec, WIT SA, IQC Assest Management et Arclan System
Participation	Proposal co-writing and participation in the workpackage concerning the realization of the recognition of emotional gestures of people in front of a showcase equipped with 2D / 3D sensors
2014 – 2015	**It's Me - Bonus Funding Research** (project leader)
Subject	Artistic 3D holographic video - gesture recognition and interaction
Funding	Funds for interactive and innovative projects with high development potential (PICTANOVO, Région Nord-Pas-de-Calais
Academic partners	CRIStAL- UMR CNRS 9189 (Université Lille 1)
Industrial partners	Idées-3Com, Acnot, Holusion.

1.5.3 Event organization and committee member

2017 Co-organizer and General Chair of the 3D Shape Retrieval Contest 2017 3D Hand Gesture Recognition Using a Depth and Skeletal Dataset SHREC2017, Lyon, April 23-24, 2017

2016 Co-organizer and General Chair of the "International Workshop on Understanding Human Activities through 3D Sensors (UHA3DS'16), in conjunction with IAPR-ICPR, Cancun, Mexico, December 4-8 2016

2015 Member of the organization committee of Shape Modeling International conference SMI 2015, Telecom, Lille June 24-26 2015

2015 Co-organizer and General Chair of the International Workshop on Understanding Human Activities through 3D Sensors UHA3DS'15, in conjunction with IEEE-FG, Ljubljana, Slovenia, May 4-8 2015

2014 Area Chair of the IAPR International Conference on Pattern Recognition (Pattern Recognition Applications Track), Stockholm, Sweden, 24-28 August 2014

Since 2014 Member of the program committee of Eurographics Workshop on 3D Object Retrieval, 2014 - 2018

2012 Member of the organization committee of French national conference COmpression et REprésentation des Signaux Audiovisuels CORESA 2012, Lille, May 24, 2012,

1.5.4 Evaluation and review panels

Ph.D. committees

06/12/2017 **Alexandre Pérez**
Institution Université de Cergy Pontoise - ENSEA
Thesis Analysis and Recognition of Gestures with a RGBD Sensor

Evaluation of research project

09/2018 Reviewer of a research project proposal on Support of Research, Development and Innovation - Czech Science Foundation.

Reviewing Activities

International Journals IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Image Processing, IEEE Transactions on Multimedia, Pattern Recognition, IEEE Transactions on Cybernetics, IEEE Transactions on Image Processing, Journal of Electronic Imaging, Applied Sciences, Computer Vision and Image Understanding

International Conferences IAPR ICPR 2016, CORESA, Eurographics Workshop on 3D Object Retrieval 3DOR 2017, 3DOR 2016, IEEE FG 2015, 3DOR 2015, Workshop DIFF-CV2015, VISUAL 2015, AVSS 2014, ICIAP 2013

1.5.5 Seminar and invited talks

September 21, 2018 Hand gesture detection and recognition by a deep learning approach, invited talk *Sequel* research team at INRIA Lille, France.

January 31, 2018 3D Human motion analysis from RGBD sensors, talk at the workshop on motion capture organized by TCTS team, University of Mons, Belgium.

February 22, 2016 Shape analysis of human motion and Pose, seminar at NUMEDIART Research institute, University of Mons, Belgium.

December 11, 2014 Approche géométrique pour la reconnaissance d'actions humaines à partir d'un capteur RGB-D, *Journée GDR-ISIS "Action Visage, geste, action et comportement"*

April 12, 2011 Reconstruction et localisation 3D en environnement intérieur pour l'indexation de vidéo issue de caméra portée, *Journée GDR-ISIS "SfM-SfX - Structure à partir du mouvement et d'autres indices visuels : état de l'art et évolution du domaine"*

January 21, 2011 Localisation 3D en environnement intérieur par caméra portée pour la détection d'événements liés aux activités, *Journée GDR-ISIS Suivi d'objets dans l'espace 3D: méthodes et applications*

April 02, 2009 Conception d'un outil complet d'aide au diagnostic clinique: de l'application à la classification couleur multi-vues, *Journée GDR ISIS du groupe SCATI : Les Systèmes de Vision: de l'acquisition à l'interprétation*

January 09, 2017 Classification tissulaire robuste appliquée au suivi thérapeutique d'escarres", *école d'hiver sur l'imagerie numérique couleur*, Campus du Futuroscope, Université de Poitiers

September 28, 2006 Evaluation et réglage d'une chaîne de traitement d'images routiéres, *Journée bilan du groupe SCATI : chaîne et pilotage de traitements GdR ISIS*

1.5.6 Scientific collaboration

Here are only mentioned the most significant collaborations that have resulted in the publication of at least one international conference or journal paper.

- Pietro Pala and Stefano Berretti (Media Integration and Communication Center MICC), University of Florence, Italy. Maxime Devanne, PhD student, was in collaboration between the University of Lille 1, France and the University of Florence, Italie (2012 - 2015)
- Anuj Srivastava (Statistical Shape Analysis and Modeling Group SSAMG), Florida State University, USA
- Francisco Florez-Revuelta (University of Alicante), Spain.

1.5.7 Distinction and awards

2015 **Ph.D. and research supervision bonus (PEDR)**: awarded by the University of Lille (application is approved by the National Council of Universities CNU) in october 2015 for a period of 4 years.

1.5.8 Synthetic report of scientific production

- **3** *submitted* papers in international journals,
- **7** international journal papers,
- **21** international conference papers,

- **5** national conference papers,
- **1** book chapters,
- **3** national medical journal papers,

1.6 Full list of publications

1.6.1 Submitted papers

[P1] H. WANNOUS, J.-P. VANDEBORRE, and Q. DE SMEDT. "Dynamic Hand Gesture Detection and Recognition using Combined Convolutional and Recurrent Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence – in review*, 2018.

[P2] S. RIBET, H. WANNOUS, and J.-P. VANDEBORRE. "Survey on Style in 3D Human Body Motion: Taxonomy, Data, Recognition and its Applications". In: *IEEE Transactions on Affective Computing – in review (minor revisions submitted)*, 2018.

[P3] Q. DE SMEDT, H. WANNOUS, and J.-P. VANDEBORRE. "Heterogeneous hand gesture recognition using 3D dynamic skeletal data". In: *Computer Vision and Image Understanding – in review (minor revisions submitted)*, 2018.

1.6.2 International journal papers

[J1] M. DEVANNE, S. BERRETTI, P. PALA, H. WANNOUS, M. DAOUDI, and A. D. BIMBO. "Motion segment decomposition of RGB-D sequences for human behavior understanding". In: *Pattern Recognition* 61, 2017, pp. 222–233. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2016.07.041.

[J2] R. SLAMA, H. WANNOUS, M. DAOUDI, and A. SRIVASTAVA. "Accurate 3D action recognition using learning on the Grassmann manifold". In: *Pattern Recognition* 48 (2), Feb. 2015, pp. 556–567.

[J3] R. SLAMA, H. WANNOUS, and M. DAOUDI. "3D human motion analysis framework for shape similarity and retrieval". In: *Image and Vision Computing* 32 (2), 2014, pp. 131–154. ISSN: 0262-8856. DOI: http://dx.doi.org/10.1016/j.imavis.2013.12.011.

[J4] M. DEVANNE, H. WANNOUS, S. BERRETTI, P. PALA, M. DAOUDI, and A. DEL BIMBO. "3D Human Action Recognition by Shape Analysis of Motion Trajectories on Riemannian Manifold". In: *IEEE Trans. on Cybernetics* 45 (7), 2014, pp. 1340–1352. ISSN: 2168-2267. DOI: 10.1109/TCYB.2014.2350774.

[J5] H. WANNOUS, Y. LUCAS, S. TREUILLET, A. MANSOURI, and Y. VOISIN. "Improving color correction across camera and illumination changes by contextual sample selection". In: *Journal of Electronic Imaging* 21 (2), June 2012, pp. 023015-1–023015-14. DOI: 10.1117/1.JEI.21.2.023015.

[J6] H. WANNOUS, Y. LUCAS, and S. TREUILLET. "Enhanced Assessment of the Wound-Healing Process by Accurate Multiview Tissue Classification". In: *IEEE Transactions on Medical Imaging* 30 (2), 2011. 12 Pages, pp. 315–326. DOI: 10.1109/TMI.2010.2077739.

[J7] H. WANNOUS, S. TREUILLET, and Y. LUCAS. "Robust tissue classification for reproducible wound assessment in telemedicine environments". In: *Journal of Electronic Imaging* 19, 2 Feb. 2010. DOI: 10.1117/1.3378149.

1.6.3 International conference papers

[C1] Q. De Smedt, H. Wannous, and J.-P. Vandeborre. "3D Hand Gesture Recognition by Analysing Set-of-Joints Trajectories". In: *International Conference on Pattern Recognition (ICPR) / UHA3DS 2016 workshop*. Cancun, Mexico: Springer International Publishing, Dec. 2018, pp. 86–97. ISBN: 978-3-319-91863-1.

[C2] M. Rhif, H. Wannous, and I.-R. Farah. "Action Recognition from 3D Skeleton Sequences using Deep Networks on Lie Group Features". In: *International Conference on Pattern Recognition (ICPR)*. Beijing, China, 2018.

[C3] Q. De Smedt, H. Wannous, J.-P. Vandeborre, J. Guerry, B. Le Saux, and D. Filliat. "SHREC'17 Track: 3D Hand Gesture Recognition Using a Depth and Skeletal Dataset". In: *10th Eurographics Workshop on 3D Object Retrieval*. Ed. by I. Pratikakis, F. Dupont, and M. Ovsjanikov. Lyon, France, Apr. 2017. DOI: 10.2312/3dor.20171049.

[C4] M. Devanne, H. Wannous, M. Daoudi, S. Berretti, A. D. Bimbo, and P. Pala. "Learning Shape Variations of Motion Trajectories for Gait Analysis". In: *International Conference on Pattern Recognition (ICPR 2016)*. Cancun, Mexico, Dec. 2016, pp. 895–900. DOI: 10.1109/ICPR.2016.7899749.

[C5] Q. De Smedt, H. Wannous, and J.-P. Vandeborre. "Skeleton-Based Dynamic Hand Gesture Recognition". In: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2016 IEEE Conference on*. Las Vegas, United States, June 2016, pp. 1206–1214. DOI: 10.1109/CVPRW.2016.153.

[C6] M. Devanne, A. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. Del Bimbo. "Combined Shape Analysis of Human Poses and Motion Units for Action Segmentation and Recognition". In: *Int. Work. on Understanding Human Activities through 3D Sensors (UHA3DS'15), in conjunction with FG*. Ljubljana, Slovenia, May 2015.

[C7] R. Slama, H. Wannous, and M. Daoudi. "Grassmannian Representation of Motion Depth for 3D Human Gesture and Action Recognition". In: *22nd International Conference on Pattern Recognition (ICPR)*. 2014, pp. 3499–3504. ISBN: 978-1-4799-5209-0.

[C8] R. Slama, H. Wannous, and M. Daoudi. "3D Human Video Retrieval: from Pose to Motion Matching". In: *Eurographics Workshop on 3D Object Retrieval*. Girona, Spain, May 2013.

[C9] R. Slama, H. Wannous, and M. Daoudi. "Extremal Human Curves: a New Human Body Shape and Pose Descriptor". In: *Proc. IEEE Int. Conf. on Automatic Face and Gesture Recognition*. Shanghai, China, 2013, pp. 1–6.

[C10] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. Del Bimbo. "Space-Time Pose Representation for 3D Human Action Recognition". In: *New Trends in Image Analysis and Processing – ICIAP 2013*. Ed. by A. Petrosino, L. Maddalena, and P. Pala. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 456–464. ISBN: 978-3-642-41190-8.

[C11] H. Wannous, V. Dovgalecs, and R. Mégret. "Place Recognition via 3D Modeling for Personal Activity Lifelog Using Wearable Camera". In: *Proceedings of the 18th International Conference on Advances in Multimedia Modeling*. Klagenfurt, Austria, 2012, pp. 244–254. ISBN: 978-3-642-27354-4. DOI: 10.1007/978-3-642-27355-1_24.

[C12] V. DOVGALECS, R. MEGRET, H. WANNOUS, and Y. BERTHOUMIEU. "Semi-Supervised Learning for Location Recognition from Wearable Video". In: *International Workshop on Content-Based Multimedia Indexing (CBMI)*. Grenoble, France, June 2010. DOI: 10.1109/CBMI.2010.5529903.

[C13] H. WANNOUS, S. TREUILLET, Y. LUCAS, A. MANSOURI, and Y. VOISIN. "Design of a Customized Pattern for Improving Color Constancy Across Camera and Illumination Changes". In: *VISAPP 2010 - Fifth International Conference on Computer Vision Theory and Applications*. Vol. 1. Angers, France, May 2010, pp. 60–67.

[C14] H. WANNOUS, Y. LUCAS, and S. TREUILLET. "Combined Machine Learning with Multi-view Modeling for Robust Wound Tissue Assessment". In: *VISAPP 2010 - Fifth International Conference on Computer Vision Theory and Applications*. Vol. 1. Angers, France, May 2010, pp. 92–104.

[C15] R. MÉGRET, V. DOVGALECS, H. WANNOUS, S. KARAMAN, J. BENOIS-PINEAU, E. EL KHOURY, J. PINQUIER, P. JOLY, R. ANDRÉ-OBRECHT, Y. GAÉSTEL, and J.-F. DARTIGUES. "The IMMED Project: Wearable Video Monitoring of People with Age Dementia". In: *Proceedings of the 18th ACM International Conference on Multimedia*. MM '10. Firenze, Italy: ACM, 2010, pp. 1299–1302. ISBN: 978-1-60558-933-6. DOI: 10.1145/1873951.1874206.

[C16] H. WANNOUS, Y. LUCAS, and S. TREUILLET. "Efficient SVM classifier based on color and texture region features for wound tissue images". In: *Medical Imaging 2008: Computer-Aided Diagnosis*. Vol. 6915. Mar. 2008, 69152T. DOI: 10.1117/12.770339.

[C17] H. WANNOUS, Y. LUCAS, S. TREUILLET, and B. ALBOUY. "Fusion of Multi-view Tissue Classification Based on Wound 3D Model". In: *Advanced Concepts for Intelligent Vision Systems*. Ed. by J. BLANC-TALON, S. BOURENNANE, W. PHILIPS, D. POPESCU, and P. SCHEUNDERS. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 924–935. ISBN: 978-3-540-88458-3.

[C18] H. WANNOUS, Y. LUCAS, S. TREUILLET, and B. ALBOUY. "A complete 3D wound assessment tool for accurate tissue classification and measurement". In: *15th IEEE International Conference on Image Processing*. San Diego, CA, USA: IEEE, 2008, pp. 2928–2931.

[C19] H. WANNOUS, Y. LUCAS, S. TREUILLET, and B. ALBOUY. "Mapping Classification Results on 3D model: a Solution for Measuring the Real Areas Covered by Skin Wound Tissues". In: *3rd International Conference on Information and Communication Technologies: From Theory to Applications*. Damascus, Syria: IEEE, 2008.

[C20] H. WANNOUS, S. TREUILLET, and Y. LUCAS. "Supervised tissue classification from color images for a complete wound assessment tool". In: *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Lyon, France: IEEE, 2007, pp. 6032–6035.

[C21] Y. LUCAS, S. TREUILLET, H. WANNOUS, and J.-C. PICHAUD. "3D and color wound assessment using a simple digital camera". In: *9th EPUAP Open Meeting*. Berlin, Germany, 2006.

1.6.4 National medical journal papers

[RN1] Y. LUCAS, S. TREUILLET, H. WANNOUS, B. ALBOUY, and J.-C. PICHAUD. "Classification colorielle déscarres par étiquetage de régions 3D". In: *L'Escarre - Revue officielle P.E.R.S.E. Prévention et Recherche en Soins d'Escarre* (36), Dec. 2007, pp. 16–18.

[RN2] Y. LUCAS, S. TREUILLET, B. ALBOUY, H. WANNOUS, and J.-C. PICHAUD. "Evaluation d'escarres avec un simple appareil photo numérique - Un outil développé avec des praticiens hospitaliers". In: *L'Escarre - Revue officielle P.E.R.S.E. Prévention et Recherche en Soins d'Escarre* (34), 2007, pp. 24–27.

[RN3] J.-C. PICHAUD, H. BARRE, Y. LUCAS, S. TREUILLET, B. ALBOUY, and H. WANNOUS. "ESCALE : ESCarre Analyse Lisibilité Evaluation Imagerie 3D couleur par simple appareil photo numérique Fruit de l'expérience d'une collaboration médico-scientifique". In: *L'Escarre - Revue officielle P.E.R.S.E. Prévention et Recherche en Soins d'Escarre* (30), 2006, pp. 26–30.

1.6.5 National conference papers

[CN1] M. DEVANNE, H. WANNOUS, S. BERRETTI, P. PALA, M. DAOUDI, and A. DEL BIMBO. "Reconnaissance dáctions humaines 3D par lánalyse de forme des trajectoires de mouvement". In: *COmpression et REprésentation des Signaux Audiovisuels (CORESA)*. Reims, France, 2014.

[CN2] R. SLAMA, H. WANNOUS, and M. DAOUDI. "Indexation et recherche dáctions humaines 3D basées sur lánalyse des courbes surfaciques". In: *COmpression et REprésentation des Signaux Audiovisuels (CORESA)*. Creusot, France, 2013.

[CN3] H. WANNOUS, Y. LUCAS, S. TREUILLET, and B. ALBOUY. "ESCALE : outil complet de mesures 2D/3D pour le suivi thérapeutique déscarres". In: *6ème Colloque Capteurs 2008*. Bourges, France, 2008.

[CN4] H. WANNOUS, Y. LUCAS, and S. TREUILLET. "Classification supervisée de régions couleur - Application au suivi thérapeutique des escarres". In: *21ème Conference sur le traitement du signal et des images (GRETSI'07)*. Troyes, France, 2007.

[CN5] H. WANNOUS, Y. LUCAS, and S. TREUILLET. "Classification tissulaire robuste appliquée au suivi thérapeutique déscarres". In: *Onzième Congrès Francophone des Jeunes Chercheurs en Vision par Ordinateur (ORASIS'07)*. Obernai, France, 2007.

1.6.6 Book chapters

[B1] H. WANNOUS, P. PALA, M. DAOUDI, and F. FLUREZ-REVUELTA. "Understanding Human Activities Through 3D Sensors". In: Springer International Publishing, 2018. ISBN: 978-3-319-91863-1.

1.6.7 Thesis

[T1] H. WANNOUS. "Multi view Classification of Color Regions – Application to the 3D Assessment of Chronic Wounds". Thesis. Université d'Orléans, Dec. 2008.

Part II

Research activities

CHAPTER 2

Introduction

Human motion analysis has been an active topic from the early beginning of computer vision [181] due to its relevance to a large variety of domains, each one deals with a specific aspect of the problem. It has received a great interest during the last decade and became currently one of the most active research topics in computer vision. Body motions are the natural way of the people to communicate with their real world environments. Hence, the interpretation and understanding of such human motions is valuable in many field of application, such as virtual reality, gaming, human-computer interfaces and assisted living, etc.

Human motion analysis concerns the detection, tracking, recognition of people activities and and more generally, the understanding of its behaviors from image sequences involving humans. Human detection involves motion segmentation and object classification, and mainly employed in video surveillance. Tracking is particularly important in human motion analysis because of its role in several methods of pose estimation and action recognition. The tracking algorithms within human motion analysis usually relates to the human motion detection in video sequences as well as its tracking over the sequence. In most cases, the overall motion of the whole body is taken into account in order to determine global attitude of people in the crowd. In contrast to human detection and tracking, body motion analysis focuses on how the movement is executed. Such issues are particularly useful for many applications, like people gait recognition, abnormal detection and sport rehabilitation. Human behavoir understanding relates to the local body parts or whole body analysis of the human motion in order to recognize it and understand its meaning.

Human behavior analysis from vision cues is composed of sub-domains of research differing in scale, both spatially –face, hand, upper-body, whole body clues– and temporally –time to perform an expression, a gesture, an action or yet an activity–. Similar approaches can be used to tackle each or a part of those problems. However, each of them has its own particularities that have to be taken into account to create robust and efficient recognition systems. The main concern of this dissertation is the issue of human behavior understanding trough vision-based analysis of the human motion limited to body behaviour. The behaviour in this document, according to the complexity of motion, can be conceptually categorized into different types of motion modalities: gestures, actions, activities and grained-fine hand gestures. However, we note from a state-of-the-art overview, the boundaries between these terminologies are often smooth as on behavior can lie between two behavior types. For instance, a simple action performed with one arm can be assimilated as a gesture. Conversely, an action performed with an object can be viewed as an activity.

The rest of the chapter presents the recent scientific context of my research directions and a summary of my contributions organized in four chapters dealing with different modalities of human behavior analysis: motion retrieval from 3D videos, human action recognition from 3D joint sequences, human activity recognition in depth video and hand gesture recognition from depth cameras.

2.1 Scientific Context

Human motion analysis has evolved substantially in parallel with major technological advancements, especially capturing technologies. Wide investigations of human behavior understanding in computer vision started with the development of techniques to operate on regular visual data, i.e. color images or videos from RGB cameras [65, 107, 161]. However, most of these methods suffer from some limitations coming from 2D videos, like the sensitivity to color and lighting conditions, background clutter and occlusions, in addition to the fact they can only capture projective information of the real world.

Besides, 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 [183, 199]. In such videos, each frame is a mesh approximation of the body surface shape often generated independently regardless of its neighboring frames.

With the recent release of RGB-D sensors, like Microsoft Kinect [91] or Asus Xtion PRO LIVE [105], that revolutionized pose estimation approaches [31, 146], new opportunities have emerged in this field. The analysis of human motion goes, however, beyond the pose extraction, where a higher level of interpretation is required in order to understand human behaviors.

Human behavior analysis has shown considerable progress in the field, yet unresolved problems remain, especially those related to motion representation and time series modeling. The complex nature of human motion makes understanding human behavior a difficult task. Indeed, human movements span a high dimensional space and motions with similar meanings performed by different subjects exhibit substantial variations. Further complications arise from the fact that the recognition system must be sufficiently robust with respect to the speed of execution and the geometric transformations of the movement, such as the size of the subject, its position and its orientation in the scene. Additionally, in complexe activities, interactions with objects add more challenges to the behavoir recognition issue. One of the main issues of recognition systems is the online capability for early detection and recognition. This capability anables the analysis of very long motion sequences of different behaviors performed successively, and makes the interaction system more natural.

The work that I present in this document is essentially based on the following research works:

- Rim Slama's Ph.D. thesis (2011-2014) *Geometric Approaches for 3D Human Motion Analysis: Application to Action Recognition and Retrieval*
- Maxime Devanne's Ph.D. thesis (2012-2015) *3D Human Behavior Understanding by Shape - Analysis of Human Motion and Pose*
- Quentin De Smedt's Ph.D. thesis (2014-2017) *Dynamic Hand Gesture Recognition - from Traditional Handcrafted to Recent Deep Learning Approaches*

2.2 Contributions of the HdR

The scientific context described above raises many challenges, including:

Human motion retrieval from 3D videos With the emergence of capture technology for motion data collection, human motion data have become available and widely used in several research areas in computer vision and computer graphics. Thus, an efficient motion data retrieval method is needed. However, 3D shape representation and similarity is critical to perform an accurate and efficient human motion retrieval. Our focus in **Chapter 3** concerns 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.

Starting point being the data representation issue, we chose to formulate the human shape representation as Extremal Human Curve descriptor extracted from both the spatial and the topological dimensions of the body surface. Its extraction is based on extremal features and geodesics between each pair of them. Being invariant to pose changes, EHC descriptors allow the comparison of pose and motion of subjects regardless of translation, rotation and scaling. Such a representation can be employed not only in pose retrieval for video annotation and concatenation but also in motion retrieval, clustering and activity analysis. The key idea behind its extension to the temporal domain was 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 and gesture recognition on the Grassmann manifold More recently, effective and inexpensive depth video cameras, much less cumbersome than multiple camera or scanning systems, are increasingly emerging. These range sensors provide 3D structural information of the scene, which offers more discerning information to recover human postures. Often compared to 2D cameras, these devices are more robust to common low-level issues in RGB imagery like background subtraction and light variations. **Chapter 4** addressed the issue of human action recognition from such depth cameras. Recognizing human actions have many potential applications including video surveillance, human computer interfaces, sport video analysis, health care, etc. 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.

To perform action recognition, we proposed to model sequence features temporally as subspaces lying in Grassmann manifold. Action recognition is performed by introducing a learning algorithm on the manifold. First, we constructed time series as a sequence of consecutive feature vectors with temporal order. Second, to capture 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 processes, in which we perform classification using features computed from depth map information using: (1) a Truncated Wrapped Gaussian model using

features computed from depth map information, one for each class in its own tangent space, and (2) a vector representation formed by 3D skeleton coordinates in tangent spaces associated with different classes in order to train a linear SVM. Our approach in terms of accuracy/latency revealed an important ability for a low-latency action recognition system.

The effectiveness of skeleton data has been proven for the analysis and recognition of relatively simple behaviors, like human actions. However, more complex behaviors like activities involving manipulation of objects. So as to characterize such human-object interactions, hybrid approaches combining description of both human motion and objects are appreciated. Such activities also involve more complex human motions. Hence, a temporally local analysis of the motion is often required. Finally, Linear Dynamic Systems 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 classification problems by modeling trajectories on the manifold.

Human activity recognition by shape analysis of motion trajectories In order to address the problem of human behavior understanding where many issues are still open, we introduced in **Chapter 5** a new approach related to the local and/or global analysis of the human motion in order to better understand its meaning. In particular, we extend the Riemannian framework presented in **Chapter 3** to deal with high dimensional curves, by considering the human action representation as a trajectory in action space over the time. First, shapes of trajectories are interpreted within a Riemannian manifold and an elastic metric is employed for computing shape similarity, thus improving robustness to the execution speed of actions.

Second, the extension to complex behaviors, like activities, became possible, by segmenting the motion into short motion units and considering both human movement and depth appearance to characterize human-object interactions. Finally, the sequence of temporal segments is modeled through a Dynamic Naive Bayesian Classifier. Extensive experiments carried out on several public datasets evaluate the potential of the proposed approach in different contexts, including action recognition and online activity detection and recognition.

Despite its usefulness in describing the depth appearance description around hand joints, its effectiveness remains limited in a complex scenario of human-object interactions. While it allows us to differentiate similar activities in terms of human motion, such method sills insufficient to interpret fine hand gestures, which is a critical problem for behaviour understanding. Among human body parts, hands are the most effective and intuitive interaction tools in Human-Computer Interaction applications. Thus, hand gesture analysis and recognition present a crucial task to achieve a deeper understanding of the behavior.

Fine-grained hand gesture recognition Like action and activity recognition, hand gesture analysis has been widely investigated in the literature, especially from 2D videos captured with RGB cameras. There were, however, challenges confronted by these methods, such as the sensitivity to color and illumination changes, background clutter and occlusions. Afterwards, thanks to the recent release of inexpensive depth sensors, new opportunities for hand pose estimation and gesture recognition emerge. The area of hand gesture analysis covers hand pose estimation and gesture recognition. Hand pose estimation is considered to be more challenging than other human part estimation due to the small size of the hand, its greater complexity and its important self occlusions. Beside, the development of a precise hand gesture recognition system is also challenging. Different occurrences of the same gesture type contain high dissimilarities derived from ad-hoc, cultural

and/or individual factors in the style, the position and the speed of gestures. In addition, gestures with different meanings contain high similarities derived from the heterogeneity of possible gestures.

All the above considerations lead us to address in, **Chapter 6**, the problem of hand gesture recognition according to two distinct approaches: handcrafted and deep learning. Hence, we investigate in the first part the gesture recognition problem by employing geometric features derived from hand posture, represented as skeletal data, for heterogeneous and fine dynamic hand gestures. The hand pose, can be either captured directly by certain depth sensors, or extracted later from depth images. In the second part, we extend the study to online dynamic hand gestures taking over the whole pipeline of the recognition process, from hand pose estimation to the recognition process, using a deep learning approach. So as to face the main challenges, we propose to revisit the feature pipeline by combining the merits of geometric shape and dynamic appearance, both extracted from a Convolutional Neural Network model trained for hand pose estimation problem. Transfer learning strategy has been employed in our approach to transfer the knowledge of a CNN model, trained using a large hand pose estimation dataset, to extract relevant features describe the gesture. The use of the transfer learning anable us to outperform state-of-the-art deep learning approaches using less than half of the number of parameters of the baseline model. However, we limited our experiments for only two hand gesture datasets simulating human-computer interface based on hand gestures acquired in online scenario.

CHAPTER 3

3D Human Motion Retrieval
Static Poses and Motion Shape Analysis

This chapter presents our contributions on shape representation and similarity in 3D human video sequences. These contributions originate from the work done by Rim Slama during her Ph.D thesis [38]. The chapter is organized as follows. After a description of the context of this work as well as the state-of-the-art methods of the domain in Section 3.1, Section 3.2 presents our 3D shape representation by extremal curve extraction. Section 3.3 describes the pose modeling in shape space and the elastic metric used for curve comparison. In section 3.4, we discuss the evaluation of our framework in terms of shape similarity, video segmentation and retrieval. Finally, Section 3.5 gives conclusion.

The contribution presented in this chapter were published in the journal paper [J3] and conference papers [C8, C9], and from where some parts of this chapter are extracted.

3.1 Context

Unlike the analysis of human body in 2D video, human body analysis in 3D video is still a little explored field. Since significant progress in multiple view reconstruction techniques has been made [183, 199], 3D video sequences of human motion are more and more available. However, the need for handling and processing such data led to several approaches using temporal shape representation and matching. In such videos, each frame is a mesh approximation of the body surface shape often generated independently regardless of its neighboring frames. Most work on 3D video have been mainly focused on performance, quality improvements and compression methods [169, 183, 192].

The acquisition of long sequences may produce massive quantity of data which necessitates efficient schemes for navigating, browsing and people and motion searching. Thus, there is a real need to develop such a retrieval method to accelerate and facilitate browsing this data. There are several retrieval scenarios but the ones we are targeting here concerne: (1) human pose retrieval in several motions, which helps to analyze repetitions in the sequence, to take decisions about motion transition and to produce character animation. (2) human subsequence retrieval 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.

In this chapter, we consider the problem of 3D shape similarity in 3D video sequences of people motions. Existing approaches use traditional global descriptors of shape to define the shape similarity using L_2 like distance. However, such a coarse representation present limits for whole and/or body part pose similarity. Besides, they are not allowing doing statistics on human body pose representations. For these rasons, For these reasons, we are interested in pose descriptors which represent and compare the pose information, in high dimentionality, using a unified geometric framework providing several processing modules within a duality pose/motion approach.

We first focus on the analysis of human pose and we propose a novel 3D human curve-based shape descriptor called Extremal Human Curves (EHC). This descriptor, extracted on body surface, is based on extremal features and geodesics between them. Every 3D mesh is represented by a collection of these open curves. The mesh to mesh comparison is then performed in a Riemannian shape space using an elastic metric between each two correspondent human curves. At this level, our ultimate goal is to be able to perform reliable reduced representation based on geodesic curves for shape and pose similarity metric. Invariant to pose changes, our EHC descriptor allows pose (and motion) comparison of subjects regardless of translation, rotation and scaling. Such descriptor can be employed not only in pose retrieval for video annotation and concatenation but also in motion retrieval, clustering and activity analysis.

Second, we are interested in the task of video segmentation and comparison between motion segments for video retrieval. As a 3D video of human motion consists of a stream of 3D models, we assume that EHC features are extracted from all 3D shape frames of the sequence, which is further segmented. For direct comparison of video sequences, the motion segmentation can play an important role in the dynamic matching by segmenting automatically the continuous 3D video data into small units describing basic movements, called clips. For the segmentation of these units, an analysis of minima on motion vector is performed using the metric employed to compare EHC representations. Finally, the motion retrieval is achieved thanks to the dynamic time warping (DTW) algorithm in the feature vector space.

3.1.1 3D human body acquisition systems and datasets

First of all, we present some of the most known 3D acquisition systems and public 3D static and dynamic human body datasets before introducing our approaches. 3D scanners are generally used to acquire real 3D human models [224, 226]. 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. Other techniques are based on silhouette extraction [217] or multi-image photogrammetry [209]. Recently, it is increasingly popular to scan the 3D human body using single or multiple depth sensors like kinect as introduced in works of [101, 124]. 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 [227]). 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. 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 [169, 204]. 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 [128, 156, 158, 184]. Many 3D static and dynamic human body datasets are published, where the most know are presented in Table 3.1.

Data	Static/Dynamic	Real/Synthetic
Ceasar [228]	static	real
Pickup et al.[50]	static	Synthetic and real
Haster et al. [174]	static	real
Liu et al. [158]	static	real
Gkalelis et al. [175]	dynamic	real
Huang et al. [167]	dynamic	synthetic
Vlasic et al. [180]	dynamic	real
Starck et al. [186]	dynamic	real
4dr [225]	dynamic	real

Table 3.1: Summary of datasets containing 3D human body in static poses and also in motion.

3.1.2 Related work

3D shape representation and similarity have been addressed in various research domains, such as computer vision, computer graphics, and for various applications, such as 3D object recognition, classification, retrieval. We address below, the most relevant works related to our approach, which only utilize the full-reconstructed 3D data for shape similarity in 3D human video. The most known

Static descriptors include: spin images, spherical harmonics, shape context and shape distribution. Spin image descriptor is proposed by Johnson et al. [212] to encode the density of mesh vertices into 2D histogram. Osada et al. [206] use a Shape Distribution, by computing the distance between random points on the surface. Ankerst et al. [213] 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. [173]. A similar representation to the Shape Histogram is presented by Kortgen et al. [202] as 3D extended shape context. Kazhdan et al. [203] 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. [167]. Other works on the 3D shape similarity 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 [123] or closed surface-based curves [136, 143, 187]. The extraction and the matching of these features have been widely investigated using different scalar functions from geodesic distances to heat-kernel [162, 172]. Tabia et al. [143] propose to extract arbitrarily closed curves amounting from feature points and use a geodesic distance between curves for 3D object classification. Elkhoury et al. [136] extract the same closed curves but they use heat-kernel distance in the 3D object retrieval process.

Temporal descriptors are presented in several works as an extension of static descriptors to temporal ones for frame retrieval in a 3D human video, using time filtering and shape flows obtained via invariant-rotation shape histograms [167]. Such approaches usually do not capture any geometrical information about the 3D human body pose and joint positions/orientations. This prevents

using them in certain applications that require accurate estimation of the pose (and the joints in some cases) of the body parts. The temporal similarity in 3D video is addressed also in the case of skeletal motion and is evaluated from difference in joint angle or position together with velocity and acceleration [208]. Huang et al. [166] demonstrate that skeleton-based Reeb-Graph descriptor has a good performance in the task of finding similar poses of the same person in 3D video. Shape similarity is also used for solving the problem of video retrieval by matching frames and comparing correspondent ones using a specified metric. In Yamasaki et al. [185], the modified shape distribution histogram is employed as feature representation of 3D models. The similar motion retrieval is realized by Dynamic Programming matching using the feature vectors and Euclidean distance. The Dynamic Time Warping algorithm (DTW), based on Dynamic Programming and some restrictions, was also widely used to resolve the 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, by a frame descriptor and the temporal alignment using DTW, many authors succeed to perform action recognition or sequence matching for indexing [116, 125, 141]. Recently, Tung et al. [123] 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.

From the above review, we can identify certain issues may be considered in our approach. The use of global description of the model ignores the local details. The aspect of motion is usually incorporated by time convolution of the distance metric itself computed from static poses. We are convinced of the extremities feature points as they are used in many state-of-the-art algorithms as an important compact semantic representation of human posture. Finally, the shape analysis of curves extracted from human body mesh may enable us to represent the shape variations. Such representative curves of the body surface may provide an efficient and a compact representation of human shape for the similarity task.

3.2 Principle of Extremal Curve

Our strategy consists of describing the body shape as a skeleton based shape representation, extracted on the surface of the mesh by connecting features located on the extremities of the body. This allows us to analyze pose variation with elastic deformation of the body, using representative curves on the surface.

3.2.1 Feature point detection

The points of the surface located at the extremity of its prominent components are well used in many applications, including deformation transfer, mesh retrieval, texture mapping and segmentation. Our strategy consists in using such feature points to represent human pose descriptor based on curves connecting each two extremities. Many existing approaches have been proposed to extract feature points. Some of them select vertices as feature points [207], where Gaussian curvature exceeds a given threshold. However, this method can miss some feature points because of the threshold parameter and cannot resolve extraction on constant curvature areas. Katz et al. [198] develop an algorithm based on multidimensional scaling, in quadratic execution complexity. Tierny et al. [191] proposed an approach based on geodesic distance evaluation to detect body extremity

points. This approach is invariant to geometrical transformations and model pose, and its process can be simply 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. [211]). 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 3.1. 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 3.2 shows different persons from three different datasets where feature extraction is stable despite change in shape, pose and clothing for each actor.

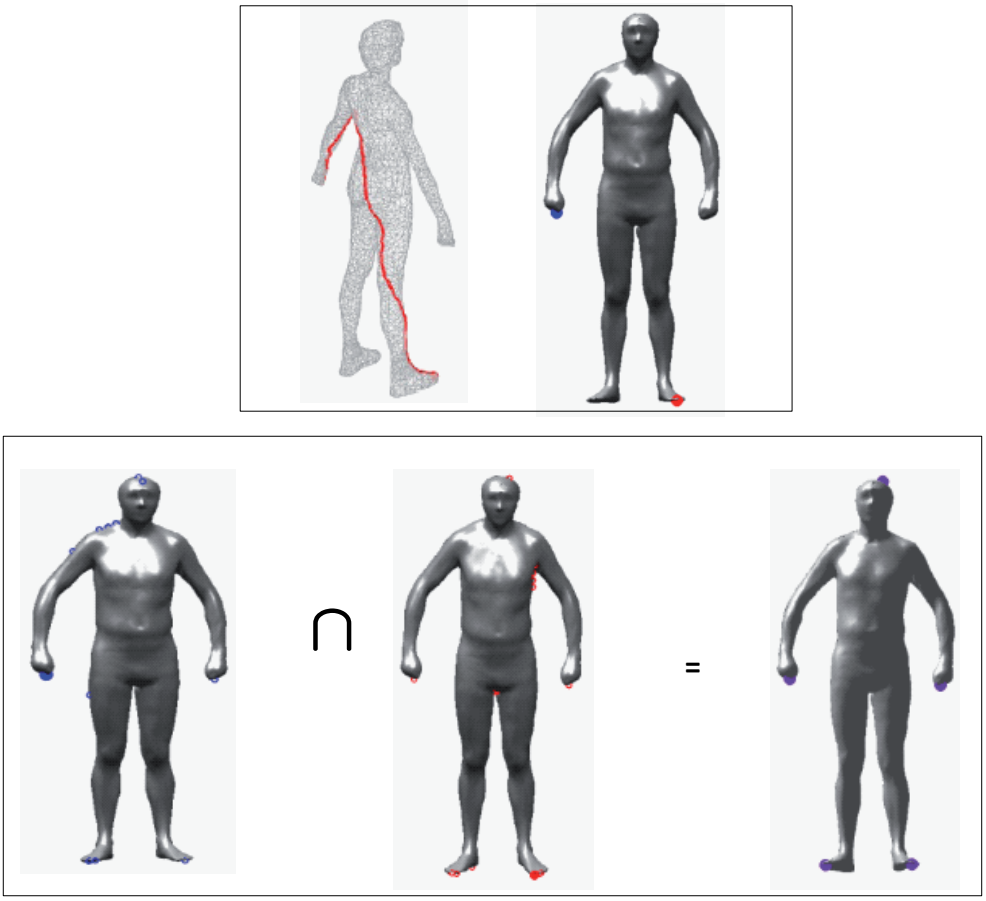


Figure 3.1: Extraction process of 3D human body extremity points. (top) The two distant vertices on the surface of the human body. (bottom) The set of local extrema and the result of their intersection.

We opted to use this extremity point process for our body shape representation approach.

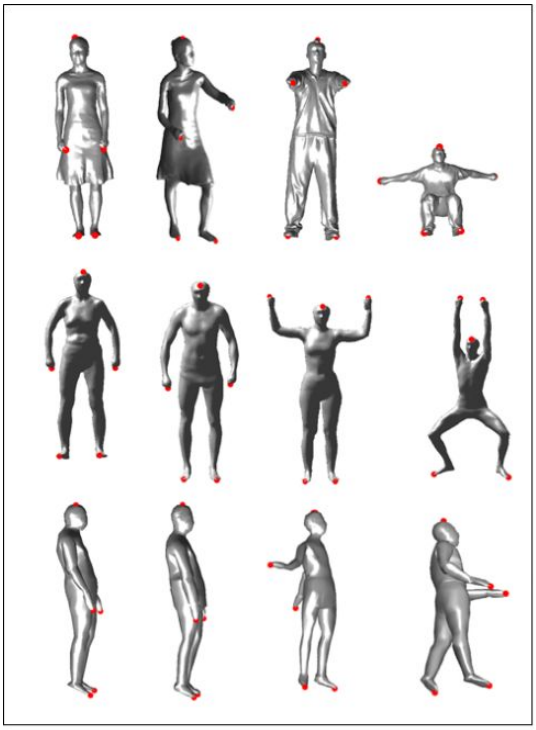


Figure 3.2: Extremity points extracted on different human body subjects in different poses.

3.2.2 Collection of extremal curve

First of all, let describe our curve extraction process allowing to extract a set of feature points on the body $E = \{e_1, e_2, e_3, e_4, e_5\}$ from For a body surface M. 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} between e_i and e_j. We repeat this step to extract ten extremal curves from the body surface so that we do all possible paths between elements of E. The body pose can be approximated by using these extremal curves $M \sim \bigcup \beta_{ij}$, as shown in the top of Figure 3.3. These curves can be categorize into 5 categories corresponding to the body part in question (Figure 3.3 bottom).

Note that modeling objects with curves is carried out for several applications; Abdelkader et al. [160] use closed curves extracted from human silhouettes to characterize human poses in 2D videos for action recognition. Drira et al. [100] use open curves extracted from nose tip and face surface as a surface parametrization for 3D face recognition.

In our approach, we have chosen to model the body pose by a collection of curves as they offer a reduced representation of the mesh surface, and allow later to analyze the shape variation using Riemannian geometry of shape space introduced by Joshi et al. [189].

3.3 Shape analysis for Human Pose Modeling

Shape analysis has been widely investigated in computer vision for different application domains, like object recognition in a scene, evolution of illness in medical imaging an facial recognition. We focus in this chapter on the analysis of human pose characterized by a set of the above presented curves. To achieve this analysis, we believe that the human shape cue is very important due to the geometric nature of human pose and motion.

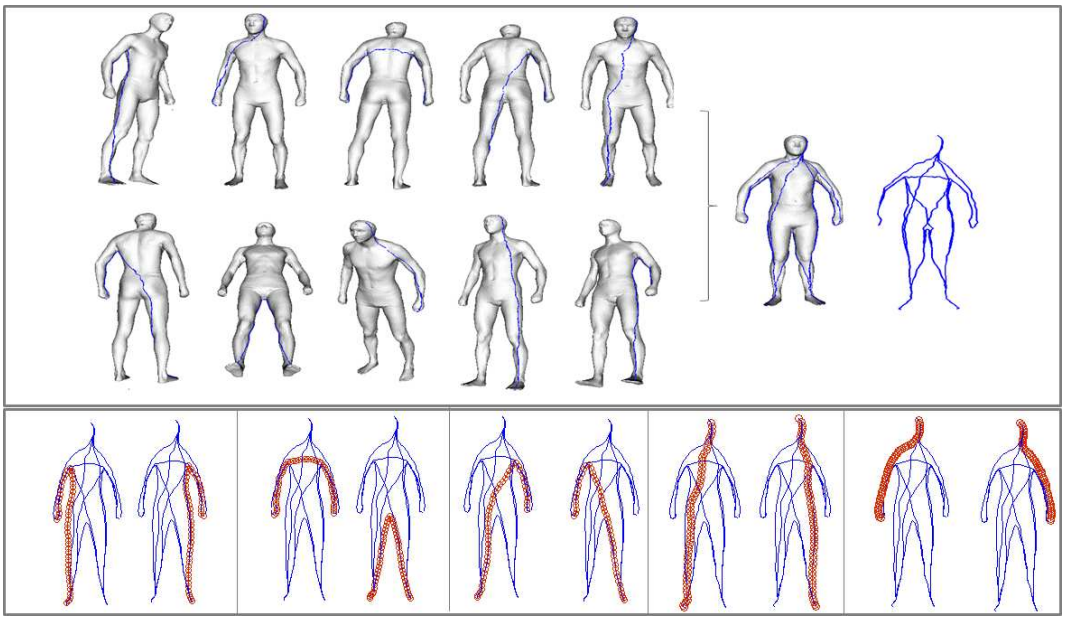


Figure 3.3: Body representation as a collection of extremal curves.

Many existing approaches have been developed in past years to analyze shapes of 2-D curves, like those 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. [193], Klassen et al. [200] and Yezzi et al. [196] show the efficiency of this approach for 2-D curves. Recently, a generalization of this work to the case of curves defined in $\mathbb{R}^n$ is proposed by Joshi et al. [189]. We adopt this work to our our collection of curves defined in $\mathbb{R}^3$.

3.3.1 A short note on Riemannian shape space framework

To analyze the shape of human body, first in static posture and later in motion, while facing the constraints related to geometrical elastic transformation, we employ a Riemannian Shape Analysis framework [189]. Such framework allows us to capture and interpret shapes of curves in $\mathbb{R}^3$ within a Riemannian manifold and provides an elastic metric to measure the similarity between such shapes. In addition, using such manifold offers a wide variety of statistical and modeling tools that can be used to improve and deepen the analysis of human pose deformation and shape similarity. Let us now giving some brief comments on the basic importance of the notion of the Shape Analysis framework [189]. For more detail, reader is referred to the Rim Slama's Ph.D thesis [38], from where this short note is extracted.

Human shape representation. 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 $\beta : I \rightarrow \mathbb{R}^3$, 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.[189] that captures the shape of β and is particularly convenient for shape analysis. The classical elastic metric for comparing shapes of curves becomes the 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 ℓ_2-metric. Hence, the SRV representation finds its potential for its ability for elastic matching. Actually, under 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):

$$\mathcal{C} = \{q : I \to \mathbb{R}^3, \|q\| = 1\} \;\subset\; \ell_2(I, \mathbb{R}^3) \,. \tag{3.4}$$

where using ℓ_2-metric on its tangent spaces, $\mathcal{C}$ becomes a Riemannian manifold.

Since the elements of $\mathcal{C}$ have a unit ℓ_2 norm, $\mathcal{C}$ is a hypersphere in the Hilbert space $\ell_2(I, \mathbb{R}^3)$. 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 [144].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 opti-

mization 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 re-parameterization 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:

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

,

where θ is now $d_s([q_1], [q_2])$.

In this way, the distance between the shape of two curves in $\mathbb{R}^3$ is invariant to their translation, scale, rotation and re-parametrization. Figure 3.4 illustrates the geodesic path on the open curve shape space between two given extremal curves.

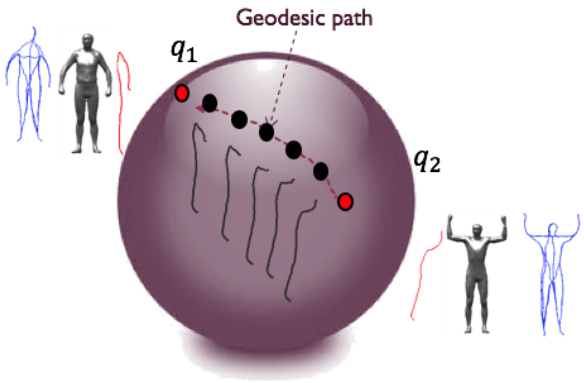


Figure 3.4: Geodesic path between extremal human curves of neutral pose with raised hands.

3.3.2 Similarity measure and average pose

Similarity measure can be defined using he elastic metric applied on extremal curve-based descriptors. 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 representing the mesh and d_s is the elastic distance of Equation 3.6. If we average the arithmetic distances between all corresponding curves, we can captures the

similarity between their postures.

Shape geometric mean of a set of data sufficiently close to each other, in Riemannian geometry, can be computed by minimization of a cost function computed from the data. A common algorithm for such mean computation on Riemannian manifold is called the Karcher mean or Riemannian center of mass [221] and employs as cost function the sum of squared geodesic distances between a given data and all other data. Here we propose to use this algorithm to identify a mean shape. For a given set of shapes $q_1, \ldots, q_n \in \mathcal{S}$, their Riemannian center of mass can be defined as:

$$\mu = \arg\min_{[q]} \sum_{i=1}^{n} d_s([q], [q_i])^2 \ . \tag{3.9}$$

So as to minimize such cost function, the algorithm employs both the exponential map and logarithm map operators in an iterative process to update the Riemannian center of mass until convergence. More specifically, at each iteration i, shapes are first projected into the tangent space at the current mean shape μ_i using the inverse exponential map. Based on the resulted velocity vectors, the average direction is computed and the mean shape is slightly moved in that direction. The exponential map is finally used to transfer the updated mean shape μ_{i+1} back on the shape space.

Average of human poses can be computed, using Riemannian center of mass [221], between different poses to represent the intermediate pose, or between similar poses done by several actors to represent a template of similar poses.

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 3.5. In the bottom of this figure, we show the average EHC representation computed using the Karcher mean.

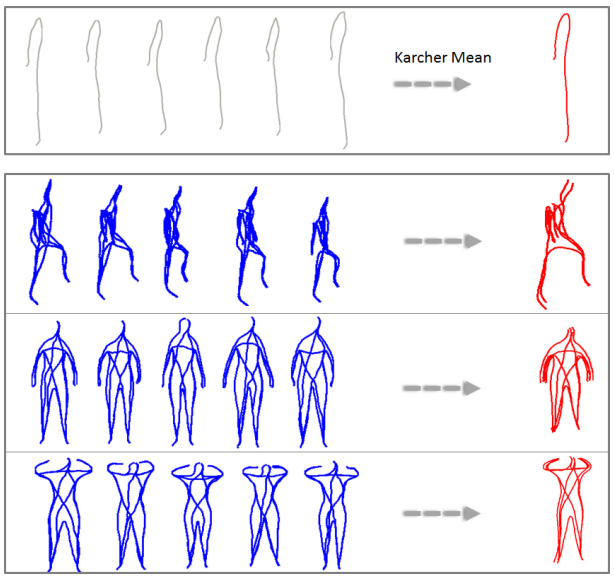


Figure 3.5: Examples of Karcher mean computation as mean curve for six extremal human curves: curve connecting hand and foot from the same side.

3.4 Application to 3D Motion Sequences

Let us present how such a reduced representation of 3D human posture by based-geodesic curves can be used for a reliable shape and pose similarity metric, which can be employed for several purposes related to 3D video analysis. To show the practical relevance of our method, we 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. We first evaluate our descriptor for shape similarity application over public static shape database [174] and evaluate the results against Spherical Harmonic descriptor [203]. 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 [167] and real [180, 186] video databases. We evaluate this later against Temporal Shape Histogram [167], Multi-resolution Reeb-graph [166] and other classic shape descriptors, using provided Ground Truth.

3.4.1 Static and temporal shape retrieval

In order to extract our curves , we need to identify the feature end-points as head, right/left hand and right/left foot, which is not affordable in practice. We can start from observations: First, the 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. Rest to distinguish between left and right, which can be done if we consider a prior knowledge on the direction of the posture of the human body for static pose and in the starting frame for video sequence.

Static shape similarity. To evaluate the effectiveness of our EHC descriptor for static shape similarity, we performed several tests on a the challenging statistical shape database [174]. Captured with a 3D laser scanner, this database contains more than hundred subjects doing more than thirty different poses. We perform our descriptor on a subset of more than 300 shape models obtained from 144 male and female subjects aged between 17 and 61 years. Only 18 consistent poses (p0-p13, p16, p28, p29, p32) are used in this experiments and some of them are illustrated in Figure 3.6.

For the purpose of static and video retrieval evaluations, we use Recall/Precision plot in addition to the four statistics indicating the percentage of the top K matches that belong to the same pose class as the query pose: nearest neighbor statistic (NN), first tier statistic (FT), second tier statistic (ST) and E-Measures. More detail about their definitions can be found in [38].

Additionally, 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. 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 $N_b = 16$) [213]. 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 selected curves outperforms SH and EHC using the 10 curves to retrieve models with the same pose.

The self similarity matrix obtained from the mean elastic distance of the five selected curves is shown in the Figure 3.8.

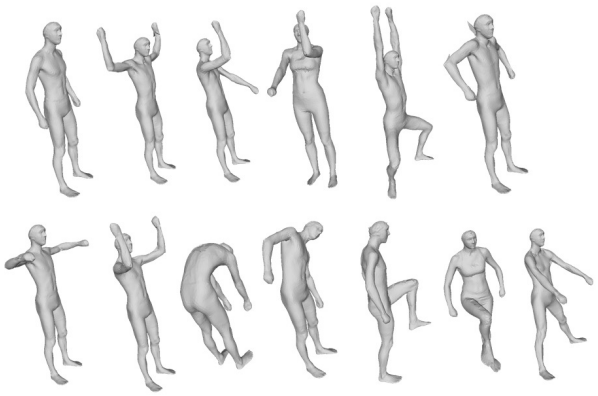


Figure 3.6: Example of body poses in the static human dataset [174].

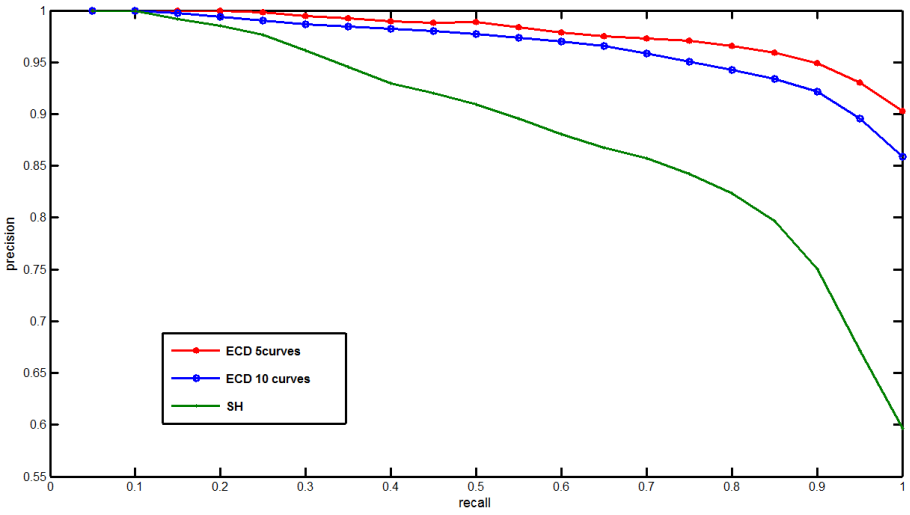


Figure 3.7: Precision-recall plot for pose-based retrieval.

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

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.

Temporal shape similarity. We used two datasets for the evaluation of the proposed temporal shape descriptor: the synthetic 3D video dataset with their ground-truth annotations proposed by Huang et al. [167] and the real captured 3D video dataset [180]. Let us presented how to compute a temporal ground-truth similarity between each two surfaces. Having two human body mesh 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

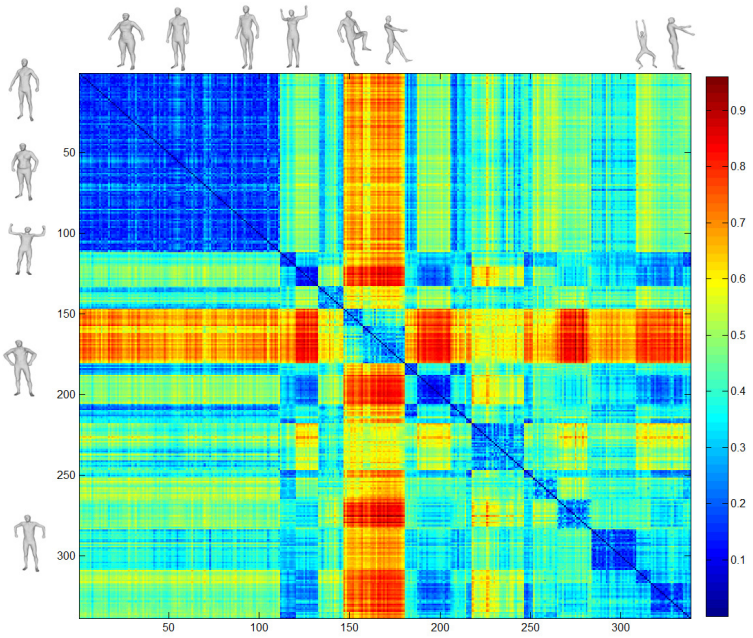


Figure 3.8: Confusion similarity matrix of pose dissimilarity between models of a 3D humans in different poses.

a shape similarity C_p and a temporal similarity C_v as follows:

$$\begin{aligned} C_T(X,Y) &= (1-\alpha)C_p(x_i,y_j) + \alpha C_v(x_i,y_j) \\ C_p(X,Y) &= \tfrac{1}{N}\textstyle\sum_{k=1}^{N} d(x_i,y_j) \\ C_v(X,Y) &= \tfrac{1}{N}\textstyle\sum_{k=1}^{N} d(\dot{x}_i,\dot{y}_j) \end{aligned} \tag{3.10}$$

where d is an Euclidean distance, $\dot{x}_i$ and $\dot{y}_j$ are the derivation of x and y between next and current frame. 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. [167]. Finally, recognition performance is evaluated using the ROC curves, and 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 descriptor, static and temporal descriptors are shown in Figure 3.9. This figure illustrates also the effect of time filtering with increasing temporal window size for EHC descriptors on a periodic walking motion.

A comparison is made between our Temporal Extremal Human Curve (TEHC) and several descriptors from the state-of-the-art, like 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. [167]) and Reeb-Graph as skeleton based shape descriptors (aMRG) (Tung et al. [197]). To measure the performance of the similarity metric results, we plot the ROC curves obtained from our EHC descriptor (see3.10). These results are compared with ROC curves obtained by all state-of-the-art descriptors presented in figure 6 at [167] where our descriptor is among the more three efficient descriptors.

We analyze these results from various points of view, including the role of the time-filer, the relative performance of the descriptors and the relative performance per action. First, we notice that recognition performance of EHC increases with the increase of the window size of time-filter

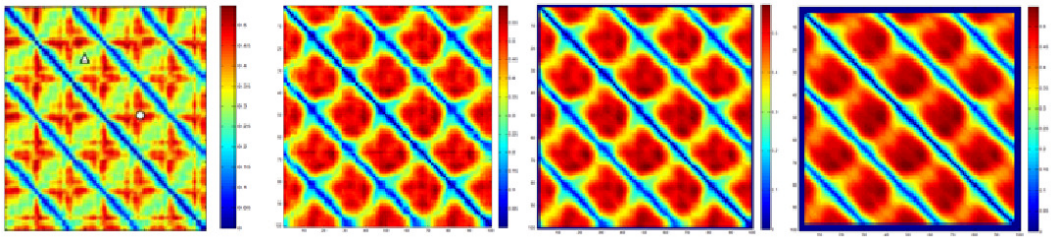


Figure 3.9: Similarity measure for "Fast Walk" motion in a straight line compared with itself. Coldest colors indicate most similar frames. 1^{st} matrix: temporal Ground-Truth (TGT). 2^{nd}, 3^{rd} and 4^{th} matrix: self-similarity matrix computed with Temporal EHC with window size 3, 5 and 7 respectively.

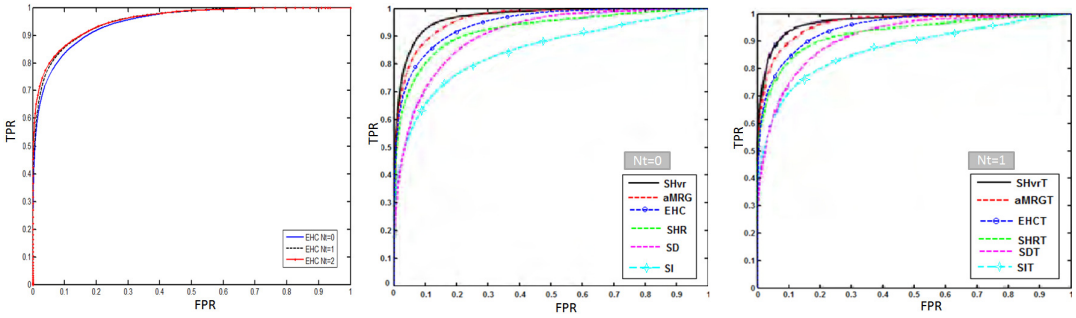


Figure 3.10: Evaluation of ROC curve for static and time-filtered descriptors on self-similarity across 14 people doing 28 motions. From right to left: ROC curves obtained by our TEHC descriptor with three different values of windows size Nt, ROC curve obtained by our EHC descriptor compared to different algorithms and ROC curves obtained with Nt= 1.

like any other descriptor. In fact, time-filter reduces the minima in the anti-diagonal direction, resulting from motion in the static descriptor. In addition, the MDS is insensitive to mesh deformation which maintains the geodesic distance and shows lower recognition performances. Our descriptor outperforms MDST and other classic shape descriptors (SI, SHRT, SD) and shows competitive results with (SHvrG/SHvrS) and aMRG. Third, multiframe shape-flow matching required in SHvrG allows the descriptor to be more robust but the computational cost will increase by the size of selected time window. Our descriptor 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. Finally, the result analysis for each action shows that TEHC gives a smooth rates that are stable and not affected by the complexity of the motion, like rock and roll, vogue dance, faint and shot arm, as illustrated in Figure 3.11

We apply the time filtering Extremal Human Curves descriptor to real captured 3D video sequences of people. Inter-person similarity across two people in a walking motion with an example similarity curve are shown in Fig. 3.12. Our temporal similarity measure identifies correctly similar frames across different people. These similar frames are located in the minima of the similarity curve.

We finally applied the time filtering Extremal Human Curves descriptor on real captured 3D video sequences of people. The first sequence is extracted from the dataset [180]. The second one is extracted from real data reconstructed by multiple camera video [186]. Inter-person similarity

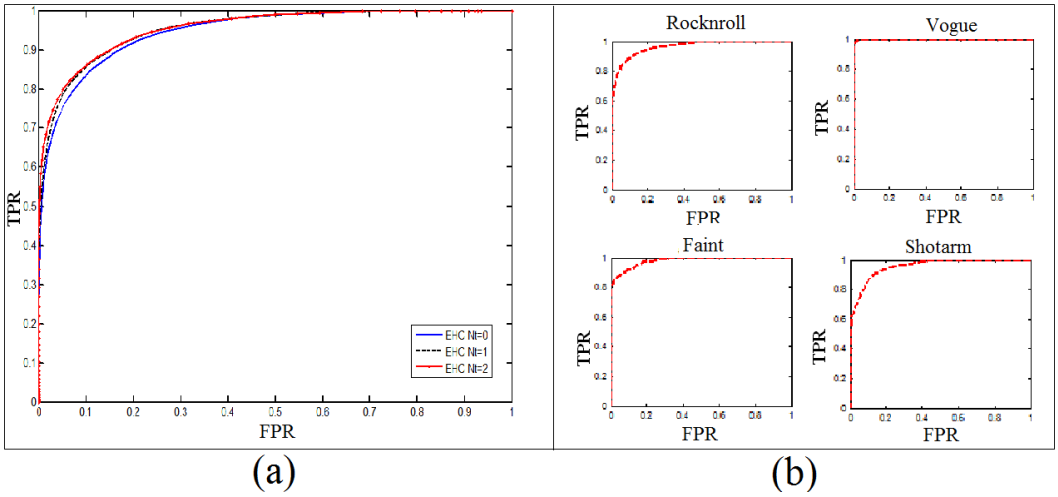


Figure 3.11: ROC curves (a) for static ($N_t = 0$) and time-filtered EHC decriptor ($N_t = 1$, $N_t = 2$) on self-similarity across 14 people doing 28 motions, (b) ROC performance for 4 complex motions obtained by EHC, for fixed window size 5 ($N_t = 2$) against Temporal Ground Truth.

across two people in a walking motion with an example similarity curve are shown in Figure 3.12 (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 3.12 (b), our algorithm succeed to identify correctly the frame in the sequence similar to the query.

3.4.2 Motion segmentation for 3D video analysis

Now we have a reliable representation of 3D human mesh in a static pose based on EHC descriptor, we can use it to compute a similarity between sequences of video. To do this, we match all pairwise correspondent EHCs using the geodesic distance in the shape space. However, a human motion sequence can be composed of several distinct actions, and each one can be repeated several times. Dividing continuous sequence into separate motion clips using our EHC can play an important role in the video analysis and matching.

In order to split automatically the continuous sequence into segments which exhibit basic movements, called clips, we use the notion of human pose representation by our EHC descriptor and the elastic distance in the shape space manifold. 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 this reason, we believe that motion speed can be an important factor [220]. 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. 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 neighbourhood. For this reason, we fix a size of window to test the efficiency of the local minimum in this condition. The speed variation can be computed using an elastic distance between each two successive EHC in the sequence, and then represented in a vector of speed for a further smoothing.

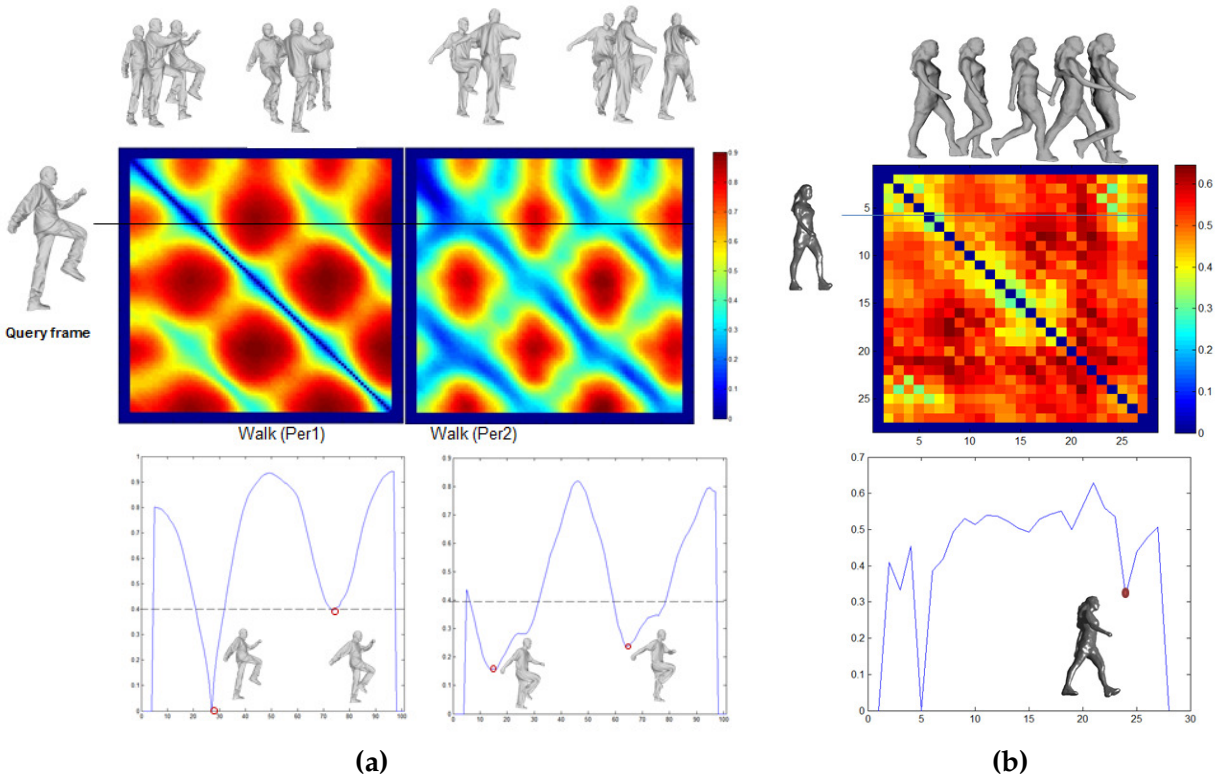


Figure 3.12: Inter-person similarity measure for real sequences. Similarity matrix, curve and example frames for (a) Walk motion across two actors [180] (b)walk motion for Roxanne [186] Game Character Walk .

Once the motion clips are obtained, we need to evaluate the segmentation process. To do this, we conduct our evaluating on the synthetic dataset [167]. In particular, 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 later retrieval experiment. To validate the segmentation step, we segment all these 3D video sequences with the proposed approach and then compare results to provided manual segmented ground-truth. Lets now present some results of motion segmentation experiments.

Practical case of motion segmentation. 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 3.13 (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 3.13 (bottom). Break points are more precise and delimits significant clips corresponding to step change in the video sequence. In order to evaluate its efficiency, we apply our segmentation method on the whole dataset [180]

and then compare the results to a manual segmentation of the base done carefully . The segmentation of the dataset 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, the a total of 144 clips have been obtained by the segmentation of the 14 motions taken from the dataset [167] performed by two actors.

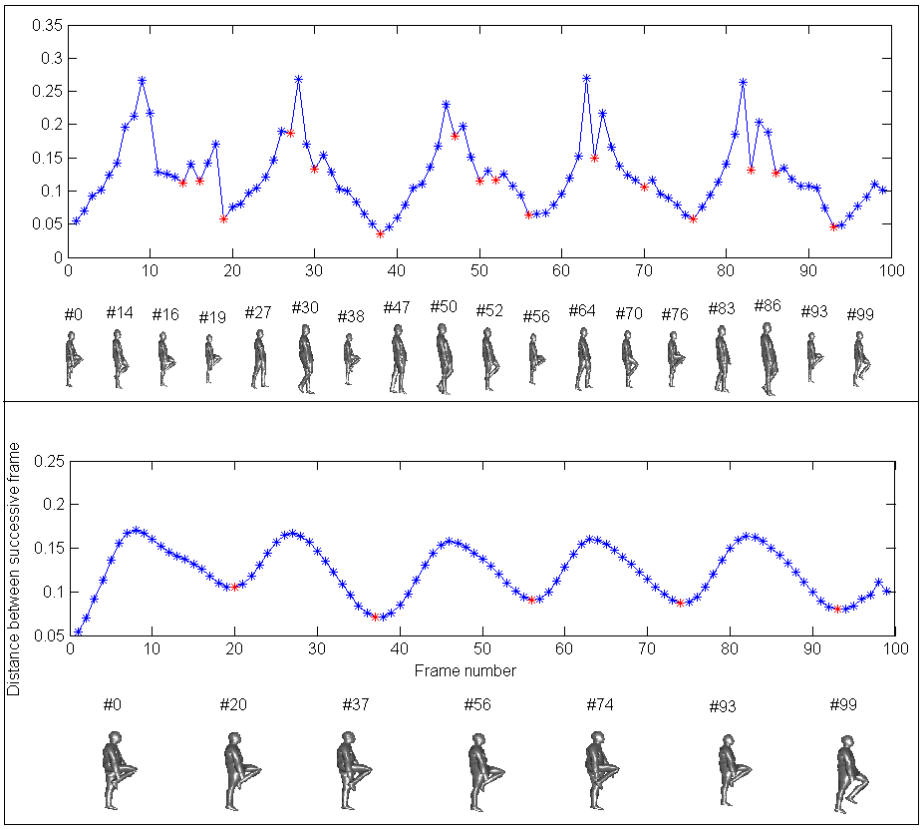


Figure 3.13: Speed curve smoothing.

Figure 3.14 shows some results of motion segmentation on a "slow walk" and a "fast walk" motions. Although the walk speed increases, the motion 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 3.14 (bottom). Thanks to the selection of local minima in a precise neighborhood, only significant break points are detected.

Analysis of motion retrieval result. 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 3.15. 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 3.15). The matching 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.

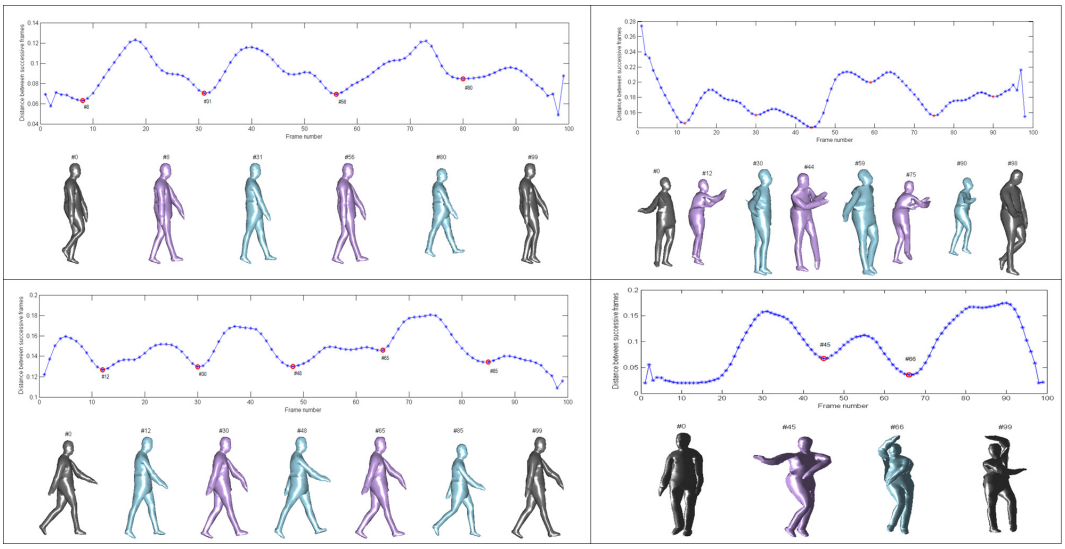


Figure 3.14: Various motion segmentation. From right top to left bottom, motions are: slow walk, Rockn'roll dance, fast walk, vogue dance.

Besides, our approach succeed to retrieve clips within motions done in different ways. For example, the walk circle clips can be matched with the clips of slow walk motion done in a straight line (see Figure 3.15). 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.

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 [180]. Self similarity example with an actor in a walking motion (walking in circular way) and its similarity curve are shown in Figure 3.16. For the query clip presented at the right of the figure, retrieved clips are found correctly in the sequence when the actor is turning.

3.4.3 Video summarization and retrieval.

In order to represent compactly a video sequence, one of the most important factor is to exploit the redundancy of information over time. The challenging task here is to find geometric relations between consecutive data stream elements, as this redundancy should be extracted from motion and not from frames separately. 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 process, like video clustering, summarization and retrieval.

Clip matching. The problem of clip matching is interesting in any time-series retrieval task where a distance metric is used to look for, in a database, the sequences whose distance to the query is below a threshold value. We firstly need to encode the motions in a specific representation that we

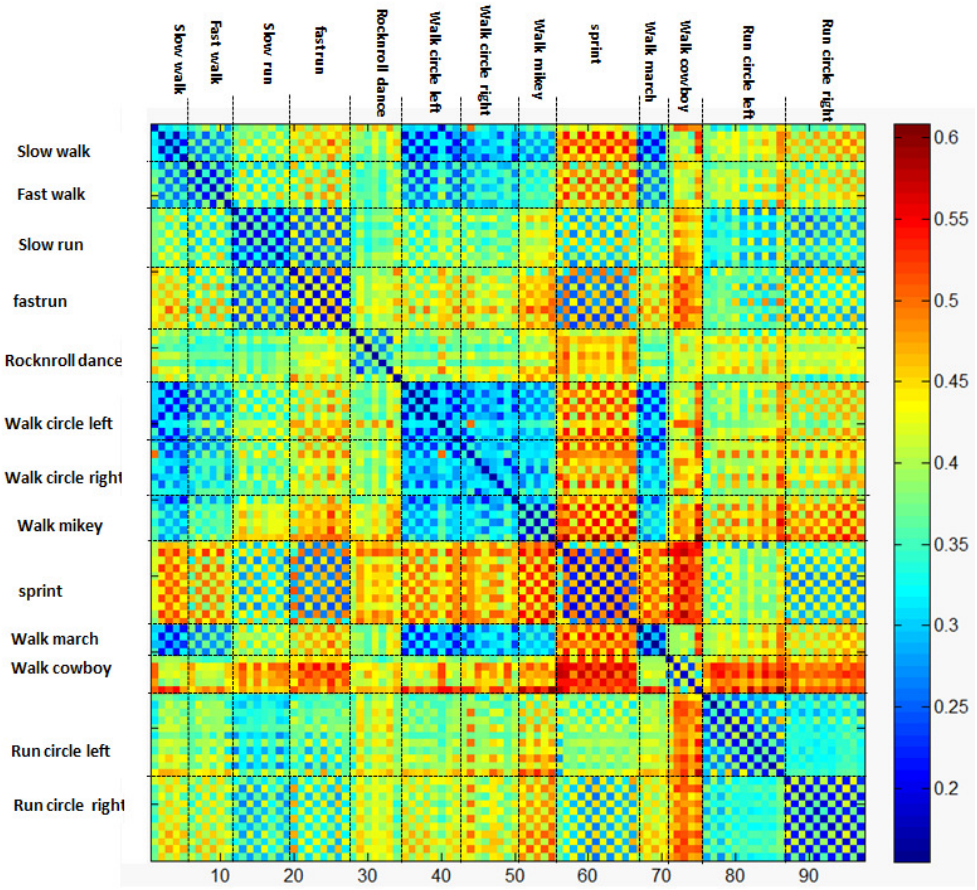


Figure 3.15: Similarity matrix evaluation between clips.

can compare regardless to certain variations. Indeed, motion clips are considered similar even if there are changes in the shape of the actor and the speed of the action execution. In particular, we represent each clip 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 3.17 (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 (see illustarion in Figure 3.17 (bottom)).

The popular Dynamic Time Warping (DTW) is an appropriated technique for this kind of temporal variation problems. Especially, we use DTW algorithm proposed by (Giorgino et al. [176]) 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 the shape space manifold. The global accumulated costs along the path define a global distance between the query clip and the motion segments found in the database.

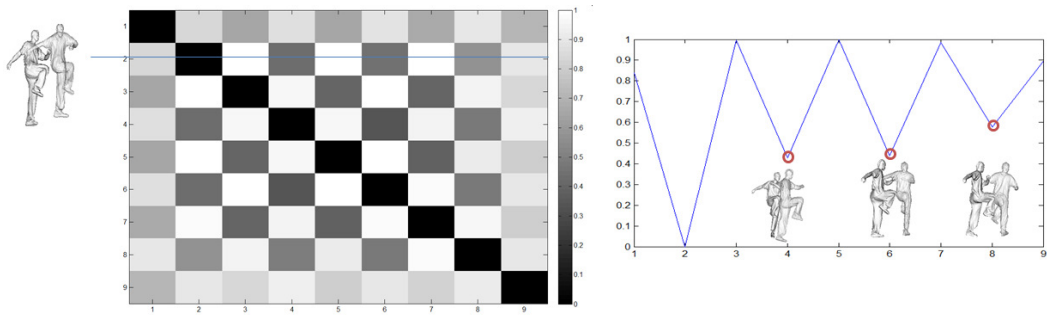


Figure 3.16: Experimental results for 3D video retrieval using motion of "walk in circle".

Average clip. Similarly to the average pose process, we introduced the notion of average clip using both Karcher mean and DTW algorithms. Thus, 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's see how can we do it. 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 3.1, 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 as points on the Riemannian manifold.

Algorithm 3.1 Computing the mean of a set of trajectories

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

Data clustering. Finally, since similarity between clips has now become possible, a clustering technique can be performed for efficient indexing, searching, and visualization. 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 3.2.

Algorithm 3.2 Data clustering

Require: $V\{e_i\}_{i=1...k}$;
At time $C_0 = \varnothing$; $R_0 = \{e_1, \ldots, e_k\}$;
  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

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 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

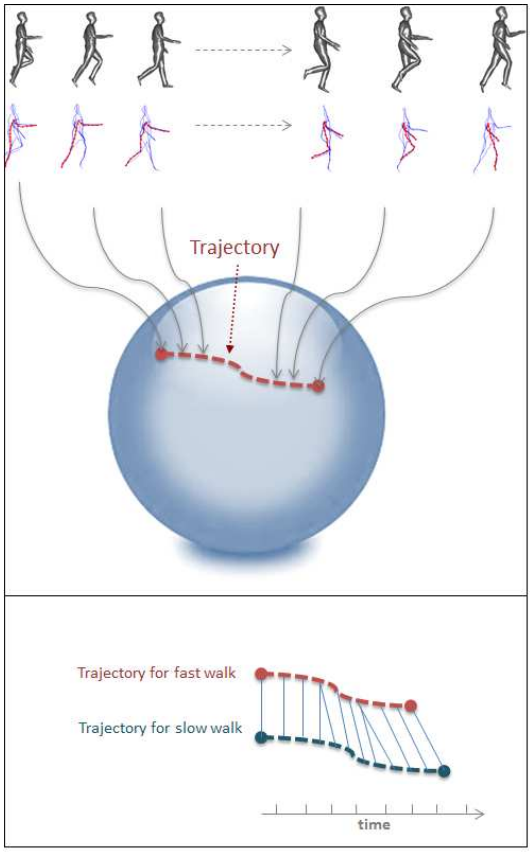


Figure 3.17: Graphical illustration of a sequence, obtained during a walking action, as trajectory on shape space manifold (top). Alignment process between trajectories of same curve index using DTW (bottom).

$l \leqslant k$). 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 [C9]. 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 sequence being a stream of clips resulting from the video segmentation approach and clusters here gather clips with similar repeated atomic actions. In this case: the feature vector used as abstraction for each clip is a trajectory on shape space of extremal human curves, and the similarity distance, used to compare clips, is based on the DTW algorithm.

Content-based summarization So that the different analysis tools of 3D human motion video have been developed, we can build a content-based summarization system. This system is consists of three phases: (1) the whole video is segmented and clustered into several clusters of clips. (2) only the most significant clip (the nearest one to all cluster elements) of each cluster is kept. (3) a subsequence is then constructed, from the starting video, where these 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.

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 [180]. For clip-based summarization experiment, the same 28 motions used for video segmentation and the retrieval have been used. The evaluation criterion of clustering method is based on the number of clusters found which should allow the identification of eventual redundant patterns.

The Figure 3.18 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 3.18.

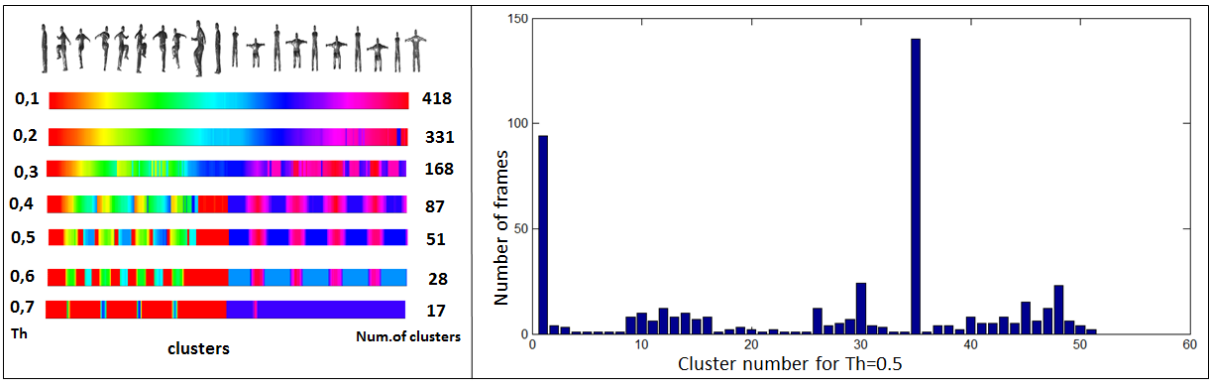


Figure 3.18: Frame clustering process with respect to threshold Th.

Pose-based clustering process can be improved by increasing the window size of the time filter

as shown in Figure 3.19.

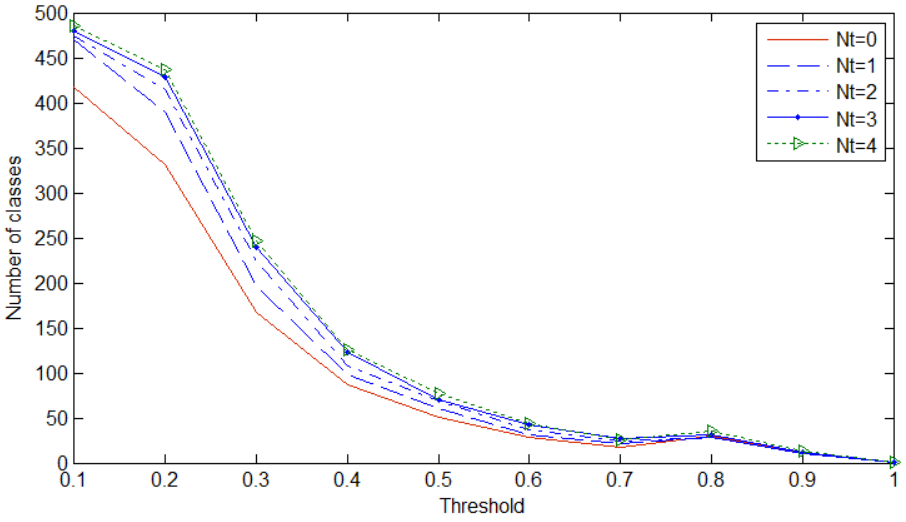


Figure 3.19: Frame clustering with respect to a threshold and with different window size.

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, we evaluate our clip-based clustering approach on the dataset [180], by applying the clustering on a set of 14 motions performed by two actors. 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 3.20). We notice that clips representing sprint or running steps are clustered together.

An hierarchical structure of video summarization process can be designed, 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. Figure 3.21 show the possess applied for the sequence of a real actor performing walking and squatting motion. 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.

Motion data retrieval. Using our data representation descriptor and related tools, we propose an hierarchical retrieval structure combining the clustering and the content-based retrieval process. 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 (as average pose) is obtained for each cluster by computing its Karcher Mean. 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.

As experimental test, we applied the hierarchical approach to the dataset in [174]. Each query model is compared to each one of the template models representing the clusters. Then, we generated a confusion matrix for all classes of pose using elastic measure values. Thanks to the provided ground truth, we evaluated the recognition performance by computing statistic retrieval measures.

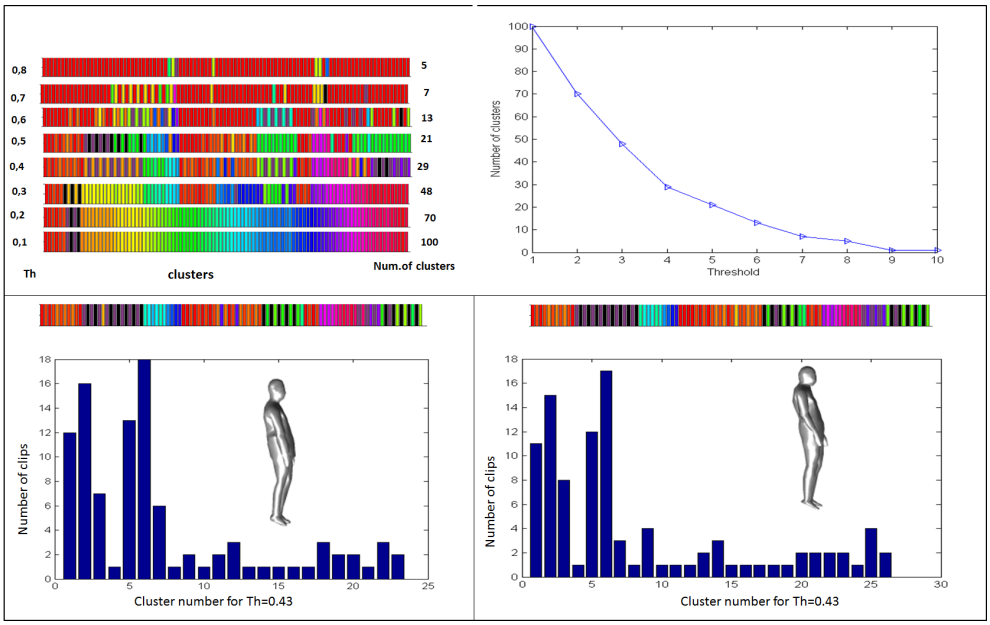


Figure 3.20: Clustering clips from a sequence of two actors performing 14 motions from the dataset (3) for a total of 1400 frames, with respect to Th. In second raw, the variation of clip number in each cluster is presented.

The matrix of comparison in the first level (model-template comparison), is shown in Figure 3.22.

The effectiveness of the summarization can be assessed by simply comparing this matrix to that already obtained for the same dataset without the use of summarization (Figure 3.8). The summarization process reduce the 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.

Furthermore, an accuracy of 90.24% is obtained in terms of pose categorization (classification), where the most significant confusion can be observed between the two classes ♯2 ♯16, both being represent people with hands outstretched.

Finally, within the hierarchical structure, the elements of cluster could be motion clips, where a video segmentation is firstly performed on the whole sequence. In this case, the template model is now computed as a "mean" for each cluster of clips thanks to its Karcher Mean. The retrieval system can then be viewed as above with hierarchical structure. To evaluate the performance of our approach in terms of clip retrieval, we conducted a similar experimentation 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 the provided ground truth. The matrix of comparison in the first level (model-template comparison) is shown in Figure 3.23.

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, 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.

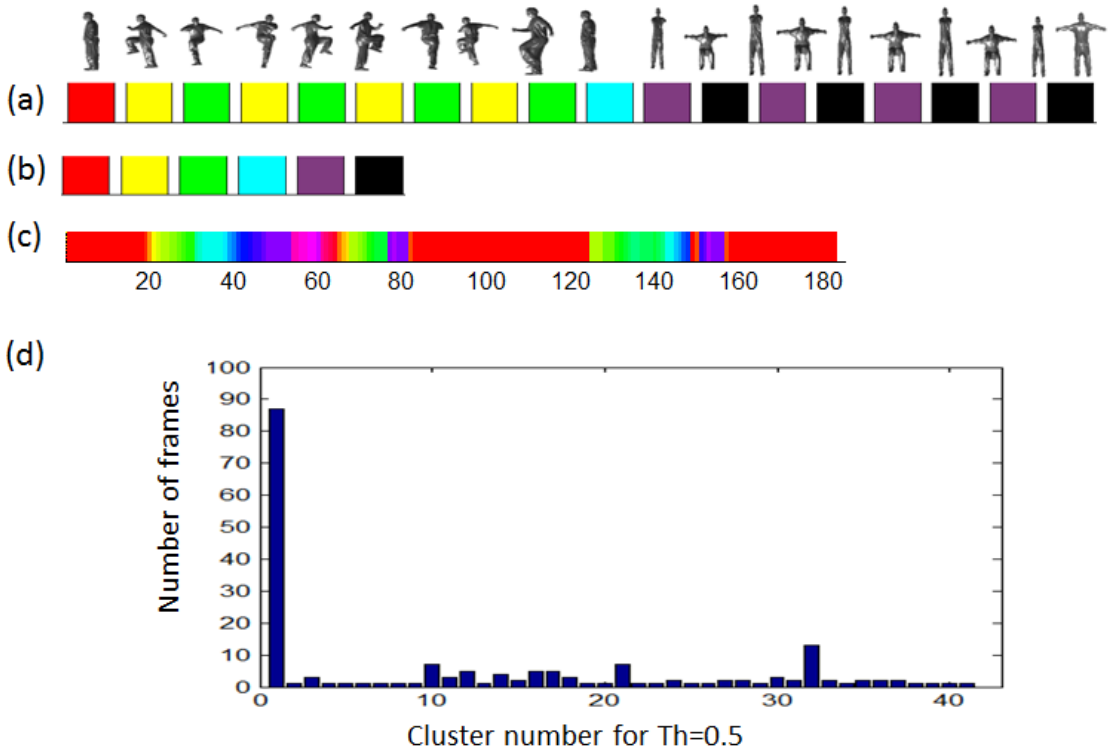


Figure 3.21: Summarization process: (a) for a sequence of 500 frames segmented into 18 clips, the clustering process returns 6 clusters of clips using $Th = 0.38$ (b) subsequence of clustered clips (180 frames) where each cluster is represented by only one clip chosen as the Karcher Mean clip of the cluster, (c) clustering of subsequence into 41 clusters of frames using $Th = 0.5$, (d) distribution of the number of frames in clusters.

3.5 Conclusion

This chapter summarizes our works on shape representation and similarity in 3D human video sequence, in which we proposed a unified framework in order to represent human body shape with a pose descriptor, as well as a sequence of frames with a specific representation. In this framework, we opted a skeletal representation of human shape based on extremal features and geodesics (in form of local open-3D-curves) between each pair of them. 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 pose of a 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. The degree of motion using feature vector, extracted from this descriptor, is then used for spliting continuous sequence into elementary motion segments called clips. Each clip describing an atomic movement is characterized by curve representation associated to human mesh. The open curves in 3D space are viewed as a point in the shape space of open curves and hence each clip is represented by a trajectory on this space. Similarity metric between each two clips is obtained by a classical Dynamic Time Warping technique to align different trajectories on the manifold.

The use of skeletal surface representation enabled us to obtain a a good performance in terms of shape similarity and motion retrieval show the potential of our approach. However, this representation has some limitations. First, it depends on the accuracy of extremities (head and limbs)

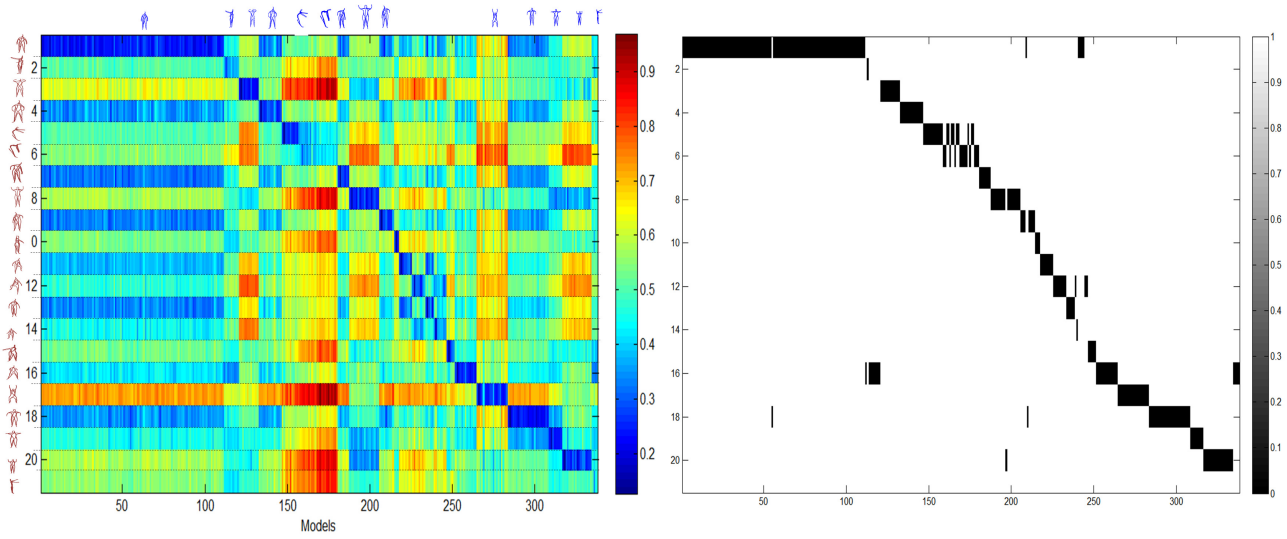


Figure 3.22: Similarity matrix and its binarization for template pose of each class against all models in the dataset.

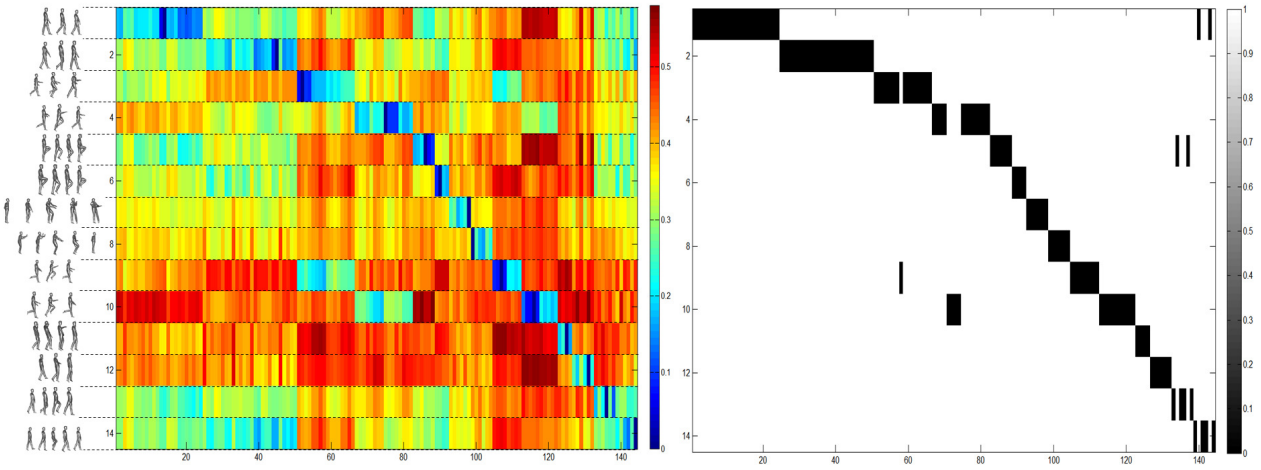


Figure 3.23: Similarity matrix and its binarization for template clip of each class against all clips in the dataset.

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. Second, we noted that our curve extraction can be sensitive to loose clothes, especially for a mesh of human body wearing a skirt. This problem will be even more critical if she wears a long skirt. Finally, a prior knowledge on the direction of the posture of the human body for the starting frame in video sequence is used to distinguish between left/right hand and foot. Other feature matching algorithms, like Heat Kernel Signature as proposed by Sun et al. [171] and Zheng et al. [67], could be used to correctly identify the right from the left side.

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 re-

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

CHAPTER 4

Human Action Recognition
Learning on the Grassmann Manifold

This chapter address the problem of modelling and analyzing of human motion by focusing on 3D body skeletons captured by a depth sensor. These contributions originate from the work done by Rim Slama during his Ph.D thesis [38]. It is organized as follows. After a description of the context of this work, as well as an overview of our approach in Section 4.1, Section 4.2 reviews related work from two points of view: manifold-based approaches and depth data representation. Section 4.3 gives a brief introduction of the Grassmann manifold. The two-fold learning process of our approach with corresponding experiments are discussed: when Truncated Wrapped Gaussian in Section 4.4 and when using Representative Tangent Vectors in Section 4.5, before concluding in Section 4.6.

The contribution presented in this chapter were published in the journal paper [J2] and conference/workshop paper [C7], and from where some parts of this chapter are extracted.

4.1 Context

Recognizing human actions in video sequences represent a task of interest for many multimedia applications, including entertainment, medicine, sport, video surveillance, human-machine interfaces and active assisted living. This wide spectrum of potential applications encouraged computer vision community to address the issue of human activity understanding from 2D videos taken from standard RGB cameras [106–108, 139, 159]. However, most of these methods suffer from some limitations, like the sensitivity to color and illumination changes, background clutter and occlusions. Since the recent release of RGB-D sensors, new opportunities have emerged in the field of human motion analysis and understanding. Hence, many research groups investigated data provided by such cameras in order to benefit from some advantages compared to RGB cameras [26, 36, 97, 149, 154]. Indeed, depth data allows a better understanding of the 3D structure of the scene and thus makes background subtraction and people detection easier. In addition, the technology behind such depth sensors provides robustness to light variations as well as the capability to work in complete darkness. Finally, the combination of such depth sensors and powerful pattern recognition algorithms [146] enables the representation of human pose at each frame as a set of 3D joints. In the past decades, human motion analysis from 3D data provided by motion capture systems has been widely investigated [41, 57, 201]. While these systems are very accurate, they present some disadvantages. First, the cost of such technology may limit its use. Second, it implies that the subject wears some physical markers so as to estimate the 3D pose. As a result, this technology is not convenient for the general public. All these considerations motivated us to focus our study of human motion from RGB-D data.

Challenges In this chapter we address the problem of modelling and analyzing human action in the 3D human joint space. Particularly, our intent is to represent skeletal joint motion in a compact and efficient way that leads to an accurate action recognition. Our ultimate goal is to develop an approach that avoids an overly complex design of feature extraction and is able to recognize

actions performed by different actors in different contexts. Additionally, we study the ability of our approach for reducing latency: in other words, to quickly recognize human actions from the smallest number of frames possible to permit a reliable recognition of the action occurring. Furthermore, we analyze the impact of reducing the number of actions per class in the training set on the classifier's accuracy. In our approach, the spatio-temporal aspect of the action is considered and each movement is characterized by a structure incorporating the intrinsic nature of the data. We believe that 3D human joint motion data captures useful knowledge to understand the intrinsic motion structure, and a manifold representation of such simple features can provide discriminating structure for action recognition. This leads to manifold-based analysis, which has been successfully used in many computer vision applications such as visual tracking [188] and action recognition in 2D video [95, 127, 129, 137].

Motivations The Grassmann manifold has long been known for its interesting mathematical properties, and as an example of homogeneous spaces of Lie groups [219]. 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 ordered 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 [46, 96]. 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.

Overview of our approach We propose in this chapter to take into consideration all the above issues, within a novel geometric approach in the Grassmann manifold, in where the analysis provides a natural way to deal with the problem of sequence matching. Especially, this manifold allows to represent a sequence by a point on its space and offers tools to compare and to do statistics on this manifold. The classification problem in this case can be transformed to point classification problem on the Grassmann manifold. Indeed, action recognition is performed by introducing a learning process on the manifold in conjunction with dynamic modelling process. Time series of consecutive feature vectors with temporal order are firstly constructed. Then, linear dynamic systems are used to capture the dynamic of the motion, before caracterizing the observability matrix of this model as an element of a Grassmann manifold. We formulate our approach through two-fold process: In the first one, we perform classification using a Truncated Wrapped Gaussian model using features computed from depth map information. In the second one, we perform the recognition using a vector representation formed by 3D skeleton coordinates in tangent spaces associated with different classes in order to train a linear classifier.

4.2 Related works

Our approach being based on a geometrical consideration related to a Riemannian manifold, but also specific to the information data captured by depth sensors, this section reviews two categories of related works from two points of view: manifold-based approaches and depth data representation.

4.2.1 Depth-based representation

Recentlty, human action recognition from RGB-D cameras has received growing attention [71, 102]. Maps obtained by RGB-D 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. We propose to group related approaches into three main categories, according to the way they use the depth channel: skeleton-based, depth map-based and hybrid approaches.

Depth map-based approaches rely on the extraction of meaningful descriptors from the entire set of points of depth images. These methods 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. Wanqing et al. [164], projected depth maps 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. 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. Local feature extraction approaches like spatiotemporal interest points (STIP) are also employed for action recognition on depth videos. Bingbing et al.[151] use depth maps to extract STIP and encode Motion History Image (MHI) in a framework combining color and depth information.

Xia et al [72] propose a method to extract STIP a on depth videos (DSTIP). Then around these points of interest they build a depth cuboid similarity feature as descriptor for each action. In the work proposed by Vieira et al. [122], each depth map sequence is represented as a 4D grid by dividing the space and time axes into multiple segments in order to extract SpatioTemporal Occupancy Pattern features (STOP). Also in Wang et al. [119], the action sequence is considered as a 4D shape but Random Occupancy Pattern (ROP) is used for features extraction. Yang et al.[109] employ Histograms of Oriented Gradients features (HOG) computed from Depth Motion Maps (DMM), as the representation of an action sequence. These histograms are then used as input to SVM classifier. Similarly, Oreifej et al. [87] compute a 4D histogram over depth, time, and spatial coordinates capturing the distribution of the surface normal orientation. This histogram is created using 4D projectors allowing quantification in 4D space.

Skeleton-based approaches have become popular thanks to the availability of 3D sensors that made possible to estimate the 3D positions of the body's joints. Especially thanks to the work of Shotton et al. [79], where a real-time method is proposed to accurately predict 3D positions of body joints. Thanks to this work, skeleton based methods have become popular and many approaches in the literature propose to model the dynamic of the action using these features. Xia et al. [114] compute histograms of the locations of 12 3D joints as a compact representation of postures and use them to construct posture visual words of actions. The temporal evolutions of those visual words are modeled by a discrete HMM. Yang et al. [111] extract three features, as pair-wise differences

of joint positions, for each skeleton joint. Then, principal component analysis (PCA) is used to reduce redundancy and noise from feature, and it is also used to obtain a compact *Eigen Joints* representation for each frame. Finally, a naïve-Bayes nearest-neighbour classifier is used for multi-class action classification.

Techniques historically will-known in speech recognition area like Dynamic Time Warping (DTW) [176] also used for action recognition. The feature vector of time series is directly constructed from human body joint orientation extracted from depth camera or 3D Motion Capture sensors, and then DTW is used to match temporal distortions between two data trajectories. Reyes et al. [148] perform DTW on a feature vector defined by 15 joints on a 3D human skeleton obtained using PrimeSense NiTE. Similarly, Sempena et al. [147], by the 3D human skeleton model, use quaternions to form a 60-element feature vector. DTW and its derivatives techniques are relatively sensitive to noise as they require all elements of the sequences to be matched to a corresponding elements of the other sequence. It also has a drawback related to its computational complexity incurring in quadratic cost. However, many works have been proposed to bypass its drawbacks by means of probabilistic models [103] or incorporating manifold learning approach [60, 155]. Finally, recognition in in-line scenario for different applications in IHM present more challenges, in which a trade-off between accuracy and latency can be highlighted. Ellis et al. [98] study this trade-off and employed a Latency Aware Learning (LAL) method, reducing latency when recognizing actions. They train 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. [64], 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 in histograms. 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 [170] and RGB-D data from MSR-Action dataset [164].

Hybrid approaches try to combine positive aspects of both skeleton data features and depth information. Azary et al. [138] propose spatiotemporal descriptors as time-invariant action surfaces, combining image features extracted using radial distance measures and 3D joint tracking. Wang et al. [117] compute local features on patches around joints for human body representation. The temporal structure of each joint in the sequence is represented through a temporal pattern representation called *Fourier Temporal Pyramid*. In Oreifej et al. [87], a spatiotemporal histogram (HON4D) computed over depth, time, and spatial coordinates is used to encode the distribution of the surface normal orientation. Similarly to Wang et al. [117], HON4D histograms [87] are computed around joints to provide the input of an SVM classifier. Althloothi et al. [49] represent 3D shape features based on spherical harmonics representation and 3D motion features using kinematic structure from skeleton. Both feature are then merged using multi kernel learning method.

4.2.2 Manifold-based 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. [142] 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. [168] 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. [153] 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. [130] 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 [95] are also used in order to transform the subspaces of a manifold onto a space where Euclidean metric can be applied. Shirazi et al. [126] 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. [95] propose to represent the spatio temporal aspect of the action by subspaces as elements of the Grassmann manifold. They embed this manifold into reproducing kernel Hilbert spaces in order to tackle the problem of action classification on such manifolds. It is important to note that, to date (2014 date of completion of the works [J2, C7]), few works have recently proposed to use Grassmann manifold analysis for 3D action recognition. Indeed, Azary et al. [104] 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 above state-of-the-art methods, 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 [95, 153] 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. [95] have checked the performance of Riemannian manifolds, in representing human activity, against several state-of-the-art methods. Conducting several experiments, including gesture recognition and person identification, Grassmann manifold has been demonstrated as the one that gives the best performance. Besides, Linear Dynamic Systems (LDS) [76] 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.

4.3 A short note on Grassmann manifolds

Let us giving some brief comments on the basic importance of the notion of the Grassmann manifold. For more detail, reader is referred to the work proposed by Gallivan et al. [205] and Rim Slama's Ph.D thesis [38], from where this short note is extracted.

4.3.1 Mathematical notations and definitions

To model, learn and compare sequences on the Grassmann manifold, we need to understand the representation of points, distance metrics and statistical models on the manifold. 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 manifold $G_{n,d}$, which can be defined as 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 document we focus on the algorithms proposed by Gallivan et al. [205]. 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.

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 [222]:

$$T_I(SO(n)) = X \in R^{n \times n} : X + X^T = 0 \tag{4.1}$$

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) [205] 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) [205] 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. [205]. 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, $\exp_\mu : T_\mu(G_{n,d}) \to G_{n,d}$ which allows to move on the manifold in certain direction. An illustration of these concepts is presented in Figure 4.1.

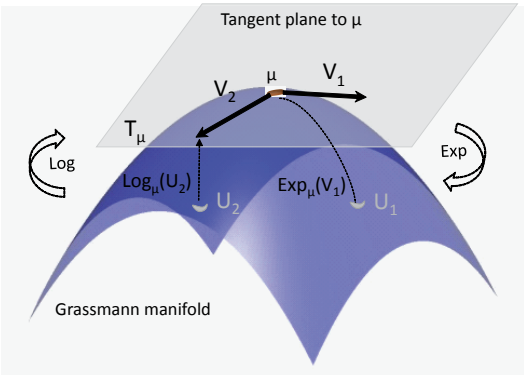


Figure 4.1: Illustration of tangent spaces, tangent vectors, and geodesics on Grassmann manifold. μ is a point on the manifold. T_μ is the tangent space at μ. Tangent vector corresponds to the velocity of the curve on the manifold. Geodesic path is constant velocity curves on the manifold. The exponential 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.

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:

$$dG(U_1, U_2) = \| [\theta_1, \theta_2, \cdots, \theta_i, \cdots, \theta_d] \|_2 \tag{4.2}$$

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 [205].

4.3.2 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 4.2 illustrates a Karcher mean of a sample of elements.

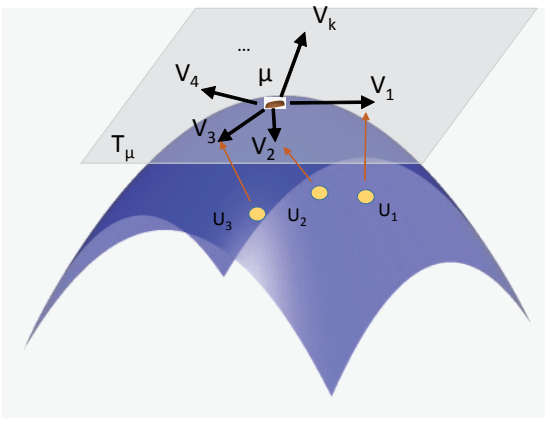


Figure 4.2: Grassmann points, their Karcher mean and their projection onto the tangent space of μ.

The algorithm exploits log and exp maps (4.3.1) 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 4.1.

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 4.1 Karcher mean computation on a Grassmann manifold

$\{U_1, U_2 \cdots U_N\}$: points belonging to $G_{n,d}$,
$\varepsilon = 0.5$, τ: threshold which is a very small number μ_j : mean of $\{U_i\}_{i=1:N}$
1- μ_0: initial estimate of Karcher mean, for example one could just take $\mu_0 = U_1$
$\|\overline{v}\| < \tau$
$i \leftarrow 1$ N **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: $\mu_{j+1} = exp_{\mu_j}(\varepsilon \overline{v})$
5- j=j+1

4.4 Recognition using depth map sequences

In this section, we present our approach presented by Slama et al. [J2, C7] to recognize human action sequences, represented in the depth map space. More particularly, we present a geometric representation of the motion from depth images, leading to a an accurate recognition. In this representation, we can consider data information from depth images, where each sequence is represented by a time series from its local displacement features. Figure 4.3 presents our first proposal approach.

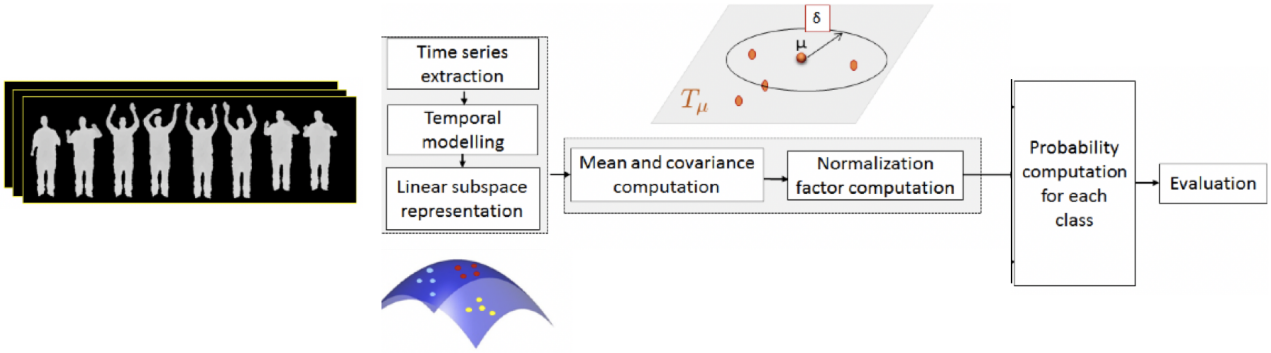


Figure 4.3: The pipeline of our first approach, which is composed of two main modules: (1) temporal modelling of time series data and manifold representation (2) learning approach using probability density function on tangent class-specific.

4.4.1 3D oriented displacement features

With a depth sensor, the distance between the pixel position and the depth sensor z, is obtained and quantized into 11-bit digits. 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].

$$\begin{aligned} &\mathbb{R}^3 \longrightarrow \mathbb{R}^1 \\ &(x, y, t) \longmapsto z = f(x, y, t) \end{aligned} \tag{4.3}$$

Since the orientation of the 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. [77], is given by:

$$n = \bigtriangledown S = (\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}, \frac{\partial z}{\partial t}, -1)^T = (n_x, n_y, n_t, -1)^T \tag{4.4}$$

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:

$$\begin{aligned} n_x &= \frac{\partial z}{\partial x} \simeq I(x - Diff, y, t) - I(x + Diff, y, t) \\ n_y &= \frac{\partial z}{\partial y} \simeq I(x, y - Diff, t) - I(x, y + Diff, t) \\ n_t &= \frac{\partial z}{\partial t} \simeq I(x, y, t) - I(x, y, t + 1) \end{aligned} \tag{4.5}$$

where $Diff$ is a positive value of displacement on image matrix. Encoding orientation information of this normal is more meaningful for describing the surface, than (x,y,z,t) coordinates. Thus, these local oriented displacements can be parametrized using spherical coordinates represented as 3 angles Θ, Φ and Ψ describing respectively zenith angle, azimuth angle and inclination angle. These angles, which are illustrated in Figure 4.4, are computed as follows:

$$\begin{aligned} \Theta &= \tan^{-1}(\sqrt{n_x^2 + n_y^2 + n_t^2}) \\ \Phi &= \tan^{-1}(\frac{n_y}{n_x}) \\ \Psi &= \tan^{-1}(\frac{n_t}{\sqrt{(n_x^2 + n_y^2)}}) \end{aligned} \tag{4.6}$$

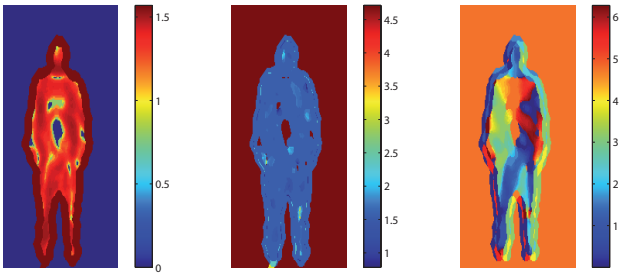


Figure 4.4: 3D angles illustration. From the left to the right the angles Θ, Φ and Ψ.

4.4.2 Temporal modeling

After feature extraction step, the sequence of depth images can be represented as a time series model of features: $F = [f(1), f(2), ..., f(\tau)]$.

3D oriented displacement features computed on each image are linearized on a vector $f(t)$ for modeling the time series.

Let $\Psi(x, y, t)$ denotes the angle orientation of a pixel computed between $I(t)$ and $I(t + 1)$. $f(t) = [\Psi(1, 1, t), \Psi(1, 2, t), \cdots \Psi(n, m, t)]$, with $n \times m = p$ the resolution of the image I. $F = f(1), f(2), , f(T)$, with T the number of frames -1 and $f \in \mathbb{R}^p$. A motion sequence can then be seen as a matrix representing a time-series from angle features. Dynamics and continuity of movement implies that action can not be resumed as a simply set of oriented 3D normal because of the temporal information contained in the sequence. Instead of directly using original time-series data, we believe that a linear dynamic system, like that often used for dynamic texture modeling, is essential before manifold analysis. Therefore, to capture both the spatial and the temporal dynamics of a motion, linear dynamical system characterized by ARMA models, is applied to the time-series matrix M. The dynamic captured by the ARMA model during an action sequence M can be represented as:

$$p(t) = Cz(t) + w(t), \quad w(t) \sim N(0, R), \tag{4.7}$$

$$z(t + 1) = Az(t) + v(t), \quad v(t) \sim N(0, Q) \tag{4.8}$$

where $z \in \mathbb{R}^d$ is a hidden state vector, $A \in \mathbb{R}^{d\times d}$ is the transition matrix and $C \in \mathbb{R}^{3*J\times d}$ 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 learn parameters of the model (A, C) given by these equations. Let $U \sum V^T$ be the singular value decomposition of M. Then, the estimated model parameters A and C are given by:

$$\begin{aligned} \hat{C} &= U \\ \hat{A} &= \sum V^T D_1 V (V^T D_2 V)^{-1} \sum^{-1} \end{aligned} \tag{4.9}$$

where $D_1 = [0 \quad 0, I_{\tau-1} \quad 0]$ and $D_2 = [I_{\tau-1} \quad 0, 0 \quad 0]$ where I represents the identity matrix. Comparing two ARMA models can be done by simply comparing their observability matrices. The expected observation sequence generated by an ARMA model (A,C) lies in the column space of the extended observability matrix given by

$$\theta_\infty^T = [C^T, (CA)^T, (CA^2)^T, ...] \tag{4.10}$$

This can be approximated by the finite observability matrix [142]:

$$\theta_m^T = [C^T, (CA)^T, (CA^2)^T, ..., (CA^{m-1})^T] \tag{4.11}$$

The subspace spanned by columns of this finite observability matrix correspond to a point on a Grassmann manifold $G_{n,d}$, and the action recognition problem is brought back to a classification problem on this manifold.

4.4.3 Learning by Truncated Wrapped Gaussian

If we follow the common learning approach on manifolds, we shall use only one-tangent space, which usually can be obtained as the tangent space to the mean (μ) of the entire data points 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. 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.

Instead of using only one tangent space of the whole data, we opted for the use of several tangent spaces, each obtained on a class of the learning dataset. In order to learn a classifier, our strategy 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 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 [142]. 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 4.5.

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

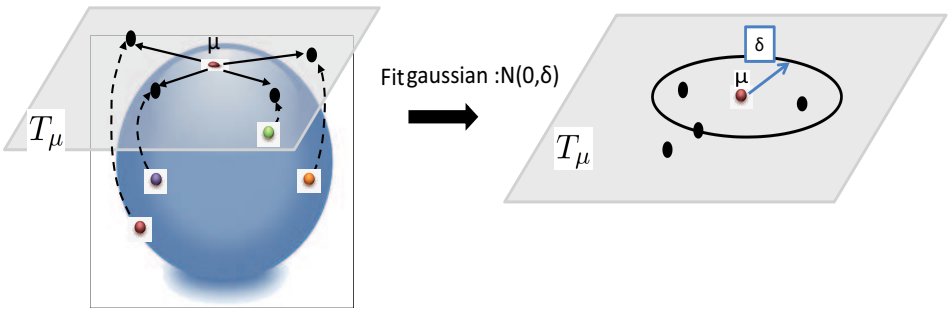


Figure 4.5: Conceptual TWG learning method on the Grassmann manifold to estimate class-conditionals on class-specific poles.

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

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-x^2}{2\sigma^2}} \tag{4.12}$$

After truncation, an approximation of f gives:

$$\hat{f}(x) = \frac{f(x) \times \mathbb{1}_{|x|<\pi}}{z} \tag{4.13}$$

where z is the normalization factor :

$$z = \int_{-\pi}^{\pi} f(x) \times \mathbb{1}_{|x|<\pi} dx \tag{4.14}$$

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

$$\hat{z} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{|x_i|<\pi} \tag{4.15}$$

In practice, we employ wrapped Gaussian in each class-specific tangent space. Separate tangent space 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, standard deviation and normalization factor in each class of actions are computed. The predicted label of an unknown action is estimated as a function of probability density in class-specific tangent spaces.

4.4.4 Experiments

We experimented our proposed approach on three public 3D action and gesture datasets containing various challenges, including MSR-action 3D [164], UT-kinect [114] and MSR-Gesture3D [119]. All datasets that do not provide depth are discarded in these experiments.

MSR-Action 3D dataset MSR-Action 3D [164] 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.

Angle normal computation is performed on cropped area around models. For each frame normal angles features computed on cropped area gives 3800 features. To reduce this feature dimension, we learn 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 vector initially contains 3800 features. This feature dimension can be reduced to 500 while kiping 100% of information. 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 $d = m = 16$. Accuracies of our approach and the state-of-the-art methods are summarized in Table 4.1. To evaluate our approach, we followed the same experimental setup as in Oreifej et al. [87] and Jiang et al. [117], where first five actors are used for training and the rest for testing.

Method	Accuracy %
Histograms of 3D Joints [**xia:2012:cvpr:HistogramofJoints**]	78.97
Eigen Joints [111]	82.33
DMM-HOG [109]	85.52
HON4D [87]	85.80
Random Occupancy patterns [119]	86.50
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 4.1: 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 feature for each image. We note that our method 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 4.1. Using Ψ angle,our approach achieves an accuracy of 86.21%, just below the best method from the state-of-the-art proposed by Oreifej et al. [87]. Knowing that our approach is based on only 3D oriented displacement features without any information about 3D joint positions, compared to other approaches, such as [87] and [119] which use the depth information around joint locations. All results in the rest of experiments are obtained using only Ψ angle as feature to represent the time series.

UT-Kinect dataset 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. To compare our results with state of the art approaches, we follow the leave-one-out cross-validation protocol proposed by Xia et al. [114]. Table 4.2 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. [C10] fail to correctly classify these actions since their approach rely totally on skeleton features. Thus, actions like throw (action with object interaction) 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 [114]	90.92
Space-time Pose Representation [C10]	91.5
Our approach [C7]	**95.25**

Table 4.2: Recognition accuracy (in %) for the UT-kinect dataset using our approach compared to the previous approaches.

MSR Gesture 3D dataset The MSR Gesture 3D dataset [133] contains 336 depth sequences of 12 hand gesture defined by American sign language (ASL). Following experiment setup used by Kurakin et al. [133], 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 4.3. 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.

Method	accuracy
Oreifej et al. [87]	92.45
Jiang et al. [109]	88.50
Yang et al. [119]	89.20
Klaser et al. [182]	85.23
Our approach [C7]	**98.21**

Table 4.3: The performance on MSR Hand gesture 3D dataset compared to previous approaches.

Discussion We obtain a good performance for action sequences with object-subject interaction (ex. UT-kinect dataset), and also when only depth images are available (ex. MSR Gesture 3D dataset). However, when actors are facing the camera in interaction with the computer as in gaming or sport action scenarios [164], 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.

4.5 Recognition using 3D skeleton sequences

In this section, we present the second process fold of our approach, which models human motion in the 3D human joint space. Here, the 3D skeletal motion representation benefits from geometric properties of Grassmann manifold. An overview of our second approach is sketched in Figure 4.6. Each action sequence 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. Then, statistical modelling of inter-classes and intra-class variations are analyzed in conjunction with appropriate tangent vectors on this manifold.

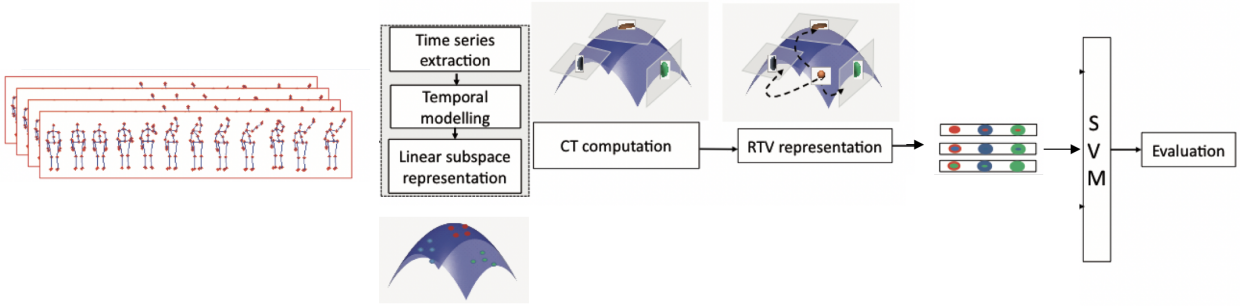


Figure 4.6: Overview of the approach. The illustrated pipeline is composed of two main modules: (1) temporal modelling of time series data and manifold representation (2) learning approach using vector representations formed by concatenating local coordinates in tangent spaces associated with different action classes.

4.5.1 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., $M = [p^1 p^2 \cdots p^\tau]$, $p \in \mathbb{R}^{3*J}$. 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).

4.5.2 Learning by Representative Tangent Vectors

Our strategy here is to consider such data points to be embedded in higher dimensional representation providing a natural and implicit separation of directions. We use the notion of tangent bundle on the manifold to formulate our learning algorithm. The tangent bundle of a manifold is defined in the literature as the manifold along with the set of tangent planes taken at all points on

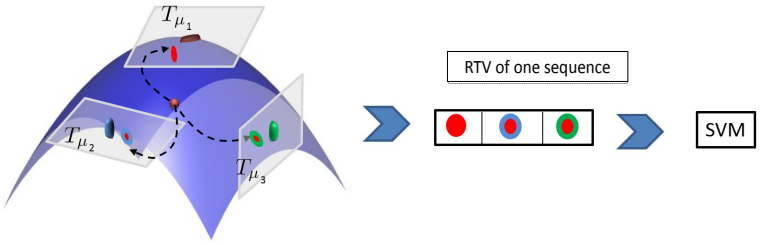


Figure 4.7: Conceptual RTV learning methods on the Grassmann manifold. An action presented by a point on the manifold is projected on all CTs, and thus construct a new observation which is the input of the SVM classifier.

it. Each such a tangent plane can be equipped with a local Euclidean coordinate system. Instead of defining an intrinsic distance for the tangent bundle like [130], we propose here to consider several "local" bundles, each one represents the tangent planes taken at all points belonging to a class from training dataset and expressed as class-specific local bundle.

We generate Control Tangents (CT) on the manifold, which represent all class-specific local bundles of data points. Each CT can be seen as the tangent space of the Karcher mean of all points belonging to the same class of points from only training data. Karcher mean algorithm can be employed here for mean computation. We then introduce an upswing of the manifold learning so-called Representative Tangent Vector (RTV), 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 LTBs can provide the input of a classifier, like the linear SVM classifier as in our case. We note here that frames of 3D joints are concatenated, similarly like in the above section, to form the input of the learning system in a time series matrix.

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 4.7.

4.5.3 Experiments

To evaluate our proposed approach, we conducted several experiments on three public 3D action datasets providing 3D skeleton sequences, including MSR-action 3D [164], UT-kinect [114] and UCF-kinect [98].

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.

MSR-Action 3D dataset The first experiment conducted on this reference dataset is presented in Figure 4.8, 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.

In addition, Table 4.4 shows the accuracy of our approach compared to several state-of-the-art

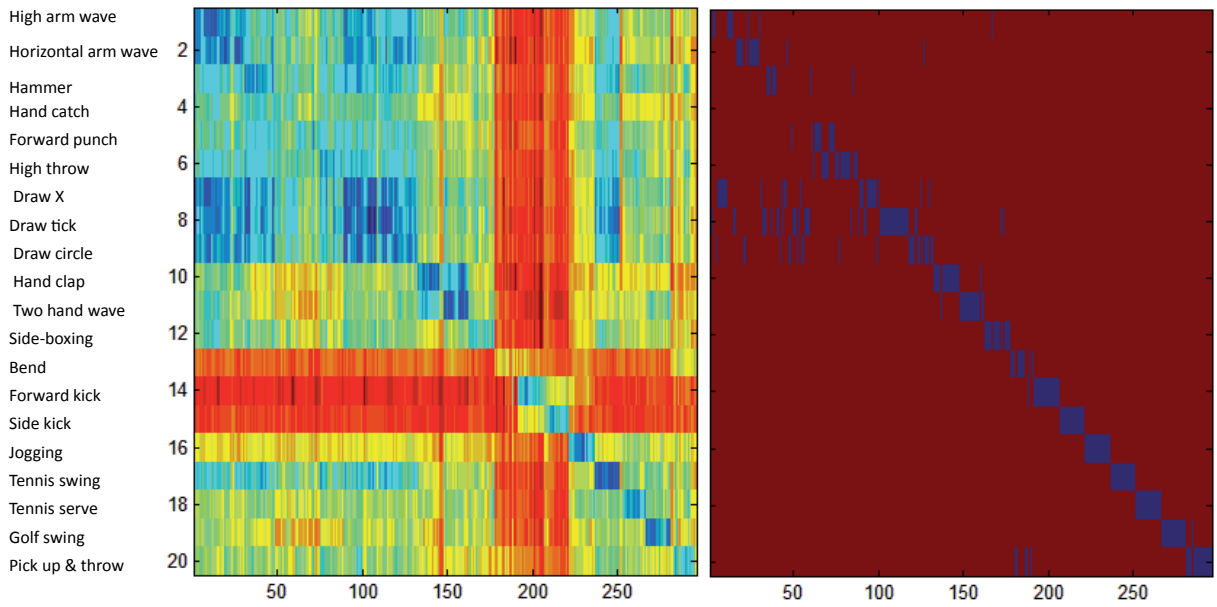


Figure 4.8: Results obtained using the template based method for classification on the MSR-3D Action dataset (a) The 20×260 similarity matrix between the 260 test sequences on the 20 action models learned (b) The same matrix binarized.

methods. We followed the same experimental setup as in Oreifej et al. [87] and Jiang et al. [117], where first five actors are used for training and the rest for testing.

Our results obtained in this table correspond to four learning methods: simple Karcher Mean

Method	accuracy %
Histograms of 3D Joints [114]	78.97
Eigen Joints [111]	82.33
DMM-HOG [109]	85.52
HON4D [87]	85.80
Random Occupancy patterns [119]	86.50
Actionlet Ensemble [117]	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 4.4: Recognition accuracy (in %) for the MSR-Action 3D dataset using our approach [J2] compared to previous approaches.

(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. [119] which use the depth map or depth information around joint locations.

To analyze results obtained by our approach according to the action type, the confusion matrix is illustrated in Figure 4.9. For most of the actions, about 11 classes of actions, video sequences are 100% correctly classified. 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

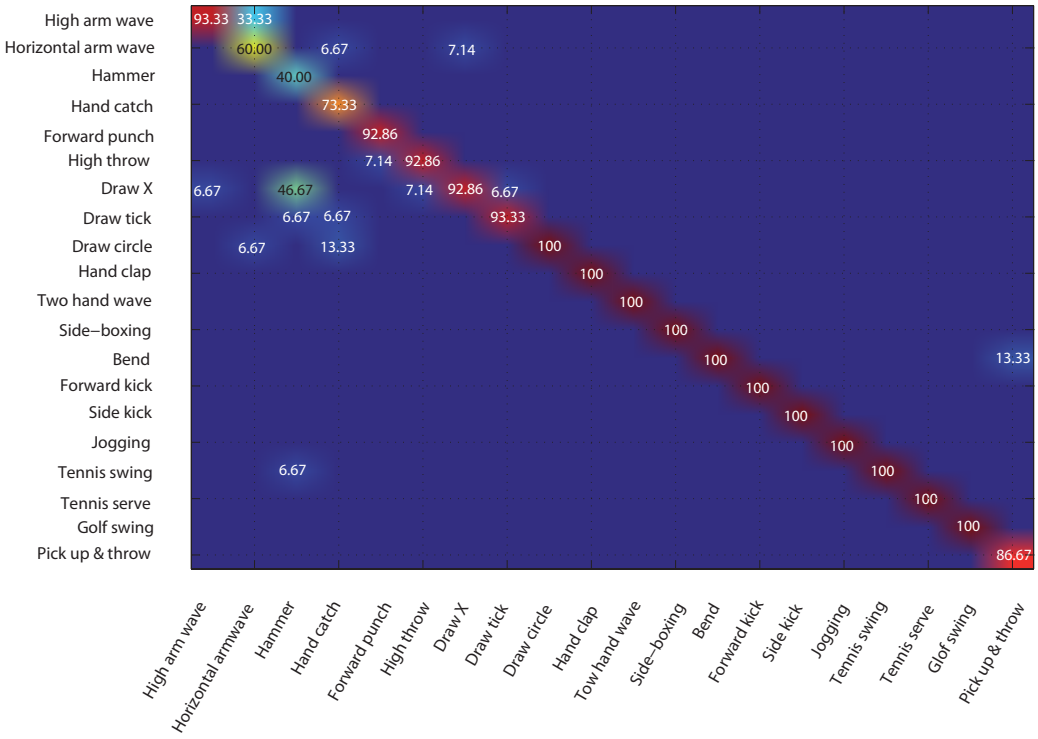


Figure 4.9: The confusion matrix for the proposed approach on MSR-Action 3D dataset.

'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.
Note here that some applications need to train with a very reduced number of data in order to reduce latency when recognizing actions. 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 4.5 shows obtained accuracy results with different size of training dataset.

As noted, 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. [114]. The protocol is leave-one-out cross-validation. In Table 4.6, we show comparison between the recognition accuracy produced by our approach and the approach presented by Xia et al. [114].

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

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 4.5: Recognition accuracy, obtained by our approach using RTVSVM on MSR-Action 3D dataset, with different size of training dataset.

Action	Acc % Xia et al. [114]	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 4.6: Recognition accuracy (per action) for the UT-kinect dataset obtained by our approach [J2] using RTVSVM compared to Xia et al.

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. [114] 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.

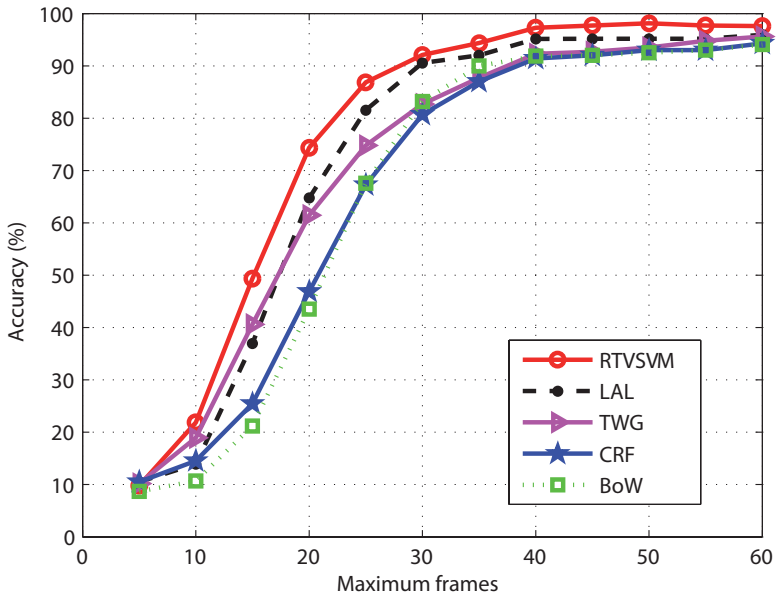
Evaluation of early recognition Our intent here is to evaluate our approach in terms of its ability for a rapid (low-latency) action recognition. The goal is to automatically determine when enough of a video sequence has been observed to permit a reliable recognition of the occurring action. For many 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. We conducted several tests on UCF-kinect dataset [98], on which skeletal joint locations (15 joints) over sequences of this dataset are estimated using Microsoft Kinect sensor and the PrimeSense NiTE. The same experimental setup as in Ellis et al. [98] 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 then compare the result obtained by our approach to those obtained by Latency Aware Learning (LAL) method proposed by Ellis et al. [98] and other baseline algorithms: Bag-of-Words (BoW) and Linear Chain Conditional Random Field (CRF), also reported by Ellis et al. [98].

As shown in Figure 4.10, our approach using RTVSVM clearly achieves improved early recognition 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 4.10 shows numerical results at several points along the curves in the figure. Thus, given only 20 frames of input, our system achieves 74.37%, while BOW, CRF recognition rate below 50% and LAL achieves 61.45%.

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. [98]). These results show that our approach can recognize actions at the desired accuracy with reducing latency. The detail of recognition rates, when using the totality of frames in the sequence, are shown through the confusion matrix in Figure 4.11. 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.

Finally, in order to visually interpret our representation of data, we analyzed the dispersion of actions in each dataset while representing actions by Grassmann representation and using the appropriate metric defined on the maifold. In Figure 4.12, 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 4.2 between each two actions presented as points on Grassmann manifold. We note that



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 [98]	13.91	36.95	64.77	81.56	90.55	95.16	95.94
CRF [98]	14.53	25.46	46.88	67.27	80.70	91.41	94.06
BOW [98]	10.7	21.17	43.52	67.58	83.20	91.88	94.06

Figure 4.10: 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.

our modeling via Grassmann manifold allows a good separation of classes especially for UCF and UT kinect datasets. In MSR-action dataset some overlapping between classes can be seen. These classes are mainly 'Hammer' and 'Draw X' actions.

4.6 Conclusion

In this chapter, we presented 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. Experimental results and the analysis of the performance of our proposed approach show promising results with high accuracies equivalent or superior to the state-of-the-art approaches on three different datasets.

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 [131] which improve TSVM results. 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

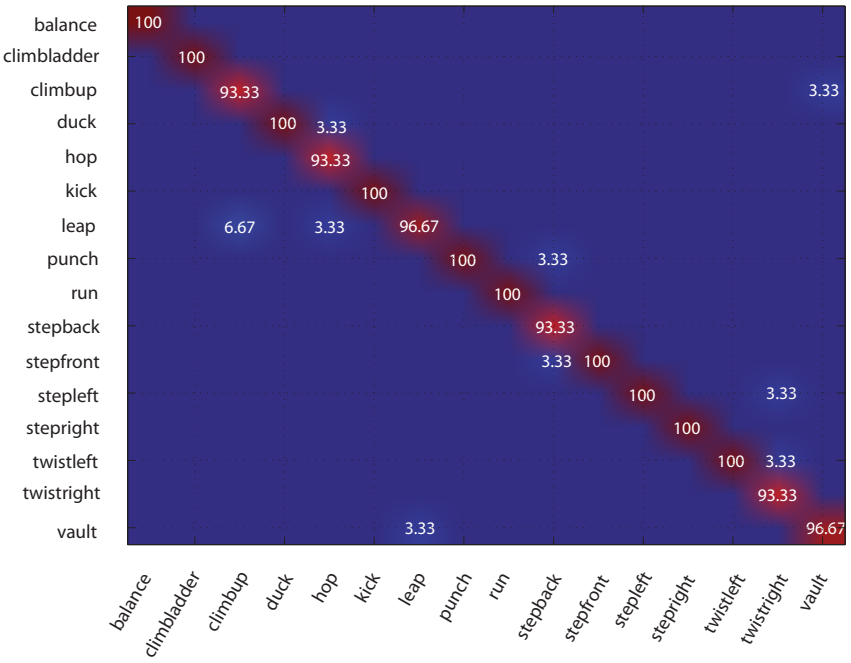


Figure 4.11: The confusion matrix for the proposed method on UCF-kinect dataset, with an overall accuracy of 97.91% is achieved.

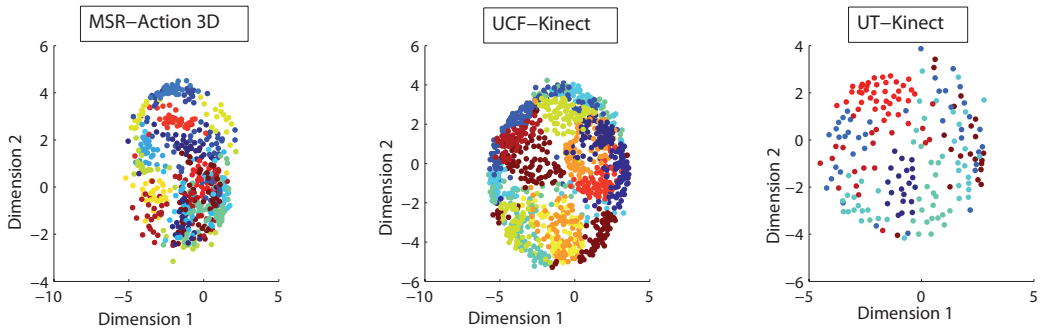


Figure 4.12: MDS plots for actions from three datasets using our proposed geometric framework. In this plot, each point is an action and each color represents a class.

local distances. Results supported our hypothesis, demonstrating that the proposed algorithm is more efficient in action recognition scenario when inter-variation classes is present as a challenge.

In terms of early recognition, the evaluations 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 [98]. 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 recognition time takes 0.26 sec to recognize a sequence of 60 frames. The joint extraction and normalization 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 $d = 12$.

Finally, the limitation of our approach comes from the fact that 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. An extension seems necessary to us at this level is to investigate more challenging problems like human activity recognition, and using additional features from depth images associated to 3D joint locations to solve the problem of human-object interaction.

CHAPTER 5

Human Activity Recognition
Shape Analysis of Motion Trajectories

In this chapter we introduce our contributions in the field of behavior recognition related the analysis of both the human pose and motion that characterize its activities. The recognition of such motions is a challenging task due to the variability of the human pose, the complexity of human motion and possible interactions with the environment. In particular, we discuss a methodological evolution of the shape analysis tools, introduced in Chapter 3, applied her for tarjectories in action space. The contributions in this chapter originate from the work done by Maxime Devanne during his Ph.D thesis [25]. The chapter is organized as follows. Section 5.1 introduces the context of this work and, gives an overview of our approach and completes the existing related methods presented in Chapter 4 by those addressing activity recognition. Section 5.2 discusses the Riemannian framework that we employ for shape analysis of human motion and presents corresponding experiments for action recognition from skeleton sequences. In Section 5.3, we describe the extension of our approach to the activity domain, combining human pose and motion analysis. Section 5.4 gives conclusion.

The contribution presented in this chapter were published in the journal papers [J1, J4] and conference/workshop papers [C4, C6, C10], and from where some parts of this chapter are extracted.

5.1 Context

Human motion analysis has evolved substantially in parallel with major technological advancements, especially capturing technology. Before the realese of RGB-D sensors, human motion analysis from 3D motion capture data has been widely investigated [41, 57, 201]. While these systems are very accurate, they present some disadvantages which may limit its use for the general public, such as to their cost and the fact that the subject must wear physical markers to estimate its 3D pose.

Recently, recognition and understanding of human activities by analyzing data provided by depth cameras have attracted the interest of several research groups [70]. While some methods focus on the analysis of human motion in order to recognize human *gestures* or *actions*, other approaches try to also model interactions with objects, so as to analyze more complex behaviors, like *activities*. Hybrid solutions are often proposed, which use depth maps for modeling scene objects, and body skeleton for modeling the human motion [118]. Other methods propose to describe and model spatio-temporal interactions between human and objects characterizing the activities [94]. Whereas these solutions study short sequences, where one single movement is performed along the sequence, additional challenges appear when several different movements/actions are executed sequentially over a long sequence.

5.1.1 Challenges

While constraints defined in Chapter 4 for action recognition, like robustness to geometric transformations remain and dynamic modeling, some additional challenges appear when it comes to

study more complex human motions, like activities. Indeed, the high degree of freedom of human motions and the variability of gesture combinations that can characterize the human activities greatly complicate the analysis task. Local analysis in time of human movement is often necessary for a thorough understanding of the action performed, knowing that certain activities involve manipulations of objects and/or interactions with the real-world environment. In addition, object manipulation also involves possible occlusions of parts of the human body, resulting in missing or noisy data. Finally, the ability to detect and recognize in online scenarios is one of the major challenges of recognition methods, but also the to analyze long continuous sequence of activities, allowing to answer to a more realistic need.

5.1.2 Our approach

In order to face all these issue, we focus our work in this chapter on the analysis of both the human pose and the human motion that characterize such activities. The motion analysis presented in Chapter 4 lakes joint local and temporal consideration. Therefore, the combination of these two analysis provides information about the human body at each time as well as its evolution along a time interval. To achieve such goals, we first propose to take into account the human pose with its particular configuration, of different body parts with respect to the others in the scene, in order to capture the geometry of the human body in order to examine its shape.

Second, human motion is characterized by the evolution of its pose along the time.

Second, to capture the dynamic evolution of the pose along the sequence as well as the geometric deformation, we propose to analyze the shape of trajectory of the human pose. As a result, we recast the problem of human pose and human motion analysis to a problem of shape analysis seen in Chapter 3.

Our contributions in this chapter are structured into two separate levels. At the first level, short presegmented human motion sequence (action) are considered within a compact representation of its 3D joint trajectories in a suitable action space. The action recognition problem is then formulated as the problem of computing the similarity between the shape of trajectories in a Riemannian manifold. At the second level, more complex human motion sequence (activity) are locally investigated by detecting short temporal segments representing elementary motion, called Motion Segments (MS). For each MS, human motion and depth appearance around human hands are described to characterize the interaction with objects. This provides a deeper analysis of the human movement and allows the recognition of human gestures, actions and activities. In particular, in this chapter, gestures indicate simple movements performed with only one part of the body, actions represent a combination of gestures with different parts of the body, and activities refer to more complex motion patterns possibly involving interaction with objects. The proposed solution can be adapted to realistic scenarios, where several actions or activities are performed subsequently in a continuous sequence. Continuous sequences should certainly be considered in order to detect the starting and ending time of actions. Therefore, our goal consists at the development of an approach operating on the data stream directly, without assuming the availability of a segmentation module that identifies the first and last frame of each action/activity.

5.1.3 Related Work

In recent years, recognition and understanding of human behavior by analyzing depth data has attracted the interest of several research groups [58, 66, 69, 73]. While some methods focus on the analysis of human motion in order to recognize human gestures or actions, other approaches try to

model more complex activities including object interaction. These solutions focus on the analysis of short sequences, where one single behavior is performed along the sequence. However, additional challenges appear when several different behaviors are executed one after another over a long sequence. In order to face these challenges, methods based on online detection have been proposed. Such methods can recognize behavior before the end of their execution by analyzing short parts of the observed sequence. Thus, these methods are able to recognize multiple behaviors within a long sequence, which may not be the case for methods analyzing the entire sequence directly. Existing methods for human behavior recognition using depth data are shortly reviewed here. Methods analyzing human motion for the task of gesture/action recognition from RGB-D sensors can be grouped into three categories: skeleton-based, depth map-based and hybrid approaches, presented in in Chapter 4. Analyzing human motion by these approaches, however, may not be sufficient to understand more complex behaviors involving human interaction with the environment (i.e., what we call *activities*). Hybrid solutions are often proposed, which use depth maps for modeling scene objects and body skeleton for modeling human motion. For example, Wang et al. [118] used Local Occupancy Patterns to represent the observed depth values in correspondence to skeleton joints. Other methods propose to describe and model spatio-temporal interaction between human and objects characterizing the activities, using Markov Random Field [66]. A graphical model is also employed by Wei et al. [74] to hierarchically define activities as combination of sub-events including description of the human pose, the object and interaction between them. Yu and Liu [44] propose to capture meaningful skeleton and depth features using a middle level representation called orderlet.

Some of the works reviewed above have also *online* action recognition capabilities, as they compute their features within a short sliding window along the sequence [44]. This challenge has recently been investigated for continuous depth sequences, where several actions or activities are performed successively. For example, Huang et al. [58] proposed and applied the Sequential Max-Margin Event Detector algorithm on long sequences comprising many actions in order to perform online detection by successively discarding not corresponding action classes.

5.2 Shape Analysis of Motion Trajectories

The dynamic evolution of the human pose characterize naturally the human motion. 3D skeletal data, which provide an accurate representation of the pose, are easy to extract from depth sensors, and they also provide local description of the human body. However, despite the availability of accurate 3D joint positions, recognizing an action is still a difficult task due to significant spatial and temporal variations in the way of performing an action. These challenges motivated us to propose an approach based on the evolution of the position of the skeleton joints detected on a sequence of 3D joints. For this purpose, a high-dimensional vector of 3D joints coordinates is computed for each frame of the sequence. Then, the trajectory described by this vector in the multi-dimensional space is regarded as a signature of the temporal dynamics of the movements of all the joints. These trajectories are then interpreted in a Riemannian manifold, so as to model and compare their shapes using elastic registration and matching in the shape space. In so doing, we recast the action recognition problem as a statistical analysis on the shape space manifold. Furthermore, by using an elastic metric to compare the similarity between trajectories, robustness of action recognition to the execution speed of the action is improved.

5.2.1 Trajectories in the action space

Since the release of the RGB-D cameras, such as the Microsoft Kinect, a 3D humanoid skeleton can be estimated in real-time al. [146], in form of 3D position of a certain number of joints representing different parts of the human body. For each frame t of a sequence, the real-world 3D position of each joint i of the skeleton is represented by three coordinates expressed in the camera reference system $p_i(t) = (x_i(t), y_i(t), z_i(t))$. Let N_j be the number of joints the skeleton is composed of, the posture of the skeleton at frame t is represented by a $3N_j$ dimensional tuple:

$$v(t) = [x_1(t)\ y_1(t)\ z_1(t)\ \ldots\ x_{N_j}(t)\ y_{N_j}(t)\ z_{N_j}(t)]^T\ . \tag{5.1}$$

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

$$M = \begin{pmatrix} v(1) & v(2) & \ldots & v(N_f) \end{pmatrix}\ . \tag{5.2}$$

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

Invariance to geometric transformations. An efficient action recognition system must be able to recognize two instances of the same action differing only for the position and orientation of the person with respect to the capture device. This goal can be achieved either by adopting a translation and rotation invariant representation of the action sequence or providing a suitable distance measure that copes with translation and rotation variations. We adopt the first approach by normalizing the position and the orientation of the subject in the scene before the extraction of the joint coordinates. For this purpose, we first define the spine joint of the initial skeleton as the center of the skeleton (*root joint*). Then, a new base B is defined with origin in the root joint: it includes the left-hip joint vector $\overrightarrow{h_l}$, the right-hip joint vector $\overrightarrow{h_r}$, and their cross product $\overrightarrow{n_B} = \overrightarrow{h_l} \times \overrightarrow{h_r}$. This new base is then translated and rotated, so as to be aligned with a reference base B_0 computed from a reference skeleton (selected as the neutral pose of the sequence). The calculation of the optimal rotation between the two bases B and B_0 is performed using *Singular Value Decomposition* (SVD). For each sequence, once the translation and the rotation of the first skeleton is computed with respect to the reference skeleton, we apply the same transformations to all other skeletons of the sequence. This makes the representation of action sequence invariant to the position and orientation of the subject in the scene.

Representation of body parts. The representation of human pose by its 3D skeleton enable us to take into consideration, not only the whole body, but also of individual body parts, such as the legs and the arms. There are several reasons that make us consider in our approach the parts of the body. First of all, many actions involve motion of just some parts of the body. For example, when subjects answer a phone call, they only use one of their arms. In this case, analyzing the dynamics of the arm rather than the dynamics of the entire body is expected to be less sensitive to the noise originated by the involuntary motion of the parts of the body not directly involved in the action. Furthermore, during the actions some parts of the body can be out of the camera field of view or occluded by objects or other parts of the body. This can make the estimation of the coordinates of some joints inaccurate, compromizing the accuracy of action recognition. Finally,

due the symmetry of the body along the vertical axis, one same action can be performed using one part of the body or another. With reference to the action "answer phone call", the subject can use his left arm or right arm. By analyzing the whole body we can not detect such variations. Differently, using body parts separately, simplifies the detection of this kind of symmetrical actions. To analyze each part of the body separately, we represent a skeleton sequence by four feature sets corresponding to the body parts. Each body part is associated with a feature set that is composed of the 3D normalized position of the joints that are included in that part of the body. Let N_{j_p} be the number of joints of a body part, the skeleton sequence is now represented by four trajectories in $3 \times N_{j_p}$ dimensions instead of one trajectory in $3 \times N_j$ dimensions. The actual number of joints per body part can change from a dataset to another according to the SDK used for estimating the body skeleton. In all the cases, $N_{j_p} < N_j$ and the body parts are disjoint (i.e., they do not share any joint).

5.2.2 Shape analysis of trajectories

The sequence of poses composing an action can be regarded as the result of sampling a continuous curve trajectory in the $3N_j$-dimensional *action space* where each frame is composed of 3D N_j joints. The trajectory is defined by the motion over time of the feature point encoding the 3D coordinates of all the joints of the skeleton, or by all the feature points coding the body parts separately. According to this, two instances of the same action are associated with two curves with similar shape in the action space. Hence, action recognition can be regarded and formulated as a shape matching task.

Furthermore, since the first and the last poses of an action are not known in advance and may differ even for two instances of the same action, the measure of shape similarity should not be biased by the position of the first and last points of the trajectory. In the following we present a framework to represent the shape of the trajectories, and compare them using the principles of elastic shape matching.

In order to capture the geometric deformation of the pose as well as the dynamics of the motion, we propose to consider the motion as a trajectory of the human pose and analyze its shape. As a result, we recast the problem of human motion analysis to a problem of shape analysis by employing the Shape Analysis framework, presented in Chapter 3. In this framework, the shape of a n-dimensional curve $\beta : I \rightarrow \mathbb{R}^n$, normalized in the interval I = [0,1], is captured through the *Square-root Velocity Function* (SRVF) [190] defined as: $q(t) \doteq \dot{\beta}(t)/\sqrt{\|\dot{\beta}(t)\|}$. As a result, each q function can be viewed as an element of a Riemannian manifold $\mathcal{C}$ and the distance between two elements q_1 and q_2 is the length of the geodesic path connecting them on $\mathcal{C}$. Such geodesic path represents the elastic deformation of the shape q_2 to correspond to the shape q_1. As $\mathcal{C}$ is a hyper-sphere, the geodesic length between two elements q_1 and q_2 is defined as $\theta = d_{\mathcal{C}}(q_1, q_2) = cos^{-1}(\langle q_1, q_2 \rangle)$.

The SRVF representation is invariant to translation and scaling, but it is not invariant to rotation and re-parametrization. To cope with this, we define the equivalence class of q as $[q]$, where elements of $[q]$ are equivalent up to rotation and re-parametrization. The set of all equivalence classes is called the *shape space* denoted as $\mathcal{S}$. To compute the geodesic distance between $[q_1]$ and $[q_2]$ on $\mathcal{S}$, we first need to find the optimal rotation and re-parametrization that register the element q_2 with respect to q_1 resulting in q_2^*. Then, the distance $d_{\mathcal{S}}([q_1], [q_2]) = d_{\mathcal{C}}(q_1, q_2^*)$ is invariant to translation, scale, rotation and re-parametrization of curves. In practice, SVD is used to find the optimal rotation, and Dynamic Programming is used to find the optimal re-parametrization.

5.2.3 Action recognition on the manifold

In our proposed approach, action recognition is performed by the K-Nearest Neighbors (kNN) algorithm applied both to whole body and separate body parts. Let us present the process of the classification using the elastic metric on the manifold. Suppose that $\{(X_i,y_i)\}$, $i = 1,\dots,N$, are the training set with respect to the class labels, where X_i belongs to a Riemannian manifold $\mathcal{S}$, and y_i is the class label taking values in $\{1,\dots,N_c\}$, with N_c the number of classes. The objective is to find a function $F(X) : \mathcal{S} \longmapsto \{1,\dots,N_c\}$ able to categorize data lying in different submanifolds of a Riemannian space, based on the training set of labeled items of the data. To this end, we propose a kNN classifier on the Riemannian manifold, learned by the points on the open curve shape space representing trajectories. Such learning method exploits geometric properties of the open curve shape space, particularly its Riemannian metric. This relies on the computation of the (geodesic) distances to the nearest neighbors of each data point of the training set.

Statistics of the trajectories. Riemannian approach provides tools for the computation of statistics of the trajectories, like the Karcher mean [221] to compute an average from several trajectories. The average trajectory among a set of different trajectories can be computed to represent the intermediate one, or between similar trajectories obtained from several subjects to represent a template, which can be viewed as a good representative of a set of trajectories.

To recognize an action, represented as a trajectory on action space and as a point on the manifold, we need to compute the total warping geodesic distances to all points from training data. For a large number of training data this can be associated to a high computational cost. This can be reduced by using the notion of "mean" of class action, and computing the mean of a set of points on the manifold. As a result, for each action class we obtain an average trajectory, which is representative of all the actions within the class. According to this, the mean can be used to perform action classification by comparing the new action with all the cluster means using the elastic metric. For a given set of training trajectories $q_1,\dots,q_n$ on the shape space, their Karcher mean can be defined as:

$$\mu = \arg\min \sum_{i=1}^{n} d_s([q],[q_i])^2 \ . \tag{5.3}$$

There is no only ways to to perform each action by actors. In fact, two different subjects can perform the same action in two different ways. This variability in performing actions between different subjects can affect the computation of average trajectories and the resulting templates may not be good representatives of the action classes. For this reason, we compute average trajectories for each subject, separately. Instead of having only one representative trajectory per action, we obtain one template per subject per action. In this way, we keep separately each different way of performing the action and the resulted average trajectories are not any more affected by such possible variations. As a drawback, with this solution the number of template trajectories in the training set increases. Let N_c be the number of classes and N_{Str} the number of subjects in the training set, the number of training trajectories is $N_c \times N_{Str}$. However, as subjects perform the same action several times, the number of training trajectories is still lower than using all trajectories.

Body parts-based classification Our classification process using kNN computes distances between corresponding parts of the training sequence and the new sequence. As a result, we obtain four distances, one for each body part. The mean distance is computed to obtain a global distance representing the similarity between the training sequence and the new sequence. We keep only the

k smallest global distances and corresponding labels to take the decision and associate the most frequent label to the new sequence. Note that in the case where some labels are equally frequent, we apply a weighted decision based on the ranking of the distances. In that particular case, the selected label corresponds to the smallest distance. However, one main motivation for considering the body parts separately is to analyze the moving parts only. To do this, we compute the total motion of each part over the sequence. We cumulate the Euclidian distances between corresponding joints in two consecutive frames for all the frames of the sequence. The total motion of a body part is the cumulated motion of the joints forming this part. We compute this total motion on the re-sampled sequences, so that it is not necessary to normalize it. Let $j^k : k = 1, \ldots, N_{j_p}$, be a joint of the body part, and N_f be the frame number of the sequence, then the total motion m of a body part for this sequence is given by:

$$m = \sum_{k=1}^{N_{j_p}} \sum_{i=1}^{N_f - 1} d_{Euc}(j_i^k, j_{i+1}^k) \ , \tag{5.4}$$

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

Once the total motion for each part of the body is computed, we define a threshold m_0 to separate moving and still parts. We assume that if the total motion of a body part is below this threshold, the part is considered to be motionless during the action. In the classification, we consider a part of the body only if it is moving either in the training sequence or the probe sequence (this is the sequence representing the action to be classified). If one part of the body is motionless in both actions, this part is ignored and does not concur to compute the distance between the two actions. For instance, if two actions are performed only using the two arms, the global distance between these two actions is equal to the mean of the distances corresponding to the arms only. We empirically select the threshold m_0 that best separates moving and still parts with respect to a labeled training set of ground truth sequences. To do that, we manually labeled a training set of sample sequences by assigning a motion binary value to each body part. The motion binary value is set to 1 if the body part is moving and set to 0 otherwise. We then compute the total motion m of each body part of the training sequences and give a motion decision according to a varying threshold. We finally select the threshold that yields the decision closest to the ground truth. In the experiments, we notice that defining two different thresholds for the upper parts and lower parts slightly improves the accuracy in some cases.

5.2.4 Experiments

To evaluate the efficiency of our approach, we conducted several experiments and compare the obtained results to state-of-the-art ones using three public benchmark datasets (MSR Action 3D [165], Florence 3D Action [81], UTKinect [113]). These three benchmark datasets differ in the characteristics and difficulties of the included sequences. This allows an in depth investigation of the strengths and weaknesses of our solution. In addition, we measure the capability of our approach to reduce the latency of recognition on UCF-kinect dataset [99], by evaluating the trade-off between accuracy and latency over a varying number of actions.

Action recognition analysis. A comparison between our approach and some existing state-of-the-art methods is reported in Table 5.1. The same experimental setup of Oreifej et al. [88] and Wang

Table 5.1: MSR Action 3D. Comparison of the proposed approach with the most relevant state-of-the-art methods.

Method	Accuracy (%)
EigenJoints [112]	82.3
STOP [121]	84.8
DMM & HOG [110]	85.5
Random Occupancy Pattern [120]	86.5
Actionlet [118]	88.2
DCSF [73]	89.3
JAS & HOG^2 [90]	**94.8**
HON4D [88]	88.9
Ours [J4]	**92.1**

Table 5.2: Florence 3D Action. We compare our method with the one presented in [81].

Method	Accuracy (%)
NBNN + parts + time [192]	82.0
Our Full Skeleton [J4]	85.85
Our Body part [C6]	**87.04**

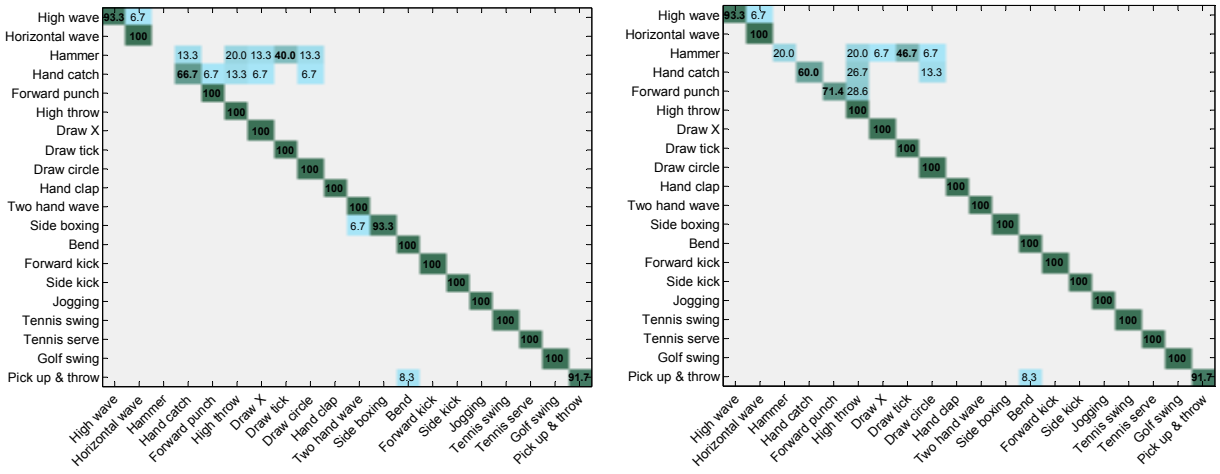
et al. [118] are followed, where the actions of five actors are used for training and the remaining actions for test. Our approach outperforms the other methods except the one proposed in [90]. However, this approach uses both skeleton and depth information. They reported that using only skeleton features an accuracy of 83.5% is obtained, which is lower than our approach.

We then conduct the same experiments exploring all possible combinations of actions used for training and for test. For each combination, we first use only kNN on body parts separately, and obtain an average accuracy of 86.09% with standard deviation 2.99% ($86.09 \pm 2.99\%$). The minimum and maximum values of the accuracy are, respectively, 77.16% and 93.44%. Then, we conduct the same experiments using the full skeleton and the Karcher mean per action and per subject, and obtain an average accuracy of $87.28 \pm 2.41\%$ ($mean \pm std$). In this case, the lowest and highest accuracy are, respectively, 81.31% and 93.04%. Compared to the work in [88], where the mean accuracy is also computed for all the possible combinations, we outperform their result ($82.15 \pm 4.18\%$). In addition, the small value of the standard deviation in our experiments shows that our method has a low dependency on the training data.

Furthermore, we computed the confusion matrix for individual actions. Figure 5.1 shows the confusion matrix when we use the kNN and the Karcher mean per action and per subject with the full skeleton (Figure 5.1a) and with body parts (Figure 5.1b). It can be noted that for each variation of our approach, we obtained very low accuracies for the actions *hammer* and *hand catch*. This can be explained by the fact that these actions are very similar to some others. In addition, the way of performing these two actions varies a lot depending on the subject. For example, for the action *hammer*, subjects in the training set perform it only once, while some subjects in the test set perform it more than once (cyclically). In this case, the shape of the trajectories is very different. Our method does not deal with this kind of variations.

Obtained results by our approach on Florence 3D Action dataset [82] are reported in Table 5.2. It can be observed that the proposed approach outperforms the results obtained in [82] using the same protocol (leave-one-subject-out cross validation), even if we do not use the body parts variant.

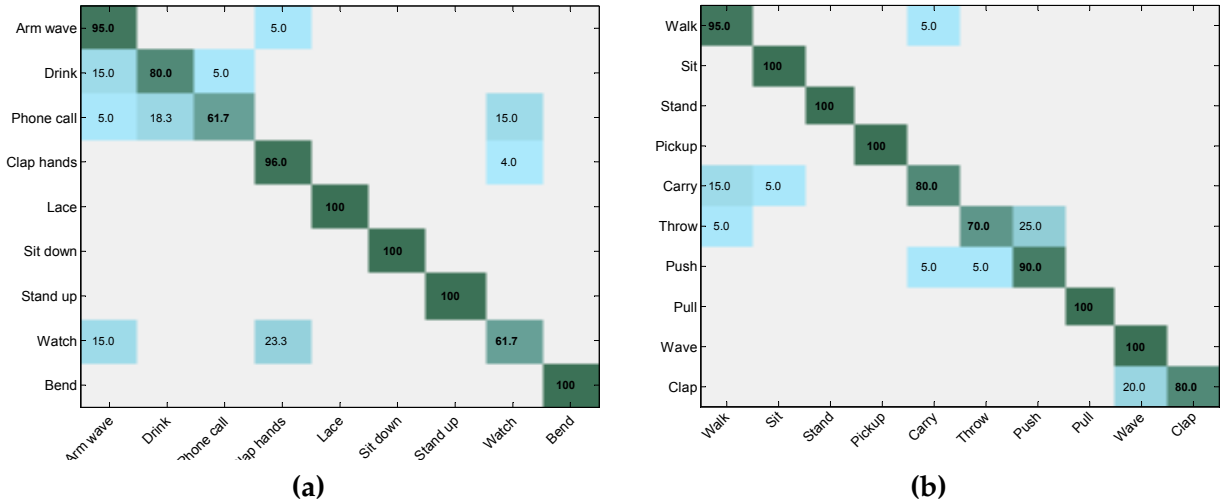
From the confusion matrix in Figure 5.2a, obtained by our method using body parts separately, we can notice that the proposed approach obtains very high accuracies for most of the actions. However, there are some confusions between similar actions using the same group of joints. This



(a) **(b)**

Figure 5.1: Confusion matrix for two variations of our approach obtained on MSR Action 3D dataset: (a) Full skeleton with kNN and Karcher mean per action and per subject; (b) Body parts with kNN and Karcher mean per action and per subject.

can be observed in the case of *read watch* and *clap hands*, and also in the case of *arm wave*, *drink* and *answer phone*. For these two groups of actions, the trajectories of the arms are very similar. For the first group of actions, in most of the cases, *read watch* is performed using the two arms, which is very similar to the action *clap hands*. For the second group of actions, the main difference between the three actions is the object held by the subject (no object, a bottle, a mobile phone). As we use only skeleton features, we cannot detect and differentiate these objects. As an example, Figure 5.3 shows two different actions, *drink* and *phone call*, that in term of skeleton are similar and difficult to distinguish.



(a) **(b)**

Figure 5.2: Confusion matrix obtained by our approach on (a) Florence 3D Action and (b) UTKinect. We can see that similar actions involving different objects are confused.

Finally, we conducted the same experiments on UTK-Kinect dataset [113] using leave one sequence out cross validation protocol. For each iteration, one sequence is used as test and all the

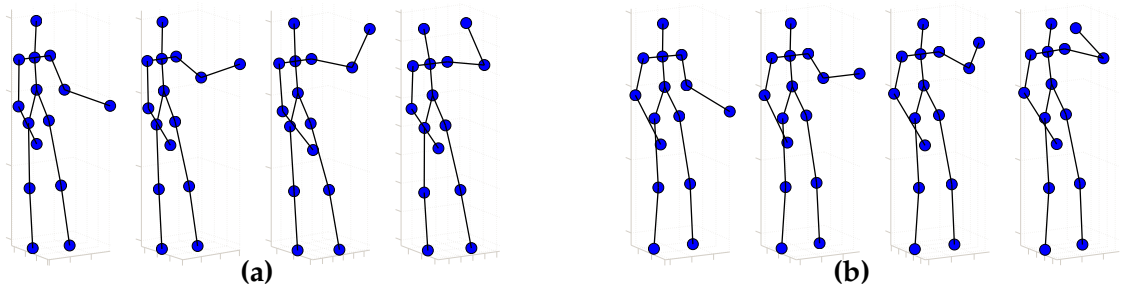


Figure 5.3: Example of similar actions from Florence action 3D dataset: (a) *drink* action where the subject holds a bottle; (b) *phone call* action, where the subject holds a phone.

other sequences are used as training. The operation is repeated such that each sequence is used once as testing. We obtained an accuracy of 91.5%, which improves the accuracy of 90.9% reported in [113]. This shows that our method is robust to different points of view and also to occlusions of some parts of the body. However, by analyzing the confusion matrix in Figure 5.2b, we can notice that lower accuracies are obtained for those actions that include the interaction with some object, for instance the *carry* and *throw* actions. These actions are not always distinguished by actions that are similar in terms of dynamics yet not including the interaction with some object, like *walk* and *push*, respectively. This result is due to the fact that our approach does not take into account any informative description of objects. Results on different datasets show that our approach outperforms most of the state-of-the-art methods. First, some skeleton based methods like [112] use skeleton features based on pairwise distances between joints. However, results obtained on MSR Action 3D dataset show that analyzing how the whole skeleton evolves during the sequence is more discriminative than taking into consideration the joints separately. In addition, the method proposed in [112] is not invariant to the execution speed. To deal with the execution speed, in [81] a pose-based method is proposed. However, the lack of information about temporal dynamics of the action makes the recognition less effective compared to our method, as shown in Table 5.2. Second, the comparison with depth-map based methods shows that skeleton joints extracted from depth-maps are effective descriptors to model the motion of the human body along the time. However, results also show that using strength of both depth and skeleton data may be a good solution as proposed in [90]. The combination of both data can be very helpful especially for the case of human-object interaction, where skeleton based methods are not sufficient as shown by the experiments on UTKinect dataset.

Discussion. The experimental results on the MSR Action 3D, Florence 3D Action and UTKinect datasets demonstrate that our approach outperforms the existing state-of-the-art methods in most of the cases. However, experiments also demonstrated some limits of our approach. Firstly, we identify a failure case when actions can be characterized by a different number of repetitions of a single gesture. In that case the shape of the resulted motion trajectories may differ and the recognition effectiveness can be affected. Secondly, as we are using only skeleton data, we only have information about the human pose and its evolution along the time. The analysis of obtained results on benchmark action dataset shown that some different actions may be very similar in term of human motion. What differentiate such similar actions is the object held by the subject. By only using skeleton data, we are unable to describe such interaction with objects and thus to differentiate these similar actions. Finally, our proposed method is a sequence-based approach: we analyze and classify a full delimited sequence. Indeed, during experiments, we consider that each

sequence contains only one action starting in the beginning of the sequence and finishing at its end. However, this consideration does not reflect a real-world context in which a camera is continuously observing a scene. Hence, the subject may perform different actions successively as well as remain still during a certain time interval. Thus our method is not appropriate for this real-world context. In the following section, we investigate a way to deal with these limits and thus we propose a method suitable for more complex cases.

5.3 Analysis of complex activities by motion segmentation

The skeleton and its changes across time provide valuable information. However, understanding the human behavior is still a difficult task due to the complexity of human and spatial/temporal variations in the way gestures, actions, or activities are performed. These challenges motivated us to analyze locally the motion sequences.

Human pose within the sequence is firstly represented by a 3D curve describing the spatial configuration of the skeleton. This representation permits to interpret the curves in a Riemannian manifold of shape space where their shapes can be modeled and compared using elastic metric of this manifold. Thanks to such shape analysis, we can identify similar human poses and group them together. As a result, a motion sequence is temporally segmented into a set of successive subsequences of elementary motions, called Motion Segments (MS). A MS is thus characterized by a sequence of skeletons, each of which is modeled as a multi-dimensional vector by concatenating the three-dimensional coordinates of its joints. Then, the trajectory described by this vector in the multi-dimensional space is regarded as a signature of the temporal dynamics of all the joints. Similarly to pose curves, the shape of such motion trajectories is analyzed in a shape space manifold. A statistical analysis on this manifold allows us to compare motion trajectories independently to their execution speed, and then to identify relevant shapes characterizing a set of MSs. It should be noted here that skeletal data do not sufficiently describe human behavior in presence of object manipulation. However, the depth appearance around subject hands if considered in MS, it can provide useful information about possible human-object interactions. Therefore, we employ a Dynamic Naive Bayes classifier to model the sequence of MSs, by combining both skeleton and depth features in order to fully describe the dynamics of human behavior.

Our strategy in this part is to propose an approach based on the analysis of both human pose and human motion. Using a shape analysis framework, an activity sequence can be analyzed and described through two steps: First, we locally regard it at the level of human poses in order to segment the full human motion into a set of Motion Segments (MSs). Then, the analysis of these segments allows us to describe the sequence as a combination of successive MSs.

5.3.1 Shape analysis of human pose

While Human motion is characterized by the evolution of the human pose across time, a pose of human body can be characterized by the spatial configuration of body parts. So, we propose to analyze the shape of such spatial configuration. In order to capture the geometric deformation of the pose as well as the dynamics of the motion, we propose to consider the motion as a trajectory of the human pose and analyze its shape. As a result, we recast the problem of human pose and human motion analysis to a problem of shape analysis by employing the Shape Analysis framework, presented in Chapter 3 and used in Section 5.2. Human body is represented by a set of 3D joints located in correspondence to different body parts. Thus, a human pose is characterized by a certain spatial configuration of these 3D joints. In order to describe human poses, we propose

to analyze the shape of the spatial configuration of 3D joints. By connecting the 3D joints, human pose can be viewed as a 3D curve representing the shape of human body. As shown in Figure 5.4, in order to keep the human shape information associated to the limbs, we keep a coherent structure linking together joints belonging to the same limb. Thus, a 3D curve representing the human pose connects successively the spine joints, then the arms joints (left/right) and finally the legs joints (left/right). In this way, a human pose is represented by a 3D curve instead of a 3D skeleton. Thus, We can perform shape analysis of curves using the shape analysis framework and the provided distance (see Sect. 5.2 for $n = 3$ as each joint is represented by the x, y, z coordinates. Note that, as we will explain later, we need to compare successive human poses from a same sequence (same subject). Hence, we can assume that the scale of skeletons as well as the orientation of the subject between two successive poses are unchanged during a short time interval. Likewise, as a 3D curve connects joints in a predefined order, the parametrization of curves remains the same along a single sequence. Since it is not necessary to find the optimal re-parametrization between two shapes, the analysis of the shape of the 3D curves is simplified. Figure 5.4 shows a geodesic path between two human poses represented by their 3D curve.

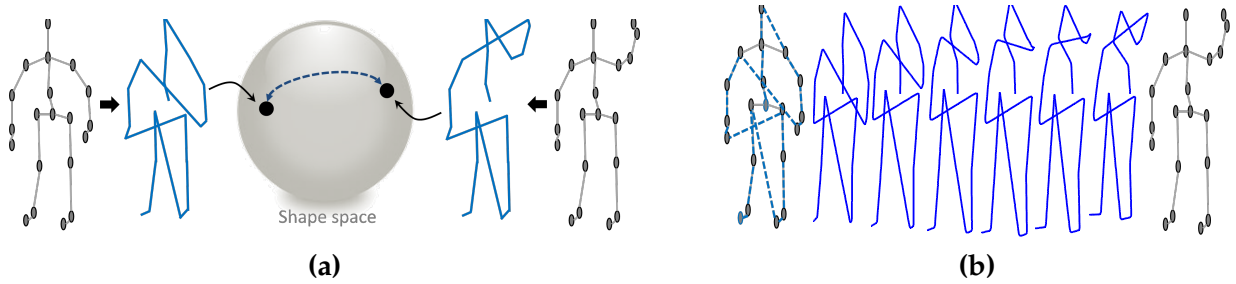


Figure 5.4: Human pose interpretation as curve in the shape space manifold. (a) Shape of 3D curves representing human poses represented in the shape space where the distance between two shapes is measured through the geodesic distance. (b) Visualization of the geodesic path representing a natural deformation between shape of poses.

5.3.2 Motion decomposition of activity sequence

To deal with the complexity of human motion activities, our approach decomposes firstly the full motion into shorter MSs. The idea of decomposing a motion sequence into a set of MSs has already been investigated by several state-of-the-art approaches. In [134] the "movelet" is proposed on accelerometer data by concatenating features within overlapping temporal intervals with fixed length. However, as the length of each temporal interval is fixed, it may not represent a relevant MS. Another idea called "dyneme" is employed in [57], where human poses are clustered to identify several temporal segments with similar poses represented by one centroid pose. However, the use of pose information only may lack of information about the dynamics of the MS. In addition, labeling successive poses independently may result in irrelevant intervals. In our approach, the decomposition process is based on the analysis of the human pose at each frame of the sequence to identify relevant MSs including continuous elementary motions.

Thanks to the elastic distance measuring the similarity between the shape of two poses is defined, we can analyze the deformation of human body along an activity sequence. Thus, we identify MSs by breaking the continuous sequence of activity in correspondence to points where the speed of change of the 3D curve has a local minimum. To compute the speed of change, we benefit

from the shape analysis framework that enables the computation of statistics, like the mean and the standard deviation, on the manifold. Hence, given the poses $p_1, \ldots, p_n$ observed over a temporal window of predefined duration, the average pose shape μ is computed as the Riemannian center of mass [221] of the pose shapes $q_1, \ldots, q_n$ on the shape space. For this purpose, the distance $d_{\mathcal{S}}$ described in Sect. 5.2 (more detail can be found in Chapter 3) is used according to the following expression:

$$\mu = \arg\min_{[q]} \sum_{i=1}^{n} d_{\mathcal{S}}([q], [q_i])^2 \ . \tag{5.5}$$

Once the mean pose shape is computed, the standard deviation σ between this mean shape and all the shapes within the window is estimated:

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} d_{\mathcal{S}}([\mu], [q_i])^2} \ . \tag{5.6}$$

Higher values of σ correspond to faster motion, while lower values correspond to slower motion, i.e., transition intervals. By detecting local minima along the sequence, we are able to temporally localize the motion transition, and thus decompose the sequence into MSs.

As an example, Figure 5.5 shows the variation of σ along a sequence and the MSs identified by breaking the sequence in correspondence to local minima of σ.

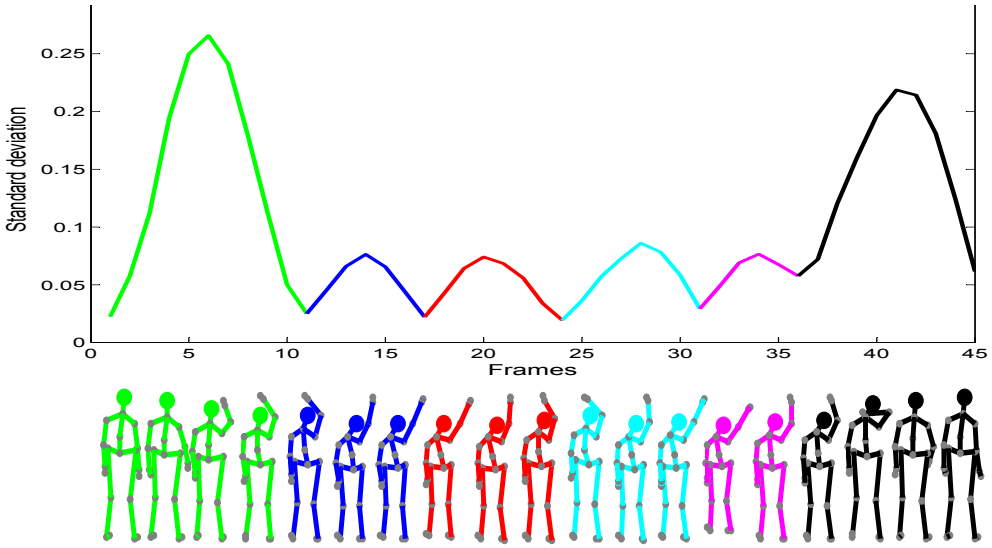


Figure 5.5: Segmentation of a sequence based on minima of the standard deviation σ. Different MSs and corresponding poses are displayed with different colors.

5.3.3 Segment features

After the segmentation of the activity sequence, all MSs are taken into consideration in the second phase in order to describe the whole sequence. The features here is twofold: motion features describing the evolution of pose over the segment, and appearance features describing the depth appearance around hands in order to characterize possible objects held by the subject.

Motion features. Here, we interpret the pose changes across a time interval corresponding to a MS. For each frame included in a MS, we concatenate the x_i, y_i, z_i coordinates of each joint to build

a feature vector. Let N_j be the number of joints of the skeleton, the posture of the skeleton at frame t is represented by a $3N_j$ dimensional tuple:

$$v(t) = [x_1(t)\ y_1(t)\ z_1(t), \ldots, x_{N_j}(t)\ y_{N_j}(t)\ z_{N_j}(t)]^T\ . \tag{5.7}$$

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

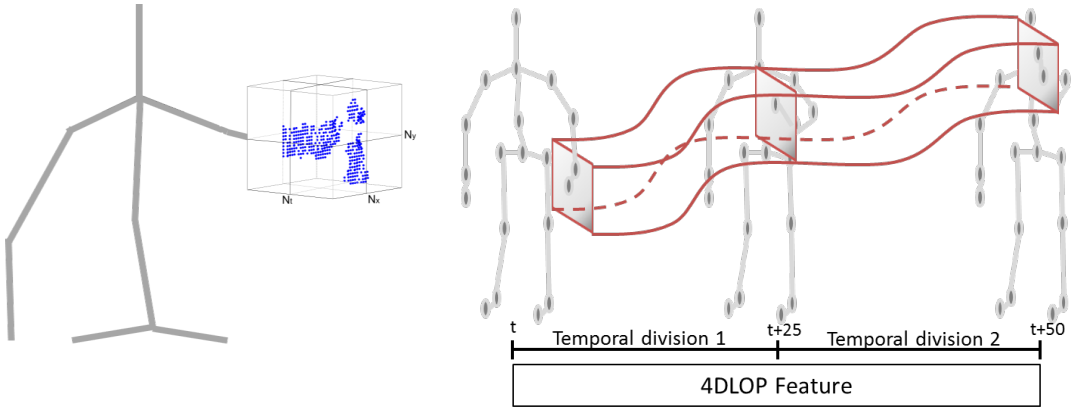
$$M = \begin{pmatrix} v(1) & v(2) & \ldots & v(N_f) \end{pmatrix}\ . \tag{5.8}$$

This matrix captures the changes of the skeleton pose across time. Hence, it can be viewed as a trajectory in R^{3N_j} representing the motion in a $3N_j$ dimensional space. The size of such feature matrix is $3N_j \times N_f$. Note that, in order to guarantee invariance to MSs translation and rotation, we normalize the position and the orientation of the subject before extracting the features. We use the spine and hips joints to form the base representing the position and orientation of the body. We align the initial pose of a segment with respect to a reference posture by finding the best rigid transformation between corresponding bases. The optimal transformation is then applied to all other poses of the segment. This makes the representation invariant to the position and orientation of the subject in the scene. With this representation, an activity sequence can be viewed as a set of short spatiotemporal trajectories in R^{3N_j} representing MSs.

Appearance features. Features of human motion are complemented with appearance features describing the objects the user is interacting with, if any. Such combination of motion and object features improves the robustness of the activity recognition, and is also necessary to discriminate between actions that would be almost identical in terms of motion patterns. Indeed, discriminating between activities like *Drink* and *Phone call* would require a description pattern capable of accurately distinguishing whether the user hand is closer to the mouth than to the ear. This level of accuracy is generally beyond the capability of commercial depth sensor, unless the user is very close to the sensor. Differently, two such actions can be easily distinguished by considering the objects with which the user interacts.

In order to describe the distribution of depth pixels within a local region around subject hands, the Local Occupancy Pattern (LOP) [120] descriptor can be used. In this approach, a depth image is viewed as a 3D point cloud, and the local regions are represented by 3D bounding boxes centered at the hand joints. As shown in Figure 5.6a, each bounding box is partitioned into $N_c = N_x \times N_y \times N_z$ 3D cells, and the number of 3D points that fall in each cell is counted. In the experimental tests, we empirically select a local region of size $0.3m \times 0.3m \times 0.3m$ divided into $5 \times 5 \times 5$ cells.

This local depth representation is combined with the motion features, which represent an activity as a sequence of successive MSs. For each frame of a MS, we compute the LOP feature for each hand joint (l_l and l_r) and concatenate them to form one global LOP feature vector $L_f = [l_l, l_r]$ for the frame f. The length of such feature vector is $2 \times N_c$. However, MSs can have different duration. As a consequence, they are described with a different number of LOP features, which is not convenient in the comparison. To deal with duration variability, we propose a compact representation of the depth appearance, which is independent from its duration. First, we assume the object held by the subject during the time interval corresponding to a MS does not change considerably, and we compute the mean of the LOP features among frames of a MS. Thus, one single feature, that we call Mean LOP (MLOP) is used to describe the average depth appearance of a MS. Then, we consider changes of depth appearance around hand joints, which can be induced by object manipulation



(a) **(b)**

Figure 5.6: LOP feature computation. (a) A 3D cuboid divided into 3D cells is extracted from the depth image around the hand joint and the number of 3D points within each 3D cell is counted. (b) Schema of the 4DLOP feature representing depth appearance evolution along a MS in two time steps.

during a MS. For instance, for the activity *Drink* a MS would consist of bringing the container to the mouth. In that case, the support where the object is located may appear in the local region around the hand, in the first part of the MS, but the face of the subject may be present in this local region at the end of the MS. To represent this depth variation, we adopt an extension of LOP feature in four dimensions called 4DLOP. The spatio-temporal volume representing the change of the local region around hands along the MS is also partitioned in N_t divisions across temporal dimension.

Note that, differently to [43], which analyzes depth variation in fixed 4D boxes, we consider depth variation in a moving spatio-temporal region following the motion of human hands. This idea is illustrated in Figure 5.6b. As a result, each MS is represented by a feature vector describing the depth appearance independently to its duration (either MLOP or 4DLOP).

5.3.4 Vocabulary of Motion Units

We propose to use a bag-of-word paradigm to describe human behaviors, so as to identify codebook of exemplar MUs (symbols) necessary to build a reference dictionary. Such codebook is usually learned from training sequences. Then our idea is that unknown complex motion sequences can be represented as a set of generic MUs from the learn codebook, thus facilitating their analysis and understanding. As each MU is described by two types of features representing human motion and depth appearance, we identify two distinct codebooks for each of the feature.

Motion codebook. Human Motion Units are represented by the shape of the corresponding motion trajectories in the shape space. To learn the codebook of exemplar shapes, we perform clustering of shapes using the k-means clustering algorithm, using elastic distance, on the shape space. Such clustering provides a mapping between trajectory shapes represented on the shape space and a finite set of symbols corresponding to clusters. In order to describe each cluster by using only its corresponding exemplar shape, we propose to learn a density function for each cluster. These density functions capture the variability between shapes belonging to the same cluster and provide a deeper modeling of each cluster. In so doing, we assume the distribution of shapes within a cluster follows a multivariate normal model.

Unfortunately, learning such density functions on the shape space is not straightforward, mainly due to the non-linearity and infinite-dimensionality of such manifold (i.e., shapes are represented by functions, so they have infinite dimension). Different methods have been proposed to deal with these two challenges [145, 195]. A common way to circumvent the non-linearity of the manifold is to consider a hyper-plane tangent to the manifold at the mean shape (i.e., *tangent space*). Such tangent space is a linear vector space, where conventional statistics applies, like the computation of density functions. We denote $T_{\mu_k}\mathcal{S}$ the tangent space at the mean shape of the k-th cluster μ_k. For each shape $q_i \in \mathcal{S}$ within the k-th cluster, we compute its corresponding tangent vector $v_i \in T_{\mu_k}\mathcal{S}$ using the logarithm map. This approximation is valid because samples belong to the same cluster. Thus, we can assume that they lie in a small neighborhood around the mean shape μ_k. To deal with the problem of infinite-dimensionality, we assume the variations in tangent vectors are restricted to an m-dimensional subspace. Using tangent vectors of each cluster, we use PCA to learn a principal subspace for each cluster. We denote n the dimension of such principal subspace. Tangent vectors v_i are then projected into the learned subspace. Let $\tilde{v}_i$ be such projected vectors, we compute the covariance matrix Σ between all projected vectors $\tilde{v}_i$ belonging to the same cluster. Finally, we use the resulting mean shape μ and covariance matrix Σ to learn a multivariate normal distribution for each cluster. Its corresponding probability density function is defined as:

$$f(\tilde{v}) = \frac{1}{(2\pi)^{n/2}\ |\Sigma|^{1/2}} e^{-\frac{1}{2}\tilde{v}^T\Sigma^{-1}\tilde{v}}\ . \tag{5.9}$$

where Σ corresponds to the covariance matrix computed on the learned principal subspace. The process of learning the distribution on the shape space is illustrated in Figure 5.7.

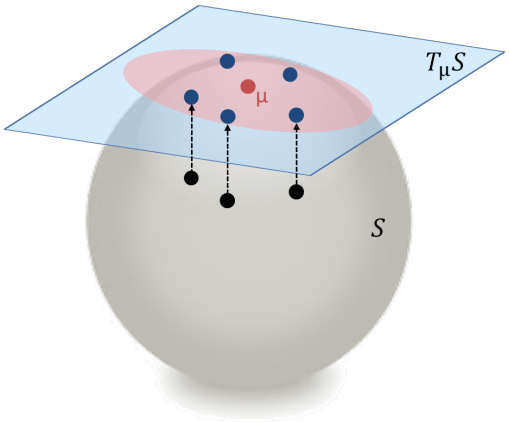


Figure 5.7: For each cluster, the mean shape μ is computed (red) from shapes q_i belonging to the same cluster. Then, the shapes q_i (black) are projected on the corresponding tangent space $T_\mu\mathcal{S}$. Such tangent vectors v_i (blue) are used to compute the covariance matrix and learn the multivariate distribution for each cluster.

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

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

Σ_k. Then the log probability that the shape q belongs to the cluster k is defined as:

$$L = -\frac{1}{2}ln(|\Sigma_k|) \ - \frac{1}{2}\tilde{v}_k^T \Sigma_k^{-1} \tilde{v}_k \ - \frac{n}{2}ln(2\pi) \tag{5.10}$$

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

$$d_l = \|l_A - l_B\|^2_{N_c} = \sum_{i=1}^{N_c} (a_i - b_i)^2 \ , \tag{5.11}$$

where a_i and b_i are the i-th components of l_A and l_B, respectively. Such clustering provides a mapping between LOP feature vectors and a finite set of LOP symbols represented by the cluster centroids. Similarly to human motion, the codebook is learned from MU segments of training sequences. For MU segments of test sequences, we first compute the distance d_l between the corresponding LOP feature and all the exemplar LOP. Finally, the labeling is done using the nearest rule.

5.3.5 Dynamic modeling of activity sequences

The activity sequence is decomposed now into MSs, and each MS is described in terms of human motion and depth appearance around subject hands. Thus, the dynamics of a sequence can be viewed as combination of two sequences of successive symbols, one corresponding to human motion, and the other corresponding to depth appearance around hands. In so doing, we assume that sequences of the same class are represented by similar arrangements of MSs. Conversely, different sequences of symbols should represent different classes. Hence, we need a method to analyze the change of symbols across time, and recognize different arrangements of MSs. To this end, we propose to use the Dynamic Naive Bayes classifier (DNBC) [179] as statistical model.

Learning. In DNBC training, we only know the sequence of observations $X = \{X_t^a \mid t = 1, \ldots, T \ , \ 1 \leq a \leq A\}$, being A the number of attributes, while the states $S = \{S_t | t = 1, \ldots, T\}$ are not available. Thus, we need tools for estimating the model parameters, i.e., the *prior*, *transition* and *emission* probabilities. The prior probability represents the initial state of the process. The transition probability is the probability to transit from one state to another state of the process. The emission probability represents, for each state, the probability of generating each attribute. Similarly to HMM, a common way to learn such parameters from training sequences of observed symbols is to use the Baum-Welch algorithm [223]. In the case of DNBC, parameters estimation is slightly modified due to the model setting, which allows the emission of several attributes per state (more details on this can be found in [140]). For our task, we assume that each activity class is modeled with a different DNBC. Let the activity class $c \in \{1, \ldots, C\}$ with C being the number of activity classes, we learn one DNBC denoted λ_c for each class c using the training sequences of to the class c.

Classification. The classification process of an observed sequence X is the performed as follows. First, the sequence is presented to each of the trained λ_c DNBC modeling different activity classes. Then, the likelihood $P(X|\lambda_c)$ that the sequence X has been generated by the λ_c DNBC is computed

using the *Forward* algorithm. Finally, the sequence is classified as the activity whose corresponding DNBC gives the highest log-likelihood: $activity(X) = \arg\max_c P(X|\lambda_c)$.

This process is then extended to perform the online classification, so that a decision can be taken before the end of a sequence. This is particularly convenient for real-time applications, permitting natural interaction with the system. In addition, it allows us to process a sequence as a continuous stream, where several activities can be performed successively, which is often the case in real-world contexts. As shown in Sect. 5.3.2, the segmentation process is based on a sliding window technique. Hence, it can also be applied in an online manner so as to detect MSs from a continuous stream. Each new frame of the sequence is given as input to the segmentation process. When a MS is detected, we compute the corresponding human motion and depth appearance features and assign a symbol to each, as described in Sect. 5.2. The resulted observation sequence of length-1 is then presented to each trained DNBC in order to compute the corresponding log-likelihoods. This process is performed for each new detected MS. Thus, the length of the observation sequence is increased by one, and the log-likelihoods are updated. If the log-likelihood of a class falls below a threshold, we discard the activity class. This allows us to gradually reduce the set of possible classes. The process is repeated until all classes are discarded. Among the remaining classes, we keep the class with the highest log-probability as the detected activity. However, transitions between activities are often smooth. Thus, when an activity is finished, its corresponding log-probability may not considerably decrease and directly fall below the threshold. In order to consider this smooth transition, we select as the ending boundary of the activity the time step when its corresponding log-probability starts to decrease instead of the time step when it falls below the threshold. Finally, we restart the detection process from the successive time step using all the classes. This is repeated until the end of the sequence. As a result, we obtain the set of detected activities along the sequence with corresponding starting and ending boundaries. This online activity detection is illustrated in Figure 5.8.

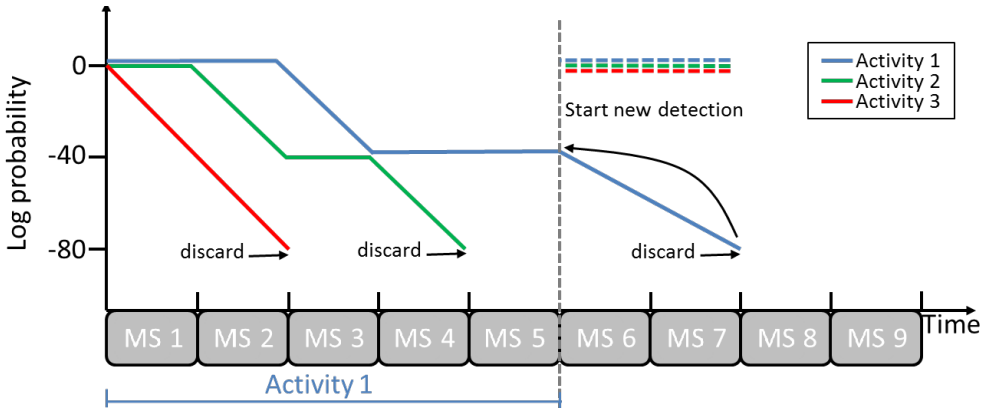


Figure 5.8: Online detection process. The *Activity-2* and *Activity-3* are discarded after the fourth and second time step, respectively, as their log-probability fall below -80. The remaining *Activity-1* is discarded after the seventh time interval. As a result, the five first time intervals are classified as *Activity-1*, and a new detection is started from the sixth time step.

5.3.6 Experiments

To evaluate the efficiency of our approach to recognize the human activity from complex sequences, we conducted several tests on four public benchmark datasets, and compare results with those obtained by state-of-the-art methods.

MSRC-12 dataset. The first test concerns the task of human gesture recognition. The main goal of this experiment is to show how the proposed method deals with actions characterized by repetitions of a single gesture. In particular, we want to evidence the proposed decomposition of a sequence into a set of MSs is capable of managing such variability.

We performed this experiment on the Microsoft Research MSRC-12 dataset, which includes 12 gestures performed by 30 subjects for a total of about 50 sequences per class, where a single gesture is performed several times along a sequence (10 times in most of the cases, but this number may vary from 2 to 15). This variability is indeed important to show how it can affect the recognition accuracy. Only skeleton data is provided in this dataset, so we only use the motion features to describe each segment. Following the same protocol as in Lehrmann et al. [40], only six gestures are considered and a 5-fold cross validation protocol is applied. Results are reported in Table 5.3 as average accuracy across folds in comparison to [40] and [J4].

Table 5.3: MSRC-12. Comparison of the proposed approach with DFM [40] and [J4]. Accuracy is reported in percentage

Class	DFM [40]	Devanne et al. [J4]	Our [J1]
Duck	96.0	**100**	**100**
Goggles	88.0	82.0	**91.6**
Shoot	**85.7**	73.5	83.0
Throw	**90.0**	88.0	**90.0**
Change weapon	87.5	**89.6**	**94.0**
Kick	**98.0**	**98.0**	**98.2**
Mean	90.9	88.5	**92.8**

From Table 5.3, we can notice the proposed approach outperforms [40] for all gesture classes except one (*Shoot*), with an overall accuracy of 92.8%, compared to 90.9% reported in [40]. In addition, the accuracy of the proposed approach increases of about 4% that reported in our previous work [J4], where the decomposition into MSs is not considered.

Cornell Activity dataset 120. The second test concerns the task of human activity recognition, and the tests are conducted on using the Cornell Activity dataset 120 (CAD120) [66]. This dataset contains 120 RGB-D sequences of ten high-level activities involving manipulation with objects, performed by four different subjects three times each. The variability of performed activities, the variability of subject orientation in the scene and the body part occlusion caused by objects make this dataset quite challenging. For a fair comparison with state-of-the-art methods, the *leave-one-person-out* cross protocol is used, and the average accuracy and standard deviation among the four folds are finally computed. Table 5.4 reports results obtained by our method in comparison to state-of-the-art. Our best accuracy is obtained by using a codebook size of 100 for both features. In particular, methods are compared by separating the case in which only the human skeleton is used, from the case in which both skeleton and depth data are considered.

From the results, we can first notice that our method significantly outperforms the other approaches when only skeleton data is used. More specifically, in comparison with [J4], which represents each activity by spatio-temporal trajectory only, the recognition accuracy is improved by more than 20%. This shows that when activities involve complex motions, it is not sufficient to analyze the global motion. Indeed, local analysis and decomposition of the activity into MSs provides a better representation of activities, thus allowing a better understanding of the human behavior. In addition, the accuracy of 69.4% obtained by our method shows that the decomposition of the

Table 5.4: Cornell Activity dataset 120. Comparison of our approach to state of the art methods

Method	Accuracy (%)
Skeleton Only	
Koppula et al. [66]	27.4
Devanne et al. [J4]	48.3
Our [J1]	**69.4 ± 4.1**
Skeleton + Depth	
Koppula et al. [66]	80.6
Koppula and Saxena [94]	**83.1**
Rybok et al. [42]	78.2
Our [J1] (Skel + MLOP)	79.0
Our [J1] (Skel + LOP4D)	**82.3 ± 3.4**

sequence allows us to quite well recognize activity sequences involving objects manipulation, even without describing any explicit information about objects held by the subject. However, results show that using only skeleton data is insufficient to be competitive with state-of-the-art methods. As we can see in Table 5.4, using depth appearance features in addition to skeleton in our DNBC allows us to improve the recognition by about 13%. As a result, we obtain competitive accuracy in comparison with other approaches. Indeed, only [94] is above by less than 1%. Note that methods in [66] and [94] use ground truth object bounding box in the training process. In our case, we do not need this information. Moreover, the small value of standard deviation among the folds shows that our method has a low dependency on training data.

Finally, by comparing the results obtained with our two different depth appearance features, we can notice that the 4DLOP feature is more effective. This observation is strengthened by the confusion matrices in Figure 5.9, and particularly by the confusion obtained for the pair of opposite activities *stacking* and *unstacking objects*. We can see that using the LOP4D feature results in less confusion between the two activities than using the MLOP feature. Indeed, in this particular case, the average depth appearance of *putting* and *taking* the object may be very similar and represented by the same symbol from the codebook. The 4DLOP feature capturing the variation of depth appearance is more suitable to discriminate the two elementary motions, and thus the two activities.

On this dataset, we also evaluate the effectiveness of our method when the value of parameters (size of the codebook and number of DNBC states) is changed. The evolution of the accuracy with respect to both parameters is displayed in Figure 5.10 for both MLOP and LOP4D features. First, it can be observed that the proposed method obtains the best accuracy using both features, when a DNBC with 10 states is trained. It can be also observed that the accuracy is relatively independent from the number of states (except when only three states are used). Second, we can notice the best accuracy is obtained with a codebook of size 50 for the MLOP feature, and a codebook of size 100 for the LOP4D feature. In addition, if too much exemplar features (i.e., 200) are used, the accuracy falls down. Indeed, learning a codebook with too much symbols may result in similar activities represented by different symbols. Hence, symbols represent more a particular sequence performed by one subject, than a generic template of one activity class.

Multi-Modal Action database. The third test concerns the task of online action detection. The Multi-Modal Action Detection (MAD) database [58], has been used to evalaute our method in this context. This RGB-D database has the advantage of including long sequences of 20 subjects per-

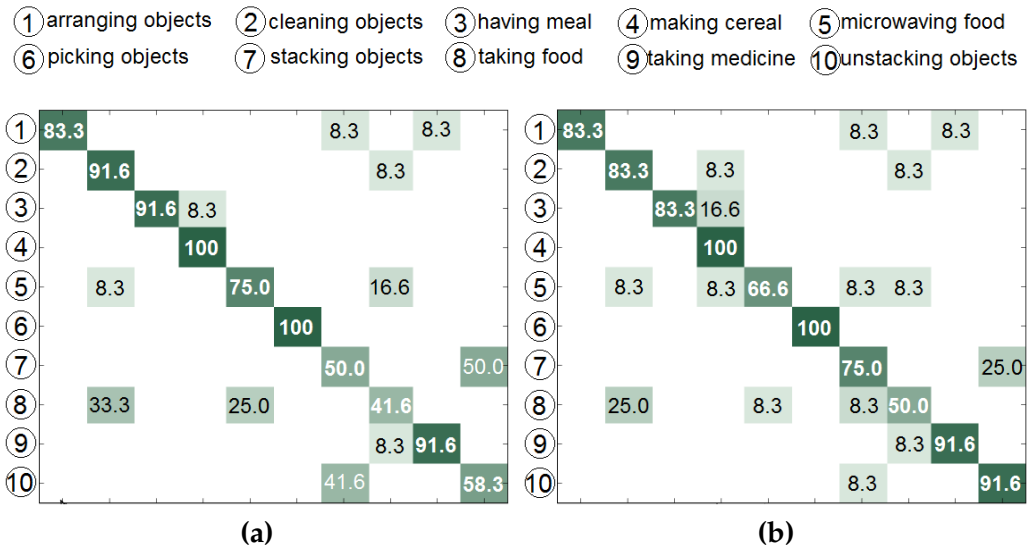


Figure 5.9: Confusion matrices obtained on CAD-120 using MLOP (a), and 4DLOP (b).

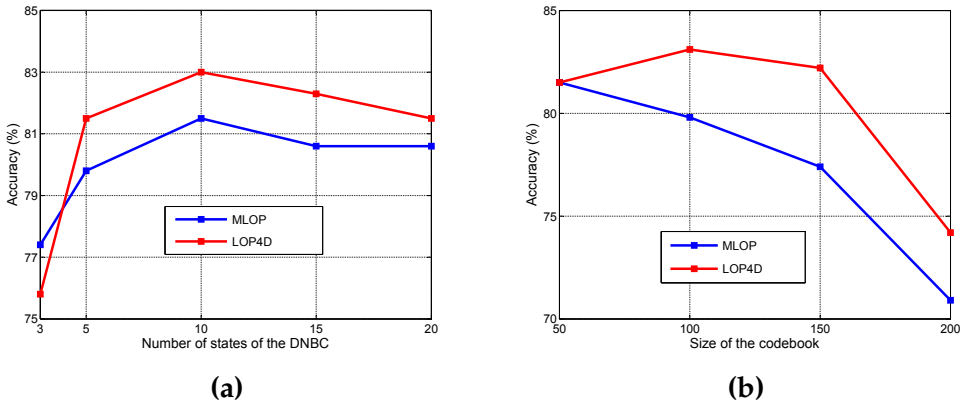


Figure 5.10: Accuracy evolution of our method with respect to varying parameters: the number of states of DNBC (a), and the size of the codebooks (b).

forming successively 35 actions, like *Running*, *Throw* and *Kicking*.

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

Our method is run in an online way as described in Sect. 5.3.5. As a result, we obtain a segmented sequence with an action label for each AU corresponding to the action we detected. In order to evaluate the method and compare it with the state-of-the-art, we compute two measures: *Precision*, which corresponds to the percentage of correctly detected actions over all the detected actions; *Recall*, that is the percentage of correctly detected actions over all the ground truth actions. An action is considered as correctly detected if it overlaps with 50% of the segments of the ground truth action. The ground truth provided by the database authors is obtained by manual labeling

of sequences. We compare these two measures with the SMMED and MSO-SVM methods, both proposed in [58]. The average and standard deviation values among the five folds are reported in Table 5.5. We can see that our method outperforms the state-of-the-art approaches for both the measures.

Table 5.5: MAD database. Comparison of the proposed online detection approach with SMMED [58] and MSO-SVM [58]. The precision and recall measures are computed

Measure (%)	MSO-SVM [58]	SMMED [58]	Our [J1]
Recall	51.4	57.4	**79.7 ± 6.4**
Precision	28.6	59.2	**72.1 ± 5.8**

Fig 5.11 also shows the detection results of one sequence in comparison with the ground truth and the best state of the art method, SMMED, proposed in [58]. We can see that while both our method and [58] are able to accurately detect actions along the time, our method detects more efficiently the end of actions, thus resulting in a duration of detected actions closer to the ground truth. As an overlap of 50% with ground truth is considered as the criterion of good detection, our method obtains higher values of recall and precision.

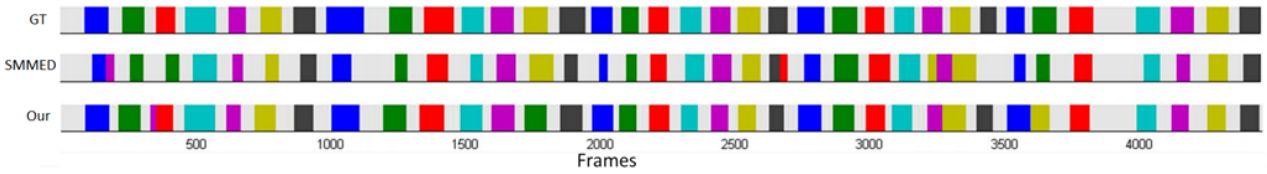


Figure 5.11: Action detection result, for the sequence-1 of subject-1 from the MAD database, of the SMMED method [58] (second row) and the proposed approach (third row) in comparison to the ground truth (first row). Our method provides segments whose duration is closer to ground truth compared to [58].

Online RGB-D dataset. The fourth test concerns the several issues related to the action and activity detection and recognition. The Online RGB-D dataset [44] proposes different types of sequences, which allow evaluation in different contexts, like activity recognition and online activity detection. The dataset contains RGB-D sequences of seven activities, like *drinking*, *eating* or *reading book*. On this dataset, we first evaluate the effectiveness of our method for activity recognition. To this end, we follow the same procedure as in [44] by employing a 2-fold cross validation. We compare our approach with state-of-the-art methods according to the type of features employed. When we use depth features in our method, we use the 4DLOP feature and learn codebooks of different sizes. The best accuracy is obtained for a codebook of size 100. Results are reported in Table 5.6.

Table 5.6: Online RGB-D dataset. Comparison of our approach with state of the art methods for the task of activity recognition

Method	Accuracy (%)		
	Depth	*Skeleton*	*Depth + Skeleton*
DCSF [73]	61.7	-	-
Moving Pose [69]	-	38.4	-
Actionlet [118]	-	-	66.0
DOM [44]	46.4	63.3	71.4
Our [J1]	**64.5 ± 0.7**	**71.8 ± 1.8**	**80.9 ± 1.1**

It can be noticed that the proposed approach outperforms the state-of-the-art methods for every combination of features. It should also be noted that if only depth features are used, our method is not fairly comparable to the others. Indeed, even if we only use depth features to describe MSs, our method still needs skeleton data to identify MSs. Nevertheless, we can see that our segmentation approach allows a good recognition of activities when each segment is only described by depth appearance feature. Compared to skeleton-based methods, our approach significantly outperforms other solutions. This shows that our segmentation approach combined with shape analysis of human motion allows us to efficiently recognize activities involving manipulation of objects. Even without considering any information about objects held by the subject, we are able to recognize 71.8% of the activities. This result is higher than that scored by [118] and [44], which combine both skeleton and depth features. Finally, if we add depth features to the skeleton, the recognition accuracy is increased to 80.9%, which is almost 10% above the best state-of-the-art method [44].

We evaluate also the latency of our approach by measuring the ability to recognize the activity without observing the whole sequence. Hence, the average recognition accuracy is computed on different observed portions of the sequence, as reported in Figure 5.12 in comparison to state-of-the-art. We can notice that the proposed approach outperforms the methods in [73] and [69] for every observation ratio. However, our method exceeds the method proposed in [44] from 40% of observation. Indeed, when we observe less than 40% of the sequence, it often results in activity sequences represented by one or two temporal segments. In these cases, the dynamics of the activity is null (one observation) or very small (two observations). Hence, the use of statistical models like DNBC is not appropriate and efficient for modeling short portions of the activity sequence. Finally, our method allows efficient recognition when half of the sequence is observed (accuracy of 75.6%). This shows that even if our method is not suitable for very early detection of activities (less than 30% of observation), we guarantee a good recognition accuracy when only half of the sequence is observed.

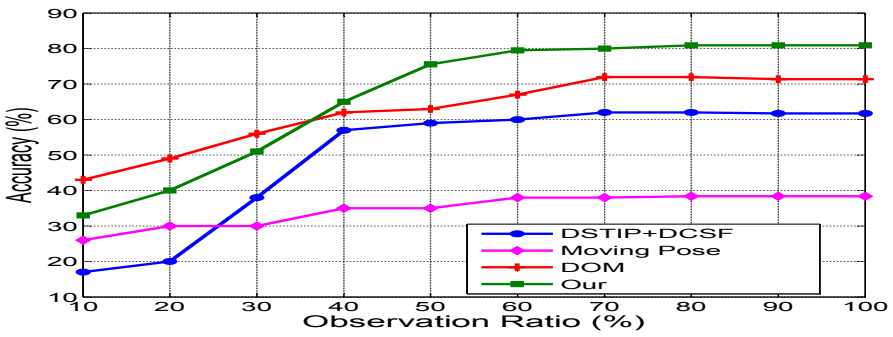


Figure 5.12: Latency analysis on Online RGB-D dataset. Accuracy obtained for different portion of the sequences is compared to and state-of-the-art methods.

Finally, we propose to evaluate our approach for online activity detection. The same set of activities as for activity recognition is used to train one DNBC for each activity class. In addition, we use a set of background activities provided by the dataset, so as to learn the null class. Finally, we run our detection method on a new set of sequences. It includes 36 long sequences from $30sec$ to two minutes, where 12 new subjects are successively performing different activities. Manual labeling provided by the dataset is used as ground truth. Detection is evaluated using a frame-level accuracy as in [44], computed by averaging the number of well classified frames out the all set of frames in the test sequences. Results are reported in Table 5.7. We can see that our method performs better than state-of-the-art approaches to detect activity in an online manner. Using an

unoptimized Matlab implementation with an Intel Core i-5 2.6GHz CPU and a 8GB RAM, we run our detection method at $7fps$.

Table 5.7: Online RGB-D Dataset. Comparison of our approach with state of the art methods for the task of online activity detection

Method	Accuracy (%)
DSTIP + DCSF [73]	32.1
Moving Pose [69]	50.0
DOM [44]	56.4
Our [J1]	**60.9**

5.4 Conclusions

In this chapter, we addressed the issue of human action and activity analysis and recognition from RGB-D data, a widely investigated topic due to its large panel of potential applications. We differentiate the study of behaviors according to their complexity.

In a first time, we focused on the recognition of relatively simple short movements, like actions. To this end, we employed skeleton data provided by RGB-D sensors, which represent the human body pose as a set of connected 3D joints for each frame of the sequence. In order to analyze the action performed by the subject during the sequence, a spatiotemporal modeling of motion trajectories in a Riemannian manifold is proposed. Each motion trajectory is expressed as a point in the open curve shape space. Thanks to the Riemannian geometry of this manifold, action classification is solved using the nearest neighbor rule. The efficiency of the proposed method, verified on three benchmark datasets, demonstrate that our approach outperforms the existing state-of-the-art methods in most of the cases, in terms of recognition and latency. However, experiments also demonstrated that the proposed solution suffers from limitations when actions involve long sequences and/or variable repetitions of a single gesture or manipulations of objects.

In a second time, we extended our approach, so as to simultaneously handle these limitations and considered more complex movements, like activities. Hence, we proposed a segmentation method in order to decompose a motion sequence into a set of short elementary Motion Units. Thanks to a proposed pose-based shape analysis, we decompose a sequence into relevant MSs, which are then represented as motion trajectories and interpreted in the Riemannian shape space in order to capture the dynamics of human motion. In addition, adding depth appearance information enables us to characterize MSs by the motion trajectory and depth appearance around hand joints, so as to describe the human motion and interaction with objects. Finally, the sequence of temporal segments is modeled through a Dynamic Naive Bayesian Classifier. Experiments on several datasets show the potential of our method for the task of human behavior recognition in comparison with state-of-the-art methods. Finally, we adapt our method to allow online behavior detection in long sequences, which is an important challenge in real-world contexts. Evaluation on two datasets demonstrate that the proposed approach outperforms state-of-the-art methods for online detection of human behavior.

Although the appearance of the depth around the articulations of the hand enables to distinguish certain gestures, based on the description of the object in interaction, its effectiveness remains limited in a complex scenario of human-object interactions. Interpreting fine hand gestures is a critical problem for understanding human behavior. Among human body parts, hands are the most effective and intuitive interaction tools in Human-Computer Interaction applications. Thus, hand

gesture analysis and recognition present a crucial task to achieve a deeper understanding of the behavior.

CHAPTER 6

On Hand Gesture Recognition *Migrate from Handcrafted to Deep Learning Approaches*

This chapter presents our contributions on 3D hand gesture recognition problems. We first discuss a traditional handcrafted approach using hand shape and motion descriptors computed on 3D hand skeletal features. Then, we extend the study of hand gesture analysis to online recognition. Using a deep learning approach, we employ a transfer learning strategy to learn hand posture and shape features from depth image dataset originally created for hand pose estimation. We propose this methodological evolution since we are convinced that the understanding of traditional approaches as well as their challenges and limitations will help to understand modern deep learning approaches and how they can be used to improve results. During the description of the approaches, we highlight challenges hand pose detection and its motion description and potential ways to overcome them. The contributions in this chapter originate from the work done by Quentin De Smedt during his Ph.D thesis [3]. It is structured as follows. After introducing the problem and reviewing the related work in Section 6.1, our handcrafted based approach is described in Section 6.2, followed by experimental testes. In Section 6.3, we present an evolution towards deep learning approaches. Finally, strengths of our approach in terms of online detection and recognition are demonstrated on two datasets before concluding in Section 6.4.

The contribution presented in this chapter were published in the conference/workshop papers [C1, C2, C5] and *under review* in the submitted journal papers [P1, P3], and from where some parts of this chapter are extracted.

6.1 Context

Among human body parts, the hand is an effective and intuitive interaction tool in most Human-Computer Interaction (HCI) applications. Consequently, hand gesture recognition is becoming a central key for different types of applications such as virtual game control, sign language recognition, HCI, robot control, etc.

Using hand gestures as a Human-Computer Interaction (HCI) modality introduces intuitive and easy-to-use interfaces for a wide range of applications in virtual and augmented reality systems, offering support for the hearing-impaired and providing solutions for all environments using touchless interfaces. However, the hand is an object with a complex topology and has endless possibilities to perform the same gesture. For example, Feix *et al.* [177] summarize the grasping taxonomy and found 17 different hand shapes to perform a grasp. Grasping is a hand gesture where we need precise information about the hand shape if we want to recognize it. Other gestures, such as *swipes*, which are defined more by hand motions than its shape, are already commonly used in tactile HCI. Thus, the heterogeneity between useful gestures have to be taken into account in a hand gesture recognition algorithm.

The area of hand gesture analysis covers hand pose estimation and gesture recognition. Hand pose estimation is considered to be more challenging than other human part estimation due to the

small size of the hand, its greater complexity and its important self occlusions. Beside, the development of a precise hand gesture recognition system is also challenging. Different occurrences of the same gesture type contain high dissimilarities derived from ad-hoc, cultural and/or individual factors in the style, the position and the speed of gestures. In addition, gestures with different meanings contain high similarities derived from the heterogeneity of possible gestures.

To date, most reliable tools used to capture 3D hand gestures are motion capture devices, which have sensors attached to a glove delivering real-time measurements of the hand. However, they present several drawbacks in terms of the naturalness of the hand gesture and cost, in addition to their complex calibration setup process. Recently, effective and inexpensive depth sensors, like the Microsoft Kinect, have been increasingly used in the domain of computer vision. By adding a third dimension into the game, depth images offer new opportunities to many research fields, one of which is the hand gesture recognition. Hence, many research work have addressed 3D hand gesture recognition challenges using depth images [19, 27, 33, 52, 53, 63] .

In meantime, deep neural networks have proven their outstanding effectiveness of many area of research. Indeed, they allowed researchers to make a jump in robustness and efficiency in hand pose estimation. However, deep learning algorithms are data-hungry and annotating hand pose datasets is very time-consuming. Similarly to hand pose estimation, methods using deep learning for the task of hand gesture recognition showed also an improvement in the robustness and the efficiency of new algorithms based on learned features compared to traditional handcrafted descriptors. However, in the same way, current available hand gesture datasets are small in size and strategies have to be used to overcome the hungriness of deep learning algorithms. Nevertheless, the existence of large datasets made for the hand pose estimation issue can be used to pre-train a deep model for the hand gesture recognition task.

Despite an increasing amount of methods proposed over the last few years, defining an online dynamic hand gesture recognition system, robust enough to work in real world applications, remains a challenge. Dynamic hand gestures can be defined by shape variations of the hand during sequences (e.g. fine gestures performed by fingers), or by hand movements (e.g. swipe gestures), and often both. These multiple characteristics, which have to be taken into account, make harder the process of feature learning as it has to learn mutually spatial and temporal information.

In order to fully extract relevant features of complex hand gestures using raw data, models of neural networks need a large number of layers which increase their computation complexity. However, the computational complexity has to be small enough so that the algorithm can predict a new incoming gesture in real time. Some methods present acceptable runtime results using very deep networks but use a powerful hardware with several GPUs. Currently, this hardware configuration is too much expensive and so not suitable for real-world applications.

6.1.1 Related Work

Gesture recognition has been a widely explored topic in computer vision. Over the past few years, advances in inexpensive 3D depth sensors have substantially promoted the research of hand gesture detection and recognition. We will focus only in reviewing the works on 3D hand gesture recognition we consider relevant to two main categories – handcrafted and deep learning based methods – using depth images.

In traditional **handcrafted** approaches, 3D depth information is used to recognize hand silhouettes or simply hand areas in order to extract features from a segmented hand region [27, 84, 93, 150]. The temporal aspect of hand motion is also considered by Kurakin *et al.* [132], where they presented the MSRGesture3D dataset containing 12 dynamic gesture from the American Sign

Language. Their recognition algorithm is based on a hand depth cell occupancy and a silhouette descriptor. They used an action graph to represent the dynamic aspect of the gestures. Recently, using a histogram of 3D facets to encode 3D hand shape information from depth maps, Zhang *et al.* [68] outperformed latest results obtained on the MSRGesture3D dataset using a dynamic programming-based temporal segmentation. One of the tracks of the *Chalearn 2014* [63] consists of using a multimodal database of 4000 gestures drawn from a vocabulary of 20 dynamic Italian Sign Language gesture categories. They provided sequences of depth images of the whole human body and body skeletons. From this dataset, Monnier *et al.* [53] employed both body skeleton and Histogram of Oriented Gradients (HOG) features computed on the depth map cropped around the hand to perform a gesture classification using a boosted cascade classifier.

In order to study hand gesture recognition in a real-time scenario for automotive interfaces, Ohn-Bar and Trivedi [52] made a publicly available dataset of 19 gestures performed in a car captured with the Microsoft Kinect. The initial resolution obtained by such a sensor is 640×480 and the final region of interest is 115×250. Moreover, at some distance from the camera, with the illumination varying in the car, the resulting depth is very noisy, making the challenge of gesture recognition tougher. They compared the accuracy of gesture recognition using several known depth features (HOG, HOG3D, HOG^2). In a previous work, De Smedt *et al.* [C5] investigate the use of a hand skeleton model in a dynamic hand gesture recognition solution. It includes three gestural features representing the hand shape and the motion information computed on the skeletal data of the hand in addition to a temporal encoding of the gesture dynamics. The evaluation of this approach on three hand gesture datasets containing a set of fine and coarse heterogeneous gestures, shows a promising way to perform hand gesture recognition with a skeletal-based approach. Nevertheless, this approach has shown some weaknesses compared to deep neural network-based models, to represent the complex dynamic and temporal information of a hand gesture.

Like many research areas in pattern recognition, **deep learning** approaches have recently shown a particularity high performances for hand gesture recognition. Their ability to learn relevant spatial and/or temporal features in addition to play the role of classifier, has been studied last years. Convolutional neural networks [214] designed to take images as input have been used for static hand gesture recognition using RGB data [54, 152] and/or depth maps [35]. Neverova *et al.* [18] designed a multi-modal deep learning framework which takes as inputs: RGB, depth, audio stream and body skeleton data. Their network captured several spatial information, such as motions of the upper body or the hand, at three distinct spatial scales in order to perform dynamic sign language recognition. Their framework classified each frame and the final label of a sequence was computed using a majority vote. Molchanov *et al.* [33] proposed a dynamic hand gesture algorithm using a two-stream 3DCNN which takes as inputs stacked image gradients and depth maps to classify sequence of images. They later enhanced their method and proposed a dynamic hand gesture algorithm – called R3DCNN [19] – using a larger 3DCNN, previously defined by Karpathy *et al.* [56], to extract features from sub-sequences followed by a recurrent layer to model the temporal aspect of gestures. The 3DCNN was composed of eight convolutions with an increasing number of filters in order to get both spatial and temporal features on sequences of RGB and depth images. In addition, they used a Connectionist Temporal Classification [194] as the cost function. While it has been initially designed to perform prediction of sequence in an unsegmented input streams, the CTC is applied here to perform online classification. To overcome the hungriness of deep learning algorithms, they pre-trained their model on the large-scale Sport-1M [56] human action recognition dataset. If they claimed to obtain real-time results, they used a powerful hardware configuration not suitable for public use.

The recognition algorithms of dynamic hand gestures based on skeletal data are not yet well represented in the literature. This is due to the fact that hand pose estimation methods begin only recently to be robust and efficient in challenging contexts. Lu *et al.* [20] used a neural based variant of the HCRF which were fed with features computed on hand skeletal data captured via a Leap Motion Controller.

However, the problem of modeling skeletal data sequences with deep neural networks has been studied in the field of action recognition. Wang *et al.* [5] used a two-stream Recurrent Neural Network (RNN) architecture for skeleton based action recognition. One stream was used in order to model temporal information while the other focus on spatial cues. Garcia *et al.* [8] used a two-stacked Long-Term Short Memory (LSTM) network as a baseline for their hand action dataset. LSTM has shown better performance over all previous traditional methods. Du *et al.* [37] proposed to divide the human body skeleton in five meaningful parts and fed each one into a distinct RNN network. They used a bidirectionnal variant [215] of the LSTM in order to use past frames but also future one to model each time step of a sequence. The recurrent layers are then fused step by step to be inputs of higher layers. Very recently, Liu *et al.* [6] defined a global context-aware attention LSTM networks for skeleton-based action recognition. The idea behind their approach is that an upstream recurrent network processes an incoming sequence and update a context memory cell which allowed to extract the potential importance of each body joint in the sequence. A second LSTM performed the classification paying attention more precisely at some joints using the context memory.

6.1.2 Challenges and motivations

The development of a precise dynamic hand gesture recognition system, able to take into account the heterogeneity of possible gesture types, presents some important challenges. Indeed, the intraclass gesture dissimilarities, which come from *ad-hoc*, cultural and/or individual factors in the style, position and speed of gestures. Indeed, two different actors rarely perform the same gesture in the same way. These variations are caused by differences of dexterity, size or yet again culture. In fact, even a particular subject never perform the same gesture twice in the same way. Other than its position relative to the camera can change, when a user performs a particular type of gesture multiple times, he makes it his own. It follows sometimes large differences with the example given at the beginning. In addition, interclass similarities, which come from high similarities between different types of gestures, represent an important factor. Furthermore, these similarities are exacerbated by deformations due to intraclass variations. Finally, some hand gestures can only be described by hand shape variations through time. However, a hand is a small object with a high degree of freedom and with a high potential of self-occlusion. It is very hard to extract precise information of the hand shape based on data captured using long-ranged depth sensors with a low image resolution such as the first version of Microsft Kinect. In addition, the noise of depth images, self-occlusions and environmental variations in the viewpoints make the study of hand shape very challenging. Nevertheless, new short-ranged depth devices enable more precise hand capture (e.g. Intel RealSense or the SoftKinetic DS325).

6.2 Presegmented Hand Gesture Recognition using 3D Dynamic Skeletal Data

In the field of action recognition, Shotton *et al.* [78] proposed a real-time method to accurately predict the 3D positions of 20 body joints from depth images. Hence, several descriptors in the literature proved how positions, motions, and orientations of joints could be excellent descriptors for human actions. Following this statement, hand skeletal data could also handle precise information of the hands that HCI needs in order to use them as a manipulation tool.

Recently, new devices, such as the Intel RealSense or the Leap Motion Controller (LMC), provide, in addition to depth images, precise **skeletal data** of the hand and fingers in the form of a full 3D skeleton corresponding to twenty or so joints in $\mathbb{R}^3$. Potter *et al.* [85] presented an early exploration of the suitability of using such data from a LMC in order to recognize and classify precise hand gestures in Australian Sign Language. However, hand pose estimation from depth images remains a prominent field of research. Many issues still have to be solved: properly recognizing the skeleton when the hand is either closed, perpendicular to the camera, or without an accurate initialization, or when the user performs a quick gesture. The hand contains more joints than there are in the rest of the human body model of Shotton *et al.* [78] and is a smaller object. The hand has also a more complex structure. If an arm, a head or a leg have different shapes, the hand is composed of a palm and five similar fingers making its pose estimation more challenging.

6.2.1 Approach overview

To face challenges of dynamic hand gesture recognition, we introduce an original approach using three features computed on hand skeletal feature sequences to classify unknown hand gestures.

Our proposed method is a hand skeleton-based approach since we consider those features contain precise information about hand motions and shape variations information. In addition, new devices are able to directly provide us hand skeletal features. However, even if 3D joint positions of hand skeleton are available, the hand gesture recognition task is still challenging due to significant spatial and temporal variations in the way of performing a gesture.

First, we use a temporal pyramid to represent the dynamic aspect of gestures. We cut sequences in overlapping sub-sequences. On each sub-sequences, we compute three set of features: a set of direction vector which the hand is taking through the sequence, a set of rotation and a hand shape descriptor called *Shape of Connected Joints*. Those sets are then transformed into a statistical representation vector using a Fisher Kernel. The final gesture descriptor is the concatenation of the three statistical representation features computed for each sub-sequence. Finally, a linear SVM is used to perform classification.

Since skeleton based approaches became popular thanks to Shotton *et al.*'s approach [78], more and more datasets dedicated to human action and activity recognition from human skeleton have been created [16, 80, 115, 163]. However, in the context of hand gesture recognition, there was no publicly released dynamic hand gestures dataset providing labeled sequences of depth *and* hand skeletal features. We are therefore encouraged to collect a dataset with this type of information data [C5].

6.2.2 3D Hand Pose Estimation

The task of hand pose estimation aims to map an observed input, generally a 2D or a 3D image, to a set of 3D joints together forming a possible hand configuration called hand skeleton or hand

pose, which takes into account the anatomic structure constraints of the hand as depicted in Figure 6.1. The hand pose estimation community has rapidly grown larger in recent years. The introduc-

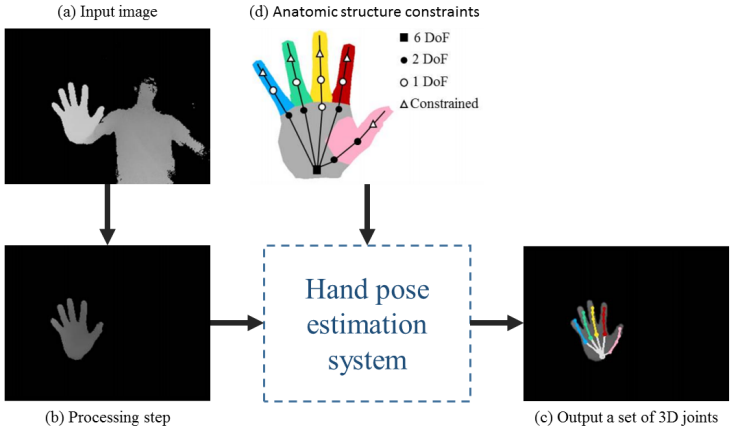


Figure 6.1: Illustration of the hand pose estimation task. (a) from an input image and after (b) a pre-processing step, a hand pose estimation system is able to (c) output a set of 3D joints called together hand skeleton or hand pose based on (d) the anatomic structure constraints of the hand.

tion of commodity depth sensors and the multitude of potential applications have stimulated new advances. However, it is still challenging to achieve efficient and robust estimation performance because of large possible variations of hand poses, severe self-occlusions and self-similarities between fingers in the depth image. The current state-of-the-art methods mostly employ deep learning approaches to estimate hand pose from a depth image [11, 14, 24, 31, 32, 47]. In meantime, the availability of a large-scale, accurately annotated dataset is a key factor for advancing this field of research. Consequently, numerous RGB-D datasets have been made publicly available last years. Currently, the widely used datasets in the literature for benchmarking purposes are the ICVL [48] and the NYU [47] datasets. In addition to depth images, the software development kit of the Intel RealSense SR300 [17] provides a stream of 3D full hand skeleton of 22 joints at 30 frames per second. Beside, in July 2013, the Leap Motion Controller (LMC) is launched on the public market, which was primarily designed for hand tracking, provides 3D full hand skeleton of 24 joints. Such data offer new opportunities and axes of research related to hand gesture analysis.

6.2.3 Skeletal feature extraction

Using 3D hand skeletal data (an example can be seen in Figure 6.3a), a dynamic gesture can be seen as a time series of hand skeletons. It describes the motion and the hand shapes along the gesture. For each frame t of the sequence, the position in the camera space of N_j joint which are represented by three coordinates, i.e. $j_i(t) = [\ x_i(t)\ y_i(t)\ z_i(t)\ ]$. N_j is the number of joints which compose the hand skeleton. The skeleton at frame t is then represented by the $3N_j$ dimension row vector:

$$s(t) = [\ x_1(t)\ y_1(t)\ z_1(t)\ ...\ x_{N_j}(t)\ y_{N_j}(t)\ z_{N_j}(t)\ ] \tag{6.1}$$

With N_f representing the number of frames in the sequence, the final representation of the sequence is a matrix of size $N_f \times 3N_j$ where each line t is the row vector $s(t)$:

$$\mathcal{M} = \begin{bmatrix} s(1) \\ \vdots \\ (N_f) \end{bmatrix} \tag{6.2}$$

This new type of data handles a lot of information on the motion and the shape of the hand along the sequence. In order to fully represent the gesture, we propose to mainly capture the hand shape variations based on skeleton joints, but also the direction of the movement and the rotation of the hand with three distinct features.

Motion features. Some gestures are defined almost only by the way the hand moves in space (e.g. *swipes*). To take this characteristic into account, we compute a direction vector in $\mathbb{R}^3$ for each frame t of our sequence using the position of the palm joint noted j_{palm}:

$$\overrightarrow{d}_{dir}(t) = \frac{j_{palm}(t) - j_{palm}(t-c)}{\|j_{palm}(t) - j_{palm}(t-c)\|} \tag{6.3}$$

where c is a constant value chosen experimentally. We normalize the direction vector by dividing it by its norm.

For a sequence of N_f frames, we have the set $\mathcal{S}_D$:

$$\mathcal{S}_D = \left\{ \overrightarrow{d}_{dir}(t) \right\}_{[\,1 < t < N_f\,]} \tag{6.4}$$

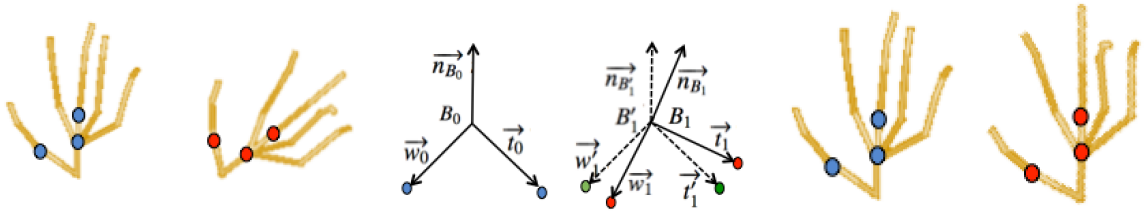
The rotation of the wrist during the gesture describes also how the hand is moving. For each frame t, we compute the vector from the wrist node to the palm node to get the rotational information in $\mathbb{R}^3$ of the hand:

$$\overrightarrow{d}_{rot}(t) = \frac{j_{palm}(t) - j_{wrist}(t)}{\|j_{palm}(t) - j_{wrist}(t)\|} \tag{6.5}$$

For a sequence of N_f frames, we have the set $\mathcal{S}_R$:

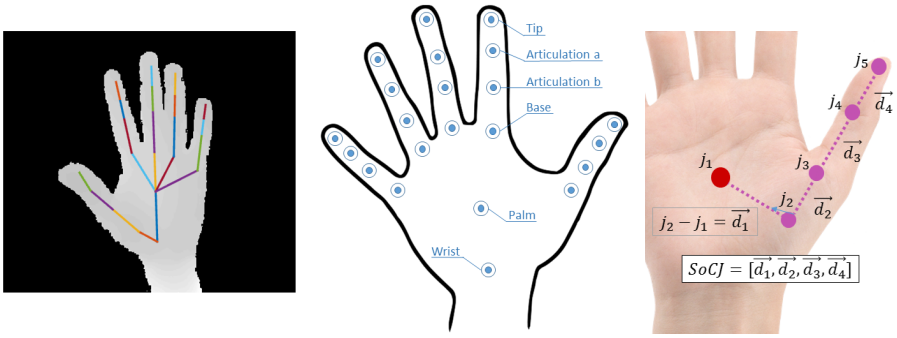
$$\mathcal{S}_R = \left\{ \overrightarrow{d}_{rot}(t) \right\}_{[\,1 < t < N_f\,]} \tag{6.6}$$

Shape features. To represent the shapes of the hand during the sequence using skeleton data, we propose a descriptor based on sets of joints, denoted as *Shape of Connected Joints* (SoCJ). Hand skeleton returned from sensors consists of 3D coordinates of joints, represented in the camera coordinate system. Therefore, they vary with the rotation and translation of the hand with respect to the camera. To make our hand shape descriptor invariant to hand geometric transformations, we propose a normalization phase. Firstly, in order to take into account the differences of hand size between performers, we estimate the average size of each bone of the hand skeleton using all hands in the dataset. Secondly, carefully keeping the angles between bones, we change their size by their respective average size found previously. Indeed, in order to be consistent with the translation and rotation transformations, we create a reference skeleton hand H_f corresponding to an open hand in front of the camera with its palm node at $[0\ 0\ 0]$ as the *root joint*. Then, we define a new base with origin in the root joint, which includes the wrist node vector $\overrightarrow{w}$, the base of the thumb node vector $\overrightarrow{t}$, and their cross product $\overrightarrow{n_B} = \overrightarrow{w} \times \overrightarrow{t}$. This new base is then translated and rotated, so as



(a) **(b)** **(c)**

Figure 6.2: The calculation of the optimal rotation between the two hand skeletons using SVD: (a) Two skeletons with different orientations; (b) Bases B_1 and B_2 are built from the two corresponding wrists; (c) The resulting aligned skeleton (right) is now aligned with respect to the first one (left).



(a) **(b)**

Figure 6.3: Hand skeletal structure: (a) Depth and hand skeletal data returned by the Intel RealSense camera, and (b) An example of the SoCJ descriptor constructed around the thumb tuple. Let be $T = (j_1, j_2, j_3, j_4, j_5)$ where $j_i \in \mathbb{R}^3$. We compute the displacements from points to their respective right neighbor resulting in the SoCJ vector $\left[\vec{d_1}, \vec{d_2}, \vec{d_3}, \vec{d_4}\right]$.

to be aligned with a reference base B_0 computed from H_f. The calculation of the optimal rotation between the two bases B_1 of a current skeleton and B_0 of the reference skeleton H_f, is performed using *Singular Value Decomposition* (SVD). This process results in a new hand which keeps its shape but centered around $[0\ 0\ 0]$ with the palm facing the camera. For each gesture sequence, we compute the translation and the rotation of the first hand skeleton with respect to the H_f and then apply the same transformations to all other hand skeletons of the sequence. This guarantees the invariance of the representation to the position and orientation of the hand in the scene. Figure 6.2 shows an example of alignment of two different hand skeletons.

Let x represent the coordinates of a joint in $\mathbb{R}^3$ and $T = [x_1\ x_2\ x_3\ x_4\ x_5]$ a tuple of five ordered different joints from the hand skeleton s. To describe the shape of the joint connections, we compute the displacement from one point to its right-hand neighbor:

$$SoCJ(T) = [\ x_2 - x_1\ ...\ x_5 - x_4\ ] \tag{6.7}$$

This results in a descriptor in $\mathbb{R}^{12}$. Figure 6.3b shows an example of a particular SoCJ using the palm's joint and the thumb's. We remind that the skeleton of the Intel RealSense camera is composed of 22 joints. Theoretically, with $\mathcal{C}$ binomial coefficient function , we can compute

$\mathcal{C}(22,5) = 26334$ different SoCJs for the hand skeleton s resulting in the set:

$$s_{socj} = \{\ SoCJ(i)\ \}_{[\ 1 < i < 26334\ ]} \tag{6.8}$$

For a sequence of N_f frames, we have the set $\mathcal{S}_{socj}$:

$$\mathcal{S}_{socj} = \{\ s_{socj}(t)\ \}_{[\ 1 < t < N_f\ ]} \tag{6.9}$$

6.2.4 Feature modeling and classification

The Fisher Vector (FV) coding method was first introduced for large-scale image classification. Its superiority against the Bag-Of-Word (BOW) method has been analyzed in the image classification [83] as it is going beyond count analysis. It encodes additional information about the distribution of the descriptors. It also has been used over the past five years in action recognition [12, 51, 61, 75]. As a particular hand gesture is so far represented by three sets of features, we aim to use the FV coding method to obtain a statistical representation vector for each of them. First, we train a K-component Gaussian Mixture Model (GMM) using all sets of a particular feature in the training set. Once we have the set of Gaussian Models, we can compute our FV which is given by the derivatives of gradient. We also normalize the final vector with a $l2$ and power normalization to eliminate the sparseness of the FV and increase its discriminability. We refer the reader to Sanchez *et al.* [83] for more details. It is also interesting to notice that the final size of a FV is $2dK$ where d is the size of the feature and K the number of clusters used in the GMM. This observation is a drawback compared to BOW, which has a size of K, when applied to a long descriptor. However, this effect can be ignored in our case where K is relatively small.

To model the dynamics of movement, we use a Temporal Pyramid (TP) representation already employed in action and hand gesture recognition approaches [61, 68]. The principle of the TP is to divide the sequence into n sub-sequences at each n^{th} level of the pyramid.

After feature extraction, we represent a sequence of hand skeletons by three sets of different features describing the direction of the hand ($\mathcal{S}_D$), its rotation ($\mathcal{S}_R$) and its shape ($\mathcal{S}_{socj}$) during the sequence. Adding more levels to the pyramid gives more temporal precision but increases the size of the final descriptor and the computing time substantially. For gesture classification, we use a supervised learning classifier SVM with a linear kernel as it easily deals with our high-dimensional representation. We employ a *one-vs-rest* strategy resulting in G binary classifiers, where G is the number of different gestures in the experiment. We make use of the implementation contained in the LIBSVM library [157].

6.2.5 Experiments

We first evaluate our proposed approach on two datasets and compare it with four state-of-the-art methods using depth images and skeletal data. We then explore its capa- bility to reduce the latency of the recognition process by evalu- ating the trade-off between accuracy and latency. We also study the impact of the hand pose estimation on a third dataset and finally discuss the promising potential of our approach and lim- itations.

In this section, we first introduce the experimental settings of our method related to descriptor encoding and the impact of the hand pose estimation algorithm on the recognition process. Second, we evaluate our proposed approach on three datasets (DHG 14-28 [C3], Handicraft-Gesture [20] NVIDIA Dynamic Hand Gesture [19]) and compare it to several state-of-the-art methods using depth images and skeletal data, before concluding the section with discussions.

Encoding settings. We choose the number of levels L_{pyr} of the TP as equal to 4 as it provides a satisfactory compromise between the temporal representation of gestures and the final size of our descriptor. The final size of our computed descriptor is then $(\sum_{i=1}^{L_{pyr}} i) \times (size_{\Phi D} + size_{\Phi R} + size_{\Phi SoCJ})$, where $size_{\Phi_x}$ is the FV representation computed from the set of features x. Note that $size_{\Phi} = 2dK$, where K is the number of models created in the GMM. d is the feature dimension: respectively in $\mathbb{R}^3$, $\mathbb{R}^3$ and $\mathbb{R}^{12}$ for the direction, the rotation and the SoCJ features. For FV encoding, we map our descriptors into a K-component GMM with K equal to 8, 8 and 256 gaussians respectively for the direction, the rotation and the SoCJ features. For all experiments conducted on the datasets, we use a *Leave-One-Subject-Out cross-validation* protocol.

Influence of hand pose estimation on gesture recognition. The introduction of commodity depth sensors and the multitude of potential applications have stimulated new advances inside the hand pose estimation community. However, it is still challenging to achieve efficient and robust estimation performance because of large possible variations of hand poses, severe self-occlusions and self-similarities between fingers in the depth image. The current state-of-the-art methods mostly employ deep neural networks to estimate hand pose from a depth image [11, 14, 24, 31, 47]. The availability of a large-scale, accurately annotated dataset is a key factor for advancing this field of research. Consequently, numerous RGB-D datasets have been made publicly available last years. The different hand pose datasets differ in the annotation protocol used, the number of samples, the number of joints in the hand skeleton representation, the view point and the depth image resolution. Currently, the widely used datasets in the literature benchmarking purposes are IVCL [39], NYU [47] and MSRA15 [29]. The IVCL [39] and the MSRA15 [29] datasets are captured using the Intel Creative depth sensor (time-of-light), and composed respectively of 180K and 76.5K ground truth annotated depth images with the 3D joint locations of the hand. The NYU [47] comprises 72K frames of multi-view depth images captured using the Primesense Carmine camera (structured light). In order to measure the effect of pose estimation on gesture recognition, we performed several experiments on the two first datasets as their capture technology corresponds to the used hand gesture datasets in this work. First, we evaluate three hand pose estimators on DHG dataset, using the methods proposed by Oberweger *et al.* [31] and Ge *et al.* [24] in addition to the Intel RealSense estimator [17]. We used in these experiments the region-of-interest of the hand returned by Intel RealSense camera as input to the hand pose estimator algorithms instead of a particular hand extraction algorithm, without any preprocessing step. Both estimators [24, 31] were trained on both datasets to select the best training one. Tests showed an improvement of 4% of the recognition accuracy for Oberweger *et al.*'s estimator using IVCL dataset for training. However, they did not reveal any significant effect of the used dataset for Ge *et al.*'s estimator, used for training the pose estimator, on the gesture recognition result. Thus, we choose the IVCL dataset for all the training phase of the two estimators. Fig. 6.4 shows the recognition accuracies on our DHG-14 dataset per class of gestures. The average accuracies by estimator, available in Table 6.1, show that our method performs well independently to the pose estimation method.

Table 6.1: Average recognition accuracies obtained on the DHG-14 dataset using three hand pose estimators.

Hand pose estimation algorithm	Accuracy (%)
Ge *et al.* [24]	86.92
Oberwerger *et al.* [31]	86.24
Intel Realsense [17]	86.86

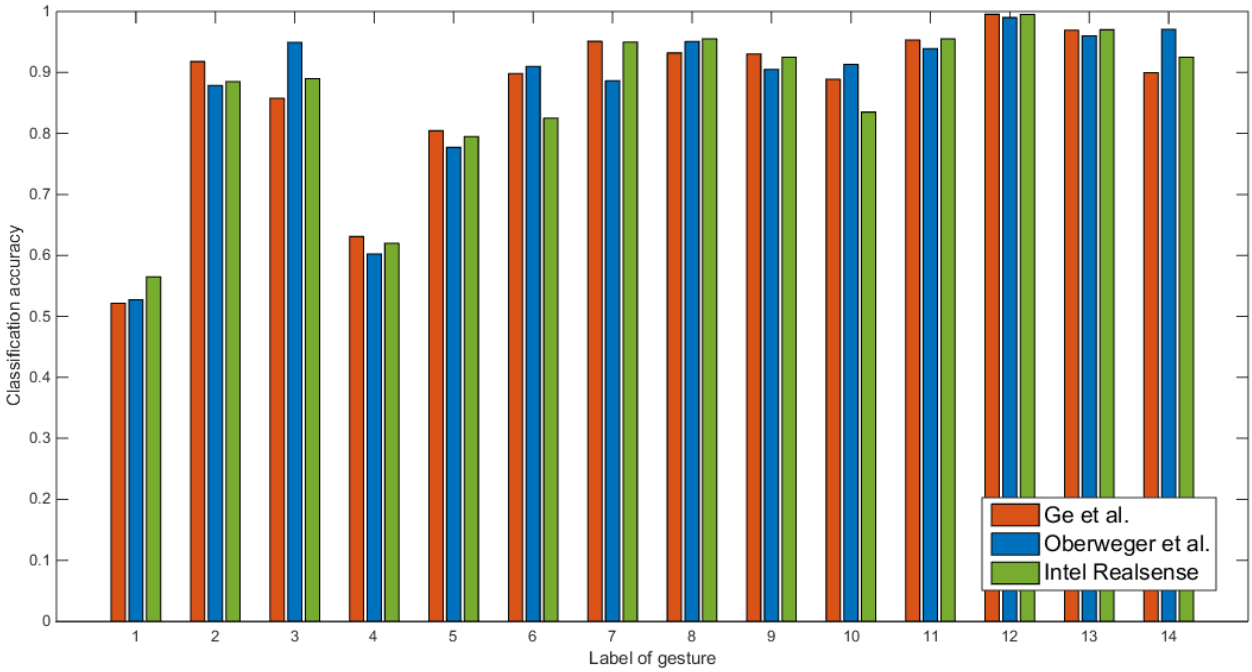


Figure 6.4: Recognition accuracies per class of gesture on the DHG-14 dataset following three hand pose estimators.

Evaluation on DHG 14-28 dataset. The Dynamic Hand Gesture DHG 14/28[1] dataset [C5] contains 14 heterogeneous dynamic hand gestures performed in two ways: using one finger or the whole hand (see Table 6.2). Each gesture is performed five times by 20 volunteers in two ways, resulting in 2800 sequences. Sequences are labeled following their gesture, the number of fingers used and the performer. The Intel RealSense short range depth camera is used to collect the dataset. Each frame contains a depth image and the coordinates of 22 positions of hand joints in the 3D camera space. To assess the effectiveness of our algorithm to classify gestures of the DHG dataset into 14 classes, we compare the results obtained by the hand shape and motion descriptors separately. Table 6.3 presents the accuracies of our approach obtained using each of our descriptors independently and by combining them. For clarity, we divide the results by coarse and fine gestures, allowing us to analyze the impact of each descriptor on each gesture category.

Using all descriptors (direction + rotation + SoCJ) presented in Section 6.2, the final accuracy of our algorithm on the DHG-14 is 86.86%. It rises to 93.77% recognition for the coarse gestures, but for the fine ones the accuracy drops below 75%. A large difference can be observed between accuracies obtained for the fine and the coarse gestures, respectively 44.60% and 88.50% when using only the direction. The analysis of the results obtained using only the SoCJ descriptor shows that the hand shape is the most effective feature for the fine gestures with an accuracy of 67.84%. On the other hand, this result shows that the hand shape is also a way to describe coarse gestures with a fair accuracy of 63.12%. If the rotation descriptor shows a low average accuracy of 50.50% for both fine and coarse gestures, it is a valuable feature for pairs of similar gestures such as *Rotation CW* and *Rotation CCW*. These results confirm the interest of using several descriptors in order to completely describe hand gestures. To better understand the behavior of our approach according to the recognition per class, the confusion matrix is illustrated in Figure 6.5. The first observation is that 11 gestures out of 14 have scored higher than 85.00%. The second observation is the low

[1] Available on: http://www-rech.telecom-lille.fr/DHGdataset

Table 6.2: Gesture list included in the DHG 14-28 dataset.

Index (14)	Index (28)	Gesture	Labelization
1	1, 2	Grab	Fine
2	3, 4	Expand	Fine
3	5, 6	Pinch	Fine
4	7, 8	Rotation CW	Fine
5	9, 10	Rotation CCW	Fine
6	11, 12	Tap	Coarse
7	13, 14	Swipe Right	Coarse
8	15, 16	Swipe Left	Coarse
9	17, 18	Swipe Up	Coarse
10	19, 20	Swipe Down	Coarse
11	21, 22	Swipe X	Coarse
12	23, 24	Swipe V	Coarse
13	25, 26	Swipe +	Coarse
14	27, 28	Shake	Coarse

Table 6.3: Accuracy comparison fine / coarse / both gesture for the DHG-14 dataset.

Features	Fine (%)	Coarse (%)	Both (%)
Direction	44.60	88.50	72.79
Rotation	50.30	50.61	50.50
SoCJ	67.84	63.12	64.88
SoCJ + Direction + Rotation	74.43	93.77	**86.86**

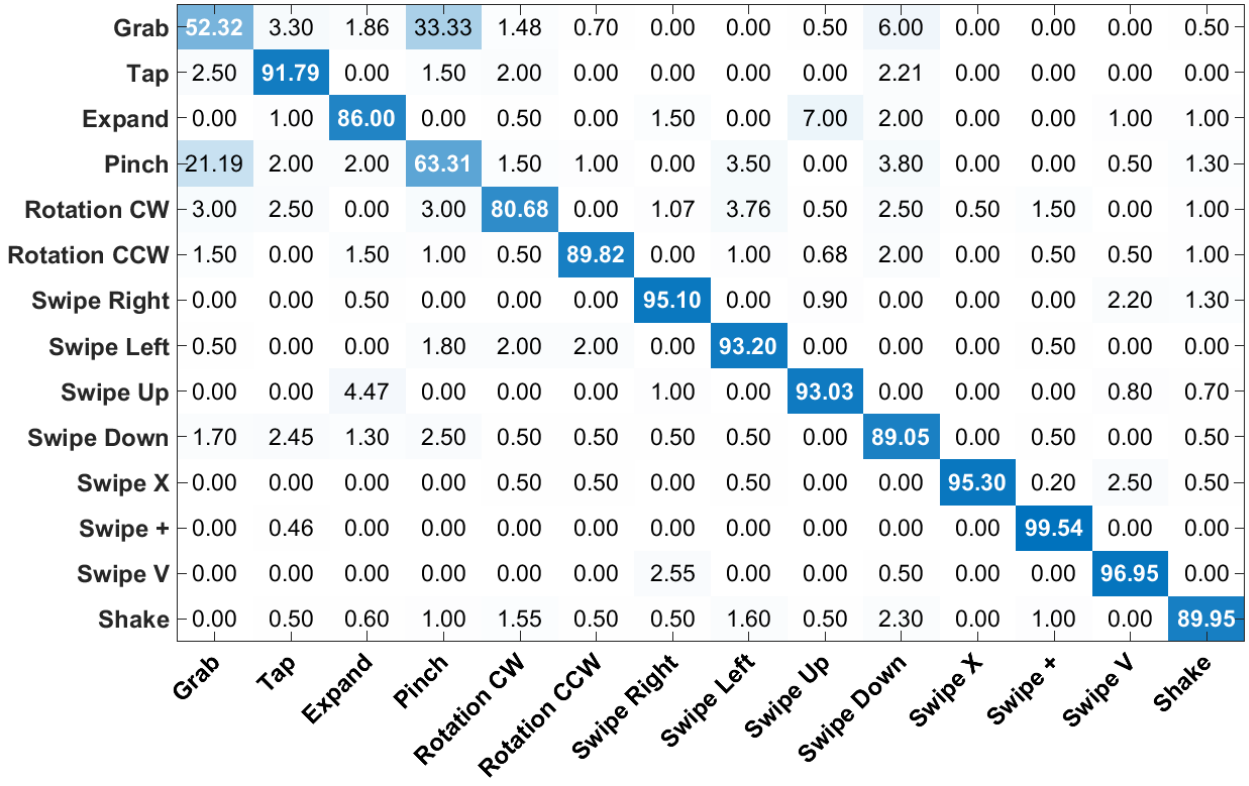


Figure 6.5: The confusion matrix of the proposed approach for the DHG-14 dataset.

accuracy obtained for certain gestures such as *Grab* is mainly due to the great confusion with *Pinch* gesture. By analyzing their sequences, we find that the algorithms of the hand pose perform well the 3D join estimation. However, we observe that these gestures are very similar and difficult to distinguish even by the human eye. The main difference between them is the hand movement amplitude and our approach does not take this characteristic into account. With a final accuracy of 86.86% obtained on DHG-14 dataset, we noticed that the recognition of dynamic hand gestures is still challenging. The recognition system has to deal with the considerable differences between gestures performed by different people, resulting in a challenging heterogeneity of the gestures.

Finally, in order to meet the challenge of gesture recognition when performed with different numbers of fingers existing in the DHG-28 dataset, we consider hand gestures as belonging to 28 classes related to the gesture type and the way it has been performed (with one finger or the whole hand). Using our approach, we obtain an accuracy of 84.22%. As shown in Table 6.4, by multiplying the number of classes by two, we lose 2.64% of accuracy. We compare our approach with four state-of-the-art methods on the DHG dataset. We chose two depth-based descriptors: HOG^2 proposed by Ohn-Bar *et al.* [89] and HON4D proposed by Oreifej *et al.* [86]. We also compare our approach to a skeleton-based method proposed by Devanne *et al.* [J4] showing a good accuracy for human action recognition. Finally, we compare the hand shape descriptor SoCJ with a similar state-of-the-art feature called Skeletal Quad defined by Evangelidis *et al.* [61]. The publicly available source codes of these methods are used in our experiments.

Table 6.4 presents the results obtained by the methods cited previously using 14 and 28 gestures and considering fine and coarse gestures separately from the DHG dataset. We note that our approach outperformed, with an accuracy of 86.86%, the two depth-based descriptors showing the promising direction of using skeletal data for hand gesture recognition. The accuracy obtained by

the action recognition method [J4] applied for 3D hand joints trajectories is 76.61%. It shows that an action recognition approach is often not appropriate for hand gesture recognition and that hand trajectories are not sufficiently distinctive enough for hand gesture classification.

When we apply these methods on 28 classes, the HOG^2 descriptor [89], which had a good result on 14 gestures, obtains 76.53% of accuracy. The depth-based methods do not handle enough hand shape information to deal with the challenge of hand gestures performed with different numbers of fingers. We note that Devanne's approach loses 14.61% of recognition rate on this experiment showing that the method, giving a good result on action recognition dataset, it is unsuitable for fine and dynamic hand gesture recognition.

Evangelidis *et al.* [61] propose a local body skeleton descriptor that encodes the relative position of joint quadruples. It requires a *Similarity Normalisation Transform* (SNT) that leads to a compact (6D) view-invariant skeletal feature, called Skeletal Quad. Because of the SNT, their descriptor takes more computation time and is less suitable for hand shape description as it lost information about distances between joints. The accuracy on the DHG-28 dataset using their hand shape descriptor decreases by 4% compared to the SoCJ descriptor.

Table 6.4: Accuracy comparisons for 14/28 and coarse/fine gestures of the DHG dataset.

Method	14 gestures (%)	28 gestures (%)	Coarse (%)	Fine (%)
Ohn-Bar *et al.* [89]	81.85	76.53	86.00	71.60
Oreifej *et al.* [86]	75.53	74.03	83.88	60.50
Devanne *et al.* [J4]	76.61	62.00	86.61	58.60
Evangelidis *et al.* [61]	84.50	79.43	92.22	70.62
Ours [P3]	**86.86**	**84.22**	**93.77**	**74.43**

We notice also that coarse gestures are defined by the motion of the hand in space and fine gestures are more distinguished by the variation of the hand shape during the sequence. The statement of a need of precision in the field of dynamic hand gesture recognition is also shown in this experiment. Except for the HOG^2 descriptor [89], Oreifej *et al.* [86] and Devanne *et al.* [J4] give honorable results in the task of coarse gesture classification but they show a lack of precision generating a recognition rate below 61% when trying to classify fine gestures. Although our approach gives the best results with 74.43% of correctly labeled fine gestures, we note that further improvements are needed.

Evaluation on Handicraft-Gesture dataset. Handicraft-Gesture is a dataset built with a Leap Motion Controller (LMC) [20]. A LMC is a device providing accurate information about the hand skeleton which contains the same 22 joints described. This dataset is made of 10 gestures, which originate from pottery skills, by 10 volunteers each one performed every gesture three times, resulting in 300 sequences.

To evaluate our approach on the Handicraft-Gesture dataset, we follow the experimental protocol proposed by Lu *et al.* [20], i.e. *Leave-One-Subject-Out* cross-validation. They compute several features based on palm direction, palm normal, fingertip positions, and palm center position. For the classification of temporal sequences, they use a Hidden Conditional Neural Field classifier. Table 6.5 shows how the hand gesture recognition accuracy has been increased by 2.11% using our approach.

Evaluation on NVIDIA Dynamic Hand Gesture dataset. Molchanov *et al.* [19] introduced a new challenging multimodal dynamic hand gesture dataset captured with depth, color and stereo-IR

Table 6.5: Recognition accuracies obtained on the Handicraft-Gesture dataset.

Method	Accuracy (%)
Lu *et al.* [20]	95.00
Ours	**97.11**

sensors in a car simulator. Using multiple sensors, they acquired a total of 1532 gestures of 25 hand gesture class (see Table 6.6. Similarly to the DHG dataset, this set contains coarse and fine gestures. A total of 20 subjects participated in data collection, performing gestures with their right hand. The SoftKinetic DS325 sensor is used to acquire frontal view color and depth videos.

Table 6.6: Gesture list included in the NVIDIA Dynamic Hand Gesture dataset.

Index	Gesture	Index	Gesture
1	move the hand left	13	show the index finger
2	move the hand right	14	show 2 fingers
3	move the hand up	15	show three fingers
4	move the hand down	16	push the hand up
5	move 2 fingers left	17	push the hand down
6	move 2 fingers right	18	push the hand out
7	move 2 fingers up	19	push the hand in
8	move 2 fingers down	20	rotate 2 fingers clockwise
9	click with index finger	21	rotate counter-clockwise
10	call someone	22	push forward with 2 fingers
11	open the hand	23	close the hand twice
12	shake the hand	24	show "thumb up"
		25	show "Ok"

To evaluate our approach on this challenging dataset, we use the Ge *et al.* hand pose estimator [24] which gives the best recognition accuracy on DHG-14 dataset (see Table 6.1). We performed the hand region-of-interested extraction step using the same algorithm proposed by [24, 31]. The extracted 3D joint positions of hand from depth images are used as input for our gesture recognition method. Following the same protocol proposed in [19], we randomly split the data by subject into training (70%) and test (30%) sets, resulting in 1050 training and 482 test videos. When considering the pre-segmented sequences of the dataset, our approach obtain an accuracy of 74%. First, with such a recognition accuracy, we went beyond the two handcrafted methods [45, 52] which extract descriptors on the sequence of depth images and obtained respectively 36.3% and 70.7%. Second, deep learning methods outperformed recent results in many domains in computer vision. Following this statement, 3D convolutional layers presented in [19, 28] show particularly reliable accuracies on the task of 3D hand gesture recognition, obtaining, respectively, 78.8% and 80.3% accuracy. Finally, in addition to the recognition challenges, the NVIDIA dataset [19] has been created to study the detection of gestures. Indeed, an unsegmented stream of gestures contains a lot of unwanted and meaningless hand motions that do not belong to none of the gesture categories. A prior gesture detection is required before the recognition process.

Finally, we notice that the accuracy obtained by our method with a prior manual gesture detection step have been significantly improved. This experiment reveals that the detection step for

Table 6.7: Comparison of our method to the state-of-the-art methods on depth images of the NVIDIA Dynamic Hand Gesture dataset.

Method	Type	Data	Accuracy (%)
HOG+HOG^2 [52]	Hand-Crafted	Depth	36.3
SNV [45]	Hand-Crafted	Depth	70.7
C3D [28]	Deep Learning	Depth	78.8
R3DCNN [19]	Deep Learning	Depth	80.3
Ours [P3]	Hand-Crafted	3D Hand Skeletal	74.0
Ours + manual detection [P3]	Hand-Crafted	3D Hand Skeletal	83.3

hand gesture recognition is essential as we improve our previous result by 9.3%. However, there is room to improve the effectiveness of gesture detection phase , where a recognition of 83.3% can be reached with a manual detection of gesture.

Comparative results with state-of-the-art methods on the three public datasets demonstrate that our approach outperforms existing handcrafted approaches. Moreover, we also revealed a lack of precision to describe the dynamic of complex hand gestures, compared with the feature learning power of modern deep learning models. In the next part of this chapters, we focus on deep learning strategies in order to better represent the complex dynamic and temporal information of hand gestures. Moreover, gesture detection in an online scenario are considered as an extension of our current approach.

6.3 Online Gesture Recognition using Combined Convolutional and Recurrent Networks

Deep neural networks have proven their effectiveness in various challenges, improving recognition rates substantially for various image classification tasks. Furthermore, motivated by their success for images and videos, many research works have appeared very recently proposing models, such as convolutional neural network, for learning hand pose features.

Convinced of the usefulness of the pose features to describe hand gestures and motivated by the success of these approaches, we extend the study to online dynamic hand gestures taking over the whole pipeline of the recognition process, from hand pose estimation to the classification step, using deep learning. We aim to structure such a framework following two statements made in section 6.2: (1) Hand postures along the sequence are relevant features to describe the gesture. (2) Hand gestures can be efficiently described by the temporal variation of both, hand shape and its motions.

6.3.1 Approach overview

The dynamic aspect of gesture sequences requires the use of time-series based models, such as 3DCNN on sub-sequences of depth images, or RNN on lighter data sequences, like hand joints resulting from a hand pose estimation method. The first one requires a powerful hardware configuration and computational complexity, whereas the second one lacks of efficiency related to the loss of information due to the lack of robustness of current hand pose estimators. On the other hand, to describe the gesture, hand postures along the sequence are relevant features, but also the

temporal variation of both, hand shape and its motions need to be considered. Taking into account these multiple characteristics makes harder the learning process as it has to learn both spatial and temporal information. All these considerations lead us to address the problem of hand gesture recognition within a framework based separately on learned features of temporal hand shape and its posture. Those data are both extracted from a Convolutional Neural Network (CNN) trained on depth frames of the gesture sequence.

Based on the previous statements, we present in this section a new framework based on deep learning for online dynamic hand gesture recognition using a transfer learning strategy. So as to face the main challenges, we propose to revisit the feature pipeline by combining the merits of geometric shape features and dynamic appearance, both extracted from a CNN trained for hand pose estimation problem. Note that, despite an increasing amount of methods proposed over the last few years, defining an online dynamic hand gesture recognition system robust enough to work in real world applications is still very challenging.

6.3.2 Model architecture

In some field of computer vision, the access to millions of data makes it possible to create very deep neural networks from scratch. For example, Deng *et al.* [178] make publicly available the ImageNet dataset which contains over 10 million images. However, it is not the case while working with 3D data making difficult the use of data-driven approaches easily. To our knowledge, the largest available dynamic hand gesture dataset contains only 2,800 gesture sequences [C5].

To overcome this problem, we first propose to use a transfer learning strategy to extract hand shape and posture features for hand gesture recognition purpose. To do so, we train a CNN for hand pose estimation using the ICVL dataset [48]. Note that a training set of depth images of this dataset labeled with the 3D joint locations is available. Once the training of the CNN is over, we use it to output two distinct representations for each time step of a hand depth image sequence: hand posture features, noted J_t, which represent hand joints locations, and a hand shape feature vector X_t which represents the coarse hand shape in a high dimensional space.

Thus, original hand depth image sequences, $s_{original} = \{I_t\}_{t=1...N}$, are transformed into two different sets of sequences as follows: $s_{posture} = \{J_t\}_{t=1...N}$ and $s_{shape} = \{X_t\}_{t=1...N}$ for a sequence of N frames. Both sequences are fed to two recurrent neural networks: RNN_{joints} and the RNN_{shape}, in order to model the temporal aspect separately of the hand poses and the shape variations over the time. Finally, results are merged to perform the recognition of hand gestures. Figure 6.6 summarizes the model architecture.

Note that a pre-processing step is first applied to extract the region-of-interest (ROI) of the hand assuming the hand is the closest object to the camera. The estimation is then refined using a 3D bounding box around the center of the mass, from which we can extract a cropped image of the hand and we compute its center of the mass P_{com} in the original image space.

6.3.3 Deep extraction of CNN features

Inspired by Oberweger *et al.* [31], we consider the hand pose estimation algorithm based on a CNN architecture using prior enforcement. The implementation of the prior enforcement is made by introducing a *bottleneck* in the penultimate layer, having a smaller size than the final one which outputs the 3D joint coordinates. The linear mapping between the lower dimensional space and the final output is kept by not adding any activation function to the bottleneck layer.

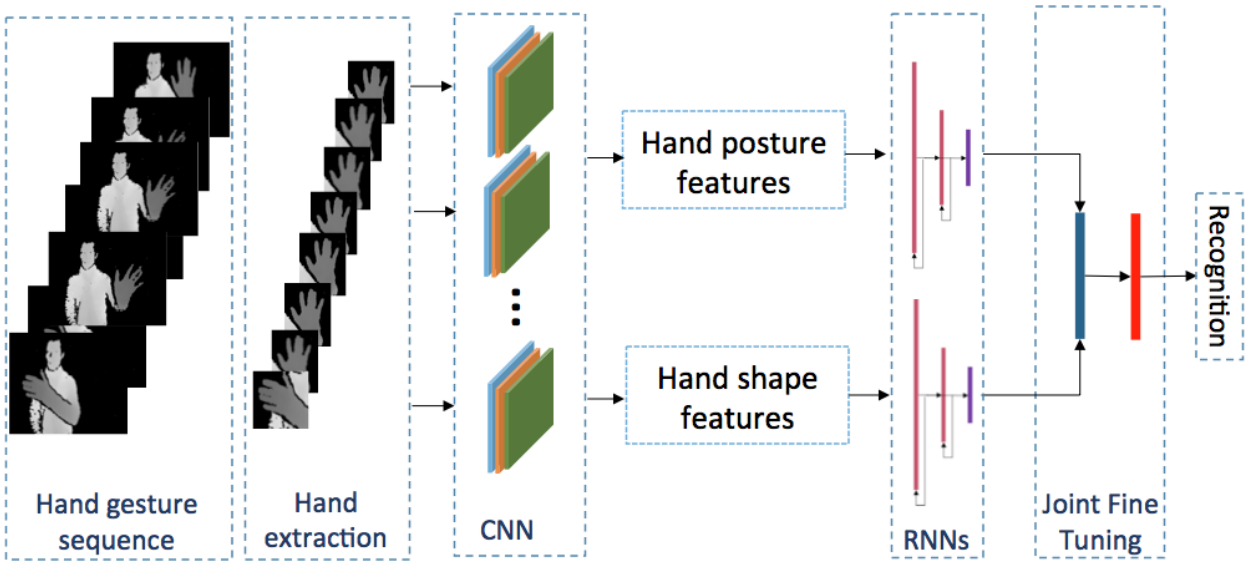


Figure 6.6: Overview of our proposed approach for hand gesture recognition.

The model takes as input a cropped depth image I^* around the hand. Given the physical constraints over the hand, there are strong correlations between 3D joint locations, and such a lower dimensional space of $3 \times K$ are sufficient to parameterize a 3D hand pose of K joints [210], but not enough to describe the gesture. Thereby, the CNN first maps the image to a high dimensional space vector X, laying in $\mathbb{R}^{1024}$ containing a coarse description of the hand shape. Then, it follows a "bottleneck" layer with a smaller size that the desired output to model the physical constraints over the hand topology. The network architecture, further called CNN_{hand_pose}, is depicted in Figure 6.7.

Using this network, we extract two feature vectors from a hand depth image at coarse and fine level, respectively hand shape features X and hand joint features $J*$. The shape feature vector X, lying in a high dimensional space $\mathbb{R}^{1024}$, is used to describe the coarse hand shape without taking into account the details of its topology.

The hand joint feature vector J^*, lying in $\mathbb{R}^{3*K}$ with K the number of joint, contains 3D hand joints locations centered around the center of mass of the hand P_{com} in the original depth image. The original joint locations into the image space can be then easily retrieved by applying an inverse transformation to the joints using P_{com} as follows:

$$J_i = \{j_i^* + P_{com}\}_{i=1...K} \tag{6.10}$$

where j_i^* is the i^{th} joint in the predicted hand pose J^*, P_{com} is the center of mass of the hand in the original depth image.

Parameters of the CNN model have to be initialized before the training step. A common way to generate the values is to use a random normal distribution. An exception is made for the bottleneck layer as we can help the network using prior knowledge. We initialize its weights with the 30 major components from a Principal Component Analysis (PCA) of the hand joint label space of the training set. As the cost function, we minimize the Huber loss to evaluate the differences between the hand pose ground-truth and the output of the network.

As the cost function, we minimize the Huber loss to evaluate the differences between the hand

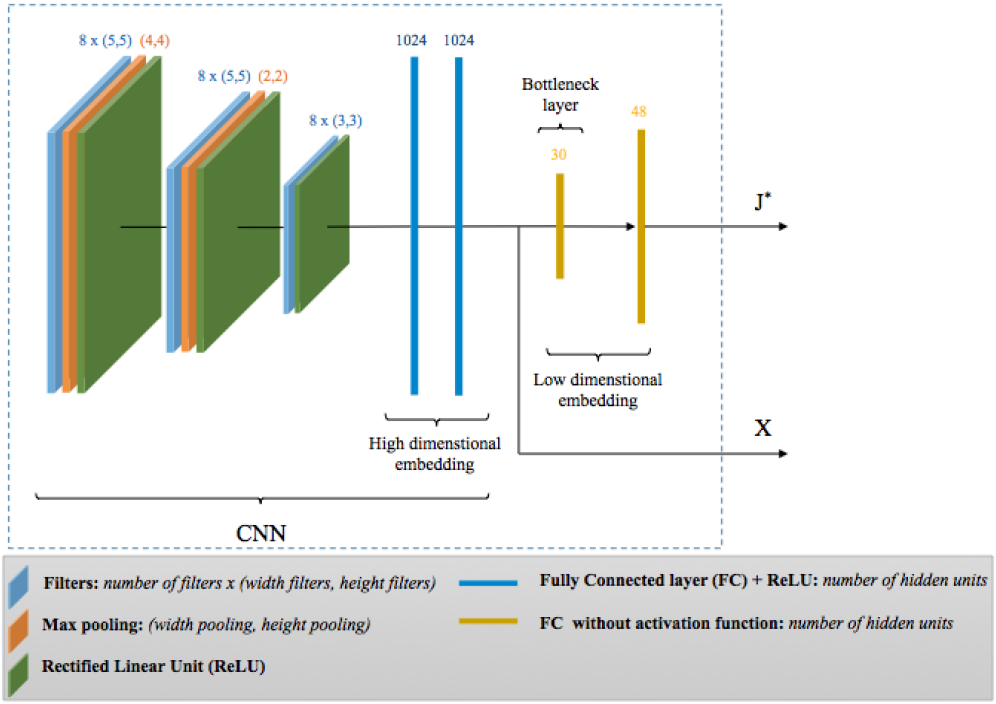


Figure 6.7: Architecture of the CNN for hand shape and posture feature extraction using prior enforcement.

pose ground-truth J and the output of the network noted $\hat{J}$:

$$H(J, \hat{J}, \delta) = \begin{cases} \frac{1}{2}(J - \hat{J})^2 & \text{for } |J - \hat{J}| \leq \delta, \\ \delta\,|J - \hat{J}| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases} \tag{6.11}$$

The Hubert loss is thus quadratic when the error is small ($\leq \delta$) and linear when it becomes larger. Consequently, this loss function is less sensitive to noisy annotations (which imply large errors) than the squared error loss function as depicted in Figure 6.8.

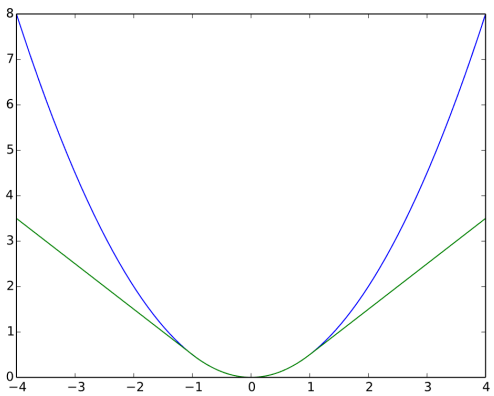


Figure 6.8: Huber loss defined in Equation 6.11 (green $\delta = 1$) and the squared error loss (blue $\frac{1}{2}x^2$) as functions of $J - \hat{J}$. Huber loss is less sensitive to large errors.

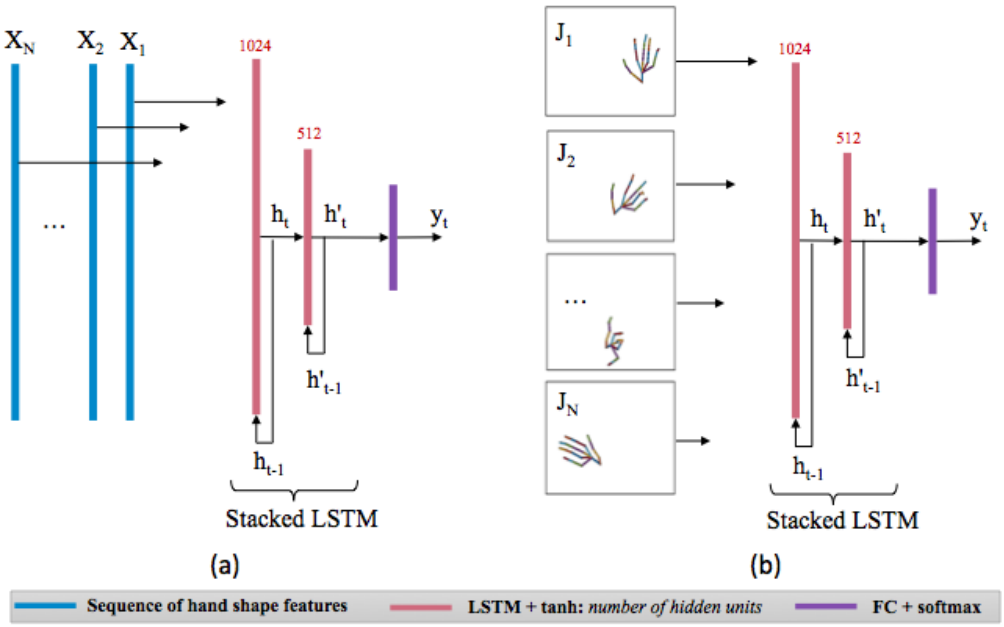


Figure 6.9: Temporal modeling by RNNs of hand shape and posture cues. (a) RNN_{shape}, (b) RNN_{joints}.

6.3.4 Temporal learning of shape and posture features

The hand gesture can be represented, according to the shape feature cue, as a sequence of shape features $s = \{X_t\}_{t=1...N}$ which is an ordered list of N vectors and $X_t \in \mathbb{R}^{1024}$. It can also be represented, according to the joint feature cue, as a sequence of joint feature vector $s = \{[j_1, \ldots, j_K]\}_{t=1...N}$ be an ordered list of N vectors and $j_i \in J$. To model the temporal aspect of gestures, we feed separately these sequences to two Recurrent Neural Models (RNNs), noted respectively RNN_{shape} and RNN_{joints}, each one composed of two stacked LSTM layers. Their output $h_t^{'}$ is finally transform by a fully connected layer, which is a softmax activation function, in order to output a class-conditional probabilities vector. The final predicted label of the sequence s is $\hat{y} = argmax_i(\hat{y}_N^i)$, where $i \in \{1 \ldots K\}$ and K the number of gesture classes. The RNN models are sketched in Figure 6.9.

During the training phase of both the RNN_{shape} and RNN_{joints}, a weight decay and a dropout strategy are applied to prevent overfitting. Networks are trained using the Back-Propagation-Through-Time (BPTT) algorithm [218]. BPTT is equivalent of unrolling the recurrent layers, transforming them into a multi-layer feed-forward network of depth N; where N is the number of frames in the gesture sequence. The standard gradient-based back-propagation is then used. We average the gradients to consolidate weight updates to duplicated unrolling. The learning rate decreases following the number of epochs ne by $lr = 0.001 \times N_0 e^{-\lambda \times ne}$. Networks try to minimize the cross-entropy cost.

To increase variability in the training examples, we apply random horizontal, vertical and depth translations on depth image sequences before each learning iteration. Since recurrent connections can learn the specific order of gesture sequences in the training set, we randomly permute the training gesture videos before each new epoch.

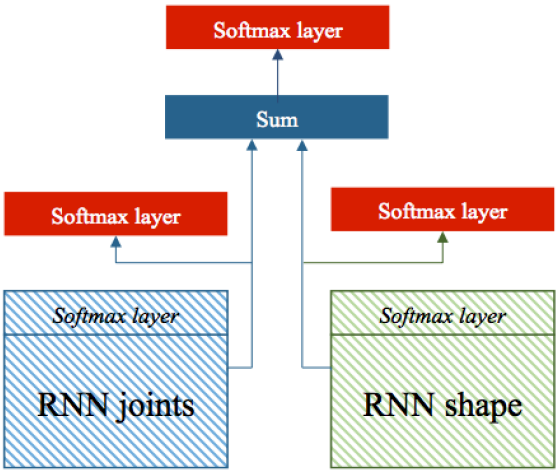


Figure 6.10: Fusing method using the joint fine-tuning strategy. Hatched boxes are elements that are fixed and not changed during the training fusion process.

6.3.5 Two-stream RNN fusion

Combining different information or modalities using deep learning is not an obvious task. Wu *et al.* [15] investigated intermediate and late fusion strategies for multimodal gesture recognition. They discover that averaging results in the last stage of the process gives accurate and more robust results. We propose in our approach a joint-fine-tuning method to fuse the outputs of the RNN_{shape} and the RNN_{joints} in order to enhance the classification process, as depicted in Figure 6.10.

The joint fine-tuning method consists in retraining the two last softmax layers of the RNN_{shape} and the RNN_{joints} while forcing their sum to be a representation of both networks. This strategy allows the network to learn a joint representation of network outputs, without adding parameters to the model and so, does not increase its complexity. Since both networks are trained separately, we retrain last fully connected layers before the softmax activation functions with a new cost function, noted $\mathcal{L}_{fusion}$, defined as follows:

$$\mathcal{L}_{fusion} = \lambda_1 \mathcal{L}_1 + \lambda_2 \mathcal{L}_2 + \lambda_3 \mathcal{L}_3 \tag{6.12}$$

where $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$ are respectively loss functions computed on the RNN_{shape}, the RNN_{joints} and the sum of both outputs. The λ_1, λ_2 and λ_3 are tuning parameters. Each cost function is a cross entropy function. Let l_1 and l_2 be respectively the output values of the network RNN_{joints} and RNN_{shape}, $\mathcal{L}_3$ is then defined as follow:

$$y_3^i = softmax(l_1^i + l_2^i) \tag{6.13}$$

$$\mathcal{L}_3 = -\frac{1}{N} \sum_{i=1}^{N} \left[y^i \log \hat{y}_3^i + (1 - y^i) \log(1 - \hat{y}_3^i) \right] \tag{6.14}$$

The final decision is obtained using y_3^i:

$$\hat{y} = \arg\max_i (y_3^i) \tag{6.15}$$

As a result, we utilize three loss functions in the training step: $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$. $\mathcal{L}_1$ and $\mathcal{L}_2$ are used to regulate, respectively, both streams and avoid that one of them vanished under the weight of the

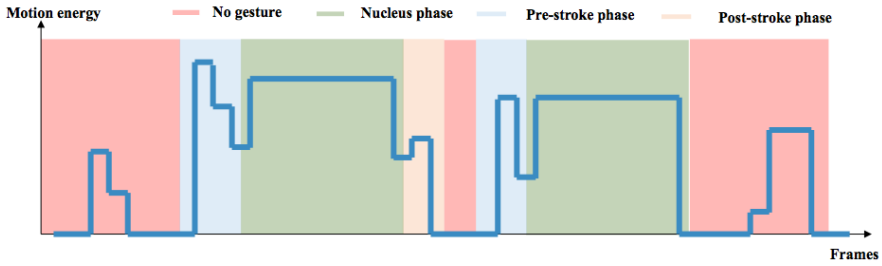


Figure 6.11: Different phases of a continuous stream of gestures. The figure depicts a fictive curve of the motion energy of a sequence of two hand gestures. Frames which do not belong to a gesture, with or without parasitical motions, are in red. Blue, green and orange squares identify, respectively, pre-stroke, nucleus and post-stroke phases. We note that the second gesture does not contain a post-stroke phase as with the pre-stroke phase, their existence is not automatic.

other. $\mathcal{L}_3$ is used to optimize the fusion of the two modalities. Consequently, we use only y_3 for prediction as it is impacted by both RNNs.

6.3.6 Problem of continuous sequence

Online hand gesture recognition requires a prior gesture detection as the sequence contains motions belonging to none of the gesture categories.

Gesture detection. An unsegmented stream of gestures contains a lot of unwanted and meaningless hand motions that do not belong to none of the gesture categories. First, hand gesture movements are often composed of three phases:

- The *pre-stroke* phase, which is composed of hand motions happening before the relevant gesture when the user needs to put his/her hand in a starting position. For example, when the user moves the hand from its restful position to a place where the camera can see the hand.
- The nucleus phase, where the hand gesture is performed and has meanings.
- The *post-stroke* phase, which is composed of hand motions happening after the relevant gesture when the user wants to move back his/her hand to a restful position.

Additionally, a stream of gestures contains motions between the gestures as depicted in Figure 6.11. For example, in a human-computer interface based on hand gestures in a car scenario, while the user is not performing a gesture, his/her hands are still moving to control the vehicle and, so, contains a lot of parasitical hands motions. A challenge of online hand gesture recognition is to detect and extract only hand motions from nucleus phases in order to improve the gesture recognition accuracy.

Gesture recognition. In real applications, we do not have information about when and where the hand gesture is going to be performed. Neverova *et al.* [18] added a binary classification step before the classification process using $\{gesture,\ no\_gesture\}$ labels. Instead of performing recognition in two steps, detection and then recognition, our approach consists of extending the dictionary of existing gestures by adding a *garbage* class such as: $Y' = Y \cup \{\ no\_gesture\}$. Consequently, the softmax layer outputs a class-conditional probability for this additional *garbage* class. All frames which do not belong to a nucleus phase are labeled with this new class.

6.3.7 Experiments

In this section, we evaluate our proposed approach on two challenging datasets – NVIDIA Dynamic Hand Gesture dataset [19] and an online version of the DHG 14-28 dataset called Online Dynamic Hand Gesture dataset [C3][2] – and compare it with four state-of-the-art methods using depth images.

We address here the online recognition of hand gestures in challenging conditions, including the heterogeneity of the hand shape depending on the set of fingers used and following two types of gesture categories: fine-grained and coarse-grained gestures. Due to the complexity of the hand movement and its potential self-occlusions, the analysis of hand gesture in such conditions becomes very difficult to achieve using long-range depth cameras like Microsoft Kinect (0.8 - 4.2 meters). Therefore, we exclude of our experimental field certain public datasets, which do not fit these conditions, as ChaLearn2014 [63], SKIG [92] and [52]. To our knowledge, only two public datasets meet these requirements/challenges which are DHG-14/28 dataset [**des**] and NVIDIA Dynamic Hand Gesture and dataset [19].

Implementation details. To extract the deep features of hand posture and its shape, we train our CNN on the ICVL dataset [48], which comprises a training set of over 180,000 depth images showing various hand poses. The dataset is recorded using a time-of-flight *Intel Creative Interactive Gesture Camera* and has 16 annotated 3D joints for each depth image. Depth images have a high quality with little noise.

We use the Hubert loss function defined in Equation 6.11. We choose a sensitive factor to error $\delta = 500$ (we remind that 3D hand coordinates annotations are given in mm). It means that errors on the joint location prediction superior to half a centimeter is linear while smaller errors are quadratic.

Weight decay is applied with a regularization factor equal to 0.001. The networks are trained with a batch size of 128 for 100 epochs. The initial learning rate lr is set to 0.01 with a momentum of 0.9.

To avoid overfitting while training the recurrent layers in the two streams of RNN, weight decay is applied with a regularization factor equal to 0.001. The dropout strategy has a probabilistic value equal to 25%. We stop the training after 30 epochs to avoid learning training dataset specification. The initial learning rate lr is set to 0.001.

For the data augmentation step, the ranges of the horizontal and vertical translations are ± 20 pixels and the range of the depth translation is ± 100. Parameters for each translation are drawn from a uniform distribution.

Pre-segmented gesture recognition (offline). Before presenting the results obtained by our approach, we analyze the individual components (streams) of our proposed approach and evaluate its usefulness according to the type of gestures and then to asses the benefits coming from their fusion. We note that the temporal aspect of the gesture includes two RNN models: the RNN_{shape} model from shape feature cue and the RNN_{joints} model from joint feature cue. We propose to evaluate separately the efficiency of each one, before the fusion process. To analyze the recognition process on our Online Dynamic Hand Gesture dataset, we extract the gesture nucleus resulting in a dataset of 2800 pre-segmented gestures of either 14 or 28 distinct labels.

[2] http://www-rech.telecom-lille.fr/shrec2017-hand

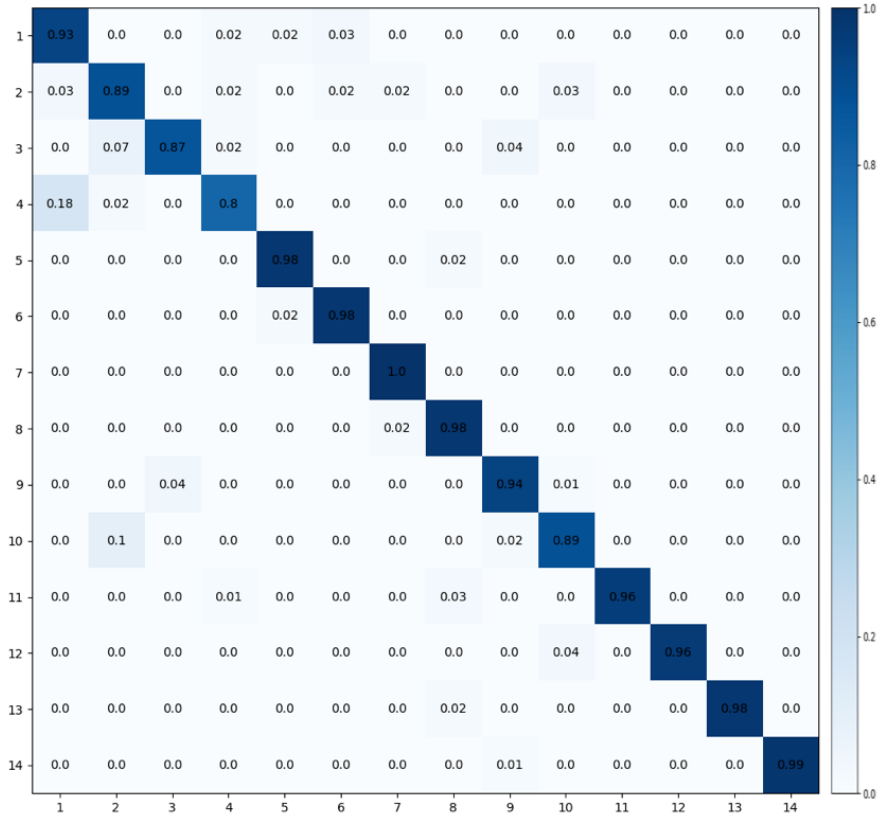


Figure 6.12: The confusion matrix obtained using a joint-fine tuning fusion of the RNN_{shape} and the RNN_{joints} on the Online DHG dataset for the task of offline recognition of 14 gesture types. Labels of gestures can be found in Table 6.2.

14 gesture classes With a vocabulary of 14 gesture types, we obtain an accuracy of 84.52% and 80% respectively using the RNN_{joints} and the RNN_{shape}. A joint fine tuning fusion of the two RNN streams provides an overall accuracy of 94.17%. The confusion matrix is depicted in Figure 6.12. It illustrates that the fusion is not only able to choose the right features to perform the classification but also takes benefit from both information to outperform the previous recognition accuracies. For example, the gesture *Swipe Down* (10) obtained 77% and 72% of accuracy using respectively the RNN_{joints} and the RNN_{shape}. Once they are merged, our system is able to correctly recognize 89% of these gestures.

28 gesture classes. Let us now study the capacity of our approach to distinguish the same gestures performed with one finger or the whole hand. With a vocabulary of 28 gesture types, we obtain an accuracy of 76.30% and 76.67% respectively using the RNN_{joints} and the RNN_{shape}. Once fused, we obtain an overall accuracy equal to 90.48%. This result illustrates the outstanding potential of fusing shape and posture features to perform fine hand gesture recognition. The confusion matrix resulting from this experiment, depicted in Figure 6.13, shows that we obtain almost no confusion between gestures with the same meaning but performed with different number of fingers, thanks to the rich shape information coming from the RNN_{shape}.

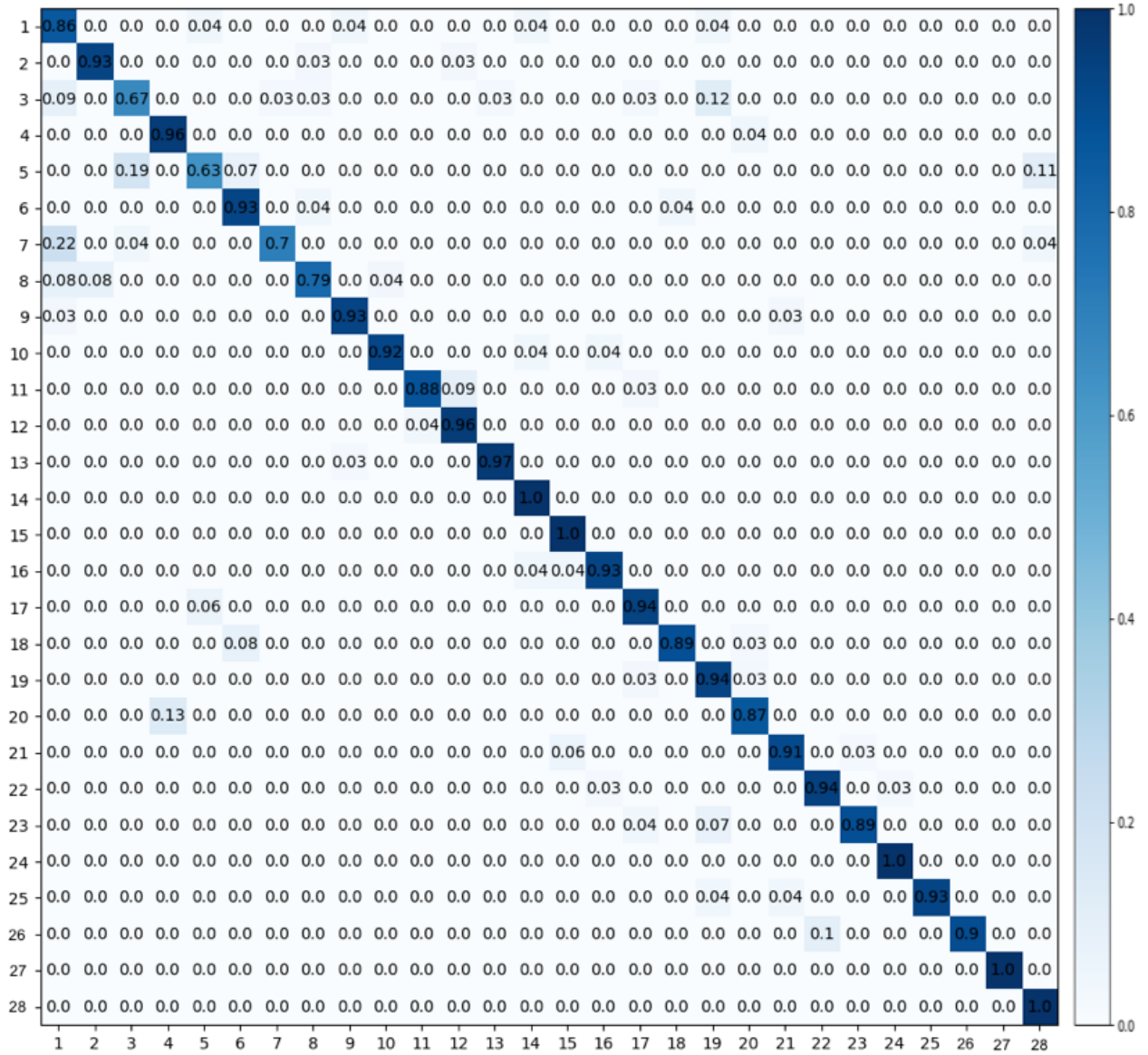


Figure 6.13: The confusion matrix obtained using a joint-fine tuning fusion of the RNN_{shape} and the RNN_{joints} on the Online DHG dataset for the task of offline recognition of 28 gesture types. Labels of gestures can be found in Table 6.2.

Comparison with state-of-the-art methods. We compare here our framework to state-of-the-art methods for the task of offline hand gesture recognition on the Online DHG dataset. First, with two handcrafted descriptors based on depth images: the HOG+HOG^2 descriptor proposed by Ohn-bar [52] and the HON4D descriptors proposed by Oreifej *et al.* [86]. Second, we compare our approach to a skeleton-based method proposed by Devanne *et al.* [J4] originally designed for human action recognition. Third, we present the results obtained by our precedent hand skeleton based approach [C1]. Finally, we report results obtained by Guerry *et al.*'s approach [C3], based on key frames detection followed by an CNN. The recognition accuracies, using both 14 and 28 gesture types of the Online DHG dataset, obtained by the state-of-the-art methods cited below, are presented in Table 6.8. We note that the publicly available source codes of these methods are used in our experiments.

Table 6.8: Comparison with state-of-the-art methods on the Online DHG dataset.

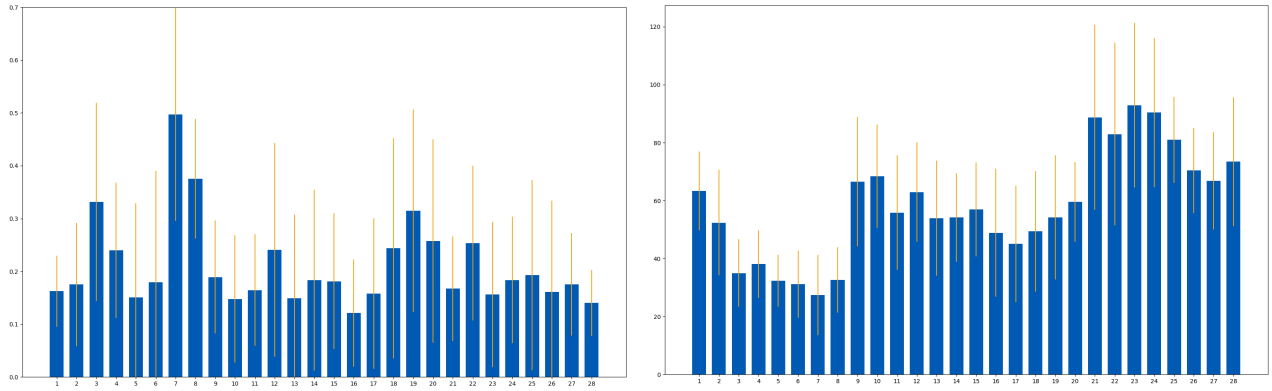
Method	**14 gestures (%)**	**28 gestures (%)**
Guerry *et al.* [C3]	82.90	71.90
Ohn-Bar *et al.* [52]	83.85	76.53
Oreifej *et al.* [86]	78.53	74.03
Devanne *et al.* [J4]	79.61	62.00
De Smedt *et al.* [C5]	88.24	81.90
Ours, RNN_{joints}	84.52	76.30
Ours, RNN_{shape}	80.60	76.67
Ours, fusion	**94.17**	**90.48**

Our method outperforms all these approaches. The key frames detection of Guerry *et al.* [C3] leads to a temporal lossy representation of the gestures. Besides, Devanne *et al.*'s action recognition method does not provide a good gesture recognizer, because it is obviously not suitable for hand gesture recognition.

We note that the RNN_{joints} alone does not outperform our previous handcrafted approach proposed in [C5]. Only by adding the shape features, our approach can outperform this method by 6%. The effectiveness of the fusion of the hand shape variations and its motions appears truly when looking at results obtained on the task of recognizing 28 gestures, where we outperform state-of-the-art methods by more than 10%.

Online detection and recognition. In this section, we analyze the behavior of our approach on the early detection and the recognition processes, on the Online DHG and the NVIDIA Dynamic Hand Gesture datasets. These two datasets have been captured in an online scenario with different phases of a continuous stream of gestures.

In order to locate properly the nucleus of gestures, as done by Molchanov *et al.* [19], our solution is considering adding a garbage class by extending the dictionary of existing gestures such as: $Y' = Y \cup \{no\_gesture\}$. Consequently, the softmax layer outputs a class-conditional probability for this additional "garbage" class. All frames which do not belong to a nucleus phase are labeled with this new class. To detect the presence of any one of the 25 gestures relative to $\{no\_gesture\}$, we compare the highest current class probability output of our approach to a threshold $\xi \in [0, 1]$. When the detection threshold is exceeded, a classification label is assigned to the most probable class.



(a) **(b)**

Figure 6.14: The Normalized Time to Detect values using a detection threshold $\xi = 0.15$ (a), and Nucleus lengths in number of frames (b), of the 28 gestures contained in the Online DHG dataset.

Evaluation metrics. To analyze the online detection and recognition capacity, we use three metrics: the Receiver Operating Characteristic (ROC) curve [216] and the Normalized Time to Detect (NTtD) [59] for the detection analysis and finally the recognition accuracy to analyze the recognition process. First, the **ROC** plots the True Positive Rate (TPR) – when the detector fires inside a nucleus phase – versus the False Positive Rate (FPR) – when the detector fires outside the nucleus phase. We use the area under the ROC for evaluating the detector accuracy. Second, the **NTtD** defines the fraction of the nucleus that has occurred, from a to b, before the system fires a successful detection, $a \leq t \leq b$, :

$$NTtD = \frac{t - a + 1}{b - a + 1} \tag{6.16}$$

By adjusting the detection threshold chosen using the ROC curve, one can achieve lower NTtD at the cost of higher FPR and vice versa. Finally, the **recognition accuracy** corresponds here to the fraction of sequences in the test set which is correctly labeled by the approach. Consequently, the predicted label in an online scenario of a particular gesture is chosen as the predicted label of the last frame which is not labeled as *no_gesture*, such as:

$$\hat{y} = \arg\max_i(\ y^i_M\ ) \mid \arg\max_i(\ y^i_{M+1\ldots N}\ ) = no\_gesture \tag{6.17}$$

where N is the number of frame in the gesture.

Online DHG dataset. To evaluate the online capability of our approach, we used the unsegmented sequences of gestures labeled following the vocabulary containing 28 labels. After plotting the ROC curve, we obtained an Area Under the Curve equal to 0.91. We choose a gesture detection threshold equal to 0.15 as it shows a good trade-off between a high TPR (85%) and a low FPR (17%).

The NTtD distribution values for various gesture types is shown in Figure 6.14a. The average NTtD across all classes is 0.2104, which means that, in average, a hand gesture can be detected after only 21% of its nucleus. The average nucleus lengths over the whole Online DHG dataset are illustrated in Figure 6.14b.

We note that nucleus of *fine* gestures (1 - 10) are shorter than those of *coarse* gestures (11 - 25). Moreover, *Swipe* gestures that contain multiple motions, such as *Swipe V*, *X* and + (21 - 28), have naturally the longest nucleus. Using the detection upstream step, we obtain an overall online accu-

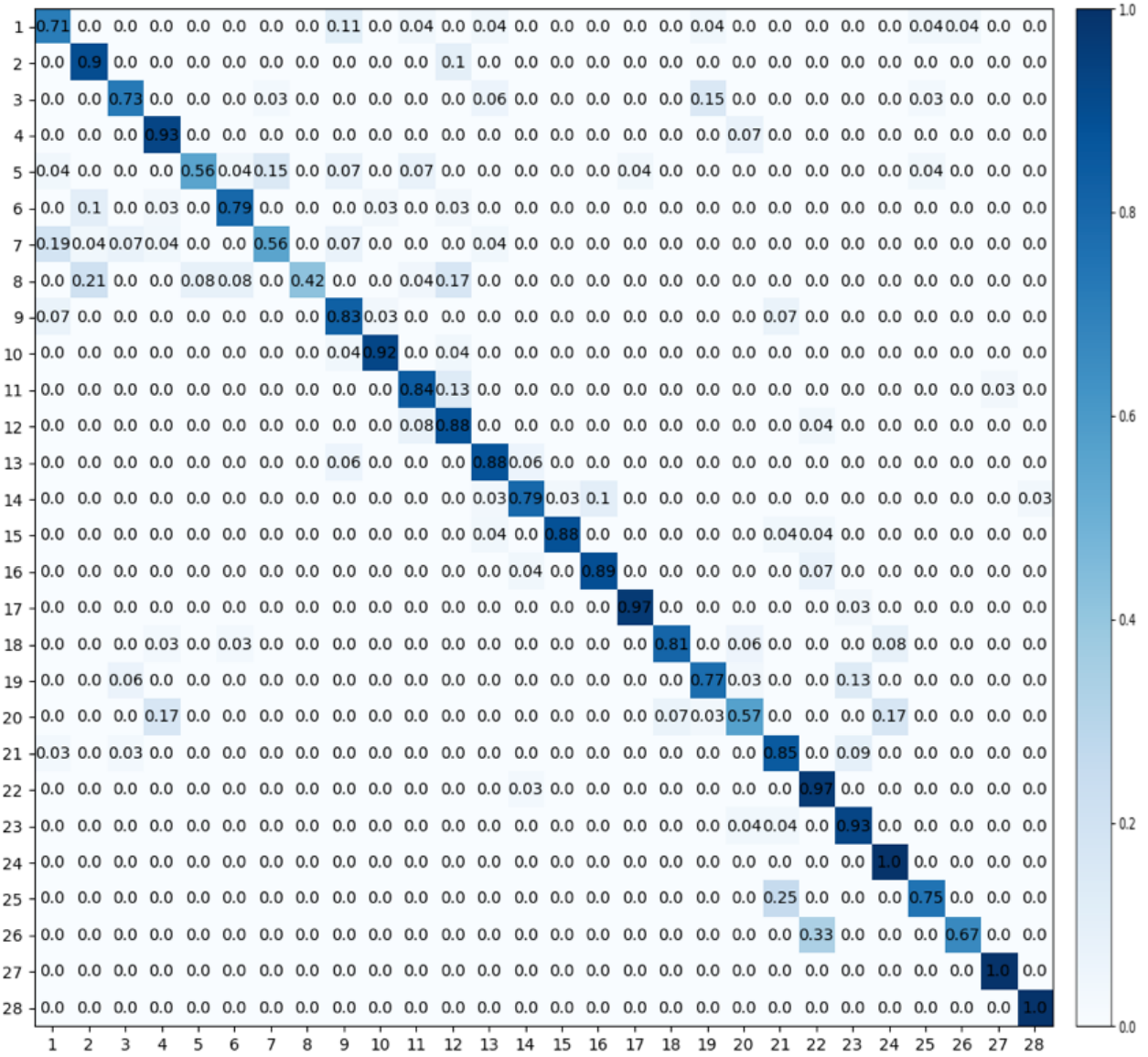
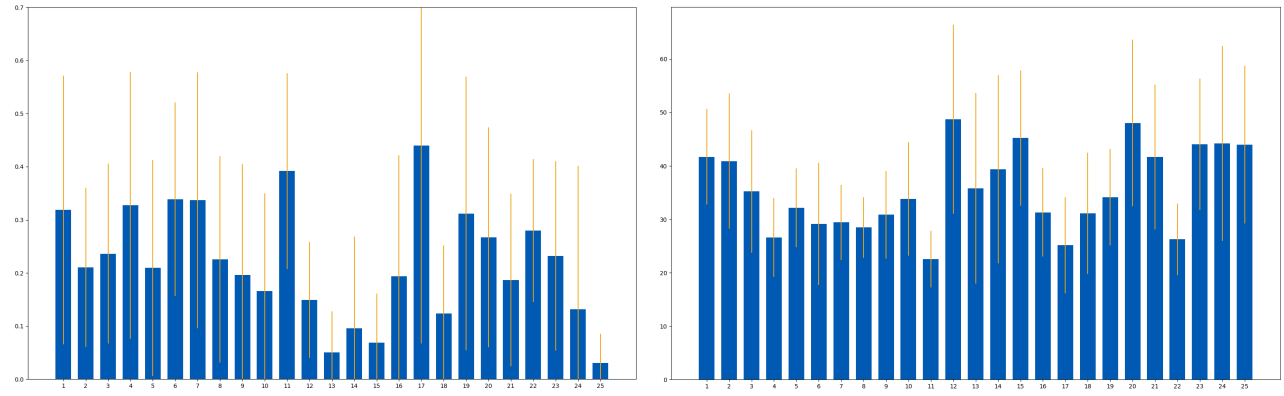


Figure 6.15: The confusion matrix obtained on the Online DHG dataset for the task of online recognition of 28 gesture types. Labels of gestures can be found in Table 6.2.

racy recognition of 82.14%. The confusion matrix is depicted in Figure 6.15.

Due to issues resulting from the detection of gestures, the recognition accuracy decreases by 8.34% compared to the easiest task of pre-segmented gesture recognition. This difference can result from incorrect gesture detection or from confusion between gestures with similar parts. For example, the *Swipe Down* gesture (20) performed with the whole hand obtains an *offline* recognition accuracy up to 87%. In the online scenario, the accuracy decreases by 57% and a high confusion up to 24% appears with the *Swipe V* gesture (24). This can be explained as the first half a *Swipe V* gesture is extremely similar to the *Swipe Down* gesture. In addition, the recognition process suffers from an incorrect prior gesture detection.

NVIDIA dataset. The dataset has been captured following a human-computer interaction based on hand gestures in a car scenario. While the user is not performing a gesture, his/her hands still move to control the vehicle and, so, is highly suitable to study gesture detection. To detect



(a) **(b)**

Figure 6.16: The Normalized Time to Detect values using a detection threshold $\xi = 0.16$ (a), and Nucleus lengths in number of frames (b), of the 25 gestures contained in the NVIDIA dataset.

the presence of any one of the 25 gestures relative to *no_gesture*, we compare the highest current class probability output of our framework to a threshold $\xi \in [0, 1]$. When the detection threshold is exceeded, a classification label is assigned to the most probable class. First, we do it in a frame-wise manner and compute the ROC curve.Using it, we choose a detection threshold ξ equal to 0.16 as it shows a good trade-off between a high TPR (85%) and a low FPR (17%). The NTtD distribution values for various gesture types is shown in Figure 6.16a. The average NTtD across all classes is 0.2158 which means that, in average, a hand gesture can be detected after only 22% of its nucleus. In general, static gestures require the finest portion of the nucleus to be seen before classification (around 10%), while dynamic gesture are classified on average within 25%. The average nucleus length over the whole dataset are given in Figure 6.16b.

Static gestures have longest nucleus phases. Intuitively, NTtD differences between dynamic and static gestures are explained as users letting their hand a long time in front of the camera to express a static gesture but the algorithm can detect it using few frames. Finally, we compute the overall recognition accuracy obtained by our approach for an online hand gesture recognition scenario. We obtained an accuracy of 81.25%. The confusion matrix is given in Figure 6.17.

Comparison with state-of-the-art methods. We compare our approach to several state-of-the-art methods: HOG+HOG^2 descriptors [52], Super Normal Vector (SNV) [45], convolutional 3D (C3D) [28] and a C3D followed by a recurrent layer (R3DCNN) [19], as well as human labeling accuracy.

To compute HOG+HOG^2 descriptors [52], all video sequences are ressampled to 32 frames and the parameters of the SVM classifier are tuned via grid search to maximize accuracy. Among the CNN-based methods, we compare against the C3D method [28], which is pre-trained with the Sports-1M [56] dataset and fine-tuned with the depth modalities of the NVIDIA dataset. The R3DCNN method uses the C3D network to extract spatiotemporal features of sub-video clip of 8 frames and fed the result sequence in a recurrent layer. Molchanov *et al.* [19] trained the whole network using a Connectionist Temporal Classification (CTC) [135, 194] loss function.

Lastly, Molchanov *et al.* [19] evaluated human performance on the NVIDIA dataset by asking six subjects to label each of the 482 gestures videos in the test set after viewing the front-view color video. Prior to the experiment, each subject familiarized themselves with all 25 gesture types. State-of-the-art method results are given in Table 6.9. We note that handcrafted methods give lower results that deep learning methods. Our approach achieves the best performances, meanwhile it is

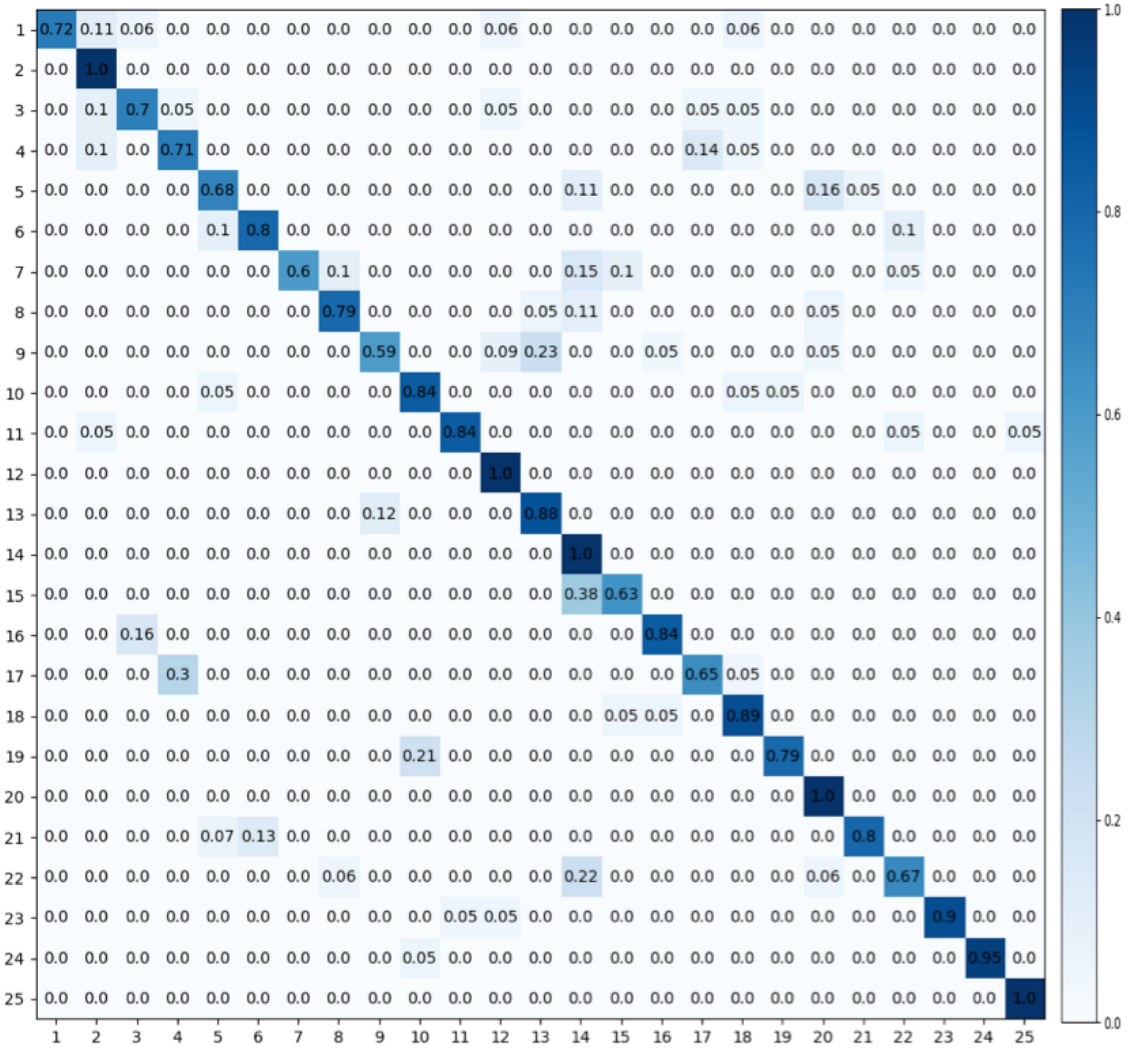


Figure 6.17: Final confusion matrix using obtained by our approach, showing its capability to perform an online hand gesture recognition on the NVIDIA dataset. The average recognition accuracy is equal to 81.25%.

Table 6.9: Comparison with state-of-the-art methods on the NVIDIA dataset.

Method	**Modality**	**Features extraction strategy**	**Accuracy**
Human	color		88.4%
HOG+HOG^2 [52]	depth	handcrafted	36.3%
SNV [45]	depth	handcrafted	70.7%
C3D [28]	depth	learned	78.8%
R3DCNN [19]	depth	learned	80.3%
Ours	depth	learned	**81.3%**

Table 6.10: Formulas of the number of parameters for different layers with a number of hidden parameters equal to n and an input of size m.

Layers	Formulas	Ex. (m=64, n=9)
Fully Connected Layer	$m \times n$	576
Recurrent layer	$m \times n + n^2$	657
Long Short Term Memory	$4 \times (m \times n + n^2)$	2628
CNN	$m \times n$	576

still below human accuracy (88.4%).

Focus on the number of parameters in networks. The capacity of a neural network model can be define following its size and its depth. Higher are the size and the depth, higher is the number of parameters. Differences in the computational complexity between models are not exactly linearly comparable to their number of parameters, as some layers can see their computational time decreases dramatically using parallel computing (e.g. convolutional layer). However, it is a good start to study the overall complexity differences between models. Formulas giving the number of parameters of different layers are shown in Table 6.10.

The R3DCNN [19] contains $79,116,288$ parameters distributed as follows: they extract spatiotemporal features using a 3D CNN of 8 convolutional steps and two fully connected layers of size 4096 which together contains $77,885,776$ parameters. They append a recurrent layer of size 256 ($1,114,112$ parameters) and a softmax layer ($6,400$ parameters).

Our approach extracts hand shape and joint features from a single light 2D CNN with 3 convolutional layers and two fully connected of size 1024 which together contains $3,182,414$ parameters. Our method uses also two-stacked LSTM layers, both containing $14,680,064$ parameters and ends on a single *softmax* layer of $12,800$ parameters. The whole pipeline of our approach contains $32,555,342$ parameters, so, less than half the number contained in the R3DCNN [19] network and still outperforms their accuracy result by 1%.

The transfer strategy using a hand pose estimator to extract hand shape and joint features allowed us to perform better while using a far less complex network

Limitations and discussion. Experiments showed that the proposed solutions guarantee an effective dynamic hand gesture recognition, but still not exceed the human performance, and gestures which contain high hand shape similarities still showed confusions.

Some of these confusion due to the fact that different phases of inverse gestures may contain high similarities. For example, as depicted in Figure 6.18, the pre-stroke phase of a *Swipe left* gesture

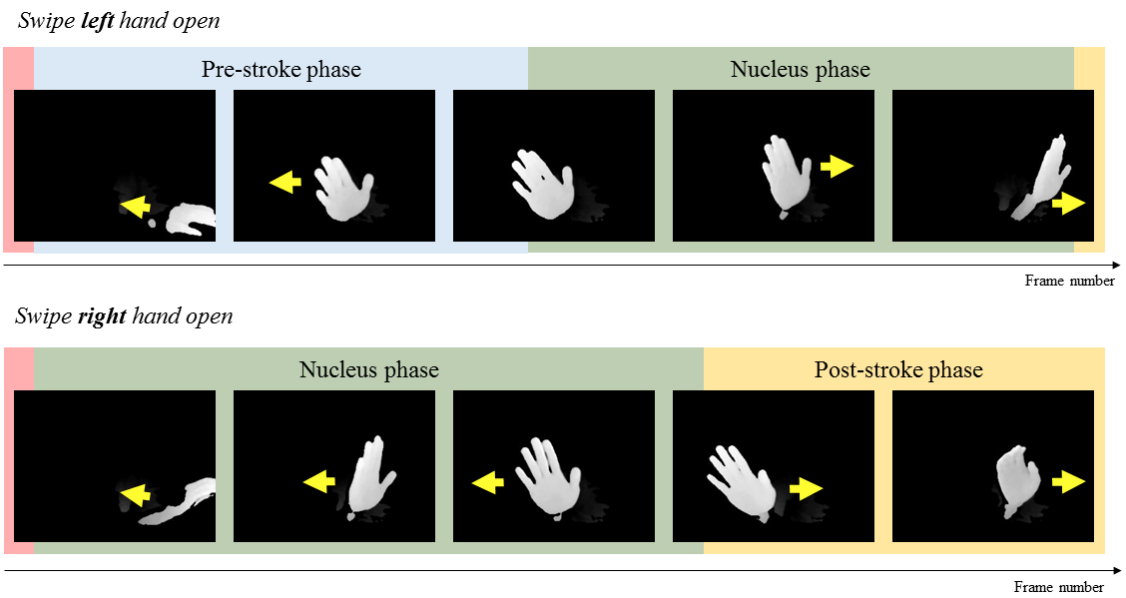


Figure 6.18: An example of a gesture *Swipe Left* (up) and *Swipe Right* (down), both hand open, and their respective phases (blue: pre-stroke, green: nucleus, orange: post-stroke). The figure shows high similarities between the pre-stroke phase of the *Swipe Left* gesture and the nucleus phase of *Swipe Right* gesture.

consists in moving the hand to the right so that the camera is able to see the entire gesture. However, this movement to the right can be seen as a *Swipe right* gesture nucleus by the localization algorithm and not as a pre-stroke phase of a *Swipe left* gesture.

Evaluation results for the online detection and recognition show that we can detect an arising gesture after only 21% of its nucleus. However, some missclassification appear during the first few frames of gestures where the algorithm has been able to detect a gesture in progress but does not yet have sufficient information to correctly recognize its type. Figure 6.19a illustrates the output of our approach on a test sequence of gestures. In this case, the result is almost perfect, each of the 10 gestures is correctly labeled after only few frames. In contrast, Figure 6.19b shows a test sequence where 5 out of 10 gestures have a misclassification during the first few frames. The issues resulting from those misclassifications could be overcome by firing an incoming gesture only if its length is longer than a threshold.

6.4 Conclusion

In this chapter, we addressed several issues of dynamic hand gesture recognition from depth data, a widely investigated topic due to its wide range of potential applications.

In the first part, we addressed the problem of dynamic hand gesture recognition as a traditional handcrafted method using three gestural features computed from hand skeletal presegmented sequences. Each set of these geometric features was encoded in a statistical representation using a Fisher Kernel followed by a temporal pyramid model before a classification process. The evaluation of our approach showed a promising potential of the use of skeletal features to perform hand gesture recognition. Evaluation results demonstrated the efficiency of our approach over the depth image based descriptors. However, the method presents limitations for online scenarios and the results also revealed a lack of precision to describe the dynamic of complex hand gestures, compared with the feature learning power of modern deep learning approaches.

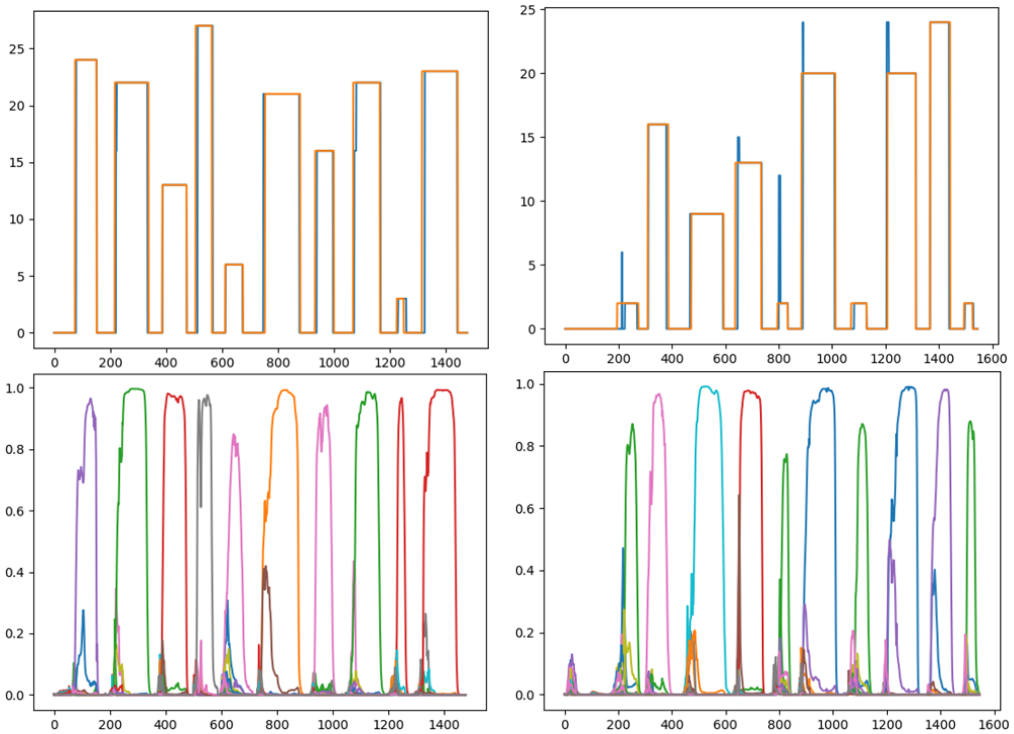


Figure 6.19: The gesture detection and recognition performance of our approach on a continuous video stream of 10 gestures. The top figures illustrate the classification outputs (blue) versus the ground-truth (orange) where the x-axis is the time in number of frames and the y-axis represents the class outputs. The bottom figures represent the accuracy for each gesture types along the time where various colors indicate different gesture types. The *no_gesture* class is not shown. (a) illustrates an almost perfect recognition result and the top curve of (b) shows that five gestures are miss-classified during the first frames of their nucleus

Then, We proposed in the second part an online recognition system capable to detect the presence of a gesture in an unsegmented video stream and to recognize the type of the gesture before its end, which is an essential capacity for real-world applications. In this approach, we have taken over the whole pipeline of the recognition process, from hand pose estimation to the classification step, and used the power of deep learning models to increase the efficiency and the robustness of our system. Our model architecture consists of combined convolutional and recurrent networks, designed in a way to extract both hand posture and shape features from depth images. The knowledge of the CNN, trained using a large hand pose estimation dataset, is transferred to extract relevant features describe the gesture. Recurrent networks model separately the temporal variations of hand postures and its shapes. Experimental results demonstrated that the proposed approach is capable to recognize hand gestures and to improve state-of-the-art results. In addition, tests on two challenging datasets showed that our designed system is able to detect an occurring gesture after only 22% of the nucleus phase, detected by adding a *garbage* class in training phase.

It is important to emphasize the contribution of the transfer learning strategy employed in our approach. Indeed, we used a CNN model which can take image as input to extract both hand posture and shape features. We transferred the knowledge of the CNN, trained using a large hand pose estimation dataset, to extract relevant features describe the gesture. Thus, hand gestures originally represented as depth image sequences are encoded into two distinct components: hand joint and hand shape feature sequences. Finally, we used two recurrent networks to model separately the temporal variations of hand postures and its shapes, before merging the outputs of both networks to obtain a single label by gesture. To perform hand gesture detection, we added a *garbage* class from all frames that do not belong to a nucleus phase, i.e. where the gesture occurs. The use of the transfer learning strategy allowed us to outperform state-of-the-art deep learning approaches using less than half of the number of parameters of the baseline model. However, we limited our experiments for only two hand gesture datasets simulating human-computer interface based on hand gestures acquired in online scenario. These datasets are captured with short-rang depth cameras (SoftKinetic DS325 and RealSense SR300) and contain a set of heterogeneous gesture types. Other datasets such as [23, 63, 92] have been discarded since hand pose estimation becomes very difficult to achieve using long-range depth cameras like Microsoft Kinect. Thus, the hand pose estimation needs to be improved and the temporal modeling using recurrent layers need also to be more investigated for more precise and fine hand gestures recognition.

CHAPTER 7

Conclusion and perspectives

7.1 Conclusion

This manuscript describes our major contributions made during the last eight years to the area of human behaviour understanding.

Our aims were to develop new theoretical and application approaches for interconnected open problems of human behaviour understanding in the context of action, gesture and activity recognition from 3D data streams. Since movements unfold in both space and time, it is mandatory to provide solutions that describe the spatial and temporal properties of human movement and examine how the the variation in both spatial and temporal influence the recognition of the meaning of the motion. To better understand these spatiotemporal data regardless of the application scenario, it is necessary to provide solutions that answer the following questions: What are the most relevant features to consider? What is the appropriate representation? How to exploit the shape of the motion? How to model its dynamics? How to make this model independent of the temporal variations of the execution? How to design the classification process and to adapt it in the real world scenario?. The research activities we have conducting are organized around these issues.

In this manuscript, a first particular focus is given to fully reconstructed human bodies in 3D videos in order to study the problem 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 motion sequences can be analyzed from mainly two perspectives, the feature space and the model space. These spaces can be described mathematically as varieties. In fact, we have followed the significant progress made over the past decade in the analytical and geometric understanding of these spaces. Therefore, we proposed different adequate geometric frameworks in order to model and compare accurately human motion acquired from 3D sensors.

In the first framework, we addressed the problem of human pose and motion retrieval in full 3D reconstructed sequences. The human shape representation is formulated using extremal curve extracted from the body surface. It allowed an efficient shape to shape comparison taking benefits from Riemannian geometry in the open curve shape space. The shape analysis idea was extended to the action recognition problem from human skeletal sequences. The embedding of action sequences in such a shape space manifold allows to capture simultaneously the body shape and the dynamics of the motion. The action recognition is then formulated as the problem of computing the similarity between shape of trajectories in a Riemannian framework.

We proposed a second Riemannian framework for modelling and recognizing human actions acquired by depth cameras. In this framework, we model sequence features temporally as subspaces lying in Grassmanns, which are manifolds of linear subspaces. Two kinds of feature are used in this framework: 3D human joints extracted directly by depth camera, and local oriented displacement features extracted from boxes around each subject in depth frames. Then, we performed a learning process on Grassmann manifold by embedding each action, presented as a point on this manifold, in higher dimensional representation. The embedding is performed using the notion of tangent spaces approximation on specific classes, providing a natural separation of action classes. Experimental results demonstrated the efficiency of the proposed methods to recognize

human gesture and actions from depth sequences, and also revealed an important ability for a low-latency recognition system. However, experiments also demonstrated that the proposed solution suffers from limitations when actions involve long sequences and/or variable repetitions of a single gesture or manipulations of objects.

Therefore, we extend our study to address more complex behaviors, like activities, by analyzing the evolution of the human posture shape in order to decompose the motion stream into short motion units. Each motion unit is then characterized by the motion trajectory and depth appearance around hand joints, so as to describe the human motion and interaction with objects. Finally, the sequence of temporal segments is modeled through a Dynamic Naive Bayesian Classifier. The combination of skeleton and depth appearance features around hand joint, as well as the modeling of the dynamics of the sequence thanks to the segmentation into motion units, show the potential of our approaches for the task of online behavior detection and recognition in long sequences, which is an important challenge in real-world contexts. However, the consideration of appearances around the hand sills insufficient to interpret fine hand gestures in an HCI scenario, which is a critical problem for behaviour understanding.

In order to go deeper into the analysis of such HCI scenarios, we focused our study on the analysis and the recognition of hand gestures. We firstly proposed a traditional geometric approach, taking into consideration the complex topology of the hand and the endless possibilities to perform the same gesture, using hand shape and motion descriptors computed on 3D hand skeletal features. Finally, motivated by the powerful capability of deep neural networks in learning compact and discriminative representations for images and videos, we moved our focus of interest towards end-to-end data-driven approaches, taking the original depth images as input. Therefore, we have taken over the whole pipeline of the recognition process, from hand pose estimation to the classification phase using the power of deep learning models. Our proposed system performs the recognition of ongoing gesture, which allows the system to detect the presence of a gesture in an unsegmented video stream and to recognize its type before its end, which is an essential capacity for an IHC application.

We have demonstrated the interest of the proposed approaches through the multiple experiments conducted on publicly available datasets in terms of action and gesture recognition (MSR Action 3D [165], Florence 3D Action [81], UTKinect [113], UCF-kinect dataset [99] and MSR Gesture 3D [119]), activity recognition (MSRC-12 [40], CAD120 [66], MAD [58] and Online RGB-D [44]) and hand gesture detection and recognition (Handicraft-Gesture [20], NVIDIA Dynamic Hand Gesture [19], DHG 14-28 [C5] and Online DHG [C3] datasets). Furthermore, the evaluation in terms of latency clearly demonstrates the efficiency of proposed approaches for a rapid recognition. Nevertheless, there are still several open problems and research leads from theoretical and practical aspects, which we would like to develop in future work.

7.2 Perspectives and future research

Generalization of deep neural network paradigm to non-Euclidean manifolds In this manuscript, we presented in Chapters 4 and 5 different geometric approaches to recognize human action and or activity from 3D stream (depth, skeleton, ...). In these approaches, we explore the characteristics of manifolds, like shape space and Grassmann, and perform recognition based on intrinsic geometry of data space, by introducing a learning algorithm on the manifold, and sometimes in conjunction with dynamic modelling process. Most of deep learning approaches, in the other hand, being applied on Euclidean structured data such as images and videos. However, there

was recently a growing interest to extend these techniques to non-Euclidean data such as manifolds [1, 2, 9]. In the meantime, preliminary results have been published in [C2], in which we proposed a deep neural network architecture taking directly as input geometric features extracted from data laying on a non-Euclidean space. These later have been recently shown to be very effective to capture the geometric structure of the human pose. In particular, we claimed in our approach to incorporate the intrinsic nature of the data characterized by Lie Group into deep neural networks and to learn more adequate geometric features for 3D action recognition problem. We believe that this field deserves to be better explored, as instead of enforcing geometry at the inputs of networks, it could be better to extend network architectures where outputs lie on manifolds.

Optimization of hand gesture detection and recognition approach The strong ability of the recent proposed deep neural networks based approaches (RNN and CNN) in learning spatiotemporal representations, outperforms the previous handcrafted feature based methods in many areas. However, despite an increasing amount of methods proposed over the last few years, defining an online dynamic hand gesture recognition system robust enough to work in real world applications is still very challenging. Thus, although many advances have taken place in recognition accuracy achieved by these approaches, there are still several challenges ahead, but also problems that can limit the performance. Already convinced by these techniques, we continue to seek ways to improve their efficiency for hand pose estimation and hand gesture recognition.

Experimental results obtained in Chapter 6 showed that the proposed solutions guarantee an effective dynamic hand gesture recognition but still need improvements, since they are still far from the human performance. First, hand pose estimation is still an active field of research and the model designed to extract features used in our framework can be enhanced. Recently, Yuan *et al.* [4] introduced the million-scale *BigHand2.2M* benchmark, that makes a significant advancement in terms of completeness of hand data variation and annotation quality compared to existing benchmarks. It should help to improve the effectiveness of hand pose estimation and also hand gesture recognition, being two interconnected problems. Second, several works successfully built well-designed RNN configurations for action recognition from skeletal human sequences [10, 21, 37]. As a new field of study, hand gesture recognition using temporal learned features on skeletal sequences has been only partially studied. Going deeper into the temporal modeling using recurrent layers could provide more efficient ways to distinguish actions gestures with high similarities. In addition, new recurrent layers have appeared recently providing an attentional system. Such networks are able to selectively focus on the informative joint skeletons along a sequence. For example, the system could automatically detect the the fingers which provides the most reliable information and focus on them to perform the gesture recognition.

Improvement of transfer learning by regularization Skeleton-based representation of motion as time series of 3D joint positions constitute models, which are computationally efficient, and substantially simplifies the view on the complex human body and/or hand locomotion. However, these models suffer from limitations related to the costly manual annotation of skeleton sequences, while automatic annotation methods may yield inaccurate predictions.

In Chapter 6, we employed a transfer learning strategy from hand pose dataset towards gesture recognition task, and focused on abstract hand gestures (i.e. each gesture has a specific meaning for the system). Meanwhile, in 2017, Garcia *et al.* [8] introduced a hand daily life activities dataset providing sequences of depth images and accurate hand poses. They obtained the best performance using a recurrent neural network on the hand skeletal features. They extracted the hand pose data

from a kind of data-glove, since state-of-the-art methods in terms of hand pose estimation still face several issues with fast moving hands or finger self-occlusions. However, the use of a data-glove in real applications is not suitable. We believe that augmenting the training data with additional information from another complementary modality could highlight characteristics, that are important to the recognition process, missing or poorly represented in the motion sequence. More particularly, we plan to exploit these annotated 3D joints extracted from the data-glove during the training phase to regularize the deep learning model in focusing on relevant features from depth images. Once the model is trained, the skeleton data are not needed anymore for the testing phase. The skeleton model of the hand, has a better ability to be dataset-invariant motion since background context is excluded, to facilitate the capture of important movement patterns of 3D finger joints, and thus to improve the performance of a deep neural network like RNN. This approach can be generalized to other motion components problems, like action and activity recognition, from depth and/or RGB streams.

Deep cross-modal learning While deep learning techniques has certainly evolved considerably the last few years, they are still hard to train and optimize and most of today's applications are typically designed to only perform a single task. The generalization using transfer learning helps such a network to reuse the representations of characteristics within the same domain, as we did in Chapter 6 to learn hand features, from depth image dataset originally created for hand pose estimation, for gesture recognition task. Recently, a new work presented by Kaiser *et al.* from Google [7], outlines a single machine learning template that can perform different tasks efficiently. The objective behind their algorithm is to create a single deep learning model that can learn tasks from multiple areas, such as machine translation, image classification, speech recognition and language parsing. Their model did not show particular improvements over individual models but it highlighted some areas on which learning processes can be drastically improved by sharing knowledge from different domains. Additionally, the approach seems to require less training data than traditional algorithms in order to achieve similar levels of efficiency.

Inspired by this work, the aim of our future work could be to create a unified motion specific deep learning model to solve tasks across multiple human motion modalities – like pose estimation, gesture, action and activity recognition –, which may be acquired from multiple stream modalities – like depth map, skeleton, color, ...–. This could be extended later it to more fonctionalities like motion prediction, synthesis and transfer. The diversity of data input types in our case is less challenging comparing to the Kaiser's MultiModel because the motion data inputs are of same nature.

The desired model consists of individual "sub-networks" to process specific inputs, like hand pose data, gesture and activity streams, and transform it into a uniform representation, which has the advantage of being variable in size. Such a model makes possible transfer learning from tasks. We do not have a single network of modalities per task, but a single network per modality, where different motion tasks share the same modality networks. Similar to Kaiser's model, our model can be seen as a combination of encoders, mixers and decoders and each one of those blocks are architectured using a combination of convolutional, attention and mixture-of-experts blocks. The outputs of the modality nets become the inputs to a shared encoder which creates the unified representation. An I/O mixer combines the encoded inputs with the previous outputs, and a decoder processes the inputs and the mixture to generate new outputs. The implementation of such an architecture offers solutions to interconnected problems, often necessary to solve in a human behavior understanding.

Style motion analysis Human motion can be considered as a combination of two sets of features: action variations –specifying actions like walking, jumping, punching, kicking, etc.– and stylistic variations –related to emotions, individual characteristics, etc.– in which the motions are performed. While actions have already been quite studied for a while, style in human body motion is a new growing topic of interest. Accurate realistic motions can be obtained either by the considerable work of 3D animators who manipulate details, or by capturing motions. The animators' work is time consuming, expensive and tedious, as they do the animations from scratch by hand most of the time [13]. Capturing motions is also time consuming and a burden for actors, especially when combinations of actions and styles are needed and should ideally be performed several times [62]. One way of overcoming this is by generating new motions, thus reducing the amount of captures needed in datasets and saving animators time [13, 22]. As a result, generating new stylistic 3D human body motion is a problem that is of concern to researchers.

Style in 3D human body motion is therefore studied, particularly when it comes to stylistic motion generation. We distinguish three types of motion style generation: motion style synthesis, motion style editing and motion style transfer. We define the motion style editing as a subpart of motion style synthesis that implies the user intervention, and the motion style transfer as the process of transforming an input motion into a new style while preserving its original content. It is also studied as a classification problem and can be a tool to identify persons. A recent work on the style in 3D human body motion is realized [P2] to address the different trends about its taxonomy, data and applications.

Following the success of deep learning methods for hand gesture recognition tasks, our ongoing work focuses on using deep learning models, like RNNs and CNN, to model human motion, with a particular focus on learning time-dependent representations, able to perform tasks not only for motion recognition, but also short-term prediction and synthesis of human motion in general and style motion in particular. We intend to introduce in the second time an attentional layer by focusing on informative body joints. We believe that this objective constitutes a perfect field of application of the unified motion specific deep learning model presented above to solve tasks across multiple human motion modalities.

HCI in virtual environments Human motion modeling is a problem at the intersection of computer vision and computer graphics, with applications spanning HCI, motion recognition, motion synthesis and motion prediction for virtual and augmented reality. The needs of precise mid-air HCI for applications, like interaction with a virtual or augmented reality world, attracted particular attention in the Computer Vision community [24, 30–32, 34, 55]. Indeed, an efficient gesture recognition system acting as interface with a virtual world can improve the quality of the interaction with the computer. A particular interest has been focused, in Chapter 6, on a finite set of abstract gestures, which imposes a very restrictive use of gestures to interact with a virtual world. Even though our developed system will be able to recognize that a user is grasping a virtual object, it does not have precise information about the required transformation to apply on the object to simulate the real world. To be able to fully reproduce the interaction in the virtual world between the object and the hand, a lot of physical rules has to be taken into account.

Our hand gesture recognition, as most of current systems, is intrinsically indirect interaction system and, so, can seem unnatural to users. The barrier of bringing the sense of touch into a virtual world is a current issue with many challenges. We can imagine futuristic applications where the limit between the real world and a virtual one is blurring. A potential issue to consider in future work is to develop approaches able to reproduce the physical rules that guide the interaction

between objects and haptic interfaces which can reproduce the sense of touch in a virtual reality application.

A first effort along this line of research was started very recently through the thesis of Théo Voillemin on *personalized augmented reality assistance by hand gesture recognition on head-mounted displays*. Indeed, recent advances in the development of optical head-mounted displays, such as Microsoft HoloLens or Epson Moverio, which overlay visual information directly in the user's field of vision, have opened up new possibilities for augmented reality applications. The aim of the proposed thesis is the development of an assistant system, that use such displays, to assist the user during activities in an intuitive and discreet manner. To this end, this system observes the hands of the user and generates contextual comments based on the recognition of his gestures. Domain of assisted surgery, self-rehabilitation, and automobile industry as advanced assistance to driver, could be potential applications of desired system. This project is at the crossroad of the computer vision, augmented reality and machine learning domains.

Bibliography

[1] R. WANG, X. WU, and J. KITTLER. "A Simple Riemannian Manifold Network for Image Set Classification". In: *CoRR* abs/1805.10628, 2018.

[2] Z. HUANG, J. WU, and L. V. GOOL. "Building Deep Networks on Grassmann Manifolds". In: *AAAI*. AAAI Press, 2018.

[3] Q. DE SMEDT. "Dynamic hand gesture recognition - From traditional handcrafted to recent deep learning approaches ". Theses. Université de Lille 1, Sciences et Technologies; CRIStAL UMR 9189, Dec. 2017.

[4] S. YUAN, Q. YE, B. STENGER, S. JAIN, and T.-K. KIM. "BigHand2. 2M Benchmark: Hand Pose Dataset and State of the Art Analysis". In: *arXiv preprint arXiv:1704.02612*, 2017.

[5] H. WANG and L. WANG. "Modeling Temporal Dynamics and Spatial Configurations of Actions Using Two-Stream Recurrent Neural Networks". In: *arXiv preprint arXiv:1704.02581*, 2017.

[6] J. LIU, G. WANG, P. HU, L.-Y. DUAN, and A. C. KOT. "Global context-aware attention lstm networks for 3d action recognition". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 7. 2017.

[7] L. KAISER, A. N. GOMEZ, N. SHAZEER, A. VASWANI, N. PARMAR, L. JONES, and J. USZKOREIT. "One Model To Learn Them All". In: *arXiv preprint arXiv:1706.05137*, 2017.

[8] G. GARCIA-HERNANDO, S. YUAN, S. BAEK, and T.-K. KIM. "First-Person Hand Action Benchmark with RGB-D Videos and 3D Hand Pose Annotations". In: *arXiv preprint arXiv:1704.02463*, 2017.

[9] M. M. BRONSTEIN, J. BRUNA, Y. LECUN, A. SZLAM, and P. VANDERGHEYNST. "Geometric Deep Learning: Going beyond Euclidean data". In: *IEEE Signal Process. Mag.* 34 (4), 2017, pp. 18–42.

[10] A. SHAHROUDY, J. LIU, T.-T. NG, and G. WANG. "NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[11] X. ZHOU, Q. WAN, W. ZHANG, X. XUE, and Y. WEI. "Model-based deep hand pose estimation". In: *arXiv preprint arXiv:1606.06854*, 2016.

[12] G. ZEN, L. PORZI, E. SANGINETO, E. RICCI, and N. SEBE. "Learning personalized models for facial expression analysis and gesture recognition". In: *IEEE Transactions on Multimedia* 18 (4), 2016, pp. 775–788.

[13] M. E. YUMER and N. J. MITRA. "Spectral style transfer for human motion between independent actions". In: *ACM Transactions on Graphics (TOG)* 35 (4), 2016, pp. 137–144. DOI: 10.1145/2897824.2925955.

[14] Q. YE, S. YUAN, and T.-K. KIM. "Spatial attention deep net with partial PSO for hierarchical hybrid hand pose estimation". In: *European Conference on Computer Vision*. Springer. 2016, pp. 346–361.

[15] D. WU, L. PIGOU, P.-J. KINDERMANS, N. D.-H. LE, L. SHAO, J. DAMBRE, and J.-M. ODOBEZ. "Deep dynamic neural networks for multimodal gesture segmentation and recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 38 (8), 2016, pp. 1583–1597.

[16] A. SHAHROUDY, J. LIU, T.-T. NG, and G. WANG. "NTU RGB+ D: A large scale dataset for 3D human activity analysis". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 1010–1019.

[17] I. REALSENSE. *http://www.intel.com/realsense*. 2016.

[18] N. NEVEROVA, C. WOLF, G. TAYLOR, and F. NEBOUT. "Moddrop: adaptive multi-modal gesture recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38 (8), 2016, pp. 1692–1706.

[19] P. MOLCHANOV, X. YANG, S. GUPTA, K. KIM, S. TYREE, and J. KAUTZ. "Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4207–4215.

[20] W. LU, Z. TONG, and J. CHU. "Dynamic hand gesture recognition with leap motion controller". In: *IEEE Signal Processing Letters* 23 (9), 2016, pp. 1188–1192.

[21] J. LIU, A. SHAHROUDY, D. XU, and G. WANG. "Spatio-temporal LSTM with trust gates for 3D human action recognition". In: *European Conference on Computer Vision*. Springer. 2016, pp. 816–833.

[22] D. HOLDEN, J. SAITO, and T. KOMURA. "A deep learning framework for character motion synthesis and editing". In: *ACM Transactions on Graphics (TOG)* 35 (4), 2016, pp. 138–149.

[23] S. GUPTA, P. MOLCHANOV, X. YANG, K. KIM, S. TYREE, and J. KAUTZ. "Towards selecting robust hand gestures for automotive interfaces". In: *Intelligent Vehicles Symposium (IV), 2016 IEEE*. IEEE. 2016, pp. 1350–1357.

[24] L. GE, H. LIANG, J. YUAN, and D. THALMANN. "Robust 3D hand pose estimation in single depth images: from single-view CNN to multi-view CNNs". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 3593–3601.

[25] M. DEVANNE. "3D Human Behavior Understanding by Shape Analysis of Human Motion and Pose". Thesis. Université Lille 1 - Sciences et Technologies, Dec. 2015.

[26] L. SUN, Z. LIU, and M.-T. SUN. "Real time gaze estimation with a consumer depth camera". In: *Information Sciences* 320 (C), Nov. 2015, pp. 346–360.

[27] C. WANG, Z. LIU, and S.-C. CHAN. "Superpixel-based hand gesture recognition with kinect depth camera". In: *IEEE transactions on multimedia* 17 (1), 2015, pp. 29–39.

[28] D. TRAN, L. BOURDEV, R. FERGUS, L. TORRESANI, and M. PALURI. "Learning spatiotemporal features with 3d convolutional networks". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 4489–4497.

[29] X. SUN, Y. WEI, S. LIANG, X. TANG, and J. SUN. "Cascaded hand pose regression". In: *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 824–832.

[30] X. SUN, Y. WEI, S. LIANG, X. TANG, and J. SUN. "Cascaded hand pose regression". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 824–832.

[31] M. OBERWEGER, P. WOHLHART, and V. LEPETIT. "Hands deep in deep learning for hand pose estimation". In: *arXiv preprint arXiv:1502.06807*, 2015.

[32] M. OBERWEGER, P. WOHLHART, and V. LEPETIT. "Training a feedback loop for hand pose estimation". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 3316–3324.

[33] P. MOLCHANOV, S. GUPTA, K. KIM, and J. KAUTZ. "Hand gesture recognition with 3D convolutional neural networks". In: *IEEE conference on computer vision and pattern recognition workshops*. 2015, pp. 1–7.

[34] H. LIANG, J. YUAN, and D. THALMANN. "Resolving ambiguous hand pose predictions by exploiting part correlations". In: *IEEE Transactions on Circuits and Systems for Video Technology* 25 (7), 2015, pp. 1125–1139.

[35] S.-Z. LI, B. YU, W. WU, S.-Z. SU, and R.-R. JI. "Feature learning based on SAE–PCA network for human gesture recognition in RGBD images". In: *Neurocomputing* 151, 2015, pp. 565–573.

[36] R. S. GHIASS, O. ARANDJELOVIĆ, and D. LAURENDEAU. "Highly Accurate and Fully Automatic Head Pose Estimation from a Low Quality Consumer-Level RGB-D Sensor". In: *Work. on Computational Models of Social Interactions: Human-Computer-Media Communication*. 2015, pp. 25–34.

[37] Y. DU, W. WANG, and L. WANG. "Hierarchical recurrent neural network for skeleton based action recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1110–1118.

[38] R. SLAMA. "Geometric Approaches for 3D Human Motion Analysis: Application to Action Recognition and Retrieval". Thesis. Université Lille 1 - Sciences et Technologies, Oct. 2014.

[39] D. TANG, H. J. CHANG, A. TEJANI, and T. K. KIM. "Latent Regression Forest: Structured Estimation of 3D Articulated Hand Posture". In: *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014, pp. 3786–3793. DOI: 10.1109/CVPR.2014.490.

[40] A. LEHRMANN, P. GEHLER, and S. NOWOZIN. "Efficient Nonlinear Markov Models for Human Motion". In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. Columbus, OH, USA, June 2014, pp. 1314–1321.

[41] F. ZHOU, F. DE LA TORRE, and J. K. HODGINS. "Hierarchical Aligned Cluster Analysis for Temporal Clustering of Human Motion". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (3), Mar. 2014, pp. 582–596.

[42] L. RYBOK, B. SCHAUERTE, Z. AL-HALAH, and R. STIEFELHAGEN. "Important Stuff, Everywhere! Activity Recognition with Salient Proto-Objects as Context". In: *IEEE Winter Conf. on Applications of Computer Vision (WACV)*. Steamboat Springs, CO, Mar. 2014, pp. 646–651.

[43] A. W. VIEIRA, E. R. NASCIMENTO, G. L. OLIVEIRA, Z. LIU, and M. F. CAMPOS. "On the improvement of human action recognition from depth map sequences using SpaceâTime Occupancy Patterns". In: *Pattern Recognition Letters* 36, Jan. 2014, pp. 221–227.

[44] G. YU, Z. LIU, and J. YUAN. "Discriminative Orderlet Mining For Real-time Recognition of Human-Object Interaction". In: *Asian Conference on Computer Vision (ACCV)*. Singapore, 2014.

[45] X. YANG and Y. TIAN. "Super normal vector for activity recognition using depth sequences". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 804–811.

[46] 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.

[47] J. TOMPSON, M. STEIN, Y. LECUN, and K. PERLIN. "Real-time continuous pose recovery of human hands using convolutional networks". In: *ACM Transactions on Graphics (ToG)* 33 (5), 2014, p. 169.

[48] D. TANG, H. JIN CHANG, A. TEJANI, and T.-K. KIM. "Latent regression forest: Structured estimation of 3d articulated hand posture". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 3786–3793.

[49] X. Z. S. ALTHLOOTHI M. H. MAHOOR and R. M. VOYLES. "Human activity recognition using multi-features and multiple kernel learning". In: *Pattern Recognition*. Vol. 47. 5. 2014, pp. 1800–1812.

[50] D. PICKUP, X. SUN, P. ROSIN, 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.

[51] X. PENG, C. ZOU, Y. QIAO, and Q. PENG. "Action recognition with stacked fisher vectors". In: *European Conference on Computer Vision*. Springer. 2014, pp. 581–595.

[52] E. OHN-BAR and M. M. TRIVEDI. "Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations". In: *IEEE transactions on intelligent transportation systems* 15 (6), 2014, pp. 2368–2377.

[53] C. MONNIER, S. GERMAN, and A. OST. "A Multi-scale Boosted Detector for Efficient and Robust Gesture Recognition." In: *ECCV Workshops (1)*. 2014, pp. 491–502.

[54] H.-I. LIN, M.-H. HSU, and W.-K. CHEN. "Human hand gesture recognition using a convolution neural network". In: *IEEE International Conference on Automation Science and Engineering*. IEEE. 2014, pp. 1038–1043.

[55] H. LIANG, J. YUAN, and D. THALMANN. "Parsing the hand in depth images". In: *IEEE Transactions on Multimedia* 16 (5), 2014, pp. 1241–1253.

[56] A. KARPATHY, G. TODERICI, S. SHETTY, T. LEUNG, R. SUKTHANKAR, and L. FEI-FEI. "Large-scale video classification with convolutional neural networks". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2014, pp. 1725–1732.

[57] K. IOANNIS and N. NIKOS. "Action recognition on motion capture data using a dynemes and forward differences representation". In: *Journal of Visual Communication and Image Representation* 25 (6), 2014, pp. 1432–1445.

[58] D. HUANG, Y. WANG, S. YAO, and F. D. LA TORRE. "Sequential Max-Margin Event Detectors". In: *European Conference on Computer Vision (ECCV)*. Zurich, Swiss, 2014.

[59] M. HOAI and F. DE LA TORRE. "Max-margin early event detectors". In: *International Journal of Computer Vision* 107 (2), 2014, pp. 191–202.

[60] 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*. Vol. 36. 2014, pp. 1414–1427.

[61] G. EVANGELIDIS, G. SINGH, and R. HORAUD. "Skeletal quads: Human action recognition using joint quadruples". In: *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE. 2014, pp. 4513–4518.

[62] S. A. ETEMAD and A. ARYA. "Classification and translation of style and affect in human motion using RBF neural networks". In: *Neurocomputing* 129, 2014, pp. 585–595.

[63] S. ESCALERA, X. BARÓ, J. GONZALEZ, M. Á. BAUTISTA, M. MADADI, M. REYES, V. PONCE-LÓPEZ, H. J. ESCALANTE, J. SHOTTON, and I. GUYON. "ChaLearn Looking at People Challenge 2014: Dataset and Results." In: *ECCV Workshops (1)*. 2014, pp. 459–473.

[64] M. BARNACHON, S. BOUAKAZ, B. BOUFAMA, and E. GUILLOU. "Ongoing human action recognition with motion capture". In: *Pattern Recognition*. Vol. 47. 1. 2014, pp. 238–247.

[65] L. LIU, L. SHAO, X. ZHEN, and X. LI. "Learning Discriminative Key Poses for Action Recognition". In: *IEEE Transactions on Cybernetics* 43 (6), Dec. 2013, pp. 1860–1870. ISSN: 2168-2267. DOI: 10.1109/TSMCB.2012.2231959.

[66] H. S. KOPPULA, R. GUPTA, and A. SAXENA. "Learning human activities and object affordances from RGB-D videos". In: *International Journal of Robotics Research* 32 (8), July 2013, pp. 951–970.

[67] Y. ZHENG, C.-L. TAI, E. ZHANG, and P. XU. "Pairwise Harmonics for Shape Analysis". In: *IEEE Transactions on Visualization and Computer Graphics*. Vol. 19. 7. Los Alamitos, CA, USA, 2013, pp. 1172–1184.

[68] C. ZHANG, X. YANG, and Y. TIAN. "Histogram of 3D facets: A characteristic descriptor for hand gesture recognition". In: *10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*. IEEE. 2013, pp. 1–8.

[69] M. ZANFIR, M. LEORDEANU, and C. SMINCHISESCU. "The Moving Pose: An Efficient 3D Kinematics Descriptor for Low-Latency Action Recognition and Detection". In: *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*. Sydney, AUstralia: IEEE, 2013, pp. 2752–2759.

[70] M. YE, Q. ZHANG, L. WANG, J. ZHU, R. YANG, and J. GALL. "A Survey on Human Motion Analysis from Depth Data". In: *CVPR Tutorial on RGBD Cameras*. Portland, USA, 2013.

[71] 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*. Vol. 8200. 2013, pp. 149–187.

[72] L. XIA and J. AGGARWAL. "Spatio-temporal Depth Cuboid Similarity Feature for Activity Recognition Using Depth Camera". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 2834–2841.

[73] L. XIA and J. K. AGGARWAL. "Spatio-Temporal Depth Cuboid Similarity Feature for Activity Recognition Using Depth Camera". In: *Proc. CVPR Work. on Human Activity Understanding from 3D Data*. Portland, Oregon, USA, 2013, pp. 2834–2841.

[74] P. WEI, Y. ZHAO, N. ZHENG, and S. ZHU. "Modeling 4D Human-Object Interactions for Event and Object Recognition". In: *International Conference on Computer Vision (ICCV)*. Sydney, Australia, 2013.

[75] H. WANG and C. SCHMID. "Action recognition with improved trajectories". In: *Proceedings of the IEEE international conference on computer vision*. 2013, pp. 3551–3558.

[76] H. WANG, C. YUAN, G. LUO, W. HU, and C. SUN. "Action recognition using linear dynamic systems". In: *Pattern Recognition*. Vol. 46. 6. 2013, pp. 1710–1718.

[77] 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*. Vol. 7725. 2013, pp. 525–538.

[78] J. SHOTTON, T. SHARP, A. KIPMAN, A. FITZGIBBON, M. FINOCCHIO, A. BLAKE, M. COOK, and R. MOORE. "Real-time human pose recognition in parts from single depth images". In: *Communications of the ACM* 56 (1), 2013, pp. 116–124.

[79] J. SHOTTON, A. FITZGIBBON, M. COOK, T. SHARP, M. FINOCCHIO, R. MOORE, A. KIPMAN, and A. BLAKE. "Real-Time Human Pose Recognition in Parts from Single Depth Images". In: *Machine Learning for Computer Vision*. Vol. 411. 2013, pp. 119–135.

[80] L. SEIDENARI, V. VARANO, S. BERRETTI, A. BIMBO, and P. PALA. "Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2013, pp. 479–485.

[81] 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*. 2013, pp. 479–485.

[82] L. SEIDENARI, V. VARANO, S. BERRETTI, A. D. BIMBO, and P. PALA. "Recognizing Actions from Depth Cameras as Weakly Aligned Multi-Part Bag-of-Poses". In: *Proc. CVPR Work. on Human Activity Understanding from 3D Data*. Portland, Oregon, USA, 2013, pp. 479–485.

[83] J. SÁNCHEZ, F. PERRONNIN, T. MENSINK, and J. VERBEEK. "Image classification with the fisher vector: Theory and practice". In: *International journal of computer vision* 105 (3), 2013, pp. 222–245.

[84] Z. REN, J. YUAN, J. MENG, and Z. ZHANG. "Robust part-based hand gesture recognition using kinect sensor". In: *IEEE transactions on multimedia* 15 (5), 2013, pp. 1110–1120.

[85] L. E. POTTER, J. ARAULLO, and L. CARTER. "The leap motion controller: a view on sign language". In: *Australian computer-human interaction conference: augmentation, application, innovation, collaboration*, 2013, pp. 175–178.

[86] 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*. 2013, pp. 716–723.

[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*. Washington, DC, USA, 2013, pp. 716–723.

[88] O. OREIFEJ and Z. LIU. "HON4D: Histogram of Oriented 4D Normals for Activity Recognition from Depth Sequences". In: *Proc. Int. Conf. on Computer Vision and Pattern Recognition*. Portland, Oregon, USA, 2013, pp. 716–723.

[89] E. OHN-BAR and M. TRIVEDI. "Joint angles similarities and HOG2 for action recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2013, pp. 465–470.

[90] E. OHN-BAR and M. M. TRIVEDI. "Joint Angles Similarities and HOG^2 for Action Recognition". In: *Proc. CVPR Work. on Human Activity Understanding from 3D Data*. Portland, Oregon, USA, 2013, pp. 465–470.

[91] MICROSOFT KINECT. http://www.microsoft.com/en-us/kinectforwindows. 2013.

[92] L. LIU and L. SHAO. "Learning Discriminative Representations from RGB-D Video Data." In: *IJCAI*. Vol. 4. 2013, p. 8.

[93] A. KUZNETSOVA, L. LEAL-TAIXÉ, and B. ROSENHAHN. "Real-time sign language recognition using a consumer depth camera". In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2013, pp. 83–90.

[94] H. S. KOPPULA and A. SAXENA. "Learning Spatio-Temporal Structure from RGB-D Videos for Human Activity Detection and Anticipation". In: *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*. Atlanta, USA, 2013.

[95] M. T. HARANDI, C. SANDERSON, S. SHIRAZI, and B. C. LOVELL. "Kernel analysis on Grassmann manifolds for action recognition". In: *Pattern Recognition Letters*. Vol. 34. 15. 2013, pp. 1906–1915.

[96] K. GUO, P. ISHWAR, and J. KONRAD. "Action Recognition From Video Using Feature Covariance Matrices". In: *IEEE Transactions on Image Processing*. Vol. 22. 6. 2013, pp. 2479–2494.

[97] K. A. FUNES MORA and J.-M. ODOBEZ. "Person independent 3D gaze estimation from remote RGB-D cameras". In: *IEEE Int. Conf. on Image Processing*. 2013, pp. 2787–2791.

[98] C. ELLIS, S. MASOOD, M. TAPPEN, J. LAVIOLA, and R. SUKTHANKAR. "Exploring the Trade-off Between Accuracy and Observational Latency in Action Recognition". In: *International Journal of Computer Vision*. Vol. 101. 2013, pp. 420–436.

[99] C. ELLIS, S. Z. MASOOD, M. F. TAPPEN, J. J. LA VIOLA JR., and R. SUKTHANKAR. "Exploring the Trade-off Between Accuracy and Observational Latency in Action Recognition". In: *Int. Journal on Computer Vision* 101 (3), 2013, pp. 420–436.

[100] 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*. Vol. 35. 9. 2013, pp. 2270–2283.

[101] Y. CUI, W. C., T. NOLL, and D. STRICKER. "KinectAvatar: Fully Automatic Body Capture Using a Single Kinect". In: *Asian Conference on Computer Vision Workshops*. Vol. 7729. 2013, pp. 133–147.

[102] L. CHEN, H. WEI, and J. FERRYMAN. "A survey of human motion analysis using depth imagery". In: *Pattern Recognition Letters*. Vol. 34. 15. 2013, pp. 1995–2006.

[103] M. BAUTISTA, A. HERN'ANDEZ-VELA, V. PONCE, X. PEREZ-SALA, X. BAR'O, O. PUJOL, C. ANGULO, and S. ESCALERA. "Probability-Based Dynamic Time Warping for Gesture Recognition on RGB-D Data". In: *Advances in Depth Image Analysis and Applications*. Vol. 7854. 2013, pp. 126–135.

[104] S. AZARY and A. SAVAKIS. "Grassmannian Sparse Representations and Motion Depth Surfaces for 3D Action Recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops*. Portland OR, 2013, pp. 492–499.

[105] ASUS XTION PRO LIVE. http://www.asus.com/Multimedia/Xtion_PRO/. 2013.

[106] B. SOLMAZ, B. E. MOORE, and M. SHAH. "Identifying Behaviors in Crowd Scenes Using Stability Analysis for Dynamical Systems". In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 34 (10), Oct. 2012, pp. 2064–2070.

[107] Y. TIAN, L. CAO, Z. LIU, and Z. ZHANG. "Hierarchical Filtered Motion for Action Recognition in Crowded Videos". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 42 (3), May 2012, pp. 313–323. ISSN: 1094-6977. DOI: 10.1109/TSMCC.2011.2149519.

[108] W. GE, R. T. COLLINS, and R. B. RUBACK. "Vision-Based Analysis of Small Groups in Pedestrian Crowds". In: *IEEE Trans. on Pattern and Analysis Machine Intelligence* 34 (5), May 2012, pp. 1003–1016.

[109] X. YANG, C. ZHANG, and Y. TIAN. "Recognizing actions using depth motion maps-based histograms of oriented gradients". In: *international conference on ACM Multimedia*. Nara, Japan, 2012, pp. 1057–1060.

[110] X. YANG, C. ZHANG, and Y. TIAN. "Recognizing actions using depth motion maps-based histograms of oriented gradients". In: *Proc. ACM Int. Conf. on Multimedia*. Nara, Japan, 2012, pp. 1057–1060.

[111] X. YANG and Y. TIAN. "EigenJoints based action recognition using Naive Bayes Nearest Neighbor". In: *Computer Vision and Pattern Recognition Workshops*. 2012, pp. 14–19.

[112] X. YANG and Y. TIAN. "EigenJoints-based Action Recognition Using Naive-Bayes-Nearest-Neighbor". In: *Proc. Work. on Human Activity Understanding from 3D Data*. Providence, Rhode Island, 2012, pp. 14–19.

[113] L. XIA, C. CHEN, and J. K. AGGARWAL. "View Invariant Human Action Recognition Using Histograms of 3D Joints". In: *Proc. Work. on Human Activity Understanding from 3D Data*. Providence, Rhode Island, USA, 2012, pp. 20–27.

[114] 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*. 2012, pp. 20–27.

[115] J. WANG, Z. LIU, Y. WU, and J. YUAN. "Mining actionlet ensemble for action recognition with depth cameras". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pp. 1290–1297.

[116] 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*. Vol. 2. 2012, pp. 498–502.

[117] 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*. 2012, pp. 1290–1297.

[118] J. WANG, Z. LIU, Y. WU, and J. YUAN. "Mining Actionlet Ensemble for Action Recognition with Depth Cameras". In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. Providence, Rhode Island, USA, 2012, pp. 1–8.

[119] 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*. 2012, pp. 872–885.

[120] J. WANG, Z. LIU, J. CHOROWSKI, Z. CHEN, and Y. WU. "Robust 3D Action Recognition with Random Occupancy Patterns". In: *Proc. Europ. Conf. on Computer Vision*. Florence, Italy, 2012, pp. 1–8.

[121] A. W. VIEIRA, E. R. NASCIMENTO, G. L. OLIVEIRA, Z. LIU, and M. F. M. CAMPOS. "STOP: Space-Time Occupancy Patterns for 3D Action Recognition from Depth Map Sequences". In: *Iberoamerican Congress on Pattern Recognition*. Buenos Airies, Argentina, 2012, pp. 252–259.

[122] A. W. VIEIRA, E. R. N., G. L. O., Z. L., and M. F. 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*. Vol. 7441. 2012, pp. 252–259.

[123] T. TUNG and T. MATSUYAMA. "Topology Dictionary for 3D Video Understanding". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 34. 8. 2012, pp. 1645–1657.

[124] 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*. Vol. 18. 4. 2012, pp. 643–650.

[125] Y.-S. TAK, J. KIM, and E. HWANG. "Hierarchical querying scheme of human motions for smart home environment". In: *Engineering Applications of Artificial Intelligence*. Vol. 25. 7. 2012, pp. 1301–1312.

[126] S. SHIRAZI, M. HARANDI, C. SANDERSON, A. A., and B. LOVELL. "Clustering on Grassmann Manifolds Via Kernel Embedding with Application to Action Analysis". In: *International Conference on Image Processing*. 2012, pp. 781–784.

[127] S. O'HARA, Y. M. LUI, and B. A. DRAPER. "Using a Product Manifold distance for unsupervised action recognition". In: *Image and Vision Computing*. Vol. 30. 3. 2012, pp. 206–216.

[128] T. MATSUYAMA, S. NOBUHARA, T. TAKAI, and T. TUNG. "Multi-camera Systems for 3D Video Production". In: *3D Video and Its Applications*. 2012, pp. 17–44.

[129] Y. M. LUI. "Advances in matrix manifolds for computer vision". In: *Image and Vision Computing*. Vol. 30. 6–7. 2012, pp. 380–388.

[130] Y. M. LUI. "Tangent Bundles on Special Manifolds for Action Recognition". In: *IEEE Transactions on Circuits and Systems for Video Technology* 22, 2012, pp. 930–942.

[131] 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*. Vol. 107. 2012, pp. 1152–1165.

[132] A. KURAKIN, Z. ZHANG, and Z. LIU. "A real time system for dynamic hand gesture recognition with a depth sensor". In: *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*. IEEE. 2012, pp. 1975–1979.

[133] 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)*. 2012, pp. 1975–1979.

[134] B. JIAWEI, G. JEFF, and C. BRIAN. "Movelets: A dictionary of movement". In: *Electronic Journal of Statistics* 6, 2012, pp. 559–578.

[135] A. GRAVES et al. *Supervised sequence labelling with recurrent neural networks*. Vol. 385. Springer, 2012.

[136] R. EL KHOURY, J.-P. VANDEBORRE, and M. DAOUDI. "Indexed heat curves for 3D-model retrieval". In: *International Conference on Pattern Recognition*. 2012, pp. 1964–1967.

[137] M. BREGONZIO, T. XIANG, and S. GONG. "Fusing appearance and distribution information of interest points for action recognition". In: *Pattern Recognition*. Vol. 45. 3. 2012, pp. 1220–1234.

[138] 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*. 2012, pp. 769–772.

[139] N. BUCH, S. A. VELASTIN, and J. ORWELL. "A Review of Computer Vision Techniques for the Analysis of Urban Traffic". In: *IEEE Trans. on Intelligent Transportation Systems* 12 (3), Sept. 2011, pp. 920–939.

[140] H. AVILÉS-ARRIAGA, L. SUCAR-SUCCAR, C. MENDOZA-DURÁN, and L. PINEDA-CORTÉS. "A Comparison of Dynamic Naive Bayesian Classifiers and Hidden Markov Models for Gesture Recognition". In: *Journal of Applied Research and Technology* 9 (1), Apr. 2011, pp. 81–102.

[141] L. WANG, L. CHENG, and L. WANG. "Elastic Sequence Correlation for Human Action Analysis". In: *IEEE Transactions on Image Processing,* vol. 20. 6. 2011, pp. 1725–1738.

[142] 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*. Vol. 33. 2011, pp. 2273–2286.

[143] 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*. Vol. 33. 4. 2011, pp. 852–858.

[144] A. SRIVASTAVA, E. KLASSEN, S. JOSHI, and I. JERMYN. "Shape Analysis of Elastic Curves in Euclidean Spaces". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 33. 2011, pp. 1415–1428.

[145] A. SRIVASTAVA, E. KLASSEN, S. H. JOSHI, and I. JERMYN. "Shape Analysis of Elastic Curves in Euclidean Spaces". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33 (7), 2011, pp. 1415–1428.

[146] 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: *IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. Colorado Springs, Colorado, USA, 2011, pp. 1–8.

[147] S. SEMPENA, N. MAULIDEVI, and P. ARYAN. "Human action recognition using Dynamic Time Warping". In: *International Conference on Electrical Engineering and Informatics*. 2011, pp. 1–5.

[148] 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*. 2011, pp. 1182–1188.

[149] Z. REN, J. YUAN, and Z. ZHANG. "Robust Hand Gesture Recognition Based on Finger-Earth Mover's Distance with a Commodity Depth Camera". In: *Proc. ACM Int. Conf. on Multimedia*. Scottsdale, Arizona, USA, 2011, pp. 1093–1096.

[150] N. PUGEAULT and R. BOWDEN. "Spelling it out: Real-time asl fingerspelling recognition". In: *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE. 2011, pp. 1114–1119.

[151] B. NI, G. WANG, and P. MOULIN. "RGBD-HuDaAct: A color-depth video database for human daily activity recognition". In: *International Conference on Computer Vision Workshops*. 2011, pp. 1147–1153.

[152] J. NAGI, F. DUCATELLE, G. A. DI CARO, D. CIREŞAN, U. MEIER, A. GIUSTI, F. NAGI, J. SCHMIDHUBER, and L. M. GAMBARDELLA. "Max-pooling convolutional neural networks for vision-based hand gesture recognition". In: *Signal and Image Processing Applications (ICSIPA), 2011 IEEE International Conference on*. IEEE. 2011, pp. 342–347.

[153] Y. M. LUI and J. R. BEVERIDGE. "Tangent bundle for human action recognition". In: *IEEE Int. Conf. Automat. Face Gesture Recog. FG*. 2011, pp. 97–102.

[154] S. HADFIELD and R. BOWDEN. "Kinecting the dots: Particle Based Scene Flow From Depth Sensors". In: *Proc. Int. Conf. on Computer Vision*. Barcelona, Spain, 2011, pp. 2290–2295.

[155] D. GONG and G. MEDIONI. "Dynamic Manifold Warping for view invariant action recognition". In: *IEEE International Conference on Computer Vision*. Barcelona, Spain, 2011, pp. 571–578.

[156] 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*. 2011, pp. 1602–1609.

[157] C.-C. CHANG and C.-J. LIN. "LIBSVM: A library for support vector machines". In: *ACM Transactions on Intelligent Systems and Technology*. Vol. 2. 2011, pp. 1–27.

[158] Q. D. 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*. Vol. 17. 2011, pp. 1082–1095.

[159] O. ARANDJELOVIĆ. "Contextually learnt detection of unusual motion-based behaviour in Â´ crowded public spaces". In: *Int. Symp. on Computer and Information Sciences*. 2011, pp. 403–410.

[160] M. 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*. Vol. 115. 3. New York, NY, USA, 2011, pp. 439–455.

[161] A. A. SALAH, T. GEVERS, N. SEBE, and A. VINCIARELLI. "Challenges of Human Behavior Understanding". In: *International Workshop on Human Behavior Understanding*. Vol. 6219. 2010, pp. 1–12.

[162] M. OVSJANIKOV, Q. MÉRIGOT, F. MÉMOLI, and L. J. GUIBAS. "One Point Isometric Matching with the Heat Kernel". In: *Computer Graphics Forum*. Vol. 29. 5. 2010, pp. 1555–1564.

[163] W. LI, Z. ZHANG, and Z. LIU. "Action recognition based on a bag of 3d points". In: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*. IEEE. 2010, pp. 9–14.

[164] 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*. 2010, pp. 9–14.

[165] W. LI, Z. ZHANG, and Z. LIU. "Action recognition based on a bag of 3D points". In: *Proc. Work. on Human Communicative Behavior Analysis*. San Francisco, California, USA, 2010, pp. 9–14.

[166] 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.

[167] P. HUANG, A. HILTON, and J. STARCK. "Shape Similarity for 3D Video Sequences of People". In: *International Journal of Computer Vision*. Vol. 89. 2010, pp. 362–381.

[168] 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*. 2009, pp. 2435–2441.

[169] 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*. 2009, pp. 1709–1716.

[170] 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*. 2009, pp. 1089–1096.

[171] 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*. Berlin, Germany, 2009, pp. 1383–1392.

[172] Y. LIPMAN and T. FUNKHOUSER. "Mobius Voting for Surface Correspondence". In: *ACM Transactions on Graphics*. Vol. 28. 2009, pp. 1–12.

[173] P. HUANG and A. HILTON. "Shape-Colour Histograms for matching 3D video sequences". In: *IEEE International Conference on Computer Vision Workshops*. 2009, pp. 1510–1517.

[174] 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*. Vol. 2. 28. 2009, pp. 337–346.

[175] 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*. 2009, pp. 159–168.

[176] T. GIORGINO. "Computing and Visualizing Dynamic Time Warping Alignments in R: The DTW Package". In: *Journal of Statistical Softwar*. Vol. 31. 7. 2009, 1–24.

[177] T. FEIX, R. PAWLIK, H.-B. SCHMIEDMAYER, J. ROMERO, and D. KRAGIC. "A comprehensive grasp taxonomy". In: *Robotics, science and systems: workshop on understanding the human hand for advancing robotic manipulation*. Vol. 2. 2.3. 2009, pp. 2–3.

[178] J. DENG, W. DONG, R. SOCHER, L.-J. LI, K. LI, and L. FEI-FEI. "Imagenet: A large-scale hierarchical image database". In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE. 2009, pp. 248–255.

[179] M. MARTINEZ and L. E. SUCAR. "Learning Dynamic Naive Bayesian Classifiers". In: *Proceedings of the Twenty-First International FLAIRS Conference*. Coconut Grove, USA, May 2008.

[180] D. VLASIC, I. BARAN, W. MATUSIK, and J. POPOVIĆ. "Articulated mesh animation from multi-view silhouettes". In: *ACM Siggraph*. Los Angeles, California, 2008, pp. 1–97. ISBN: 978-1-4503-0112-1.

[181] R. KLETTE and G. TEE. "Understanding Human Motion: A Historic Review". In: *Human Motion: Understanding, Modelling, Capture, and Animation*. Springer Netherlands, 2008, pp. 1–22. ISBN: 978-1-4020-6693-1.

[182] A. KLASER, M. MARSZALEK, and C. SCHMID. "A Spatio-Temporal Descriptor Based on 3D-Gradients". In: *British Machine Vision Conference*. 2008, pp. 1–10.

[183] 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*. Vol. 27. 3. 2008, pp. 1–10.

[184] A. ZAHARESCU, E. BOYER, and R. HORAUD. "TransforMesh A Topology-Adaptive Mesh-Based Approach to Surface Evolution". In: *Asian Conference on Computer Vision*. Vol. 4844. 2007, pp. 166–175.

[185] T. YAMASAKI and K. AIZAWA. "Motion segmentation and retrieval for 3D video based on modified shape distribution". In: *Journal on Applied Signal Processing EURASIP*. Vol. 2007. 1. 2007, pp. 211–211.

[186] J. STARCK and A. HILTON. "Surface Capture for Performance-Based Animation". In: *Computer Graphics and Applications*. Vol. 27. 3. 2007, pp. 21–31.

[187] S. MAHMOUDIA and M. DAOUDIB. "A probabilistic approach for 3D shape retrieval by characteristic views". In: *Pattern Recognition Letters*. Vol. 28. 13. 2007, pp. 1705–1718.

[188] C.-S. LEE and A. M. ELGAMMAL. "Modeling View and Posture Manifolds for Tracking". In: *IEEE International Conference on Computer Vision*. 2007, pp. 1–8.

[189] S. 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*. 2007, pp. 1–7.

[190] S. H. JOSHI, E. KLASSEN, A. SRIVASTAVA, and I. JERMYN. "A Novel Representation for Riemannian Analysis of Elastic Curves in R^n". In: *Proc IEEE Int. Conf. on Computer Vision and Pattern Recognition*. Minneapolis, MN, USA, 2007, pp. 1–7.

[191] 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)*. Los Alamitos, CA, USA, 2006, pp. 105–112.

[192] S. M. SEITZ, B. CURLESS, J. DIEBEL, D. SCHARSTEIN, and R. SZELISKI. "A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms". In: *IEEE International Conference on Computer Vision and Pattern Recognition*. Vol. 1. Washington, DC, USA, 2006, pp. 519–528.

[193] P. W. MICHOR and D. MUMFORD. "Riemannian geometries on spaces of plane curves". In: *Journal of the European Mathematical Society*. Vol. 8. 2006, pp. 1–48.

[194] A. GRAVES, S. FERNÁNDEZ, F. GOMEZ, and J. SCHMIDHUBER. "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks". In: *Proceedings of the 23rd international conference on Machine learning*. ACM. 2006, pp. 369–376.

[195] A.SRIVASTAVA, S. JOSHI, W. MIO, and X. LIU. "Statistical Shape Analysis: Clustering, Learning, and Testing". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (4), Apr. 2005, pp. 590–602.

[196] A. YEZZI and A. MENNUCCI. "Conformal Metrics and True Gradient Flows for Curves". In: *IEEE International Conference on Computer Vision*. 2005, pp. 913–919.

[197] T. TUNG and F. SCHMITT. "The Augmented Multiresolution Reeb Graph Approach for Content-based Retrieval of 3D Shapes". In: *International Journal of Shape Modeling*. Vol. 11. 2005, pp. 91–120.

[198] S. KATZ, G. LEIFMAN, and A. TAL. "Mesh segmentation using feature point and core extraction". In: *The Visual Computer*. Vol. 21. 2005, pp. 649–658.

[199] K. CHEUNG, S. BAKER, and T. KANADE. "Shape-From-Silhouette Across Time Part I: Theory and Algorithms". In: *International Journal of Computer Vision*. Vol. 62. 3. 2005, pp. 221–247.

[200] 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*. Vol. 26. 2004, pp. 372–383.

[201] J. BARBIC, A. SAFONOVA, J. PAN, C. FALOUTSOS, J. K. HODGINS, and N. S. POLLARD. "Segmenting Motion Capture Data into Distinct Behaviors". In: *Proc. Graphics Interface*. 2004.

[202] 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.

[203] M. KAZHDAN, T. FUNKHOUSER, and S. RUSINKIEWICZ. "Rotation invariant spherical harmonic representation of 3D shape descriptors". In: *ACM SIGGRAPH Symposium on Geometry Processing*. 2003, pp. 156–164.

[204] J. 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*. 2003, pp. 49–56.

[205] K. GALLIVAN, A. SRIVASTAVA, L. XIUWEN, and P. VAN DOOREN. "Efficient algorithms for inferences on Grassmann manifolds". In: *IEEE Workshop on Statistical Signal Processing*. 2003, pp. 315–318.

[206] R. OSADA, T. FUNKHOUSER, B. CHAZELLE, and D. DOBKIN. "Shape Distributions". In: *ACM Transactions on Graphics*. Vol. 21. 2002, pp. 807–832.

[207] M. MORTARA and G. PATANE. "Affine-Invariant Skeleton of 3D Shapes". In: *Proceedings of the Shape Modeling International*. Washington, DC, USA, 2002, pp. 245–252.

[208] J. LEE, J. CHAI, P. S. A. REITSMA, J. K. HODGINS, and N. S. POLLARD. "Interactive control of avatars animated with human motion data". In: *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. San Antonio, Texas, 2002, pp. 491–500.

[209] N. D'APUZZO. "Modeling human faces with multi-image photogrammetry". In: *Proceedings of SPIE*. Vol. 4661. 2002, pp. 191–197.

[210] Y. WU, J. Y. LIN, and T. S. HUANG. "Capturing natural hand articulation". In: *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*. Vol. 2. IEEE. 2001, pp. 426–432.

[211] F. LAZARUS and A. VERROUST. "Level set diagrams of polyhedral objects". In: *ACM symposium on Solid modeling and applications*. Ann Arbor, Michigan, USA, 1999, pp. 130–140.

[212] A. JOHNSON and M. HEBERT. "Using spin images for efficient object recognition in cluttered 3D scenes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 21. 1999, pp. 433–449.

[213] M. ANKERST, G. KASTENMULLER, 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*. 1999, pp. 207–226.

[214] Y. LECUN, L. BOTTOU, Y. BENGIO, and P. HAFFNER. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86 (11), 1998, pp. 2278–2324.

[215] M. SCHUSTER and K. K. PALIWAL. "Bidirectional recurrent neural networks". In: *IEEE Transactions on Signal Processing* 45 (11), 1997, pp. 2673–2681.

[216] A. P. BRADLEY. "The use of the area under the ROC curve in the evaluation of machine learning algorithms". In: *Pattern recognition* 30 (7), 1997, pp. 1145–1159.

[217] J. Y. ZHENG. "Acquiring 3-D Models from Sequences of Contours". In: *IEEE transactions on Pattern Analysis and Machine Intelligence*. Vol. 16. 2. 1994, pp. 163–178.

[218] P. J. WERBOS. "Backpropagation through time: what it does and how to do it". In: *Proceedings of the IEEE* 78 (10), 1990, pp. 1550–1560.

[219] 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: Springer, 1983.

[220] L. V. D. MARR. "Representation and recognition of the movements of shapes". In: *Proe. R. Soc. Lond. B*. 1982, pp. 501–524.

[221] H. KARCHER. "Riemannian center of mass and mollifier smoothing". In: *Comm. on Pure and Applied Math.* 30, 1977, pp. 509–541.

[222] W. BOOTHBY. "An Introduction to Differentiable Manifolds and Riemannian Geometry". In: *Academic Press*. 1975.

[223] L. BAUM, T. PETRIE, G. SOULES, and N. WEISS. "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains". In: *The Annals of Mathematical Statistics* 41 (1), 1970, pp. 164–171.

[224] "Vitus: http://www.vitus.de/english/ [October 2002]". In:

[225] "http://4drepository.inrialpes.fr, September 2010." In:

[226] "Cyberware: http://www.cyberware.com [October 2002]". In:

[227] "3D Studio Max, http://www.3dmax.com/ [October 2002]". In:

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

Towards Understanding Human Behavior by Time-Series Analysis of 3D Motion

Hazem WANNOUS

Abstract Human motion analysis has been an active topic from the early beginning of computer vision due to its relevance to a large variety of domains. It is becoming a central key for different types of application including gaming, monitoring, sign language recognition and medical applications. These applications extend from simple gesture detection to complex behavior understanding, and depend on body parts involved and duration of movement. This topic has evolved substantially in parallel with major technological advancements, especially capturing technologies and machine learning techniques. The main concern of this dissertation is the issue of human behavior understanding trough vision-based analysis of the motion limited to body behavior, which can be conceptually categorized into different types of motion modalities: gestures, actions, activities and grained-fine hand gestures. Our aims were to develop new theoretical and application approaches advancing the motion representation and the recognition of human behavoir involving different body part and based on various sources of information, such as, 3D mesh, depth and skeleton data. Since movements unfold in both space and time, it is mandatory to provide solutions that describe its spatial and temporal properties and examine how variations in both spaces influence the recognition of the meaning of the motion. For this purpose, we proposed a number of motion representation and recognition frameworks, developed new theoretical and application approaches, and demonstrated their efficiency on several tasks of motion recognition, including gestures, actions and activities.

Vers une compréhension du comportement humain par l'analyse en série temporelle de mouvements 3D

Résumé L'analyse du mouvement humain est un sujet actif dans la communauté de la vision par ordinateur en raison de sa pertinence pour une grande variété de domaines. Il devient un élément clé pour différents types d'applications, notamment les jeux, la surveillance, la reconnaissance du langage des signes et les applications médicales. Ces applications vont de la simple détection de gestes à la compréhension de comportements complexes et dépendant des parties du corps impliquées ainsi que de la durée du mouvement. Ce sujet a considérablement évolué parallèlement aux avancées technologiques majeures, notamment dans la technologie de capture et les techniques dâapprentissage automatique. Cette Habilitation a pour thème principal la compréhension du comportement humain par l'analyse du mouvement limitée au comportement corporel, qui peut Ã[a]tre catégorisée conceptuellement en différents modalités de mouvement: gestes, actions, activités et gestes fins de la main. L'objectif est de développer de nouvelles approches théoriques et applicatives faisant progresser la représentation du mouvement et la reconnaissance du comportement humain impliquant différentes parties du corps, en se basant sur diverses sources d'informations, telles que des maillages 3D, des images de profondeur et des squelettes 3D. Étant donné que les mouvements se déroulent à la fois dans lâespace et dans le temps, il est impératif de proposer des solutions décrivant ces propriétés spatiales et temporelles et dâexaminer comment les variations dans les deux espaces influencent la reconnaissance du mouvement. À cette fin, nous avons proposé un certain nombre de méthodes de représentation et de reconnaissance de mouvement, développé de nouvelles approches théoriques et applicatives et démontré leur efficacité dans plusieurs tâches de reconnaissance de mouvement, notamment les gestes, les actions et les activités.