Année 2019

Université Lille

THÈSE

pour obtenir le grade de
Docteur,
spécialité Informatique et Applications

présentée et soutenue publiquement par

Amor Ben Tanfous

le 3 Décembre 2019

Représentations parcimonieuses dans les variétés de formes pour la classification et la génération de trajectoires humaines

Sparse Representations in the Shape Manifold for Human Trajectories Classification and Generation

COMPOSITION DU JURY

Mme Alice CAPLIER	Rapporteur	Pr., Grenoble INP, Univ. Grenoble Alpes, France
M. Sylvain CALINON	Rapporteur	Dr., Idiap Research Institute, Switzerland
M. Boulbaba BEN AMOR	Directeur de la thèse	Pr., IMT Lille Douai, France
M. Hassen DRIRA	Encadrant de la thèse	Dr., IMT Lille Douai, France
M. Josef KITTLER	Examinateur	Pr., University of Surrey, England
Mme Bernadette DORIZZI	Président	Pr., Télécom SudParis, France

Abstract

Designing intelligent systems to understand video content has been a hot research topic in the past few decades since it helps compensate the limited human capabilities of analyzing videos in an efficient way. In particular, human behavior understanding in videos is receiving a huge interest due to its many potential applications. At the same time, the detection and tracking of human landmarks in video streams has gained in reliability partly due to the availability of affordable RGB-D sensors. This infer time-varying geometric data which play an important role in the automatic human motion analysis. However, such analysis remains challenging due to enormous view variations, inaccurate detection of landmarks, large intra- and inter- class variations, and insufficiency of annotated data. In this thesis, we propose novel frameworks to classify and generate 2D/3D sequences of human landmarks. We first represent them as trajectories in the shape manifold which allows for a view-invariant analysis. However, this manifold is nonlinear and thereby standard computational tools and machine learning techniques could not be applied in a straightforward manner. As a solution, we exploit notions of Riemannian geometry to encode these trajectories based on sparse coding and dictionary learning. This not only overcomes the problem of nonlinearity of the manifold but also yields sparse representations that lie in vector space, that are more discriminative and less noisy than the original data. We study intrinsic and extrinsic paradigms of sparse coding and dictionary learning in the shape manifold and provide a comprehensive evaluation on their use according to the nature of the data (*i.e.* face or body in 2D or 3D). Based on these sparse representations, we present two frameworks for 3D human action recognition and 2D micro- and macro- facial expression recognition and show that they achieve competitive performance in comparison to the state-of-the-art. Finally, we design a generative model allowing to synthesize human actions. The main idea is to train a generative adversarial network to generate new sparse representations that are then transformed to pose sequences. This framework is applied to the task of data augmentation

allowing to improve the classification performance. In addition, the generated pose sequences are used to guide a second framework to generate human videos by means of pose transfer of each pose to a texture image. We show that the obtained videos are realistic and have better appearance and motion consistency than a recent state-of-the-art baseline.

Résumé

Concevoir des systèmes intelligents pour comprendre le contenu des vidéos est devenu un axe de recherche très important car il permet de compenser les capacités humaines limitées de l'analyse efficace des vidéos. En particulier, la compréhension du comportement humain à partir de vidéos suscite un intérêt considérable en raison de ses nombreuses applications potentielles. Au même temps, la détection et le suivi de marqueurs humains dans les flux vidéo sont devenus de plus en plus fiables, et c'est en partie grâce à la disponibilité de capteurs abordables. Cela permet de déduire des données géométriques qui varient dans le temps et qui jouent un rôle important dans l'analyse automatique du mouvement humain. Cependant, une telle analyse reste difficile en raison des énormes variations de vue, de la détection inexacte des marqueurs, des variations importantes des classes ainsi que de l'insuffisance des données annotées. Dans cette thèse, nous proposons de nouvelles méthodes permettant de classifier et de générer des séquences 2D/3D de marqueurs humains. Tout d'abord, nous représentons ces séquences comme étant des trajectoires dans des variétés de formes permettant ainsi une analyse invariante à la vue. Toutefois, ces variétés sont non linéaires et, par conséquent, les algorithmes classiques telles que les techniques d'apprentissage automatique standards ne pourraient pas être appliqués d'une manière directe vu qu'ils sont conçus pour des données de nature linéaire. En guise de solution, nous exploitons des notions de la géométrie Riemannienne pour coder ces trajectoires en appliquant une technique de codage parcimonieux et d'apprentissage de dictionnaires. Cela permet non seulement de résoudre le problème de non-linéarité des variétés de formes mais aussi de représenter les trajectoires comme étant des représentations parcimonieuses qui sont définies dans des espaces vectoriels, qui sont plus discriminantes et moins bruitées que les données originales. Nous étudions des paradigmes intrinsèques et extrinsèques de codage parcimonieux et d'apprentissage de dictionnaire dans les variétés de formes et nous présentons une étude comparative de leur utilisation en fonction de la nature des données

(*i.e.* visage ou corps en 2D ou 3D). D'autre part, en se basant sur ces représentations parcimonieuses, nous proposons deux approches de reconnaissance d'actions humaines en 3D et de reconnaissance d'expressions faciales en 2D, et nous montrons que les résultats obtenus sont compétitifs avec les méthodes récentes de l'état de l'art. Enfin, nous présentons un modèle génératif permettant de synthétiser des actions humaines dont l'idée principale est de concevoir un réseau antagoniste génératif afin de générer de nouvelles représentations parcimonieuses qui sont ensuite transformées en des séquences de poses. Nous appliquons cette méthode pour l'augmentation des données ce qui permet d'améliorer les performances de la classification d'actions. De plus, les séquences de pose générées sont utilisées pour guider un second modèle générateur dans le but de générer des vidéos humaines par transfert de chaque pose d'une séquence en une image texture. Nous montrons que les vidéos obtenues sont réalistes et présentent une meilleure cohérence en termes d'apparence et de mouvement qu'une méthode récente de l'état de l'art.

Dedicated to my beloved parents

Table of contents

List of figures

List of tables

Publications

[P1] **A. Ben Tanfous**, H. Drira, B. Ben Amor, Coding Shape Trajectories for Facial Expression and Action Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), special issue on RGB-D Vision: Methods and Applications. Accepted in July 2019. [103]

[P2] **A. Ben Tanfous**, H. Drira, B. Ben Amor, Coding Kendall's Shape Trajectories for 3D Action Recognition, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2840–2849, 2018. [11]

[P3] **A. Ben Tanfous**, H. Drira, B. Ben Amor, Reconnaissance d'actions 3D par codage parcimonieux sur l'espace de Kendall, Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP), 2018. [10]

Chapitre 1

Introduction

1.1 Motivation

In recent years, video cameras have become part of our everyday life with the availability of mobile phones, tablets, and digital photo cameras which all can create, store and share high quality videos. With great storage capacities and fast internet access, videos have become easily accessible. Accordingly, designing intelligent systems to understand their content is attracting more and more attention with the availability of powerful and affordable computers. Such intelligent systems are also needed to compensate the limited human capabilities of analyzing videos in an efficient way. In particular, human behavior understanding in videos is receiving a huge interest due its many potential applications. For instance, analyzing the motion of the human body and face plays a key role in several areas such as: Human-Computer Interaction and Entertainment, Visual Surveillance, Video Retrieval, and Healthcare.

— **Human-Computer Interaction and Entertainment:** In computer graphics, video cameras are used for human-machine interaction since they represent a natural and intuitive way to communicate with a device. Therefore, the accurate recognition of

human gestures and facial expressions can guarantee a more natural, less-restrictive and effective human-computer interfaces. In the entertainment industry, a new generation of games based on full body play such as dance and sports games have increased the appeal of gaming to family members of all ages [65]. An example of popular video camera that enables accurate perception of human actions is the Microsoft Kinect sensor (see Figure 1.4).

— **Visual Surveillance** Video surveillance cameras can be found everywhere (see Figure 1.1). Visual surveillance in dynamic scenes attempts to detect, recognize and track certain objects from image sequences, and more generally to understand and describe object behaviors including humans. The aim is to develop intelligent visual surveillance to replace the traditional passive video surveillance that is proving ineffective as the number of cameras exceeds the capability of human operators to monitor them [49]. Therefore, human action and identity recognition are at the core of video surveillance and monitoring systems.

FIGURE 1.1 – Examples of video surveillance cameras and a sample monitoring center.

— **Video Retrieval** Managing and retrieving videos according to video content is becoming challenging due to the huge amount of videos uploaded on the internet with every second. Most of the search engines use text data to manage video data which are usually inaccurate or even incorrect [85]. A potential solution for efficient video retrieval could be human activity and action recognition systems.

— **Healthcare** In healthcare, assessing humans physical performance and behavior is today frequently used for the monitoring of elderly people, rehabilitation, and medical

examination. In fact, detecting behavior changes in human everyday life allows medical scientists to suggest strategies related to diet, fall prevention, exercise and medication adherence. On the other hand, automatic facial expression analysis can assist clinicians in the estimation of pain intensity from faces which is useful for monitoring patients in intensive care units and also the assessment of chronic lower back pain. Furthermore, facial expression analysis can help psychologists to measure the level of depression severity from faces since reduced facial expressiveness is considered as a behavioral indicator of depression according to the Diagnostic and Statistical Manual of Mental Disorders [6].

Choice of data modality In this dissertation, we are interested in the classification and generation of visual data that describe human motion, *i.e.* actions and facial expressions, and we mainly focus our analysis on landmark data that are extracted from the body or face, *i.e.* 2D/3D human skeletons and 2D facial landmark configurations, respectively. Our motivation behind this choice arises from the ability of these data to summarize motion patterns since humans can recognize many actions and facial expressions directly from skeletal and facial landmark sequences, respectively. In fact, in the psycho-physical research work of Johansson [56] on visual perception of motion patterns characteristics of living organisms in locomotion, it was shown that humans can recognize actions from the motion of a few moving light displays attached to the human body, describing the motions of the main body joints. He has found that between 10 and 12 moving light displays in adequate motion combinations in proximal stimulus evoke a compelling impression of human walking, running, dancing, etc. (see Figure 1.2).

On the other hand, using landmark data is expected to reduce the computational complexity of the designed algorithms as instead of treating high-dimensional RGB images that may contain irrelevant information for the desired tasks, landmark data summarize

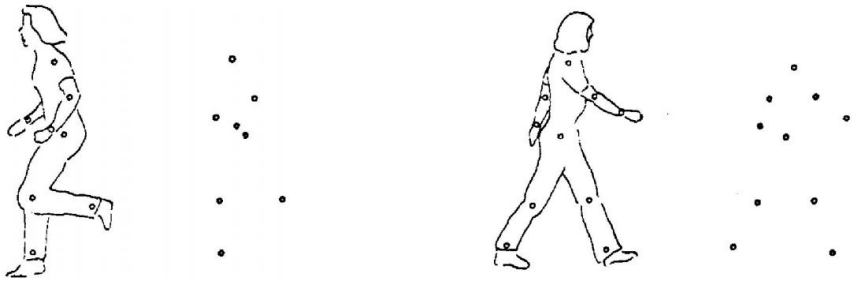
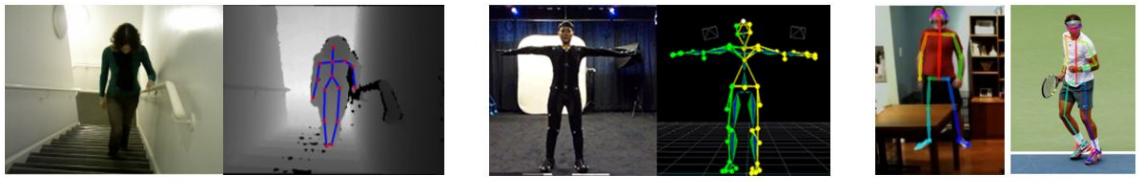


FIGURE 1.2 – Outline contours of a walking (right) and a running (left) subject and the corresponding moving light displays attached on the body [56].

the image content in a few 2D/3D landmark points that only describe human-related information. As a consequence, landmark-based solutions are expected to be relevant and less computationally expensive than other modalities making them more suitable for real-time applications. Furthermore, skeletal data estimation solutions and facial landmark detectors has seen many advancements in recent years making the obtained landmarks often reliable and accurate. On one hand, a human skeleton which is composed of a set of landmarks (or joints) that are located on different articulations of the body can be obtained with different approaches as illustrated in Figure 1.3. For instance, 3D skeletons can be tracked



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

FIGURE 1.3 – Skeleton estimation from: (a) depth images [84], (b) Motion Capture systems, (c) RGB images [17].

in a depth video by extracting the 3D locations of body joints in each frame. This can be achieved by using cost-effective depth cameras such as the Microsoft Kinect or ASUS Xtion PRO LIVE sensor as popular examples (see Figure 1.4). However, these sensors can produce inaccurate results and can also fail to estimate the body joints in the presence of serious

FIGURE 1.4 – Depth cameras: ASUS Xtion PRO LIVE (left) and the Microsoft Kinect (right).

occlusion. More accurate estimation can be yielded with Motion Capture systems based on body markers for an automatic tracking of 3D skeletons, but this solution is more expensive in cost and time than depth cameras. Recently, pose estimation approaches [3, 17, 13] have reached impressive performances inferring 2D/3D skeletons that are in many cases robust to occlusion and body profile. On the other hand, facial landmarks are located in different positions of the face such as the mouth, nose, eyes, eyebrows and also the chin in some methods. The latter can be detected from texture images in 2D [7] or 3D [8]. Figure 1.5 shows examples of 2D facial landmark tracking from RGB images where we can notice the ability to deal with profile faces and occlusion in some cases.

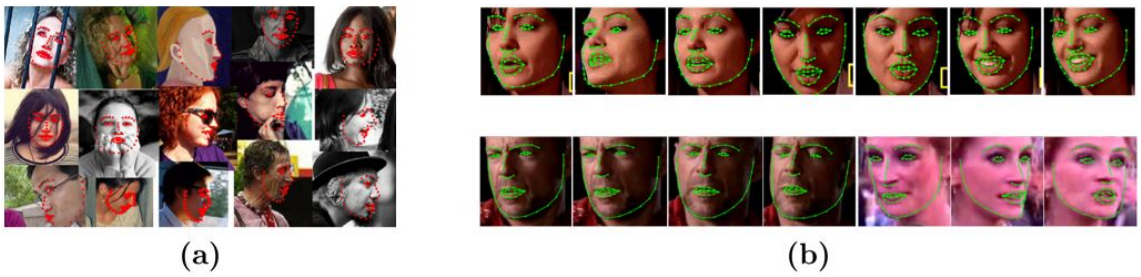


FIGURE 1.5 – Examples of facial landmark detection in: (a) 3D [8] and (b) 2D [7].

1.2 Scientific challenges

Human action and facial expression analysis (*i.e.* recognition and generation) are important but challenging research topics due to several major issues [65, 89].

Intra- and inter-class variations It is well known that people perform the same action in various styles and express their emotions differently through facial expressions. For a given action such as "*hand wave*", one can use his left or right hand, can rise it slowly or quickly, or even turns while waving. On the other hand, a person can smile with different intensities, can simultaneously close his eyes or move his head. That is to say, one action or facial expression category may contain various styles of motion which results in large intra-class pose variations. Moreover, different categories can present strong similarities. As an example, actions "walking" and "running" involve similar motion patterns. These similarities would result in small inter-class variations which makes the analysis more challenging. As a consequence, these issues should be carefully addressed in the analysis in order to make human motion representations more discriminative.

View-variations Pose sequences belonging to the same categories can be captured from various viewpoints, see Figure 1.6. They can be taken in front of the subjects, on their side, or even on top of them. Similarly, facial expression analysis may encounter several head pose variations making it more complex. In this thesis work, our goal is to go beyond the non-trivial view-variation challenge.

(a) **(b)**

FIGURE 1.6 – (a) Samples taken from [94] showing body pose with a variety in camera views. (b) Facial images with different poses.

Annotated data insufficiency Action and facial expression recognition approaches, especially those that are based on deep neural networks, usually require large amount of

annotated training data. Even though some large scale datasets became recently available, *e.g.* the NTU-RGBD 120 [72] which contains 114,480 actions belonging to 120 classes, sample annotation requires a huge human effort. As a remedy, other datasets such as the Youtube-8M database [1] are annotated by retrieval methods which are not usually accurate, thus hindering the performance of the designed algorithms. In this thesis work, we address this problem by presenting a generative framework for data augmentation which alleviates the burden of data annotation. Given a small annotated dataset, we aim to generate new samples for each category, yielding new variabilities in terms of motion style and thereby improving the classification performance.

1.3 Main contributions

The goal of this dissertation is to analyze dynamic human landmarks of the body and face. We explore sparse representations to model human actions and facial expressions and present new frameworks to classify and generate them. To summarize, we provide the following main contributions:

Action and facial expression representations We introduce sparse representations for landmark-based human actions and facial expressions which are used throughout this thesis work. Tackling the problem of camera view-variations, we represent sequences of landmarks as trajectories in the shape manifold to allow for a view-invariant analysis. Since the resulting manifold is nonlinear, we exploit its well-known Riemannian geometry to study sparse coding and dictionary learning techniques. The application of these approaches to encode trajectories yields representations that lie to vector spaces which are more suitable for the analysis than nonlinear ones. In addition, the obtained time-series are sparse which represents a suitable computational property for several vision problems.

Action and facial expression recognition frameworks We apply the proposed sparse representations to classify 3D actions and 2D facial expressions. We propose two recognition approaches: 1) a deep learning framework in which we apply Long Short-Term Memory (LSTM) to model and classify our sparse representations; and 2) a classical pipeline of methods designed for Euclidean time-series. We perform extensive evaluation on seven commonly-used datasets that present different challenges such as the MSR-Action 3D [69] and the NTU-RGBD [94] databases for action recognition, and the CK+ [76] and the Oulu-CASIA [105] for facial expression recognition. In addition, we evaluate the proposed framework on the more challenging task of micro-expression recognition. We compare our methods with each other and with state-of-the-art approaches where we demonstrate their competitiveness. Moreover, we perform various baseline experiments and ablative studies to evaluate some properties of the presented techniques.

A comparative study of intrinsic and extrinsic coding techniques in the shape manifold In Chapter 2, we study two techniques of Riemannian sparse coding and dictionary learning in the shape manifold known as *intrinsic* and *extrinsic*. We use these approaches to model human actions and facial expressions. Based on the classification experiments (Chapter 3), we try to find an answer to the question "Depending on the nature of the data (*i.e* body or face) and its dimension (*i.e* 2D or 3D), when and which technique should we apply?". This comparative study is provided in Chapter 3.

A generative framework for human motion synthesis We propose a new framework to generate human pose sequences which is based on the proposed sparse representations. Specifically, we use a Generative Adversarial Network (GAN) [35] which we train to generate new sparse representations of human actions. Then, we transform the obtained samples to pose sequences. We conduct extensive experiments to show the effectiveness of adopting a sparse representation in a generative model. In addition, we apply this framework to augment

an action recognition dataset and demonstrate that this improves our classification results.

Furthermore, we use the generated pose sequences to guide the generation of human videos. Specifically, given an input person image, we transfer each pose of a generated sequence to an image to obtain a new video. To this end, we use a recently-proposed person image generation method [130] which we extend in our work to the temporal domain. Experimentally, we show that our framework yields realistic human videos in terms of motion and content. We also compare our approach with state-of-the-art and demonstrate that it improves the quality of the generated videos.

1.4 Organisation of the dissertation

The rest of this dissertation is structured as follows. Chapter 2 begins with formal studies of sparse representations, Riemannian geometry and the Kendall's shape space with various tools that are exploited in this chapter. We also discuss existing solutions to extend sparse coding and dictionary learning to Riemannian manifolds as well as different approaches to represent Riemannian trajectories. Then, we present the human motion modeling frameworks that are used throughout this thesis work. In Chapter 3, we start by providing a brief literature review of 3D action recognition and 2D facial expression recognition approaches. Next, we present the adopted temporal modeling and classification schemes that we apply to our sparse representations. This is followed by an extensive evaluation of the proposed frameworks with a comparison to state-of-the-art. Moreover, we present a deep analysis of the proposed methods and the obtained results which includes a comparison between the adopted intrinsic and extrinsic coding approaches. Chapter 4 begins with a study of generative adversarial networks and a brief literature review of existing human video generation approaches. Then, we describe the proposed pose sequence and video generation frameworks followed by their experimental evaluation with a comparison to state-of-the-art. Finally, Chapter 5 concludes this dissertation, discusses the main limitations of the

proposed methods and presents possible future directions and perspectives of the proposed work.

Chapitre 2

Coding Human Motion Trajectories in the Shape Manifold

2.1 Introduction

The availability of real-time skeletal data estimation solutions [17, 97] and reliable facial landmarks detectors [7, 8, 118] has pushed researchers to study shapes of landmark configurations and their dynamics. For instance, 3D skeletons have been widely used to represent human actions due to their ability in summarizing the human motion. Another example is given by the 2D facial landmarks and their tremendous use in facial expression analysis. However, human actions and facial expressions observed from visual sensors are often subject to view variations which makes their analysis complex. Considering this non-trivial problem, an efficient way to analyze these data takes into account view-invariance properties, giving rise to shape representations often lying to nonlinear shape spaces [9, 16, 62]. David G. Kendall [62] defines the shape as the geometric information that remains when location, scale, and rotational effects are filtered out from an object. Accordingly, we represent 2D landmark faces and 3D skeletons as points in the 2D and 3D

Kendall's spaces, respectively. Further, when considering the dynamics of these points, the corresponding representations become trajectories in these spaces [9]. However, inferencing such a representation remains challenging due to the *nonlinearity* of the underlying manifolds. In the literature, two alternatives have been proposed to overcome this problem for different Riemannian manifolds – they are either *Intrinsic* [18, 19, 45, 107] or *Extrinsic (kernel-based)* [42, 44, 55, 68]. On one hand, intrinsic solutions tend to project the manifold-valued data to a tangent space at a reference point [4, 9, 107]. While it solves the problem of nonlinearity of the manifold of interest, this solution could introduce distortions, especially when the projected points are far from the reference point. On the other hand, extrinsic solutions are based on embeddings to higher dimensional Reproducing Kernel Hilbert Spaces (RKHS), which are vector spaces where Euclidean geometry applies. These methods bring the advantage that, as evidenced by kernel methods on $\mathbb{R}^n$, embedding a lower dimensional space in a higher dimensional one gives a richer representation of the data and helps capturing complex patterns. Nevertheless, to define a valid RKHS, the kernel function must be positive definite according to Mercer's theorem [90]. Several works in the literature have studied kernels on the 2D Kendall's space. For instance, the Procrustes Gaussian kernel is proposed in [55] as positive definite. In contrast, to our knowledge, such a kernel has not been explored for the 3D Kendall's space.

Motivated by the success of sparse representations in several recognition tasks [20, 42, 45], we propose to code shape trajectories using Riemannian sparse coding and dictionary learning (SCDL) in the shape manifold. We will explore both intrinsic and extrinsic paradigms of this technique and provide the main benefits of the resulting representations to model human actions and facial expressions.

In this chapter, we start by presenting some preliminaries and mathematical definitions. Later on, we review some existing representations of trajectories on Riemannian manifolds along with scientific challenges on the matter. In Section 2.4, we describe the solutions that we adopt to code trajectories in the shape manifold before presenting their properties in the

following section.

2.2 Background and definitions

Data encoding techniques have been broadly studied in literature as they play a crucial role in data analysis. In this dissertation, we aim to represent human skeletons and facial landmark configurations with a coding technique that yields sparse representations. This classical approach assumes that the data are defined in Euclidean space. However in our study, we are applying important shape transformations on the data which turn to be elements of a nonlinear space. As a consequence, classical coding approaches could not directly apply to these data and their nonlinear extension is required. As a solution, the Riemannian geometry of the well-known shape manifold can be used, offering several geometric tools which enables the extension of coding to nonlinear spaces. In the following, we start by describing the aforementioned coding technique. Then, we present basic notions of Riemannian geometry with a particular focus on the Kendall's shape space which is our manifold of interest in this dissertation.

2.2.1 Sparse representations

The notion of sparsity as a way to model signals has generated significant interest in the past decade [20, 42, 45]. In particular, sparse representations have proved to be successful in various computer vision and machine learning problems including classification [30], segmentation [80], image denoising and inpainting [95], visual tracking [71] and face recognition [125], to cite a few. The basic assumption of this model is that, given a dictionary $\mathcal{D} = \{d_1, d_2, ..., d_N\}$ with N elements called *atoms*, a query x has a sparse vector representation $w \in \mathbb{R}^N$ with k non-zero elements under the dictionary $\mathcal{D}$:

$$x = \mathcal{D}w, \tag{2.2.1}$$

with $\|w\|_0 \leq k$, where $\|.\|_0$ represents the ℓ_0-pseudonorm which counts the number of non-zero elements in a vector. In Equation 2.2.1, the query x is represented as a linear combination of k atoms of the dictionary $\mathcal{D}$. The set of indices of these k active atoms is called the support Su. As such, this model assumes that the signal x resides in a low dimensional subspace, which is spanned by the k atoms of $\mathcal{D}$ that correspond to the support of w. An illustration of this model is given in Figure 2.1.

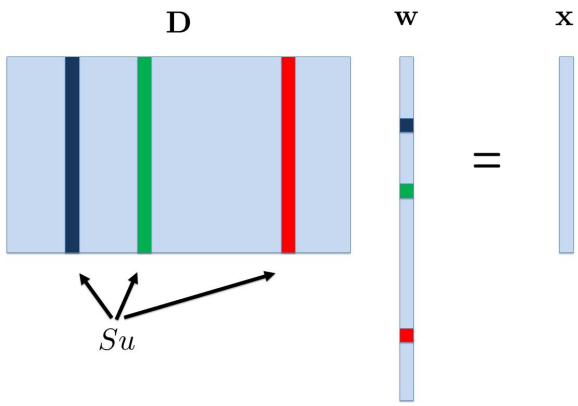


FIGURE 2.1 – Schematic sparse model.

A common approach to estimate the optimal representation w is known as *sparse coding* which includes a penalty function $f(w)$ in the following optimization problem:

$$l_E(x, \mathcal{D}) = \min_w \|x - \sum_{i=1}^{N} [w]_i \, d_i\|_2^2 + \lambda f(w), \tag{2.2.2}$$

where $w \in \mathbb{R}^N$ denotes the vector of codes comprised of $\{[w]_i\}_{i=1}^N$, $f : \mathbb{R}^N \to \mathbb{R}$ is the sparsity inducing function defined as the ℓ_1 norm, and λ is the sparsity regularization parameter. Eq. (2.2.2) seeks to optimally approximate x (by $\hat{x}$) as a linear combination of atoms, *i.e.*, $\hat{x} = \sum_{i=1}^{N} [w]_i \, d_i$, while tacking into account a particular sparsity constraint on the codes, $f(w) = \|w\|_1$. This sparsity function has the role of forcing x to be represented as only a small number of atoms.

The problem of sparse coding assumes that the dictionary is known. In practice, when the dictionary is learned from the data, significant improvements can be made on the

reconstruction of x. This problem is of wide interest and it is known as the dictionary learning problem. Several techniques have been developed to solve this problem and train dictionaries from data. Popular examples are the Method of Optimal Directions (MOD) [32] and the K-SVD method [2], which generalizes the K-means clustering algorithm to learn overcomplete dictionaries. Besides, using adaptive dictionaries [86] is usually the best choice in terms of the reconstruction performance that can be achieved. In addition, it enables to control the amount of induced sparsity over the reconstruction performance.

Formally, given a finite set of t training observations $\{x_1, x_2, ..., x_t\}$ in $\mathbb{R}^k$, learning adaptive dictionaries [86] is defined as to jointly minimize the coding cost over all choices of atoms and codes according to:

$$l_E(\mathcal{D}) = \min_{\mathcal{D},w} \sum_{i=1}^{t} \left\| x_i - \sum_{j=1}^{N} [w_i]_j d_j \right\|_2^2 + \lambda f(w_i). \tag{2.2.3}$$

To solve this non-convex problem, a common approach alternates between the two sets of variables, $\mathcal{D}$ and w, such that: (1) Minimizing over w while $\mathcal{D}$ is fixed is a convex problem (*i.e.*, sparse coding). (2) Minimizing Eq. (2.2.3) over $\mathcal{D}$ while w is fixed is similarly a convex problem.

In our work, we are interested in the extension of SCDL to the Kendall's shape space. Before introducing this manifold, in the following section, we present basic notions on Riemannian geometry.

2.2.2 Riemannian geometry

A manifold $\mathcal{M}$ is a Hausdorff topological space which locally resembles a Euclidean space $\mathbb{R}^n$, where n is the dimension of the manifold. The tangent space at a point on the manifold provides us with a vector space of tangent vectors that give an idea of direction on the manifold. A Riemannian manifold is differential and equipped with a metric on the tangent spaces. A Riemannian metric allows us to compute angle and length of directions (tangent

vectors). A Riemannian metric is a continuous collection of inner products on the tangent space at each point of the manifold. It is usually chosen to provide robustness to some geometrical transformations. Furthermore, it makes it possible to define several geometric notions on a Riemannian manifold such as the geodesic distance between points on the manifold. In fact, points on a Riemannian manifold are connected with smooth curves. Assuming the Riemannian metric, one can compute the length of a given curve. The curves yielding the minimum distance for any two points of the manifold are called geodesics which are analogous to straight lines in $\mathbb{R}^n$. The length of a geodesic defines the *geodesic distance*. The geodesic distance induced by the Riemannian metric is the most natural measure of dissimilarity between two points lying on a Riemannian manifold. However, in practice, many other nonlinear distances which do not necessarily arise from Riemannian metrics can also be useful for measuring dissimilarity on manifolds [54]. Two other essential operations on Riemannian manifolds are the *logarithm map* (log) and *exponential map* (exp). As illustrated in Figure 2.2, the exponential map $\exp_Z(.) : T_Z\mathcal{M} \rightarrow \mathcal{M}$ projects a tangent vector from the tangent space at a point Z to the manifold. It guarantees that the length of the tangent vector is equal to the geodesic distance. The logarithm map $\log_Z(.) = \exp^{-1}(.) : M \rightarrow T_Z\mathcal{M}$ projects a point on the manifold to the tangent space $T_Z\mathcal{M}$ at another point.

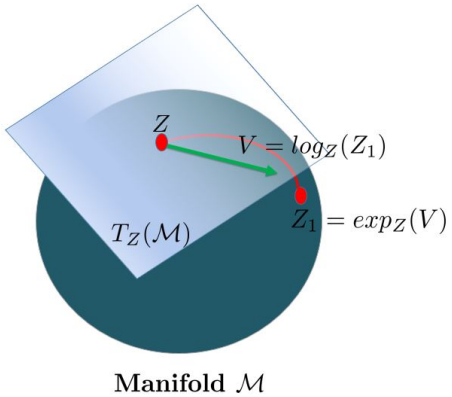


FIGURE 2.2 – Exponential map and logarithm map operations on a Riemannian manifold. The red curve represents the geodesic connecting the points Z and Z_1. The green arrow represents the tangent vector V on the tangent space $T_Z(\mathcal{M})$.

Several Riemannian manifolds have been widely studied in the computer vision literature. Examples include the Lie group SO(3) formed by 3D rotation matrices, the unit n-sphere S^n formed by normalized histograms, the manifold of symmetric positive definite (SPD) matrices, the Grassmann manifold defined as linear subspaces of $\mathbb{R}^n$, and the Kendall's shape space (also known as the shape manifold). For an overview on Riemannian geometry and manifolds, we refer the reader to useful resources on the topic [14, 77]. In the following section, we will outline the geometric properties of the manifold considered in this work, the Kendall's shape space.

2.2.3 Kendall's shape space

Let us consider a set of n landmarks in $\mathbb{R}^m$ $(m = 2, 3)$. To represent its shape, Kendall [62] proposed to establish equivalences with respect to shape-preserving transformations that are translations, rotations, and global scaling. Let $Z \in \mathbb{R}^{n\times m}$ represent a configuration of landmarks. To remove the translation variability, we follow [26] and introduce the notion of Helmert sub-matrix, a $(n-1)\times n$ sub-matrix of a commonly used Helmert matrix, to perform centering of configurations. For any $Z \in \mathbb{R}^{n\times m}$, the product $HZ \in \mathbb{R}^{(n-1)\times m}$ represents the Euclidean coordinates of the centered configuration. Let $\mathcal{C}_0$ be the set of all such centered configurations of n landmarks in $\mathbb{R}^m$, *i.e.*, $\mathcal{C}_0 = \{HZ \in \mathbb{R}^{(n-1)\times m} | Z \in \mathbb{R}^{n\times m}\}$. $\mathcal{C}_0$ is a $m(n-1)$ dimensional vector space and can be identified with $\mathbb{R}^{m(n-1)}$. To remove the scale variability, we define the pre-shape space to be: $\mathcal{C} = \{Z \in \mathcal{C}_0 | \|Z\|_F = 1\}$; $\mathcal{C}$ is a unit sphere in $\mathbb{R}^{m(n-1)}$ and, thus, is $m(n-1)-1$ dimensional. The tangent space at any pre-shape Z is given by: $T_Z(\mathcal{C}) = \{V \in \mathcal{C}_0 | \text{trace}(V^T Z) = 0\}$. To remove the rotation variability, for any $Z \in \mathcal{C}$, we define an equivalence class: $\bar{Z} = \{ZO | O \in SO(m)\}$ that represents all rotations of a configuration Z. The set of all such equivalence classes, $\mathcal{S} = \{\bar{Z} | Z \in \mathcal{C}\} = \mathcal{C}/SO(m)$ is called the *shape space* of configurations. The tangent space at any shape $\bar{Z}$ is $T_{\bar{Z}}(\mathcal{S}) = \{V \in \mathcal{C}_0 | \text{trace}(V^T Z) = 0, \quad \text{trace}(V^T ZU) = 0\}$, where U is any $m \times m$ skew-symmetric matrix. The first condition makes V tangent to $\mathcal{C}$ and the second makes V perpendicular

to the rotation orbit. Together, they force V to be tangent to the shape space $\mathcal{S}$. Assuming standard Riemannian metric on $\mathcal{S}$, the geodesic between two points $\bar{Z}_1, \bar{Z}_2 \in \mathcal{S}$ is defined as:

$$\alpha(t) = \frac{1}{\sin(\theta)}(\sin((1-t)\theta)Z_1 + \sin(t\theta)Z_2O^*), \tag{2.2.4}$$

where $\theta = \cos^{-1}(\langle Z_1, Z_2O^* \rangle)$, $\langle ., . \rangle$ is the inner product on $\mathcal{S}$, and O^* is the optimal rotation that aligns Z_2 with Z_1: $O^* = argmin_{O \in SO(m)} \|Z_1 - Z_2O\|_F^2$. θ is also the geodesic distance between $\bar{Z}_1$ and $\bar{Z}_2$ in the shape space $\mathcal{S}$, representing the optimal deformation to connect $\bar{Z}_1$ to $\bar{Z}_2$ in $\mathcal{S}$. For $t = 0$, $\alpha(0) = \bar{Z}_1$ and for $t = 1$ we have $\alpha(1) = \bar{Z}_2$. Note that Kendall's shape space is a complete Riemannian manifold such that the logarithm map operator $\log_{\bar{Z}}$ is defined for all $\bar{Z} \in \mathcal{S}$ (see Section 2.2.3.1 for its definition). As a consequence, the geodesic distance between two configurations $\bar{Z}_1$ and $\bar{Z}_2$ can be computed as $\mathbf{d}_{\mathcal{S}}(\bar{Z}_1, \bar{Z}_2) = \| \log_{\bar{Z}_1}(\bar{Z}_2) \|_{\bar{Z}_1}$, where $\|.\|_{\bar{Z}_1}$ denotes the norm induced by the Riemannian metric at $T_{\bar{Z}_1}(\mathcal{S})$.

The case of planar shapes For $m = 2$, a 2D landmark configuration can be initially represented as a n-dimensional complex vector whose real and imaginary parts respectively encode the x and y coordinates of the landmarks. In this case, the pre-shape space is defined, after removing the translation and scale effects, as: $\mathcal{C} = \{z \in \mathbb{C}^{n-1} | \|z\| = 1\}$; $\mathcal{C}$ is a complex unit sphere of dimension $2(n-1)-1$. The rotation removal consists of defining, for any $z \in \mathbb{C}^{n-1}$, an equivalence class $\bar{z} = \{zO | O \in SO(2)\}$ that represents all rotations of a configuration z. The final shape space $\mathcal{S}$ is the set of all such equivalence classes $\mathcal{S} = \{\bar{z} | z \in \mathcal{C}\} = \mathcal{C}/SO(2)$. To measure the distance between two shapes $\bar{z_1}$ and $\bar{z_2}$, we define the most popular distance on the 2D Kendall's shape space, named the full Procrustes Distance [62], as

$$d_{FP}(\bar{z_1}, \bar{z_2}) = (1 - |\langle z_1, z_2 \rangle|^2)^{1/2}, \tag{2.2.5}$$

where $\langle \cdot, \cdot \rangle$ and $|.|$ denote the inner product in $\mathcal{S}$ and the absolute value of a complex number, respectively.

2.2.3.1 Geometric tools on the manifold

Considering the spherical structure of $\mathcal{C}$, analytic expressions of the logarithm and exponential maps are well defined [26, 62] and can be easily adapted to $\mathcal{S}$. These operations allow to compensate the lack of vector structure in the shape manifold by working on tangent spaces. In this section, we define these operators, then use them to define a useful algorithm allowing to compute the mean shape of manifold-valued points, namely intrinsic mean.

Exponential map allows projection from a tangent space to the manifold. It applies the shooting vector to a source shape and provides the deformed (target) shape. It is defined, for any $V \in T_{\bar{Z}}(\mathcal{S})$, by,

$$\exp_{\bar{Z}}(V) = \left[\cos(\theta)Z + \frac{\sin(\theta)}{\theta}V\right]. \tag{2.2.6}$$

Logarithm map allows to map a point on the manifold to the tangent space at another point. It represents the *shooting vector* at the first shape (source) to the second shape (target). Its mathematical expression is given explicitly by,

$$\log_{\bar{Z}_1}(\bar{Z}_2) = \frac{\theta}{\sin(\theta)}(Z_2O^* - \cos(\theta)Z_1), \tag{2.2.7}$$

for source shape $\bar{Z}_1$ and target $\bar{Z}_2$, with θ as above.

Intrinsic mean An important advantage of the Riemannian approach is the ability to compute statistics on a set of manifold-valued points. One can use the notion of Karcher mean [60] to define an average shape. This represents an intrinsic mean and can be used as representative of a group of points on the manifold. Let $\{\bar{Z}_1, ..., \bar{Z}_k\}$ be a set of points on $\mathcal{S}$. We define an objective function $\Psi : \mathcal{S} \to \mathbb{R}$, $\Psi(\bar{Z}) = \sum_{i=1}^{k} d_{\mathcal{S}}(\bar{Z}_i, \bar{Z})^2$. The intrinsic mean is obtained by minimizing this objective function, which is commonly solved using a standard algorithm that we describe in Algorithm 1.

As discussed previously, mappings to a tangent space allow to compensate the lack of

Algorithm 1: Computing intrinsic mean on $\mathcal{S}$

input : A set of shapes $\mathcal{Z} = \{\bar{Z}_j\}_{j=1}^m$ and ϵ_1, ϵ_2 small
Initialize $\hat{\mu} \leftarrow Z_1, i \leftarrow 0$
repeat
  Compute $v_j \leftarrow \log_{\hat{\mu}_i}(\bar{Z}_j)$
  Compute average tangent vector $\bar{v} \leftarrow \frac{1}{k}\sum_{j=1}^m v_j$
  Update $\hat{\mu}_i$ according to $\hat{\mu}_{i+1} \leftarrow \exp_{\hat{\mu}_i}(\epsilon_2 \bar{v})$
  $i \leftarrow i + 1$
until $|\bar{v}| < \epsilon_1$
return $\hat{\mu}$, a sample Mean of $\mathcal{Z} = \{\bar{Z}_j\}_{j=1}^m$

vector structure on the shape manifold. We refer to this solution as an *intrinsic* approach. Another solution to seek a vector representation of the data is known as an *extrinsic* approach which embeds the manifold-valued data to a higher-dimensional vector space namely the Hilbert space.

2.2.3.2 Hilbert space embedding of the manifold

A Hilbert space $\mathcal{H}$ is an (often infinite-dimensional) inner product space which is complete with respect to the norm induced by the inner product. A Reproducing Kernel Hilbert Space (RKHS) is a special kind of Hilbert space of functions on some nonempty set $\mathcal{S}$ in which all evaluation functionals are bounded and hence continuous. The inner product of an RKHS of functions on $\mathcal{S}$ can be defined by a bivariate function on $\mathcal{S} \times \mathcal{S}$, known as the reproducing kernel of the RKHS [54].

The SCDL algorithms depend only on the notion of inner product, which allows us to measure distances. Therefore, they can be easily extended to Hilbert spaces. The embedding of the shape manifold to RKHS brings the main advantage of transforming the nonlinear manifold into a vector space where one can directly apply standard (Euclidean) algorithms. In addition, it gives a richer representation of the data in a higher-dimensional space which helps identifying complex patterns. To define an inner product in $\mathcal{H}$, one can use a kernel function $f : (\mathcal{S} \times \mathcal{S}) \rightarrow \mathbb{R}$ which makes the resulting space a RKHS without the need of

computing the actual vectors. This procedure, known as the *kernel trick*, is commonly used in machine learning when the designed algorithm only relies on measures of similarities, i.e., on inner products. However, to define a valid RKHS, the kernel function must be positive definite according to Mercer's theorem [90]. In particular, for the 2D shape manifold, the authors in [55] proved the positive definiteness of the Procrustes Gaussian kernel $k_P : (\mathcal{S} \times \mathcal{S}) \rightarrow \mathbb{R}$ which is defined as

$$k_P(\bar{z_1}, \bar{z_2}) := exp(-{d_{FP}}^2(\bar{z_1}, \bar{z_2})/2\sigma^2), \tag{2.2.8}$$

where d_{FP} is the full Procrustes Distance defined in Eq.(2.2.5) and σ is the kernel parameter. This kernel is positive definite for all $\sigma \in \mathbb{R}$. In Section 2.4.2, it will be used for the extension of SCDL to Hilbert space.

2.3 Related work

In this section, we provide a brief literature overview of Riemannian SCDL approaches. Then, we discuss different representations of trajectories on Riemannian manifolds and highlight their limitations.

2.3.1 Riemannian sparse coding and dictionary learning

The basic definition of SCDL as defined in Section 2.2.1 assumes that the query points as well as the dictionary atoms are defined in vector space. However, most suitable image descriptors often lie to nonlinear manifolds [77]. Thus, to perform SCDL on these data while respecting the geometric structure of Riemannian manifolds, the classical problem of SCDL needs to be extended to its nonlinear counterpart. The main issue here arises from the fact that the linear combination of atoms is not possible on a nonlinear manifold since the resulting point may not even be in the manifold, as illustrated in Figure 2.3. Previous works addressed this problem [20, 40, 41, 42, 45, 68, 131]. For instance, a straightforward

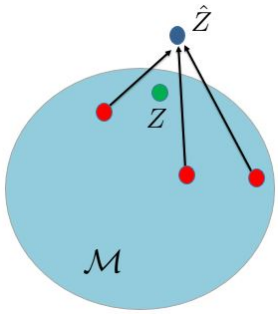


FIGURE 2.3 – Z is a data point on the manifold $\mathcal{M}$ and the red circles represent the dictionary atoms. The linear approximation $\hat{Z}$ with the atoms may be out of the manifold $\mathcal{M}$.

solution was proposed in [40, 122] by embedding the manifolds of interest into vector space, the tangent space at a reference point. However, this solution does not take advantage of the entire Riemannian structure of the manifold since on a tangent space, only distances to the reference point are equal to true geodesic distances. To overcome this problem, Ho *et al.* [45] proposed a general framework for SCDL in Riemannian manifolds by working on the tangent bundle. Here, each point is coded on its attached tangent space into which the atoms are mapped. By doing so, only distances to the tangent point are needed. Their proposed dictionary learning method includes an iterative update of the atoms using a gradient descent approach along geodesics. This general solution essentially relies on mappings to tangent spaces using the logarithm map operator. Although it is well defined for several manifolds, analytic formulation of the logarithm map is not available or difficult to compute for others. Therefore, some studies [41, 42, 44, 68] proposed to embed the Riemannian manifold into RKHSs which are vector spaces where linear SCDL becomes possible. Recently, Harandi *et al.* [42] proposed to map the Grassmann manifolds into the space of symmetric matrices to overcome the latter problem and preserve several properties of the Grassmann structure. They also proposed kernelized versions of the SCDL algorithms to handle the nonlinearity of the data, similarly proposed in [43] for symmetric positive definite matrices. Throughout this dissertation, we will investigate two paradigms of Riemannian SCDL in the shape manifold with the aim of coding trajectories while tackling different challenges that we will discuss in

the following section.

2.3.2 Trajectory representations on Riemannian manifolds

A sequence of points that evolve over time on a manifold can be seen as a time-series. In the case of a Riemannian manifold, a time-series is usually denoted by *trajectory*. Analysis of these trajectories is challenging due to the nonlinearity of underlying spaces. Several representations of landmark sequences lie to nonlinear manifolds. In many approaches, the Riemannian geometry of these manifolds is exploited to analyze the corresponding representations and a common solution to solve for their nonlinearity consists on mapping the manifold-valued data to a common tangent space. A popular example is given by the Lie group and its use for skeletal trajectory analysis. For instance, Vemulapalli *et al.* [107] proposed to represent 3D skeletal sequences in the product space of Special Euclidean (Lie) groups $SE(3)^n$. To this end, for each frame of a sequence, the Euclidean transformation matrices encoding rotations and translations between different joint pairs are computed. Hence, the dynamics of these matrices is seen as a trajectory on $SE(3) \times ... \times SE(3)$. To overcome the nonlinear nature of this manifold, this representation is mapped to the Lie algebra $\mathfrak{se}(3)^n$ which is a vector space, the tangent space at the identity element. However, mapping points to a common tangent space may introduce undesirable distortions, especially when the mapped points are far from the tangent point. Aware of this limitation, the authors in [108] proposed a mapping of trajectories on Lie groups combining the usual logarithm map with a rolling map that guarantees a better flattening of trajectories on Lie groups. Taking another direction, Anirudh *et al.* [4] extended the framework of Transported Square-Root Velocity Fields (TSRVF) [101] by modeling trajectories of human actions on the Grassmann manifold and the product space of Lie groups $SE(3) \times ... \times SE(3)$. They tackled the problems of high-dimensionality of the feature space and its nonlinearity and proposed to learn a low-dimensional embedding using a manifold functional variant of principal component analysis. Hence, each trajectory is mapped to a single point in a low-dimensional Euclidean space.

Another approach [9] proposed a different solution by extending the Kendall's shape theory to trajectories. Accordingly, translation, rotation, and global scaling are filtered out from each skeleton to quantify the shape. Then based on the TSRVF, they defined an elastic metric to jointly align and compare trajectories.

Riemannian trajectories were also considered in the analysis of 2D facial landmark sequences. Taheri *et al.* [102] proposed to represent 2D facial landmarks in the Grassmann manifold which makes the resulting representation invariant to affine transformations and hence robust to view variations. To capture facial expressions from these representations, the authors computed the velocity vectors between successive frames using the logarithm map. To obtain velocities in the same tangent space, they applied a parallel transport of these velocity vectors to a common tangent space. However, their method depends on the choice of this tangent space. In another work [59], 2D facial landmark sequences were first represented as trajectories of Gram matrices in the manifold of positive semidefinite matrices of rank 2. A similarity measure is then provided by temporally aligning trajectories while taking into account the geometry of the manifold.

Most of the methods described above share a common drawback which consists on mapping all the manifold-valued data to a reference tangent space. The major problem of this strategy is that distortions could be introduced especially when the mapped points are far from the tangent point. Moreover, comparing the resulting tangent vectors by computing distances between them is not accurate since only distances to the tangent point are equal to true geodesics. Therefore, our goal is to go beyond this drawback in the proposed intrinsic representation. On the other hand, to our knowledge, extrinsic approaches were not studied in the literature of human modeling with Riemannian trajectories. Hence, we also aim to explore this direction.

2.4 Human motion modeling framework

In this section, we present our human motion modeling, *i.e.* representation of actions and facial expressions, which is used throughout this thesis work for the tasks of human motion classification and generation. Given a set of sequences of 2D/3D skeletons or facial landmarks, our approach consists in three main steps:

— Embedding each frame (*i.e.* landmark configuration) of a sequence to the Kendall's pre-shape space $\mathcal{C}$ by filtering out translation and scale.

— Learning a dictionary from all training samples (*i.e.* static pre-shapes on $\mathcal{C}$). In this step, when an operation involves two configurations on $\mathcal{C}$ such as the logarithm map or the geodesic distance computation, rotation is filtered out by aligning one pre-shape into the second by applying the Procrustes algorithm [62]. By doing so, we consider dictionary learning on the Kendall's shape space $\mathcal{S}$.

— Using the learned dictionary, each frame (*i.e.* pre-shapes on $\mathcal{C}$) of a sequence is coded using Riemannian sparse coding in $\mathcal{S}$ after filtering out rotation as done in the previous step.

As a main result of applying the proposed framework, each frame of an input sequence gives rise to a sparse code vector and frames that have similar shapes are expected to have similar code vectors. As a consequence, the input sequence will turn to a smoothly-evolving time-series (see Figure 2.4).

(A) **(B)**

FIGURE 2.4 – Example results of the proposed framework. An input sequence is transformed to a smoothly-evolving sparse time-series. The X-axis represents the time frame. (A) 3D skeletal sequence. (B) 2D facial landmark sequence.

In this work, we investigate two approaches of Riemannian SCDL. The first is intrinsic and is based on projections to tangent spaces, while the second is extrinsic and relies on embeddings to RKHS. We illustrate these two approaches in Figure 2.5. In the following, we describe each of them.

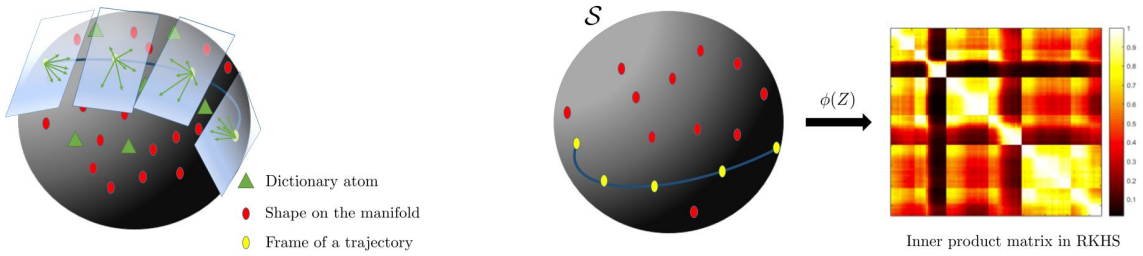


FIGURE 2.5 – Illustration of the adopted solutions to overcome the nonlinearity of the shape manifold. (A) The intrinsic approach maps the data on the manifold to tangent spaces using the logarithm map opertor. (B) The extrinsic approach embeds the manifold-valued data to RKHS by computing the inner product matrix using a positive definite kernel function.

2.4.1 Intrinsic approach

We propose to adapt a general intrinsic formulation of Riemannian SCDL [45] to the case of Kendall's shape space. This allows to transform a 2D/3D landmark configuration lying on the shape manifold to a sparse vector.

2.4.1.1 Intrinsic Sparse Coding

Let $\mathcal{D} = \{\bar{d}_1, \bar{d}_2, ..., \bar{d}_N\}$ be a dictionary on $\mathcal{S}$, and similarly the query $\bar{Z}$ is a point on $\mathcal{S}$. Accordingly, the problem of sparse coding involves the geodesic distance defined on $\mathcal{S}$ and thus the Euclidean formulation in Equation (2.2.2) (in Section 2.2.1) becomes

$$l_{\mathcal{S}}(\bar{Z}, \mathcal{D}) = \min_{w}(d_{\mathcal{S}}(\bar{Z}, F(\mathcal{D}, w))^2 + \lambda f(w)). \tag{2.4.1}$$

Here, $F : \mathcal{S}^N \times \mathbb{R}^N \rightarrow \mathcal{S}$ denotes an encoding function that generates the approximated point $\hat{\bar{Z}}$ on $\mathcal{S}$ by combining atoms with codes. Note that in the special case of Euclidean space, $F(\mathcal{D}, w)$ would be a linear combination of atoms. However, in the Riemannian manifold $\mathcal{S}$, we have forsaken the structure of vector space which makes the linear combination of atoms lying on $\mathcal{S}$ no longer applicable, since the approximated $\hat{\bar{Z}}$ may lie out of the manifold. An interesting alternative is the intrinsic formulation of Eq. (2.4.1), when considering that $\mathcal{S}$ is a complete Riemannian manifold, thus, the geodesic distance $d_{\mathcal{S}}(\bar{Z}, \bar{d}) = \| \log_{\bar{Z}}(\bar{d}) \|_{\bar{Z}}$ (as explained in Section 2.2.3). As a consequence, the cost function in (2.4.1) can be written as

$$l_{\mathcal{S}}(\bar{Z}, \mathcal{D}) = \min_{w} \| \sum_{i=1}^{N} [w]_i \log_{\bar{Z}}(\bar{d}_i) \|_{\bar{Z}}^2 + \lambda f(w), \tag{2.4.2}$$

where $\log_{\bar{Z}}$ denotes the logarithm map operator that maps each atom $\bar{d} \in \mathcal{S}$ to the tangent space $T_{\bar{Z}}(\mathcal{S})$ at the point $\bar{Z}$ being coded, and $\|.\|_{\bar{Z}}$ is the norm induced by the Riemannian metric at $T_{\bar{Z}}(\mathcal{S})$. Mathematically, this allows to partially compensate the lack of vector space structure on $\mathcal{S}$, as illustrated in Figure 2.6. To avoid the solution $w = 0$, we imposed in Eq. (2.4.2) an important additional affine constraint defined as $\sum_{i=1}^{N} [w]_i = 1$. By this formulation of sparse coding, we only compute distances to the tangent point, hence we avoid the commonly induced distortions when working in a reference tangent space. By substituting the logarithm map by its explicit formulation in Eq. (2.4.2), we have

$$l_{\mathcal{S}}(\bar{Z}, \mathcal{D}) = \min_{w} \| \sum_{i=1}^{N} [w]_i \frac{\theta}{\sin(\theta)} (d_i O^* - \cos(\theta) Z) \|_{\bar{Z}}^2 + \lambda f(w). \tag{2.4.3}$$

In practice, Eq. (2.4.3) is computed by first finding the optimal rotation O^* between Z and each atom d_i via the Procrustes algorithm [62]. Then, we solve for w using the state-of-the-art CVXPY optimizer [25]. In Algorithm 2, we provide a summary of the sparse coding approach on the shape manifold.

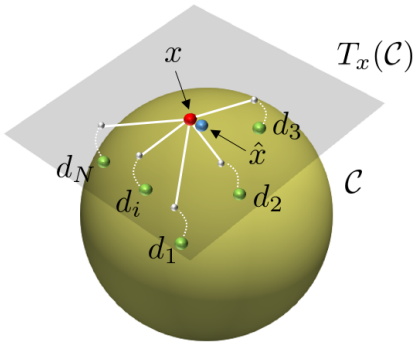


FIGURE 2.6 – Pictorial of the sparse coding approach on the pre-shape space $\mathcal{C}$. The approximation of $x \in \mathcal{C}$ could be viewed as a weighted intrinsic mean of the atoms of a dictionary $\mathcal{D} = \{d_i\}_{i=1}^{N}$.

Algorithm 2: Riemannian sparse coding algorithm

input : Dictionary $\mathcal{D} = \{\bar{d}_i\}_{i=1}^{N}$, $\bar{d}_i \in \mathcal{S}$; $\bar{Z} \in \mathcal{S}$ (query)
output: Sparse codes vector $w^*$of the query $\bar{Z}$
/ Projection of $\mathcal{D}$ into $T_{\bar{Z}}(\mathcal{S})$ */*
for $i = 1$ *to* N **do**
  $\mathcal{V}_i \leftarrow \log_{\bar{Z}}(\bar{d}_i)$
$w^* = \text{argmin}_w \| \sum_{i=1}^{N} [w]_i \mathcal{V}_i \|_2^2 + \lambda f(w)$
return Sparse codes vector $w^*$of the query $\bar{Z}$

2.4.1.2 Intrinsic Dictionary Learning

Learning a discriminative dictionary $\mathcal{D}$ typically yields accurate reconstruction of training samples and produces discriminative sparse codes. We propose a dictionary learning algorithm based on the sparse coding framework described above. Let $\mathcal{D} = \{\bar{d}_1, \bar{d}_2, ..., \bar{d}_N\}$ be a dictionary on $\mathcal{S}$, and similarly $\{\bar{Z}_1, \bar{Z}_2, ..., \bar{Z}_t\}$ is a set of t training samples on $\mathcal{S}$. Similarly to the sparse coding problem, we introduce in Eq. (2.2.3) the geodesic distance defined on $\mathcal{S}$ computed as $d_{\mathcal{S}}(\bar{Z}, \bar{d}) = \| \log_{\bar{Z}}(\bar{d}) \|_{\bar{Z}}$. As a consequence, the problem of dictionary learning on Kendall's shape space is written as

$$\min_{\mathcal{D},w} \sum_{i=1}^{t} \left\| \sum_{j=1}^{N} [w_i]_j \log_{\bar{Z}_i} \bar{d}_j \right\|_{\bar{Z}_i}^2 + \lambda f(w_i), \tag{2.4.4}$$

with the important affine constraint $\sum_{j=1}^{N} [w]_j = 1$. Similarly to the Euclidean case, the optimization problem can be solved by iteratively performing sparse coding while fixing $\mathcal{D}$, and optimizing $\mathcal{D}$ while fixing the sparse codes. Algorithm 3 summarizes the different steps of dictionary learning.

Algorithm 3: Riemannian dictionary learning algorithm

input : Training set $\mathcal{Z} = \{\bar{Z}_j\}_{j=1}^{m}$, where $\bar{Z}_j \in \mathcal{S}$;
output: Dictionary $\mathcal{D} = \{\bar{d_i}\}_{i=1}^{N}, \bar{d}_i \in \mathcal{S}$
Dictionary initialization using Bayesian clustering and PGA (see Section 2.4.1.2))
/ Processing */*
for $k = 1$ *to nIter* **do**
  Sparse Coding using Algorithm 2; w_j^* are the output sparse codes.
  */*Dictionary update Step */*
  for $a = 1$ *to* N **do**
    Updating atom i using line-search algorithm
    $w^* = \text{argmin}_w \| \sum_{i=1}^{N} [w]_i \mathcal{V}_i \|_2^2 + \lambda f(w)$
    $\bar{d_a}^* = \text{argmin}_{\bar{d_a}} \sum_{j=1}^{m} \| [w_a] \log_{\bar{Z}_j}(\bar{d}_a) \|_{\bar{Z}_j}^2 + \| \sum_{i=1;i\neq a}^{N} [w_i] \log_{\bar{Z}_j}(\bar{d}_i) \|_{\bar{Z}_j}^2 + \lambda f(w_j)$
return Dictionary $\mathcal{D} = \{\bar{d_i}\}_{i=1}^{N}, \bar{d}_i \in \mathcal{S}$

An efficient dictionary initialization for faster learning The performance of sparse coding depends on the number of the dictionary elements N, and an empiric choice of N can be time consuming, especially when it comes to large datasets. As a solution, we propose an initialization step that enables an automatic inference on N and accelerates the convergence of the dictionary learning algorithm. To this end, we propose to cluster the training shapes by adapting the Bayesian clustering of shapes of curves method proposed in [127]. The latter brings the advantage of automatically inferring the number of clusters from a set of data. From each cluster, main representatives will then be selected to constitute the initial dictionary.

Kernel-based clustering of shapes for dictionary learning – In Figure 2.7, we show a qualitative result of clustering 3D skeletal shapes and 2D facial shapes. To achieve it, an inner product matrix is first computed from the training data based on the kernel function

defined in Section 2.2.3.2. Note that in the 3D case, this kernel is positive definite for only certain values of the kernel parameter σ. Thus, its empiric choice is required to seek positive definiteness. The inner product matrix is then modeled using a Wishart distribution. To allow for an automatic inference on the number of clusters, prior distributions are carefully assigned to the parameters of the Wishart distribution. Then, posterior is sampled using a Markov chain Monte Carlo procedure based on the Chinese restaurant process for final clustering. For details, we refer the reader to [127], where the authors presented the Bayesian clustering method to segment shapes of curves. In our work, we propose to adapt their approach to cluster shapes of 2D/3D landmark configurations where the only difference resides on the computation of the distance matrix.

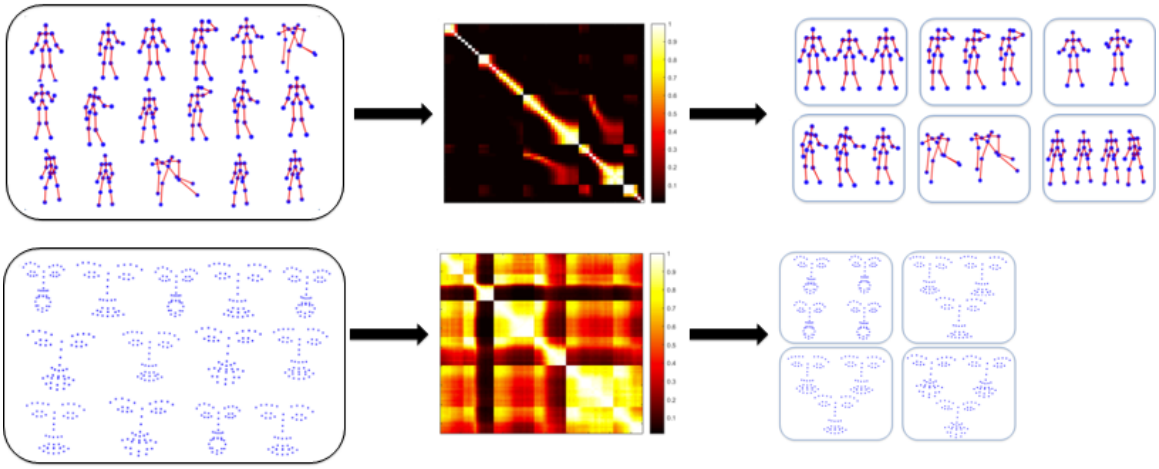


FIGURE 2.7 – Pictorial of the proposed clustering approach. Landmark configurations (left) are mapped from the the shape manifold to RKHS by computing the inner product matrix from the data (Middle). Bayesian clustering is then applied on this matrix to construct the final clusters (right) whose number is automatically inferred.

Atoms inference from clusters – To select the best representatives of a cluster, the mean shape is a suitable candidate but it is not sufficient to summarize the intra-cluster variability. For that, we propose to apply principal geodesic analysis (PGA), first proposed by [33]. Specifically, all elements of a culster are mapped to the tangent space at the mean

shape $T_{\bar{\mu}}(\mathcal{S})$. Then, principal component analysis (PCA) is applied in this vector space to induce the main components (tangent vectors). Finally, the induced vectors are mapped back to the manifold $\mathcal{S}$ using the exponential map operator to represent inital atoms. This procedure is applied to all the clusters. Note that an important advantage of performing PGA in each cluster rather than in the whole training set is to avoid the problematic case of having points in the manifold that are far from the tangent point.

2.4.2 Extrinsic approach

The SCDL algorithms only depend on the notion of inner product. In the following, we will discuss how they can be easily extended to RKHS using the Procrustes Gaussian Kernel.

2.4.2.1 Kernel Sparse Coding

A closed-form solution of kernel sparse coding is proposed in [42]. To derive it, let us first define $\phi : \mathcal{S} \rightarrow \mathcal{H}$ a mapping to RKHS induced by the kernel $k(\bar{z_1}, \bar{z_2}) = \phi(\bar{z_1})^T \phi(\bar{z_2})$, where $\bar{z_1}, \bar{z_2} \in \mathcal{S}$. For a query shape $\bar{z} \in \mathcal{S}$, extending Eq. 2.2.2 to RKHS yields

$$l_{\mathcal{H}}(\bar{z}, \mathcal{D}) = \min_{w} \|\phi(\bar{z}) - \sum_{i=1}^{N} [w]_i \, \phi(\bar{d}_i)\|_2^2 + \lambda f(w), \tag{2.4.5}$$

with $\sum_{i=1}^{N} [w]_i = 1$. In Eq. 2.4.5, since the sparsity term depends entirely on w, only the reconstruction term needs to be kernelized. Expanding the latter gives

$$\begin{aligned} \|\phi(\bar{z}) - \sum_{i=1}^{N} [w]_i \, \phi(\bar{d}_i)\|_2^2 &= \phi(\bar{z})^T \phi(\bar{z}) \\ &= -2 \sum_{i=1}^{N} [w]_i \, \phi(\bar{d}_i)^T \phi(\bar{z}) + \sum_{i,j=1}^{N} [w]_i \, [w]_j \, \phi(\bar{d}_i)^T \phi(\bar{d}_j) \\ &= k(\bar{z}, \bar{z}) - 2 w^T k(\bar{z}, D) + w^T K(D, D) w, \end{aligned} \tag{2.4.6}$$

where $k(\bar{z}, \mathcal{D})$ is the N-dimensional kernel vector computed between the query $\bar{z}$ and the dictionary atoms, and $K(\mathcal{D}, \mathcal{D})$ is the $N \times N$ kernel matrix computed between the atoms.

An efficient solution of kernel sparse coding can be obtained by considering $U\Sigma U^T$ as the SVD of the symmetric positive definite kernel $K(D, D)$, and $k(\bar{z}, \bar{z})$ as a constant term (independent on w). Thus, Eq. 2.4.6 can be written as the least-squares problem in $\mathbb{R}^N$: $\min_w \|\tilde{z} - \tilde{D}w\|_2^2$, where $\tilde{D} = \Sigma^{1/2}U^T$ and $\tilde{z} = \Sigma^{-1/2}U^T k(\bar{z}, D)$ (we refer to [42] for the proof). In this work, this approach is applied in the Kendall's shape space by using the Procrustes Gaussian Kernel defined in Section 2.2.3.2.

2.4.2.2 Kernel Dictionary learning

Similarly to Euclidean dictionary learning, the extrinsic Riemannian formulation is based on an alternating optimization strategy to update weights and atoms. While the first step is obtained with extrinsic sparse coding presented above, the second is presented in what follows. Given the codes from the first step, the problem of dictionary learning can be viewed as optimizing Eq. (2.4.5) over $\mathcal{D}$. The main idea here is to represent $\mathcal{D}$ as a linear combination of the training samples Y in RKHS according to the Representer theorem [91]. The resulting weights for the M training samples are stacked in a $M \times N$ matrix V, which gives $\phi(D) = \phi(Y)V$. Since only the first term in Eq. 2.4.5 depends on $\mathcal{D}$, the problem of dictionary update can be written as $U(V) = \|\phi(Y) - \phi(Y)VW\|_2^2$, where W is the $N \times M$ matrix of sparse codes obtained from the first step. The latter can be expanded to:

$$U(V) = Tr(\phi(Y)(I_M - VW)(I_M - VA)^T\phi(Y)^T)$$

$$= Tr(K(Y,Y)(I_M - VW - W^TV^T + VWW^TV^T)).$$

To obtain the updated dictionary that is now defined by V, the gradient of $U(V)$ is zeroed out w.r.t V. This gives $V = (WW^T)^{-1}W = W^\dagger$, where † is the pseudo-inverse operator.

2.5 Properties of the latent space

Sparse coding of trajectories give rise to time-series defined in vector space which we refer to as the latent space. In this section, we describe the main properties of the obtained representations in this space.

2.5.1 Reconstruction of trajectories

As illustrated in Figure 2.8, an important advantage of the intrinsic approach is that it enables to recover a sparse code back to the original manifold. This can be extremely useful for visualization purposes and for certain tasks such as motion generation, as will be studied in Chapter 4. Shape reconstruction is achieved with respect to the pre-learned dictionary by applying the weighted intrinsic mean algorithm described in Algorithm 4. The idea here is based on the intrinsic mean, *i.e.* Algorithm 1, where we now: 1) Initialize the approximated shape $\hat{\mu}$ as the linear combination of codes and the corresponding atoms from the dictionary. By doing so, we can obtain a good approximation of the original shape, knowing that the code vector is sparse and supposing that atoms with non-zero coefficients (elements of code vector) are the closest to the tangent point; 2) Compute tangent vectors as $\log_{\hat{\mu}_i}(w_j \bar{D}_j)$ and the mean tangent vector $\bar{v}$. Then update $\hat{\mu}$ by moving $\bar{v}$ to the average direction. This is done iteratively until the norm of $\bar{v}$ is sufficiently close to zero.

Algorithm 4: Weighted intrinsic mean for shape reconstruction from code

input : A vector of codes shapes $\{w_j\}_{j=1}^m$, a dictionary $\mathcal{D} = \{\bar{D}_j\}_{j=1}^m$ and ϵ_1, ϵ_2 small
Initialize $\hat{\mu} \leftarrow \frac{1}{m}\sum_{j=1}^m w_j D_j, i \leftarrow 0$
repeat
  Compute $v_j \leftarrow \log_{\hat{\mu}_i}(w_j \bar{D}_j)$
  Compute average tangent vector $\bar{v} \leftarrow \frac{1}{k}\sum_{j=1}^m v_j$
  Update $\hat{\mu}_i$ according to $\hat{\mu}_{i+1} \leftarrow \exp_{\hat{\mu}_i}(\epsilon_2 \bar{v})$
  $i \leftarrow i + 1$
until $|\bar{v}| < \epsilon_1$
return $\hat{\mu}$, the reconstructed shape

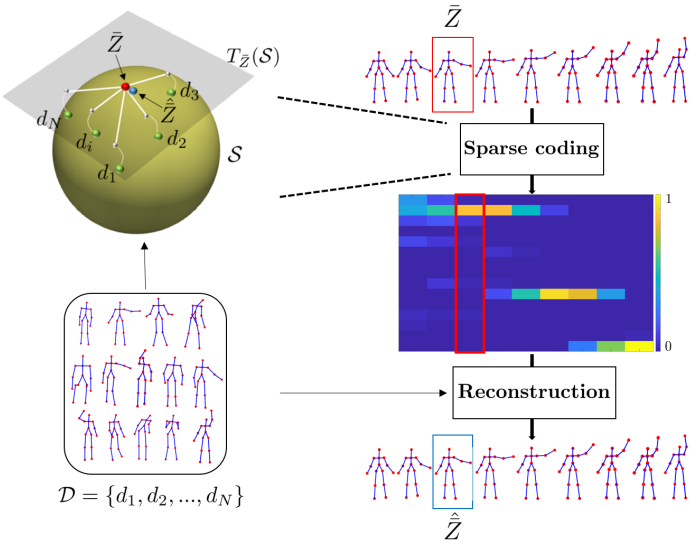


FIGURE 2.8 – Given the pre-trained dictionary $\mathcal{D}$, skeletal trajectories in the shape manifold can be reconstructed from the space of sparse codes using the weighted intrinsic mean algorithm.

2.5.2 Efficient tangent space projections

As explained in Section 2.3.2, many approaches that model sequences of human landmarks as trajectories on Riemannian manifolds [9, 102, 107] share a major drawback, that of mapping manifold-valued points to a reference tangent space which may introduce distortions especially when points are far from the tangent point. In contrast, in our coding approach, each point is coded on its attached tangent space, where the dictionary atoms are mapped (see Figure 2.9 for illustrations of tangent space approximation strategies). By doing so, we only compute distances to the tangent point which are equal to true geodesics. Furthermore, even though we map all the atoms to a tangent space where some of them may be far from the tangent point, in practice, distortions are usually avoided since our sparse coding scheme tends to code a point using the closest atoms to it and attributes zeros to the rest, distant atoms. As a consequence, assuming that the dictionary is well learned (*i.e.* the atoms cover all the space of training shapes), our approach considerably alleviates the non-trivial problem of trajectory distortions caused by tangent space approximations.

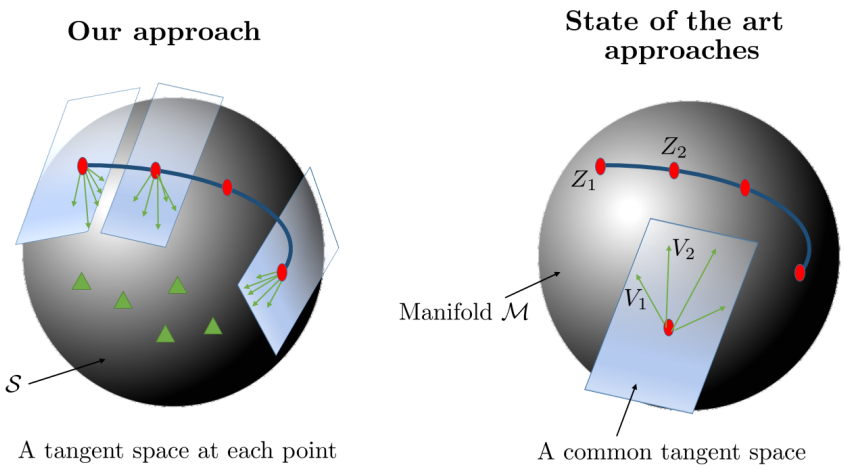


FIGURE 2.9 – Schematic tangent space projections using our method compared to state-of-the-art.

2.5.3 Denoising of skeletal shapes

Skeletal joints obtained from low cost sensors are often noisy leading to abnormal skeletal poses. This is a non-trivial issue for applications as action recognition since the performance of a landmark-based recognition approach relies essentially on the accuracy of the extracted landmarks. One advantage of the proposed sparse coding approach is that it naturally enables denoising of skeletons when assuming a clean dictionary. Figure 2.10 presents an illustration of this property using data collected from the state-of-the-art MSR-Action 3D dataset [69]. Here, only certain joints are poorly estimated in a body pose, *e.g.,* right and left knees, hence the global shape is preserved. Sparse coding attempts to approximate this shape using the closest atoms. Assuming that the dictionary does not contain abnormal shapes, the resulting approximation is expected to recover the input abnormal shape. The question now is how to obtain a clean dictionary? Recall that to train a dictionary, skeletons are collected from all training sequences. We point out that in general, noise appears in only certain frames of a sequence. In addition, actions usually evolves smoothly over time. Considering these two information, one can compute distances between successive frames and discard skeletons with a relatively great distance to the previous.

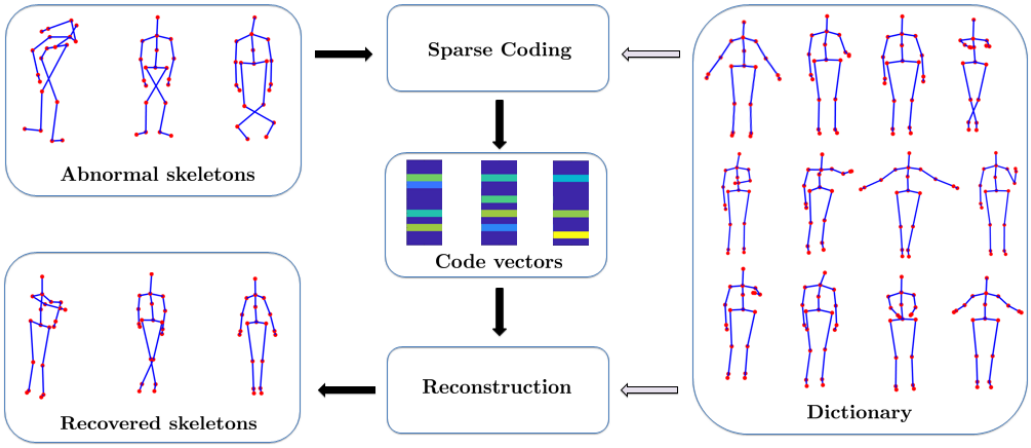


FIGURE 2.10 – Denoising of skeletal shapes using sparse coding. Abnormal skeletons are presented in the top left. Code vectors (middle) are obtained after sparse coding. They are then reconstructed with respect to the dictionary (right) to recover the abnormal skeletons.

2.5.4 On the vector structure of the latent space

Standard notions of statistics (*e.g.,* mean computation, interpolation, etc.) and analysis of time-series (*e.g.*, temporal alignment, temporal modeling, etc.) need significant modification to account for the nonlinearity of the shape manifold. In most cases, these operations become highly involved in terms of computational complexity, and often result in iterative procedures further increasing the computational load. The proposed sparse coding approach allows to exploit the linear nature of the latent variables as well as their low-dimensionality to compute standard statistics on the data and apply standard techniques to process time-series, rather than performing them directly on the manifold. For instance, one can compute the mean code linearly and recover it back to the manifold and still obtain a point on the manifold that is very close to the intrinsic mean shape of the same data. An illustration of this is given in Figure 2.11. One can also interpolate linearly between latent variables and obtain meaningful interpolates when mapped back to the shape manifold, see Figure 2.12. Now, we turn our attention to our main concern which is to consider time-evolving shapes that represent actions or facial expressions. As explained previously, sparse coding of each

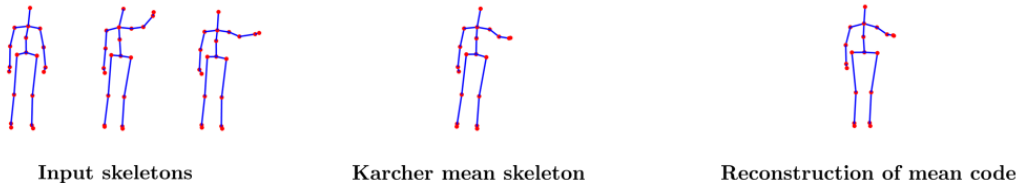


FIGURE 2.11 – Mean shape computation on a set of skeletons. Left: Input 3D skeletons in the Kendall's shape space. Middle: Mean shape computed using the intrinsic mean algorithm. Right: Mean shape computed by first sparse coding the input skeletons, then computing the mean code and reconstruct it with the dictionary.

shape of a trajectory gives rise to a smoothly-varying time-series that is naturally defined in vector space. Thereby, assuming the linearity of the latent space, one can apply machine learning techniques as well as post-processing methods dedicated to Euclidean time-series (up or down temporal resampling, denoising of time-series, temporal alignment of different time-series, etc.), without any manifold assumption.

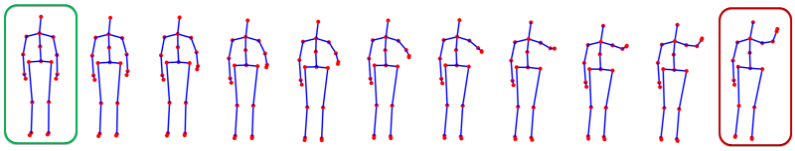


FIGURE 2.12 – An example of linear interpolation between two latent variables: source (left) and target (right). Shapes in this figure are the result of mapping the interpolates from the latent space to the manifold.

2.6 Summary

In this chapter, we proposed a novel action and facial expression modeling based on Riemannian sparse coding and dictionary learning in the shape manifold. This solution allows to overcome the nonlinear structure of the manifold by mapping a Riemannian trajectory to an Euclidean time-series. In addition to its sparsity and vector structure, this representation allows to reconstruct original trajectory from its latent representation

thanks to the dictionary. In addition, it presents a natural denoising tool allowing to alleviate the noise present in the data. We explored both intrinsic and extrinsic solutions of SCDL. The first extends sparse coding to tangent spaces while avoiding a commonly encountered problem which is to map all manifold-valued data to a common tangent space. The second is based on embedding the manifold-valued data to Hilbert space via a positive definite kernel function. In Chapter 3, these two approaches will be further studied and evaluated in the context of two recognition tasks: 3D action recognition and 2D facial expression recognition. In Chapter 4, we will exploit the intrinsic approach to generate new skeletal sequences and show how we can use them to guide the generation of human videos.

Chapitre 3

Facial Expression and Action Recognition with Sparse Representations

3.1 Introduction

In the previous chapter, we proposed a novel framework to encode trajectories in the shape manifold. We explored two alternatives of Riemannian SCDL allowing to represent human actions and facial expressions as sparse times-series in vector space. The first is an intrinsic solution where SCDL is performed on the manifold tangent spaces while the second is based on embedding the manifold-valued data to Hilbert spaces. In this chapter, we demonstrate the effectiveness of these sparse representations in two recognition tasks: the 3D action recognition and 2D facial expression recognition (both micro and macro expressions). Specifically, we will show that the intrinsic coding approach is efficient to code 3D shape trajectories while the extrinsic method is suitable for 2D trajectories. In the context of the addressed classification problems, these coding techniques bring two

main advantages: (1) Sparse coding of shapes is performed with respect to a Riemannian dictionary. Hence, the resulting sparse times-series are expected to be more discriminative than the data themselves. In addition, they are robust to noise, knowing that SCDL is a powerful denoising tool as demonstrated in Section 2.5.3 of Chapter 2; (2) Using sparse time-series as discriminative features allows us to perform both temporal modeling and classification in vector space, avoiding the more difficult task of classification on the manifold. To this end, we will study and compare two different pipelines for temporal modeling and classification. The first is based on a standard machine learning technique while the second is a deep learning approach based on RNNs. An overview of the proposed approaches is given in Figure 3.1.

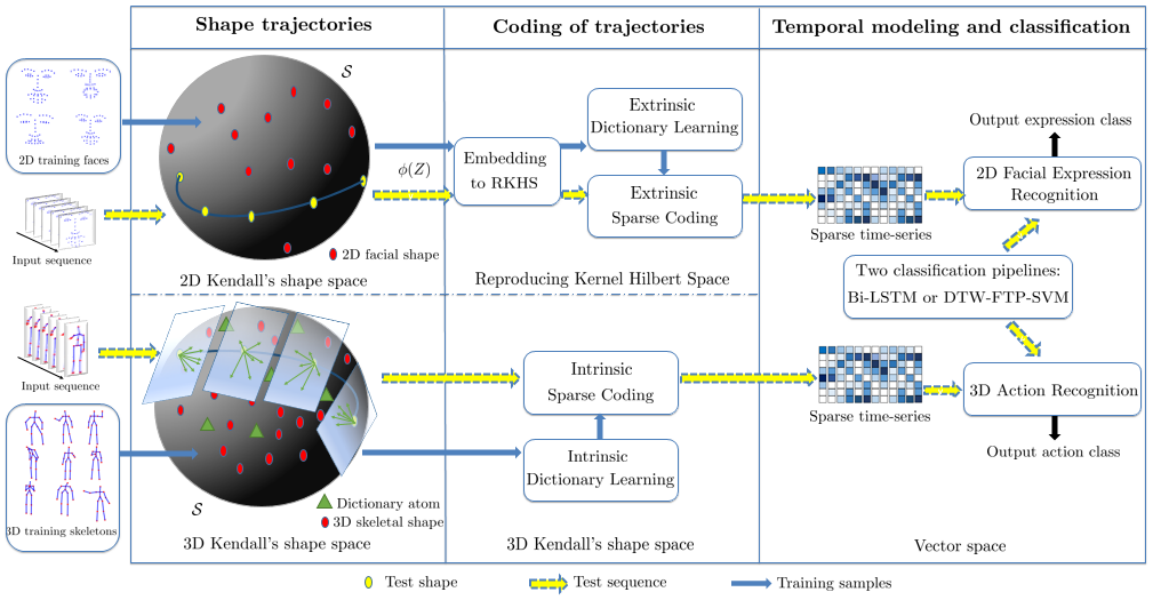


FIGURE 3.1 – Overview of the proposed frameworks. Trajectories of 2D facial expressions (respectively 3D actions) are encoded using extrinsic (respectively intrinsic) SCDL in the Kendall's shape space. Temporal modeling and classification are then performed on the obtained time-series in vector space.

Our main contributions in this chapter are:

— Application of the framework proposed in Chapter 2 to: 3D action recognition,

2D micro- and 2D macro- facial expression recognition. Extensive experiments are conducted on seven commonly-used datasets to show the competitiveness of the proposed approach to state-of-the-art.

— A comparative study on the intrinsic and extrinsic paradigms of Riemannian SCDL in the 2D and 3D shape manifolds. To the best of our knowledge, this work is the first to apply and compare both approaches to dynamic 2D and 3D shapes.

3.2 Related work

The typical framework of human motion recognition using skeletons or faces comprises the following phases, see Figure 3.2. A feature extraction step to capture lower-level information from the data. This step can account to the spatial information only or also considers the temporal dimension. We consider two main categories of features: hand-crafted features and learned features by means of deep learning. Such representations can be used as input to a temporal modeling stage to capture the temporal dependencies in time-series. The output is then fed to a classification stage which can consist on a standard classifier (e.g. k-nearest neighbors, SVM, etc.) or a deep learning framework.

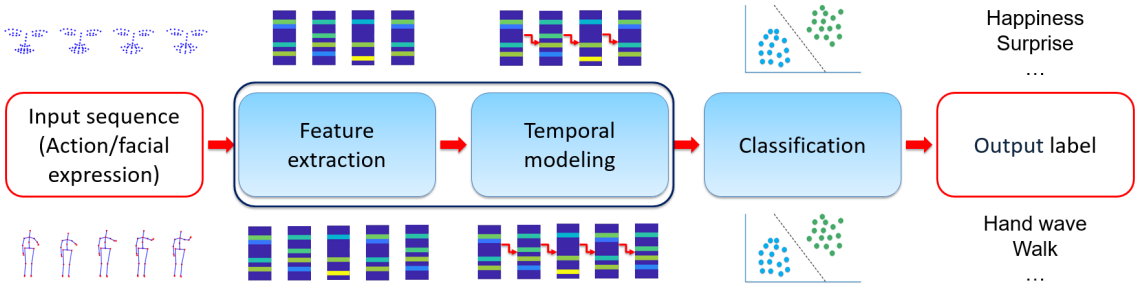


FIGURE 3.2 – Overview of a typical landmark-based action/facial expression recognition approach.

Most of the landmark-based approaches in the literature follow the above pipeline to represent and classify 3D actions or 2D facial expressions. At large, we can regroup them into

two main categories: classical methods which are based on hand-crafted feature extraction and deep learning methods which automatically learn features by designing suitable network architectures and objectives for the task at hand, see Figure 3.3.

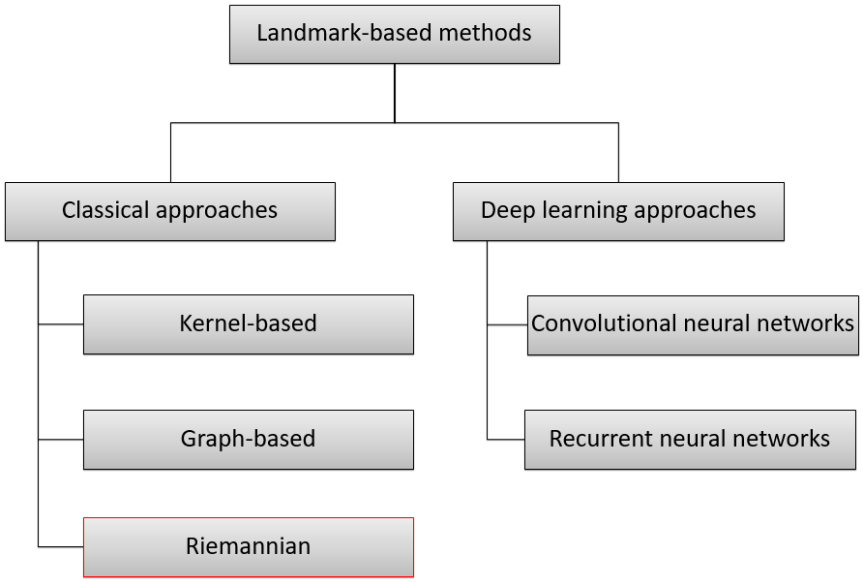


FIGURE 3.3 – Our categorisation of state-of-the-art approaches. Representations of 2D/3D sequences of landmarks (skeletons or faces) can be either hand-crafted or learned. The former representations can be categorised into: kernel-based, graphical models or Riemannian. The latter can be achieved using deep learning techniques.

In one hand, we categorise the classical methods into three groups:

Graph-based methods Graph is a powerful tool for modeling structured objects. Since the representation of a human skeleton or face is in essence composed of points that are connected to each other, it is natural to perceive it as a graph. Therefore, several approaches opted for graphs to model the spatial dependencies between human joints as well as their temporal evolution. Probabilistic models such as Hidden Markov models (HMMs), which also falls in this category, have been widely used to model the temporal evolution of human pose sequences.

Kernel methods attempt to compute similarities between data-points (*e.g.* landmark

configurations, features extracted from them, etc.) using kernel functions. This enable them to operate in a high-dimensional, implicit feature space. The latter is also called the inner product space since only inner products between data-points are computed, without ever computing the coordinates of the data in the new space. These approaches are known to bring a richer representation of the original data since the inner product space is usually higher-dimensional which helps classification methods to identify complex patterns.

Riemannian methods Several representations of landmark sequences may lie to nonlinear manifolds where traditional computational tools and machine learning techniques cannot be directly applied. In fact, in contrast to vector spaces, these manifolds are characterized with nonlinear topology that algorithms have to take into account. As an example, to compute the similarity between two points on a nonlinear manifold, the Euclidean distance is no longer suitable since it does not represent the real proximity between them. As a solution, several approaches studied the Riemannian geometry of these manifolds to define a metric which is obtained by defining a smoothly varying inner product on each tangent space of the manifold. Hence, the vector space structure of these tangent spaces can be exploited to overcome the nonlinearity of Riemannian manifolds.

On the other hand, feature learning using deep learning techniques has been receiving increasing attention in recent years due to their ability of designing powerful features without the need of heavy human labor and domain expert knowledge to develop effective feature extraction methods. Their success is also attributed to the access to Graphics Processing Units (GPUs) as well as to large scale labeled datasets which allow to design networks with millions of parameters. These deep learning techniques consist of multi-layer neural networks with a number of connected units (neurons). These layers represent three types: input layer with input units receiving information to be processed, output layer with output units giving the result of the network, and hidden layers with hidden units which process the data. The training objective of these neural networks consists of learning the weights of the connections between neurons in order to determine the mapping between the input

and the output. The most popular deep learning methods that are commonly used in action and facial expression recognition are convolutional neural networks (CNNs) and recurrent neural networks (RNNs). The former are based on the powerful convolution operation and belong to the category of feed-forward neural networks where the information moves only in one direction, that of the output layer. The latter are a kind of neural networks that takes sequential input and infer sequential output by sharing recurrent connections (parameters) between time steps, in contrast to CNNs.

In the following, we review some examples of methods from the categories discussed above in the context of 3D action recognition and 2D facial expression recognition, with a focus on Riemannian approaches since our proposed framework is also Riemannian.

3.2.1 3D action recognition

Over the last decade, many techniques have been proposed for action recognition from 3D skeletal data. As shown in Figure 3.3, several methods are based on the extraction of hand-crafted features. Examples include the use of spatio-temporal graphical models to represent and classify action sequences. Considering a human action as transitions between body poses over time, G. Hernando *et al.* [34] proposed a forest-based classifier called transition forests to discriminate both static pose information and temporal transitions between pairs of two independent frames. Another work [114] modeled a human action as a set of semantic parts called *motionlets* obtained by tracking then segmenting the trajectory of each joint. By combining the motionlets and their spatio-temporal correlations, they proposed an undirected complete labeled graph to represent a video, and a subgraph-pattern graph kernel to measure the similarity between graphs, then to classify videos.

In the category of kernel-based approaches, two kernel-based tensor representations named sequence compatibility kernel (SCK) and dynamics compatibility kernel (DCK) were introduced in [66]. These can capture the higher-order relationships between the joints. The

first captures the spatio-temporal compatibility of joints between two sequences, while the second models a sequence dynamics as the spatio-temporal co-occurrences of the joints. Tensors are then formed from these kernels to train SVM.

The above-mentioned approaches did not make any manifold assumptions on the data representation. However, several shape representations and their dynamics often lie to nonlinear manifolds. As discussed in the category of Riemannian approaches, many approaches exploited the Riemannian geometry of nonlinear manifolds to analyze skeletal sequences. The latter are considered in different Riemannian manifolds such as the Lie group, the Grassmann manifold, the shape manifold, etc. Since the work presented in this dissertation is based on a representation in the shape manifold, we consider it as part of this category. In Section 2.3.2 of Chapter 2, we described some representations of 3D skeletal sequences on Riemannian manifolds. Here, we discuss their corresponding temporal modeling and classification steps. In [107], sequences are represented as trajectories in the product space of Lie groups $SE(3) \times ... \times SE(3)$ then mapped to the Lie algebra $\mathfrak{se}(3)^n$, the tangent space at the identity element. To handle the rate variability in human actions, the obtained time-series in $\mathfrak{se}(3)^n$ are aligned by means of the popular dynamic time warping (DTW) algorithm. To handle temporal misalignment as well as the noise present in the data, the aligned sequences are processed using Fourier temporal pyramid (FTP). Finally, the obtained features are classified using a one-vs-all SVM. In [108], the authors also applied DTW on the Lie group representation of actions then classified final curves using a one-vs-all linear SVM after mapping the curves to the Lie algebra using rolling maps which ensures a better flattening of the Lie group. DTW and SVM were also used in [4] to classify their low-dimensional representation of actions which were initially represented as trajectories in the Lie group. Based on the same trajectory representation on $SE(3)$, the authors in [51] proposed a deep learning framework in Lie groups to recognize actions. The proposed architecture includes several special layers (*e.g.* RotMap layer, RotPooling layer, etc.) which accounts to the geometry of the manifold. Taking another direction, the

authors in [9] represented skeletal sequences as trajectories in the Kendall's shape space then modeled them using TSRVF. For classification, they computed the mean trajectories of each class and for each trajectory they extracted a feature vector formed by distances to mean trajectories of each class. Finally, they used SVM to classify these feature vectors. Recall that a common drawback of many of these approaches is the mapping of all manifold-valued data to a reference tangent space which may introduce distortions. In contrast, our coding approach in the shape manifold avoids such a problem.

On the other hand, RNNs, which belong to the category of deep learning approaches according to our categorisation in Figure 3.3, have showed promising performance when applied to 3D action recognition. For instance, HBRNN-L [28] applied bidirectional RNNs hierarchically by dividing a skeleton into five parts of neighboring joints. Then, each is separately fed into a bidirectional RNN before fusing their outputs to form the upper-body and the lower-body. Similarly, these latter were fed into different RNNs and their outputs fusion form the global body representation. More recently, the spatio-temporal LSTM (ST-LSTM) [73] extended LSTM to spatio-temporal domains. To this end, the analysis of a 3D skeleton joint considers spatial information from neighboring joints and temporal information from previous frames. In addition, a tree-structure based method allows to better describe the adjacency properties among the joints. This method is further improved by a gating mechanism to handle noise and occlusion.

3.2.2 2D facial expression recognition

The task of facial expression recognition (FER) consists in recognizing the basic emotions, *e.g.*, fear, surprise, happiness, etc. Recall that facial landmarks are located in certain regions of the face such as the mouth, the eyes, eyebrows, etc. As stated in [23], the motion of these landmarks defining facial expressions allows to characterize the emotional state of humans. Therefore, representing and classifying sequences of facial landmarks has

been widely used in the literature of FER.

Falling in the category of graph-based approaches, a geometric approach was proposed in [116] which introduced a unified probabilistic framework based on an interval temporal Bayesian network (ITBN) built from the movements of landmark points. Aware of the small variations along a facial expression, the authors in [53] proposed a method to capture the subtle motions within facial expressions using a variant of Conditional Random Fields (CRFs) called Latent-Dynamic CRFs (LDCRFs) on geometric features.

Taking the direction of Riemannian methods, Taheri *et al.* [102] proposed to represent 2D facial sequences as parameterized trajectories on the Grassmann manifold of 2-dimensional subspaces in $\mathbb{R}^n$ (n is the number of landmarks) which is an affine-invariant shape representation. To capture the facial deformations, they used geodesic velocities between facial shapes and finally, classification was performed by applying LDA then SVM. In the work of Kacem *et al.* [59], 2D facial landmark sequences were first represented as trajectories of Gram matrices in the manifold of positive semidefinite matrices of rank 2. A similarity measure is then provided by temporally aligning trajectories while taking into account the geometry of the manifold. This measure is finally used to train a pairwise proximity function SVM.

More recent approaches exploited deep neural networks. In [57], two neural network architectures were proposed for image videos (DTAN) and 2D facial landmark sequences (DTGN) which are combined (forming DTAGN) to predict final emotions. In particular, DTGN showed to be efficient by using only 2D landmark sequences, when applied separately. Another approach is proposed in [123] where a Part-based Hierarchical Bidirectional Recurrent Neural Network (PHRNN) is responsible for analyzing the temporal information of facial landmark sequences and a Multi-Signal Convolutional Neural Network (MSCNN) is designed to extract spatial features from still frames. These two networks are combined to boost the performance of facial expression recognition.

Although the macro facial expression recognition problem has seen considerable advances, micro-expression recognition is still a relatively challenging task [83]. Micro-expressions are brief facial movements characterized by short duration, involuntariness and subtle intensity. In the literature, previous methods opted for extracting hand-crafted features from texture videos such as LBP-TOP and HOOF [128].

More recently, deep learning methods were proposed to tackle the problem by applying CNNs [15, 63] and RNNs [63]. To our knowledge, only the method of [21] is entirely based on analyzing 2D facial landmark sequences. Their work is based on computing the point-wise distances between adjacent landmark configurations along a sequence which is stacked in a matrix. The latter was seen as an input image to a CNN-LSTM-based classifier. However, their approach was only evaluated on a synthesized dataset produced from a macro-expression dataset. In our work, we will show that we achieve state-of-the-art results on a commonly-used micro-expression dataset using only 2D landmark data.

3.3 Temporal modeling and classification

Let $\{\bar{Z}_1, \bar{Z}_2, ..., \bar{Z}_L\}$ be a sequence of skeletons representing a trajectory on $\mathcal{S}$. As described previously, we code each skeleton $\bar{Z}_i$ into a sparse vector $w_i \in \mathbb{R}^N$ with respect to a dictionary $\mathcal{D}$. As a consequence, each trajectory is mapped to an N-dimensional function of sparse codes and the problem of classifying trajectories on $\mathcal{S}$ is turned to classifying N-dimensional sparse codes functions in Euclidean space, where standard tools designed for Euclidean time-series (*e.g.*, temporal modeling, machine learning techniques, etc.) could be directly applied. We adopt and evaluate two temporal modeling and classification schemes to recognize actions and facial expressions.

3.3.1 Dynamic time warping, Fourier pyramid and SVM

A non-trivial challenge in recognizing actions and facial expressions resides in their rate variability since they can be executed at different speed. Thus, a typical landmark sequence representation has to be invariant to the execution rate. In other words, sequences belonging to the same class should typically have similar parametrization. A commonly used technique, namely Dynamic Time Warping (DTW) can temporally align one sequence into another by finding the optimal re-parameterization that minimizes a similarity measure between the two. In our work, we exploit the vector structure of our latent space to directly apply DTW on the sequential data, avoiding the more difficult task of temporal alignment of trajectories on the manifold. Another important post-processing is to further filter out noise present in the data. For instance, Fourier Temporal Pyramid (FTP) [113] have shown to be very effective for recognition tasks involving noisy data as it maps a time-series to the Fourier domain and eliminates the high frequency elements. Moreover, FTP is known to be robust to temporal mis-alignment. In our work, we apply a pipeline of DTW and FTP, then classify final features using a one-vs-all linear SVM. By doing so, we handle rate variability, temporal misalignment and noise, and classify final features, respectively.

It is important to note that to be able to apply the FTP approach, the pose sequences should have the same temporal length. For that, we need to apply a temporal re-sampling to all sequences which can be achieved in two alternatives:

— After projecting the input sequences to the Kendall's shape space, we can apply a Riemannian re-sampling algorithm as described in Algorithm 5. This alternative is better applied when the dataset contains long sequences. Thus, one can perform down-sampling of trajectories which will reduce the computational cost of the sparse coding step.

— As explained in Section 2.5.4 of Chapter 2, one can exploit the vector structure of the latent space to apply algorithms designed for Euclidean time-series which are in

most cases faster than their nonlinear equivalent. Thus, one can apply SCDL on the input trajectories, then perform a Euclidean temporal re-sampling on the obtained sequences of sparse codes.

Temporal re-sampling of trajectories Temporal up-sampling allows to increase time-steps in a sequence for more accuracy, while temporal down-sampling decreases them to reduce the computation complexity of algorithms as an example. Besides, one may need to have a set of sequences with the same length by applying up or down sampling to each sequence. This can be achieved in the shape manifold by applying the algorithm described in Algorithm. 5.

Algorithm 5: Re-sampling of trajectories on $\mathcal{S}$

input : Given a trajectory $\alpha(t)_{t=t_1,...,t_n}$, we seek $\alpha(s)_{s=s_1,...,s_l}$ where $l < n$ for down-sampling and $l > n$ for up-sampling.

for $i = 1$ *to* l **do**

Find t_{i_1}, t_{i_2} such that $t_{i_1} \leq s_i \leq t_{i_2}$

Compute $w_1 = \frac{s_i - t_{i_1}}{t_{i_2} - t_{i_1}}$ and $w_2 = \frac{t_{i_2} - s_i}{t_{i_2} - t_{i_1}}$

$x = \alpha(i_1), y = \alpha(i_2), \theta = d_{\mathcal{S}}(x, y)$ then

$\alpha(s_i) = \frac{1}{sin(\theta)}(sin(w_2\theta)x + sin(w_1\theta)y)$

return Re-sampled trajectory

3.3.2 Long short-term memory network

Modeling sequential data using recurrent neural networks (RNNs) has been widely used in different computer vision tasks and has led to breakthrough results in natural language processing [99], speech recognition [36], etc. RNNs are a kind of neural networks that take sequential input and infer sequential output by sharing parameters between time steps. They are trained using back-propagation over time. However, standard RNNs lack the ability of learning long-term dependencies as they suffer from the problem of vanishing gradient [46]. Tackling this problem, Long short-term memory (LSTM) network [47] is equipped with a gating mechanism that learns which information is relevant to keep or forget during training.

Thereby they better handle the problem of learning long-term dependencies in sequential data. In the context of action recognition, many works in the literature opted for LSTMs to model and classify actions [73, 93, 124]. In [93], the authors propose a part-aware LSTM where they divide a skeleton configuration into body parts, hence instead of keeping a long-term memory of the entire body's motion in the cell, they split it to part-based cells. Another work [124] tackles the problem of view variations in the captured actions, similarly to our work but instead of seeking a view-invariant representation then use it for classification, they proposed a view-adaptive LSTM-based network that automatically regulates observation viewpoints during the occurrence of an action. In our work, we aim to take advantage of the view-invariant nature of our representation as well as its compactness (sparsity) and vector space structure to apply an LSTM directly on the SCDL time-series. Further, we explore the use of Bi-directional LSTM (Bi-LSTM). Bi-LSTM is an extension of LSTM that presents each sequence backwards and forwards to two separate recurrent networks, providing context both from the future and past, respectively [37]. We will experimentally show that Bi-LSTM can achieve slight improvements over the traditional LSTM in recognizing human motion.

3.3.3 Dictionary structure

In the context of classification, one may exploit the important information of data labels to construct more discriminative feature vectors. To this end, we propose to build *class-specific* dictionaries, similarly to [38]. Formally, let S be a set of labeled sequences on $\mathcal{S}$ belonging to q different classes $\{c_1, c_2, ..., c_q\}$, we aim to build q class-specific dictionaries $\{D_1, D_2, ..., D_q\}$ in $\mathcal{S}$ such that each D_j is learned using skeletons belonging to training sequences from the corresponding class c_j. In this scenario, coding a query shape $\bar{Z} \in \mathcal{S}$ is done with respect to each $D_{j,1\leq j\leq q}$, independently. As a result, q vectors of codes are obtained. These vectors are then concatenated to form a global feature vector W. As will be discussed in Section 3.4.3, this yields more discriminative feature vectors for classification.

3.4 Experimental evaluation

We perform extensive experiments to evaluate the effectiveness of the proposed frameworks in the tasks of: 3D action recognition, 2D macro- and micro- facial expression recognition. We provide comparisons to some state-of-the-art approaches on several publicly available datasets in addition to comparisons between the intrinsic and extrinsic paradigms of the proposed SCDL approach. Moreover, we perform baseline experiments to evaluate some properties of our recognition frameworks.

3.4.1 3D action recognition

3.4.1.1 Datasets

We evaluate the proposed skeletal representation using four benchmark datasets presenting different challenges: Florence3D-Action [92], UTKinect-Action [117], MSR-Action 3D [69], and the large-scale NTU-RGBD dataset [94].

Florence3D-Action dataset consists of 9 actions performed by 10 subjects. Each subject performed every action two or three times for a total of 215 action sequences. The 3D locations of 15 joints collected using the Kinect sensor are provided. The challenges of this dataset consist of the similarity between some actions and also the high intra-class variations as same action can be performed using left or right hand.

UTKinect-Action dataset consists of 10 actions performed twice by 10 different subjects for a total of 199 action sequences. The 3D locations of 20 different joints captured with a stationary Kinect sensor are provided. The main challenge of this dataset is the variations in the view point.

MSR-Action 3D dataset consists of 20 actions performed by 10 different subjects. Each subject performed every action two or three times for a total of 557 sequences. The 3D locations of 20 different joints captured with a depth sensor similar to Kinect are provided

with the dataset. This is a challenging dataset because of the high similarity between many actions (*e.g.*, *hammer* and *hand catch*).

NTU-RGB+D is one of the largest 3D human action recognition datasets. It consists of 56, 000 action clips of 60 classes. 40 participants have been asked to perform these actions in a constrained lab environment, with three camera views recorded simultaneously. Each Kinect sensor estimates and records 25 joints coordinates reported in the 3D camera's coordinate system.

3.4.1.2 Experimental settings

For the first three datasets, we followed the cross-subject test setting of [112], in which half of the subjects was used for training and the remaining half was used for testing. Reported results were averaged over ten different combinations of training and test data. For Florence3D-Action and UTKinect-Action datasets, we followed an additional setting for each: Leave-one-actor-out (LOAO) [92, 111] and Leave-one-sequence-out (LOSO) [117], respectively. For MSR-Action3D dataset, we also followed [69] and divided the dataset into three subsets AS1, AS2, and AS3, each consisting of 8 actions, and performed recognition on each subset separately, following the cross-subject test setting of [112]. The subsets AS1 and AS2 were intended to group actions with similar movements, while AS3 was intended to group complex actions together. In all experiments, we performed recognition based on the two classification schemes: Bi-LSTM and DTW-FTP-SVM.

For NTU-RGB+D, the authors of this dataset recommended two experimental settings that we follow: 1) Cross-subject (X-Sub) benchmark with 39,889 clips from 20 subjects for training and 16,390 from the remaining subjects for testing; 2) Cross-view (X-View) benchmark with 37,462 and 18,817 clips for training and testing. Training clips in this setting come from the camera views 2 and 3 while the testing clips are all from the camera view 1. Due to the huge amount of data in this dataset, we construct dictionaries using the

kernel clustering approach presented in Section 2.4.1.2 since it is less time consuming than the dictionary learning optimization problem. Note that NTU-RGBD dataset contains two types of actions: daily activities where performed by one actor and interactions between two actors. For the latter, we perform sparse coding of each actor's skeleton separately. Further, since the closeness between the two actors is a relevant information, we compute the Euclidean distance between their center of mass and concatenate it to the feature vector obtained after sparse coding. Moreover, we compute displacement vectors, as described in Section 3.4.2.2 for micro-expressions, and fuse them with the rest of the features. Finally, we perform temporal modeling and classification using Bi-LSTM. For this dataset, we do not suggest using the first classification pipeline as it would be highly consuming in terms of computation of DTW and FTP due to the huge amount of data that NTU-RGBD contains.

3.4.1.3 Results and discussions

TABLE 3.1 – Overall recognition accuracy (%) on MSR-Action 3D, Florence Action 3D, and UTKinect 3D datasets. In the first column: $^{(R)}$: Riemannian approaches; $^{(N)}$: other recent approaches; Last row: our approach.

Dataset	**MSR-Action 3D**		**Florence 3D**		**UTKinect 3D**	
Protocol	**H-H**	**3 Subsets**	**H-H**	**LOAO**	**H-H**	**LOSO**
$^{(R)}$ T-SRVF on Lie group [4]	85.16	–	89.67	–	94.87	–
$^{(R)}$ T-SRVF on $\mathcal{S}$ [9]	89.9	–	–	–	–	–
$^{(R)}$ Lie Group [107]	89.48	92.46	90.8	–	97.08	–
$^{(R)}$ Rolling rotations [108]	–	–	91.4	–	–	–
$^{(R)}$ Gram matrix [58]	–	–	–	88.85	–	**98.49**
$^{(N)}$ Graph-based [114]	–	–	–	91.63	97.44	–
$^{(N)}$ ST-LSTM [73]	–	–	–	–	95.0	97.0
$^{(N)}$ JLd+RNN [126]	–	–	–	–	95.96	–
$^{(N)}$ SCK+DCK [66]	**91.45**	93.96	**95.23**	–	**98.2**	–
$^{(N)}$ Transition-Forest [34]	–	**94.57**	–	94.16	–	–
$^{(R)}$ **Ours** (SVM)	**90.01**	**94.19**	92.85	92.27	**97.39**	97.50
$^{(R)}$ **Ours** (Bi-LSTM)	86.18	86.18	**93.04**	**94.48**	96.89	**98.49**

Comparison to Riemannian methods on MSR-Action, Florence3D and UTKinect datasets The first row of methods in Table 3.1 reports the recognition results of different Riemannian approaches. Since in [9] human actions are also represented as trajectories in the Kendall's shape space, we report additional results of [9] on Florence3D and UTKinect datasets to give more insights about the strength of our coding approach compared to the method of [9]. In Table 3.1, it can be seen that we obtain better results than all Riemannian approaches on the three datasets. We recall that one common drawback of these methods is to map trajectories on manifolds to a reference tangent space, where they compute distances between different points (other than the tangent point). This may introduce distortions, especially when points are not close to the reference point. However, our method avoids such a non-trivial problem as coding of each shape is performed on its attached tangent space and the only measures that we compute are with respect to the tangent point. Now, we discuss our results obtained with the first classification scheme, *i.e.*, DTW-FTP-SVM, similarly used in [4, 107, 108]. In the three datasets, it is clearly seen that our approach outperforms existing approaches when using the same classification pipeline, which shows the effectiveness of our skeletal representation. For instance, we highlight an improvement of 1.73% on MSR-Action 3D (following protocol [69]) and 1.45% on Florence3D-Action. Now, we discuss the results we obtained using Bi-LSTM. Note that although we do not perform any preprocessing on the sequences of codes before applying Bi-LSTM, our approach still outperforms existing approaches on Florence3D, with 1.64% higher accuracy. However, it performs less well on UTKinect yielding an average accuracy of 96.89% against 97.08% obtained in [107]. In MSR-Action 3D, our approach performs better than the method of [4] using the first protocol. Note that in [4], results were averaged over all 242 possible combinations. However, our average accuracy is lower than other approaches following both protocols on this dataset (around 3.5% in the first and 0.62% in the second). Here, it is important to mention that data provided in MSR-Action 3D are noisy [75]. As a consequence, using Bi-LSTM without any additional processing step to handle the noise (*e.g.*, FTP) could

not achieve state-of-the-art results on this dataset.

Comparison to State-of-the-art We discuss our results with respect to recent non-Riemannian approaches. In all datasets, our approach achieved competitive results.

Florence3D-Action – On this dataset, our method outperforms other methods using Bi-LSTM in the case of LOAO protocol, as shown in Table 3.1. However, using the second protocol, it is 2.19% lower than [66]. The authors of [66] combine two kernel representations: sequence compatibility kernel (SCK) and dynamics compatibility kernel (DCK) which separately achieved 92.98% and 92.77%, respectively. The proposed approach achieves good performance for most of the actions. However, the main confusions concern very similar actions, *e.g.*, *Drink from a bottle* and *answer phone*, as demonstrated by the confusion matrix in Figure 3.4.

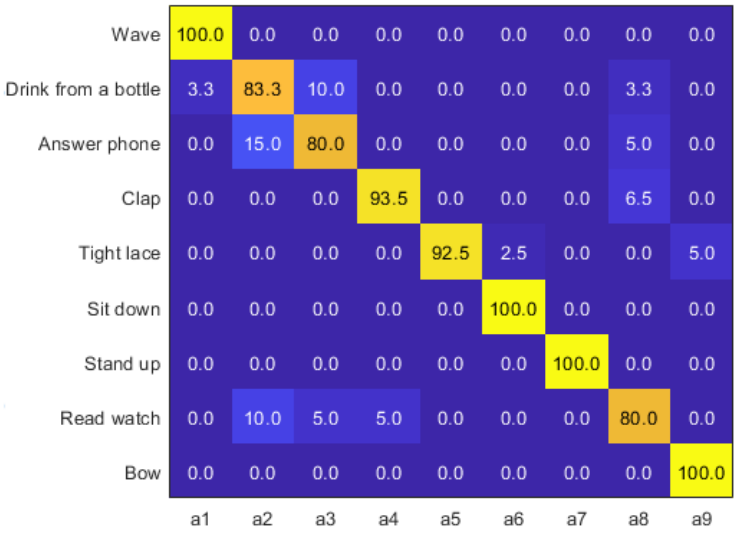


FIGURE 3.4 – Confusion matrix on the Florence 3D dataset.

UTKinect – Following the LOSO setting, our approach achieves the best recognition rate,

yielding an improvement of 2.49% compared to the method of [73], which is based on an extended version of LSTM. For the second protocol, our best result is competitive to the accuracy of 98.2% obtained in [66]. Considering the main challenge of this dataset, *i.e.*, variations in the view point, our approach confirms the importance of the invariance properties gained by adopting the Kendall's representation of shape, hence the relevance of the resulting functions of codes generated using the geometry of the manifold.

MSR-Action 3D – For the experimental setting of [69], our best result is competitive to recent approaches. In particular, on AS3, we report the highest accuracy of 100%. This result shows the efficiency of our approach in recognizing complex actions, as AS3 was intended to group complex actions together. On AS1, we achieved one of the highest accuracies (95.87%). However, our result on AS2 is about 8.9% lower than state-of-the-art best result. This shows that our approach performs less well when recognizing similar actions, as AS2 was intended to group similar actions together. Although our best result is slightly higher than [66], it is lower than the same method when following the experimental setting of [113]. This shows that our approach performs better in recognition problems with less classes.

TABLE 3.2 – Overall recognition accuracy (%) on NTU-RGB+D following the X-sub and X-view protocols. In the first column: $^{(R)}$: Riemannian approaches; $^{(RN)}$: RNN-based approaches; $^{(CN)}$: CNN-based approaches.

Protocol	X-sub	X-view
$^{(R)}$ Lie Group [106]	50.1	52.8
HB-RNN-L [29]	59.1	64.0
$^{(R)}$ Deep learing on $SO(3)^n$ [51]	61.3	66.9
$^{(RN)}$Deep LSTM [93]	60.7	67.3
$^{(RN)}$Part aware-LSTM [93]	62.9	70.3
$^{(RN)}$ST-LSTM+Trust Gate [73]	69.2	77.7
$^{(RN)}$View Adaptive LSTM [124]	79.4	87.6
$^{(CN)}$Temporal Conv [64]	74.3	83.1
$^{(CN)}$C-CNN+MTLN [61]	79.6	84.8
$^{(CN)}$ST-GCN [119]	**81.5**	**88.3**
$^{(R)}$ **Intrinsic SCDL**	**73.89**	**82.95**

NTU-RGB+D – We report the obtained results for this dataset in Table. 3.2. For both

benchmarks, X-view and X-sub, our approach remarkably outperforms other Riemannian representations. For instance, it outperforms the Lie group representation by 23% and 30% on X-sub and X-view protocols. It also surpasses the deep learning on Lie groups method by 12% and 16%. This could demonstrate the ability of our approach to deal with large scale datasets compared to conventional Riemannian approaches. Besides, our method outperforms RNN-based models, HB-RNN-L, Deep LSTM, PA LSTM and ST-LSTM+TG, with the exception of [124]. Knowing that we also used an RNN-based model (Bi-LSTM) for temporal modeling and classification, this shows the efficiency of our action modeling. In fact, sparse features obtained after SCDL in Kendall's shape space are remarkably more discriminative than the original data. In order to have a better insight into their corresponding data distributions, we used the t-distributed stochastic neighbor embedding (t-SNE) [1] to visualize original data and SCDL features. From Fig. 3.5, we can observe that the SCDL features are better clustered than the original data in terms of class labels (colors in the figure). Besides, it is worth noting that SCDL is an efficient denoising tool, which is an important advantage when dealing with the often-noisy skeletons extracted with the Kinect sensor.

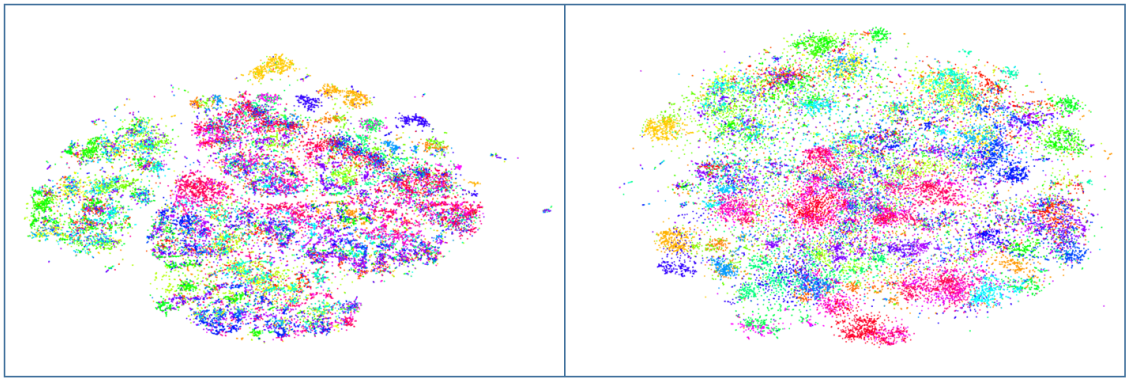


FIGURE 3.5 – Visualization of 2-dimensional features of the NTU-RGB+D dataset. Left: original data. Right: the corresponding SCDL features. Each class is represented by a different color.

1. t-SNE is a nonlinear dimensionality reduction technique that allows for embedding high-dimensional data into two or three dimensional space, which can then be visualized in a scatter plot.

3.4.1.4 Comparison to extrinsic SCDL

To further evaluate the strength of the proposed intrinsic approach in the context of 3D action recognition, we compare it to extrinsic SCDL. Recall that instead of coding on tangent spaces, the extrinsic approach tends to embed the manifold-valued data into Hilbert spaces which are higher dimensional vector spaces where linear coding becomes possible. The main difficulty here arises from the fact that this embedding relies on a kernel function which, according to Mercer's theorem, should be positive definite. For 2D shapes, we presented in Section 2.2.3.2 a positive definite kernel. However, to the best of our knowledge, the existence of such a kernel has not been proved in the literature for 3D shapes. As a remedy, we adapted the extrinsic SCDL formulation by applying the Procrustes Gaussian kernel defined in Section 2.2.3.2, in which we also adapted the full Procrustes distance to 3D shapes as $d_{FP}(\bar{Z}_1, \bar{Z}_2) = sin(\theta)$ (see Section 4.2.1 of [27]) (θ is the geodesic distance defined in Section 2.2.3). Note that the kernel function relies on a parameter σ. Experimentally, we checked the positive definiteness of the adapted kernel and found out that it is only positive definite for some values of σ. We empirically chose 0.1 for Florence3D, 0.2 for UTKinect, and 0.5 for MSR-Action 3D, as to have valid positive definite kernels. Results reported in Table 3.3 show superiority of the intrinsic method. We argue that this difference comes from the fact that for the 3D case, the positive definiteness constraint on the kernel function reduced the valid space of the kernel parameter σ. Hence, intrinsic SCDL is in this case a better coding solution. In contrast, for the 2D case where we possess a PD kernel, we will show in the next section that extrinsic SCDL is more efficient in 2D recognition tasks.

TABLE 3.3 – Comparative evaluation of intrinsic and extrinsic SCDL in recognizing 3D actions.

Dataset	**MSR-Action 3D**		**Florence 3D**		**UTKinect 3D**	
Protocol	H-H	3 Subsets	H-H	LOAO	H-H	LOSO
Extrinsic SCDL	82.52	88.53	85.76	89.03	93.97	94.97
Intrinsic SCDL	**90.01**	**94.19**	**92.85**	**92.27**	**97.39**	**97.50**

3.4.2 2D Facial Expression Recognition

In this application, we extract 49 facial landmarks from human faces in 2D and with high accuracy using a state-of-the-art facial landmark detector [7]. We first represent the sequences of landmarks as trajectories in the Kendall's shape space. Extrinsic SCDL is then applied to produce sparse time-series that are finally classified in vector space. We evaluate this approach on two different 2D facial expression recognition tasks, the macro and micro.

3.4.2.1 Macro-Expression Recognition

The task here is to recognize the basic macro emotions, *e.g.*, fear, surprise, happiness, etc. To this end, we applied our approach on two commonly-used datasets namely the Cohn-Kanade Extended dataset and the Oulu-CASIA dataset. Our obtained results are then discussed with respect to state-of-the-art approaches as well as to intrinsic SCDL. For both datasets, we followed the commonly-used experimental setting in [31, 57, 74, 129] consisting on a ten-fold cross validation.

Cohn-Kanade Extended (CK+) dataset [76] consists of 327 image sequences performed by 118 subjects with seven emotion labels: *anger*, *contempt*, *disgust*, *fear*, *happiness*, *sadness*, and *surprise*. Each sequence contains the two first temporal phases of the expression, *i.e.*, neutral and onset (with apex frames).

Oulu-CASIA dataset [105] includes 480 image sequences performed by 80 subjects. They are labeled with one of the six basic emotions (those in CK+, except the *contempt*). Each sequence begins with a neutral facial expression and ends with the expression apex.

Results and discussions Table 3.4 gives an overview of the obtained results on both datasets. Overall, our approach achieved competitive results compared to the literature. For instance, our best result on CK+ (obtained with Bi-LSTM) is by 1.52% lower than the best state-of-the-art result obtained by the method of [57]. The latter is based on two neural

network architectures trained on image videos and facial landmark sequences. However, when using only the landmark architecture (DTGN), our approach obtained a higher accuracy. Similarly, on Oulu-CASIA, our best result is lower than DTAGN and higher than DTGN. On the other hand, the method of [59] achieved a better performance on both datasets compared to our method. Comparing the confusion matrices (see Table 3.5), the same method seems to better recognize the *sadness* expression while our method is clearly more efficient in recognizing the *contempt* expression. This will be further discussed later on. From Fig. 3.6 and the confusion matrices in Tables 3.6 and 3.5, we can observe that the two expressions: *happiness* and *surprise* are well recognized in the two datasets while the main confusions happened in the two expressions: *fear* and *sadness*, conforming to state-of-the-art results [57, 59]. Besides, we highlight the superiority of extrinsic SCDL compared to intrinsic SCDL. The first is performed in RKHS which is a higher dimensional vector space. This helps capturing complex patterns in facial expressions and identifying subtle differences between similar expressions. For instance, an interesting observation could be seen for the *contempt* expression. As stated in [76], the latter is quite subtle and it gets easily confused with other, strong emotions. For this expression, the recognition accuracy obtained with intrinsic SCDL is 55%, compared to 90% obtained with extrinsic SCDL, as shown in Figure 3.6. We argue that this remarkable improvement comes from the mapping to RKHS for the same reasons mentioned above. This observation has pushed us to further evaluate the performance of our approach in the task of micro-expression recognition.

3.4.2.2 Micro-Expression Recognition

Micro expressions are brief facial movements characterized by short duration, involuntariness and subtle intensity. We argue that to recognize them, in contrast to macro-expressions, we are more interested in detecting subtle shape changes along a sequence. To this end, we applied the extrinsic SCDL framework as in macro-expression recognition, and to further detect the subtle deformations, we computed displacement vectors as the difference between

TABLE 3.4 – Comparison with state-of-the-art on CK+ and Oulu-CASIA datasets. $^{(A)}$: Appearance-based approaches; $^{(G)}$: Geometric approaches; $^{(R)}$: Riemannian approaches; Last row: our approach.

Method	CK+	Oulu-CASIA
$^{(A)}$ CSPL [129]	89.89	–
$^{(A)}$ ST-RBM [31]	95.66	–
$^{(A)}$ STM-ExpLet [74]	94.19	74.59
$^{(G)}$ ITBN [116]	86.30	–
$^{(G)}$ DTGN [57]	92.35	74.17
$^{(A+G)}$ DTAGN [57]	**97.25**	81.46
$^{(R)}$Shape velocity on Grassmannian [102]	82.80	–
$^{(R)}$Shape traj. on Grassmannian [59]	94.25	**80.0**
$^{(R)}$Gram matrix trajectories [59]	**96.87**	**83.13**
$^{(R)}$**Intrinsic SCDL** (SVM)	91.26	70.37
$^{(R)}$**Intrinsic SCDL** (Bi-LSTM)	89.43	70.24
$^{(R)}$**Extrinsic SCDL** (SVM)	95.62	**77.06**
$^{(R)}$**Extrinsic SCDL** (Bi-LSTM)	**95.73**	73.09

TABLE 3.5 – Confusion matrix on the CK+ dataset.

	An	**Co**	**Di**	**Fe**	**Ha**	**Sa**	**Su**
An	**97.50**	0	2.5	0	0	0	0
Co	10	**90.0**	0	0	0	0	0
Di	2.5	0	**95.83**	0	0	1.67	0
Fe	0	0	0	**91.67**	8.33	0	0
Ha	0	0	0	0	**97.5**	2.5	0
Sa	5.0	0	5.0	2.5	1.67	**85.83**	0
Su	0	1.11	0	0	0	0	**98.89**

successive sparse codes of L-dimensional time-series. Then, the resulting sequences of length $L-1$ are finally used for classification. We evaluate our approach on the most commonly-used dataset, namely CASME II.

CASME II dataset [120] contains 246 spontaneous micro-expression video clips recorded from 26 subjects and regrouped into five classes: happiness, surprise, disgust, repression and others. We performed classification based on the commonly used Leave-one-subject-out protocol.

Recall that previous methods that tackled the problem of micro-expression recognition are appearance-based (*i.e.*, using texture images) and to our knowledge, only [21] has studied

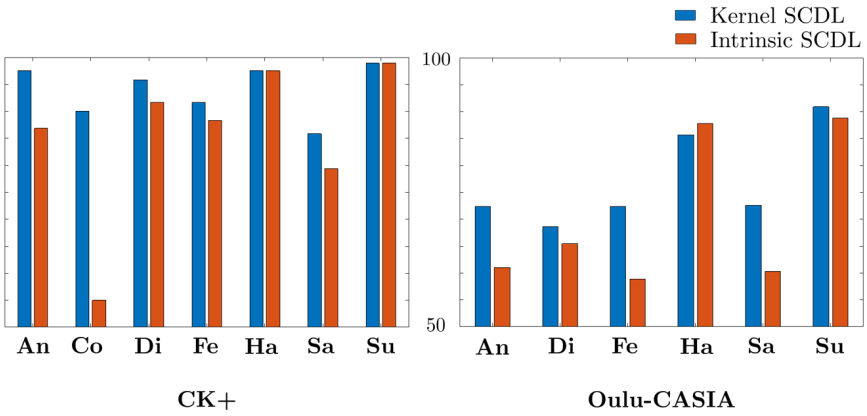


FIGURE 3.6 – Recognition accuracy achieved for each emotion class in the CK+ (left) and the CASIA (rigth) datasets, and comparison between extrinsic and intrinsic SCDL approaches.

TABLE 3.6 – Confusion matrix on the Oulu-Casia dataset.

	An	**Di**	**Fe**	**Ha**	**Sa**	**Su**
An	**72.33**	12.33	2.11	1.0	12.22	0
Di	14.22	**68.56**	6.22	3.0	8.0	0
Fe	5.22	2.0	**72.33**	5.11	9.33	6.0
Ha	4.0	0	9.33	**85.67**	1.0	0
Sa	15.22	4.11	6.11	2.0	**72.56**	0
Su	0	2.1	5.0	0	2.0	**90.89**

the problem using 2D facial landmarks. However, their approach was only evaluated on a synthesized dataset produced from CK+ (macro) videos, by selecting the three first frames of an expression, then interpolating between them. For this reason, we compare our results with respect to appearance-based methods, as shown in Table 3.7.

We point out the recognition accuracy of 64.62% achieved by our method outperforming state-of-the-art approaches, with the exception of [50]. This shows the effectiveness of the adopted extrinsic SCDL in detecting subtle deformations from 2D landmarks, without any apperance-based information as other approaches in the literature.

Compared to the intrinsic approach, it is clear from Table 3.7 that the extrinsic SCDL method is better in recognizing micro-expressions. Recall that the use of extrinsic SCDL to tackle the problem of micro-expression recognition was driven by its good performance

TABLE 3.7 – Recognition accuracy on CASME II dataset and comparison with state-of-the-art methods. In the first column: $^{(A)}$: Appearance-based approaches. $^{(R)}$: Riemannian approaches. Last row: our approach.

Method	Accuracy (%)
$^{(A)}$ STCLQP[50]	58.39
$^{(A)}$ CNN [15]	59.47
$^{(A)}$ CNN (LSTM) [63]	60.98
$^{(A)}$ LBP-TOP, HOOF [128]	63.25
$^{(A)}$ Optical Strain [70]	63.41
$^{(A)}$ DiSTLBP-IIP [50]	**64.78**
$^{(R)}$**Intrinsic SCDL** (SVM)	43.65
$^{(R)}$**Extrinsic SCDL** (SVM)	**64.62**

in recognizing the contempt emotion, in the CK+ dataset which is characterized by subtle changes along the expression. The obtained results on CASME II hence supports our previous claims.

3.4.3 Ablation study

We examine the effectiveness of the proposed Kendall SCDL schemes by performing several baseline experiments on different datasets.

A. Kendall's shape representation – We evaluate the necessity of the Kendall's shape projection. To this end, we perform temporal modeling and classification on raw data, after a scale and translation normalization, against their application on Kendall SCDL features. On NTU-RGB+D, we applied Bi-LSTM while on Florence 3D, MSR Action 3D and UTKinect, we applied the pipeline DTW-FTP-SVM. Performances are reported in the second and fourth rows of Table 3.8. In all datasets, improvements are remarkably gained with the Kendall's space projection. This is clearly seen in particular on the large scale NTU-RBD+D dataset which presents different view-variations and where the improvement is more than 26%.

B. Nonlinear SCDL – In this experiment, we evaluate the importance of the nonlinear formulation of SCDL that we applied on Kendall's space. For that, we compare it to the use of linear SCDL, i.e., by solving for Eq.2.2.2. Obtained results on the four action recognition

TABLE 3.8 – Evaluation of the Kendall's shape space representation.

Approach	NTU-RGB+D	Florence	MSR 3D	UTKinect
Raw data	56.5	84.29	87.36	92.67
Linear SCDL	79.20	87.94	89.23	93.58
Kendall SCDL	**82.95**	**92.85**	**90.01**	**97.5**

datasets, reported in the third row of Table 3.8, clearly show the interest of accounting for the nonlinearity of the manifold when applying SCDL.

C. Sparsity regularization – In this experiment, we evaluate the effect of the sparsity regularization parameter λ (in Eq. (2.4.1) and Eq. (2.4.4)) on recognition accuracies obtained using both of the adopted classifiers. To do so, we used half of a training set for learning the dictionary and training the classifiers and the other half for validation. The first graph of Fig. 3.7 shows the impact of increasing λ from 10^{-4} to 1 at steps of 10^{-2}. Further, we report the average sparsity percentage (*i.e.*, number of non-zero codes divided by the total number of codes) for some values of λ to show the coherence of the obtained codes with the proposed theory. As expected, the sparsity percentage increases when increasing λ. We remark that the accuracy reached a maximum value at $\lambda = 0.01$ (37% of sparsity) and $\lambda = 0.02$ (49% of sparsity) for SVM and Bi-LSTM, respectively. Note that in all previous experiments, λ was chosen empirically so to correspond to these latter percentages of sparsity.

D. Dictionary structure – As described in Section 3.3.3, we build class-specific dictionaries. To show the relevance of this structure in the context of classification, we compare it to the case of using a global dictionary, *e.g.*, when label(s are not taken into account. The obtained recognition accuracies using Bi-LSTM and following the LOAO setting are 94.48% and 91.53% for class-specific and global dictionary, respectively. These results clearly prove that the adopted structure is better in classifying actions.

E. Dictionary initialization – In this experiment, we evaluate the performance of our proposed initialization step based on Bayesian clustering of shapes and PGA. To this end, we

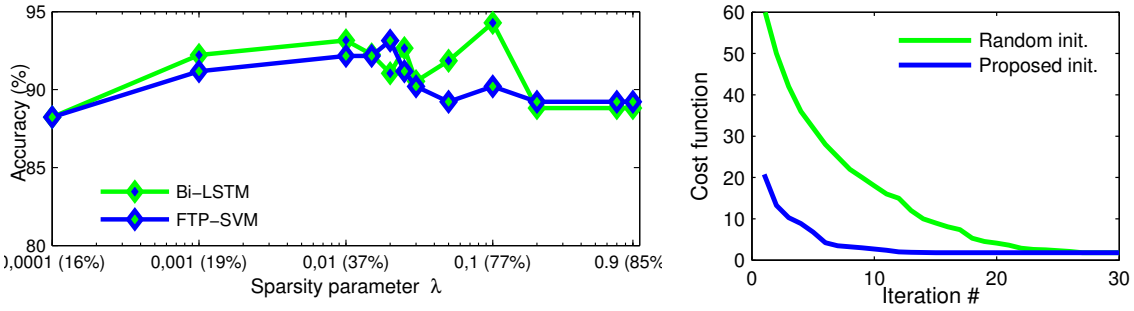


FIGURE 3.7 – Left: Accuracy when varying the sparsity regularization parameter λ (% values in the x-axis represent the average sparsity). Right: Dictionary learning objective over iterations for: (1) Random initialization; (2) Our proposed initialization based on Bayesian clustering and PGA.

.

compare it to the case of random initialization, where atoms are randomly selected from the training set. We train a class-specific dictionary (for class *tight lace* in Florence3D dataset) with the same training data in both cases. For the case of random initialization, we set the number of atoms N to 41 to be equal to that of our proposed initialization. Recall that in our approach, N is automatically inferred to avoid its empiric choice, especially as we build class-specific dictionaries. In Fig. 3.7, on the right graph, we plot the two corresponding dictionary learning objectives over iterations. As it is expected, the proposed initialization shows faster convergence, dividing the overall dictionary learning processing time by approximately two times, when taking into account the execution time of our initialization step.

F. Performance of Bi-LSTM – We compared average accuracies yielded by Bidirectional LSTM and a traditional LSTM. Following LOAO experimental setting, using Bi-LSTM shows an improvement of around 0.7% on Florence Action 3D and 1.2% on NTU-RGBD dataset, indicating the positive effect of learning both future and past contexts to recognize actions.

G. Evaluation on the facial landmark detector The task of facial expression recognition from landmark data relies essentially on the accuracy of the landmark detector. In this

experiment, we evaluate the performance of the landmark detector that we used in our experiments (*i.e.*, Chehra [7]) by comparing it to the newly-released Openface2.0 [8], which gives the option of extracting either 49 or 68 landmarks. In Table 3.9, we report the classification accuracy obtained by applying the pipeline DTW+FTP+SVM on raw landmark data (after a simple scale and translation normalization). Results obtained on CK+ and Oulu-CASIA datasets clearly show a better performance using landmarks extracted with the Chehra detector.

TABLE 3.9 – Classification performances when using different landmark detectors.

Landmark detector	Oulu-CASIA dataset	CK+ dataset
Cherha [7] - **49 landmarks**	**76.41**	**93.68**
Openface [8] - **49 landmarks**	70.85	83.73
Openface [8] - 68 landmarks	71.26	82.92

3.5 Discussions

The Kendall's shape representation has proven the efficiency of adopting a view-invariant analysis of the given data. Because of the nonlinearity of the Kendall's manifold, intrinsic and extrinsic solutions of SCDL were comprehensively studied and compared. Regarding the extrinsic solution, the advantage of embedding data from the Kendall's shape space to RKHS is twofold. First, the latter is vector space, thus it enables the extension of linear SCDL to the nonlinear Kendall's space. Second, embedding a lower dimensional space in a higher dimensional one gives a richer representation of the data and helps extracting complex patterns. However, to define a valid RKHS, the kernel function must be positive definite according to Mercer's theorem. On one hand, for the 2D Kendall's space, we have used the Procrustes Gaussian Kernel which is positive definite and shown that for the task of 2D macro facial expression recognition, extrinsic SCDL performs better than intrinsic SCDL. We argue that this is due to the kernel embedding. For instance, we highlight the clear improvement in recognizing the *contempt* emotion in the CK+ dataset. The latter is

characterized with subtle deformations that are well captured using the extrinsic approach. This has drove us to evaluate it on the task of 2D micro-expression recognition where the shape deformations along expressions are known to be subtle as well. As expected, the performance of extrinsic SCDL was promising. On the other hand, for the 3D Kendall's space, a positive definite kernel function has not been proposed in the literature. Nevertheless, adapting the PGk to 3D shapes prevented us from exploring the whole space of σ as in this case, this kernel is positive definite for only certain value of this parameter. As a consequence, the performance of extrinsic SCDL in the 3D Kendall's space can be hindered since the quality of the produced codes depends on the value of σ. We argue that this is the main reason behind the better performance obtained using intrinsic SCDL for the task of 3D action recognition. Besides, intrinsic sparse coding of a shape is performed on its attached tangent space, by mapping atoms into it. Compared to Riemannian approaches of the literature, this avoids the common drawback of mapping points to a common tangent space at a reference point which may introduce distortions.

3.6 Conclusion

In this chapter, we have used the sparse representations that we presented in the previous chapter to classify 3D actions and 2D micro and macro facial expressions. We have used two temporal modeling and classification schemes on top of the obtained sparse time-series: a deep learning framework based on Bi-LSTM and a pipeline of DTW-FTP-SVM. We have conducted extensive experiments on seven commonly-used datasets and showed that our obtained results are competitive to state-of-the-art. We have compared the two adopted temporal modeling and classification pipelines and discussed the obtained results for different datasets. Further, we presented a comprehensive comparative study on the use of intrinsic and extrinsic SCDL approaches by providing an answer to the question: "Depending on the nature of the data (*i.e* body or face) and its dimension (*i.e* 2D or 3D), when and which

technique should we apply?".

Chapitre 4

Pose-guided Human Video Generation with Sparse Representations

4.1 Motivation

Due to the emergence of Generative Adversarial Networks, the task of human video generation is attracting increasing attention in recent years as it is suitable for several vision problems. For instance, it can serve as a data augmentation tool which remarkably relieves the burden of manual annotations and thereby contributes to the development of various video understanding tasks such as action and activity recognition. On the other hand, human video synthesis allows for many human-centric applications such as avatar animation. Problems related to realistic image generation with GANs has already shown impressive results [52, 67, 82]. However, the extension from generating images to generating videos turns out to be a highly challenging task with the introduction of the temporal dimension. Considering this fact, a typical generative model needs to learn the plausible

physical motion models of objects in addition to learning their appearance models. To this end, some approaches in the literature opted for a disentangled solution by first generating plausible motion then synthesizing coherent appearances. For instance, Yang *et al.* [121] proposed a two stage approach. In the first stage, pose sequences are used to learn a generative model due to their ability to encode motion dynamics. The newly generated pose sequences are then used to guide the generation of video frames while preserving coherent appearances in the input image. Regarding the first stage, they proposed a pose sequence GAN (PSGAN) to generate skeletal sequences. Their generator transforms an input pose into a pose sequence by adopting a encoder-decoder architecture. The output of the decoder is then fed into a LSTM module for temporal pose modeling. However, their resulting sequences may contain corrupted poses which would affect the synthesis of coherent appearance in the second stage. In this chapter, we follow the same disentangled solution. In the first stage, we propose an encoder-GAN model which generates new samples in the encoder latent space which are then transformed into skeletal sequences. In contrast to previous works, it guarantees the shape and motion consistencies of the generated pose sequences while having a simpler architecture. In the second stage, given an input image, each pose of a sequence is transformed to an image, thereby constitutes the final video. The latter procedure is based on a recently proposed generative model that learns to transfer a person image to new poses [130]. The main contributions in this chapter are,

— A novel framework, namely shapeGAN, to generate dynamic human poses. Our model has a encoder-GAN architecture. The encoder aims to represent the data using SCDL in the shape manifold yielding sparse time-series that are defined in a latent space with a vector space structure. These latent variables are then used to train a GAN to generate new samples in the latent space. The generated samples are then mapped back to the shape manifold to represent the generated pose sequences.

— A pose-guided human video generation framework, namely shapeVGAN, which exploits the shapeGAN model to generate novel pose dynamics, then uses a pose transfer

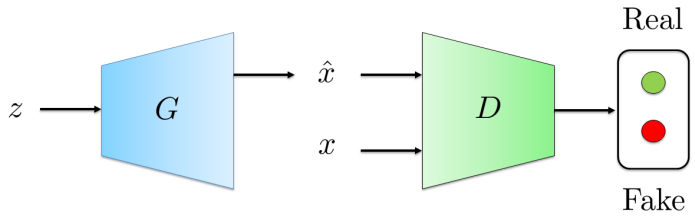


FIGURE 4.1 – Schema of the original GAN framework. The generator G takes a noise vector z as input and outputs generated images $\hat{x}$. The discriminator D distinguishes $\hat{x}$ from real samples x.

model [130] to map each generated pose to an image. We differentiate our work as an extension of [130] to video generation, while our main novelty consists on the shapeGAN framework.

— Application of the proposed frameworks to: data augmentation for landmark-based action recognition and human video generation.

4.2 Background and related work

4.2.1 Generative Adversarial Networks

In recent years, GANs [35] have led to breakthroughs in tasks related to realistic image generation. Examples include image synthesis [82] which infers multiple possible outputs for a given input, super-resolution [67], and image to image translation [52]. The original GAN framework, as demonstrated in Figure 4.1 consists of a generator and a discriminator, two neural networks that are trained against each other. The generator G aims to generate samples that resemble to real ones, while the discriminator D tries to distinguish whether a sample comes from the training data (*i.e.* real samples) or from G. In other words, G captures the data distribution by taking a latent variable z sampled from a prior distribution $p_z(z)$ (*e.g.* a Gaussian distribution), and generates a fake image $\hat{x} = G(z)$. $\hat{x}$ can be seen as a sample from the learned generative distribution p_G. D discriminates between $\hat{x}$ (as fake data) and x sampled from real data distribution $p_{data}(x)$ (as real data). Accordingly, the

training of G and D is similar to a two-player minimax game where D is trained to maximize the probability of assigning the correct label to both real samples and generated samples, and simultaneously, G is trained to minimize $\log(1 - D(G(z)))$, according to the following objective function:

$$\min_G \max_D \mathcal{L}_a(G, D) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))], \tag{4.2.1}$$

where $\mathcal{L}_a(G, D)$ denotes the adversarial loss for G and D, and $D(x)$ denotes a probability that x is real. Note that at early training stage, when G only generates poor images, D tends to assign them low probability, which makes $\log(1-D(G(z)))$ saturate and unable to provide sufficient gradient to update G. A standard way in practice is to maximize $\log(D(G(z))$ for better training G.

Training GANs In practice, training the original GANs can be remarkably challenging and unstable. Empirically, there are three most common failures occurred within the training process [22]:

1. Non-convergence: the generator and the discriminator fail to reach an equilibrium.
2. Mode collapse: the generator model generates similar samples given different inputs.
3. Quick convergence of the discriminator loss to zero: the optimal discriminator will be unable to provide sufficient gradient to update the generator.

To address these issues, several authors suggested heuristic approaches to improve GANs training. For a description of these tricks, we refer the reader to [81, 88]. More recently, a variant of GANs, called the Wasserstein GAN (WGAN) has shown to be more stable to train [5]. Different from the above-described vanilla GAN, WGAN optimizes the Wasserstein distance using Kantorovich Rubinstein duality, instead of optimizing the Jensen-Shannon divergence in the standard GAN. However, WGANs can sometimes generate low-quality samples or fail to converge in some settings due to the use of weight clipping to enforce a Lipschitz constraint on the critic. This problem is alleviated in [39] by the use of a gradient

penalty. An overview of how GANs and their variants work can be found in [48]. In this chapter, we will use the improved version of WGAN [39] to train our model.

4.2.2 Human video generation

Early video generation approaches attempted to generate videos directly in pixel space using GANs or Variational Autoencoders (VAEs) [87, 104, 109]. By doing so, they model all the structure and scene dynamics at once which may yield uninterpretable results. Recently, many works proposed to guide the video generation using pose sequences making it a two-step approach. In the first, human landmarks are exploited to generate novel pose sequences which, along with an input image, are used to generate frame videos. For instance, Walker et al. [110] proposed to guide the video forecasting with pose sequences, breaking the video generation problem into two steps. First, they exploit the effectiveness of landmarks to model human dynamics and used a VAE to generate possible future poses. Then, the latter are used to conditioning a GAN to predict future frames of a video in pixel space. The same idea was applied in [121] to synthesize human videos with a two-stage pose-guided video generation method. In the first stage, they train a Pose Sequence GAN (PSGAN) to generate sequences of poses conditioned on the class label. The latter, along with an input image, are used in the second stage to guide the generation of coherent video frames. Since PSGAN may yield corrupted poses, a Semantic Consistent GAN (SCGAN) is proposed to impose semantic consistency between generated pose and ground-truth at a high feature level. In another work, Wang et al. [115] proposed a Conditional Multi-Mode Network for a diverse smile video generation, conditioned on landmark sequences. They used a VAE to learn a facial landmark embedding. A recurrent network is then used to generate landmark embedding sequences conditioned on a specific expression. The latter are fed into a multi-mode recurrent landmark generator to produce a set of landmark sequences still associated to the given smile class but clearly distinct from each other. Finally, these landmark sequences are translated into face videos. The latter procedure can be achieved by transferring each frame of a pose

sequence to an image which is know as pose to image translation or pose transfer.

Pose to image translation Transferring a person from one pose to another was first proposed by [78] to synthesize person images in arbitrary poses. Based on an image of a person and novel poses, the generation consists of two steps: pose integration and image refinement. First, the condition image and the target pose are fed into a U-Net-like network to generate an initial but coarse image of the person with the target pose. Then, the image refinement is performed on the initial and blurry result by training a U-Net-like generator in an adversarial way. Based on the same idea, the authors in [98] introduced deformable skip connections in the generator of the GAN to deal with pixel-to-pixel misalignments caused by the pose differences. In addition, they proposed a nearest-neighbour loss instead of the common L_1 and L_2 losses in order to match the details of the generated image with the target image. More recently, a progressive pose attention transfer was proposed in [130] to generate the person image progressively. To do so, the generator of the network comprises a sequence of Pose-Attentional Transfer Blocks that each transfers certain regions it attends to. This yields more realistic images with better appearance and shape consistency than previous works.

4.3 The proposed approach

Given T training videos $\{v_i\}_{i=1}^{T}$ presenting one human action, we aim to train a model that learns the training data distribution and generates a set of new videos $\{h_i\}_{i=1}^{M}$. Typically, the generated samples would present variations in terms of the style of performing actions that are different than those of the training videos. This includes variations in terms of the execution rate of actions. To achieve this goal, we propose a two-step approach. In the first, pose sequences $\{p_i\}_{i=1}^{T}$ are extracted from videos and used to train a generative model, namely *shapeGAN*, which learns to generate novel motion dynamics $\{q_i\}_{i=1}^{M}$. In the second,

given an input image I_c of a person, each pose of a generated sequence q is mapped to I_c giving rise to a new sequence of images (*i.e.*, video). In the following, we give more details of each step.

4.3.1 shapeGAN: a pose sequence generation framework

Human landmarks have shown to be effective in several vision tasks including action recognition, as shown in the previous chapter of this thesis. Inspired by its capability to encode human motion dynamics, we aim to use human skeletons to synthesize novel motion dynamics. However, the latter are often characterized with high variability, partly due to the view variations. As a consequence, it is difficult for a generative model to identify shape and motion patterns directly from raw data. Considering this problem, we propose to encode landmark data to a more compact representation that lie to a lower-dimensional space by adopting the sparse coding and dictionary learning scheme that we presented in Chapter 2. In the context of pose sequence generation, this approach brings three main advantages: 1) Landmark configurations are represented in the shape manifold which discards the view variations within the data (*i.e.,* rotations, translations, and global scaling of configurations), hence keeping only information about the shape and its dynamics; 2) SCDL is performed in the shape manifold and yields a vector space representation while accounting to the nonlinear nature of the data. Hence, the resulting representation is compact and it is expected to be more suitable to train a generative model than the original data; 3) This coding technique enables to recover the latent variables back to the shape manifold. Thereby, we propose to design a generative model that synthesizes new samples in the latent space, then reconstruct them to obtain skeletal sequences in the shape manifold.

Figure 4.2 gives an overview of the proposed approach. Given an input video presenting a human action, we first extract a skeleton from each video frame using a state-of-the-art detector [17] where each skeleton is represented by 18 body landmarks in 2D. Then, training

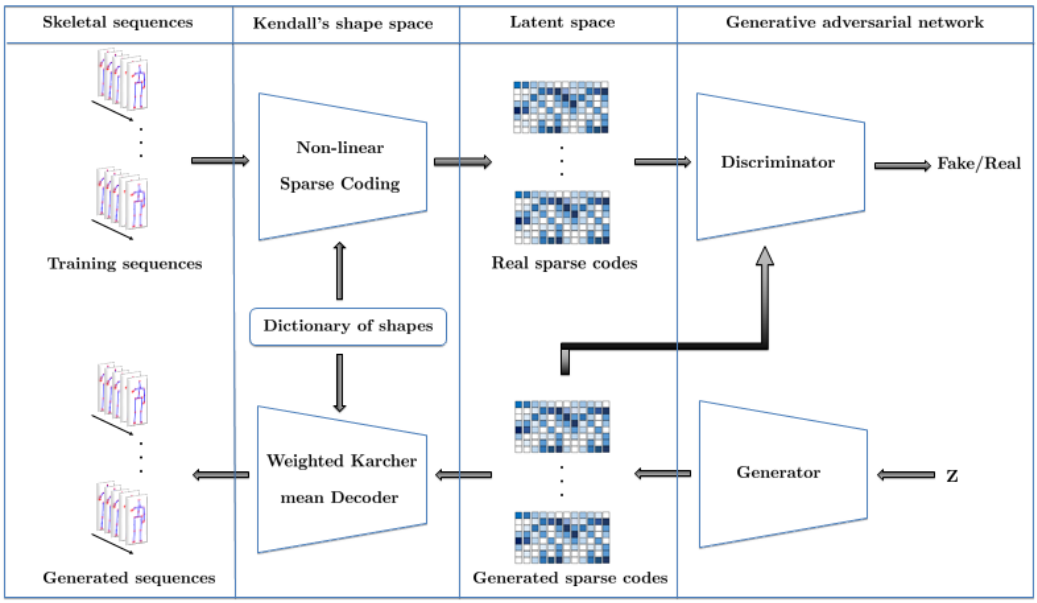


FIGURE 4.2 – Overview of the proposed shapeGAN framework.

sequences are projected to the shape manifold where they are re-sampled using Algorithm 5 of Chapter 3 to have equal length. The resulting trajectories are encoded using SCDL in the shape manifold. The obtained latent samples are used to train a Wasserstein GAN [39]. This allows to generate new samples which are then transformed to skeletal sequences in the shape manifold using the weighted intrinsic mean algorithm as described in Algorithm 4 in Chapter 2. This reconstruction procedure is performed with respect to the pretrained dictionary which insures that the obtained samples lie in the space of skeletons avoiding any noisy or corrupted poses. Moreover, the trained GAN model shows its capability to generate smoothly varying time-series which characterizes human actions. More properties of the generated samples will be investigated in Section 4.4.2.

4.3.2 shapeVGAN: a pose-guided video generation framework

Once the shapeGAN model is trained, we are able to generate novel pose sequences that present new variations in terms of style and temporal evolution of human actions. The

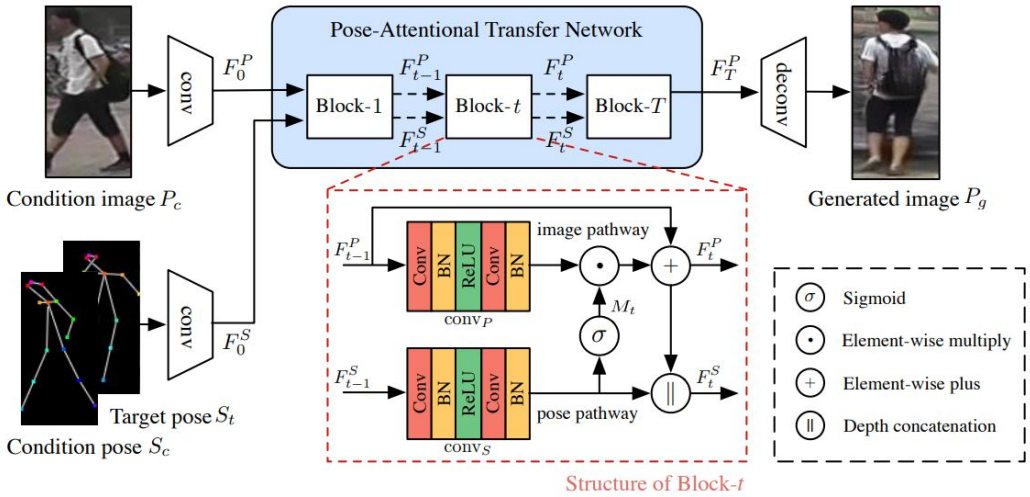


FIGURE 4.3 – Generator architecture of the progressive pose attention transfer method [130].

following step is to transform each pose of a sequence to an image, hence obtain a video. Given a generated pose sequence $P = \{P_1, P_2, ..., P_l\}$, where l is the sequence length, and an input (condition) image I_c, our goal is to transfer the pose of the person in I_c to a target pose $\{P_i\}_{i=1}^{l}$. By doing so, we obtain a sequence of images $I = \{I_1, I_2, ..., I_l\}$ which represents the final human video. In Figure 4.4, we show an example of transferring a pose (c) to an input image (a) to obtain a new image (d). This step is achieved by applying a recently proposed state-of-the-art approach [130] which is based on a GAN whose generator comprises a sequence of Pose-Attentional Transfer Blocks that each transfers certain regions it attends to, generating the person image progressively. An overview of the generator architecture of this method is given in Figure 4.3. In practice, the latter approach produces more realistic images compared to previous solutions which motivates us to use it as part of our framework.

During training, the locations of the 18 body landmarks of a human pose are encoded to a 18-channel heat map H. This model requires different combinations of condition and target image (I_c, I_t) and their corresponding condition and target pose heat map (H_c, H_t). The generator learns a mapping from the condition domain to the target domain. It outputs

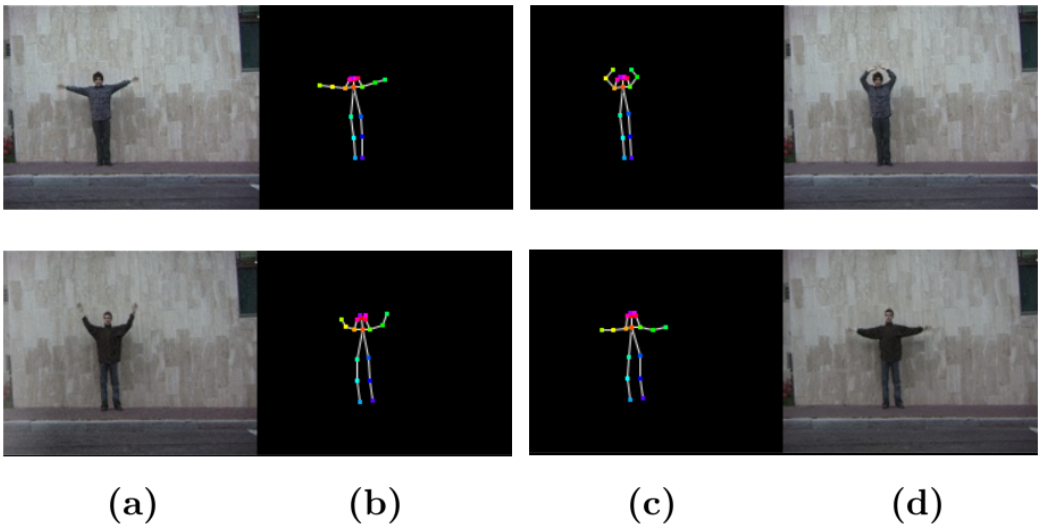


FIGURE 4.4 – Pose to image transfer. (a): Condition image. (b): Condition pose. (c): Target pose. (d): Target image.

a person image, which is challenged by the discriminator for its realness. Once this model is trained, the condition images, along with their corresponding condition heat maps, are taken from the training set while the poses of a generated sequence obtained with shapeGAN are used as target poses (heat maps) allowing to obtain target images that will finally represent a generated video.

4.4 Experimental evaluation

In this section, we conduct extensive experiments to evaluate the efficiency of the proposed frameworks. We first evaluate the performance of the proposed shapeGAN model by applying it in a data augmentation task. For that, we used the Florence 3D action dataset [92] that we presented in Chapter 3. Later on, we evaluate the shapeVGAN model using data collected from the Weizmann Action database [12] which contains 81 videos of 9 people performing 9 actions, including jumping and waving-hands.

4.4.1 Implementation details

shapeGAN implementation Recall that shapeGAN has an encoder-GAN architecture. Regarding the encoder (SCDL), for both datasets, we performed a temporal re-sampling of pose sequences to have equal length $l = 32$ using Algorithm 5. For each dataset, we constructed a dictionary comprised of $k = 32$ atoms using the k-means approach as presented in Chapter 2. We set the sparsity parameter λ to 0.1 to perform sparse coding. For a given pose sequence, the sparse coding step would result in a matrix of dimension 32×32 which can be seen as a one-channel image that we transform to a tensor of dimension $1 \times 32 \times 32$.

Our GAN implementation is based on the improved Wasserstein variant [39]. It is built upon the popular Pytorch framework. We used batches of size 4 and we trained our model for 10^5 epochs. We set the dimension of the input random vector z of the generator to 100.

We design the architecture of our generator as follows. The input latent variable z first goes through two residual blocks of linear and LeakyRelu [79] layers, followed by a linear transformation. The obtained vector after these blocks is of dimension *sequence length* (T) $\times$ *feature dimension (dim)*. It can be seen as T sub-vectors each of dimension *dim*. Each sub-vector finally goes through a softmax layer which maps each component to the range $(0, 1)$ where the components add up to 1. This output layer is important since it guarantees that the generated sparse code vectors have the same structure as the real ones.

ShapeVGAN implementation We used the Weizmann dataset to validate this model for which we scaled each image frame to 180×140. Our implementation is based on that of [130] following the same parameter setting.

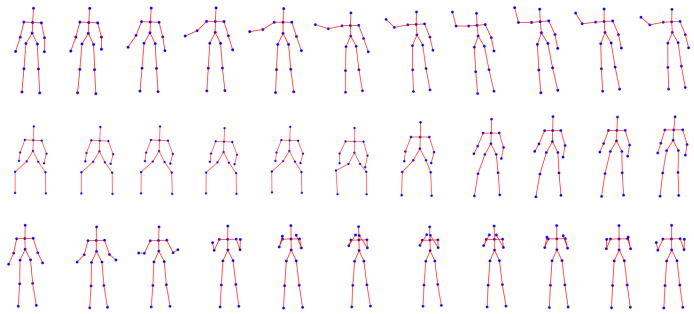


FIGURE 4.5 – Examples of generated pose sequences of actions with shapeGAN. From top to bottom are actions: hand wave, sit-down, and clap hands.

4.4.2 Pose sequence generation for data augmentation

4.4.2.1 Qualitative evaluation

Figure 4.5 shows some examples of generated pose sequences with shapeGAN. We can clearly observe that the pose sequences change in a smooth and typical way under each action scenario, without the presence of any corrupted poses (noise) or temporal interruption. Recall that the shapeGAN model learns to generate new samples in the latent space (after coding initial sequences using SCDL in the shape manifold). To help insure that it generalizes well on the training data and creates variability on the generated samples, in Figure 4.6 and Figure 4.7, we show two plots of both training and generated data in the latent space after embedding them in a 2D space. This is achieved by applying the t-SNE embedding that allows to better visualize data distributions. In the first figure, we have data that belong to one class (*i.e.*, hand wave). The training data (in blue) represent eight sequences. Around each of them, we can see the newly generated sequences (in red) which give us evidence on the generalization capability of our shapeGAN model. In the second figure, we have data that belong to nine classes of the Florence 3D dataset. The left panel representing training data (in the encoder latent space) again demonstrates the discrimination ability of our sparse representation since data-points from each class are regrouped together. The right panel shows that the generated samples are also regrouped together in terms of class

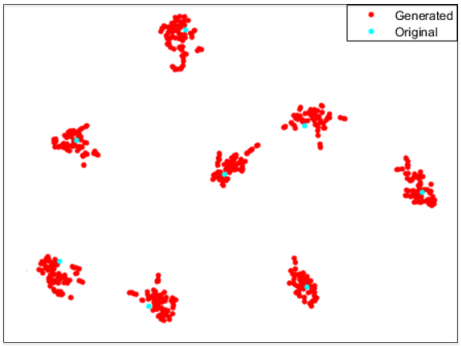


FIGURE 4.6 – 2D visualization of training samples (blue) and generated samples (red).

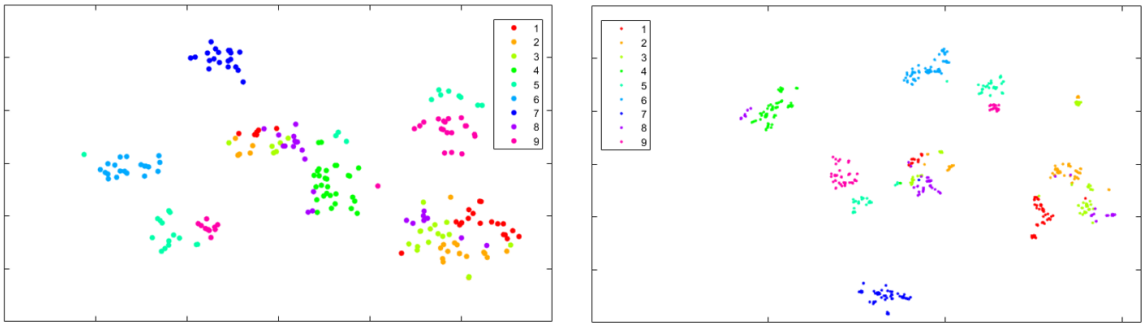


FIGURE 4.7 – 2D visualization of data belonging to nine action classes of the Florence 3D dataset. Left: training samples. Right: training and generated samples.

labels. Note that we train a shapeGAN model for each class separately and the obtained results encourages us to apply a class-conditional model in a future work. This will allow to condition the generation process on the class label. Specifically, as input to the generator, in addition to the random vector z, we can include the class label as one-hot encoded vector. Similarly, with each sample, the discriminator will receive the same information as input.

Another important criterion that we want to evaluate is the generator ability to create a temporal variability. Recall that in our model, we do not perform any temporal modeling of the data. Nevertheless, our model still captures the temporal variability within the data and is capable of generating new ones. To show our previous claim, in Figure 4.8, we visualize two feature maps corresponding to two samples in the encoder latent space (*i.e.,* sequences of sparse codes after SCDL). The first is from the training set while the second is from

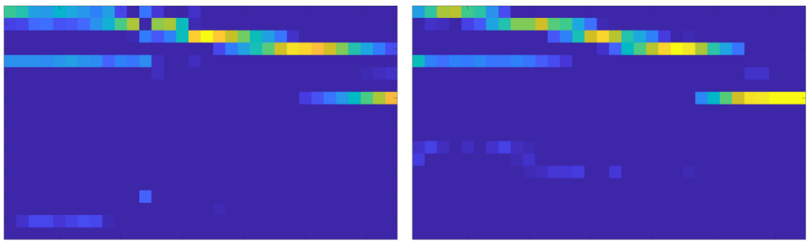


FIGURE 4.8 – Visualization of a training (left) and a generated (right) feature maps.

the generated set. Note that for each, the X-axis represents the time frame while the Y-axis represents the codes (or weights) which correspond to the activation of the dictionary atoms. Comparing these two feature maps, we can notice similar atoms activation in the two while the main difference resides in the temporal shift between them.

The effect of sparsity In our shapeGAN framework, sparsity plays a crucial role in making the data representation more compact which facilitates the task of the generator to learn the training data distribution. In this experiment, we evaluate this property by performing different experiments with different values of the sparsity regularization parameter λ. In Figure 4.9, we present some qualitative results. The training samples are collected from the class *hand wave* of the Florence 3D dataset. Note that this class presents a considerable variation since actors can perform the action either with the left or the right hand, see the first raw of Figure 4.9. We trained three shapeGAN models with the values 0, 10^{-4} and 10^{-2} of the parameter λ, respectively. Recall that λ controls the amount of sparsity in the sparse coding step. The obtained results show that the more λ is higher, the more the generated samples are realistic. We can notice that for values 0 and 10^{-4} of λ, the generated samples combine the two movements: *raising the left hand* and *raising the right hand* in the same action, and even mix both of them by raising the two hands, which do not resemble to real samples. This phenomenon does not occur for relatively higher value of λ.

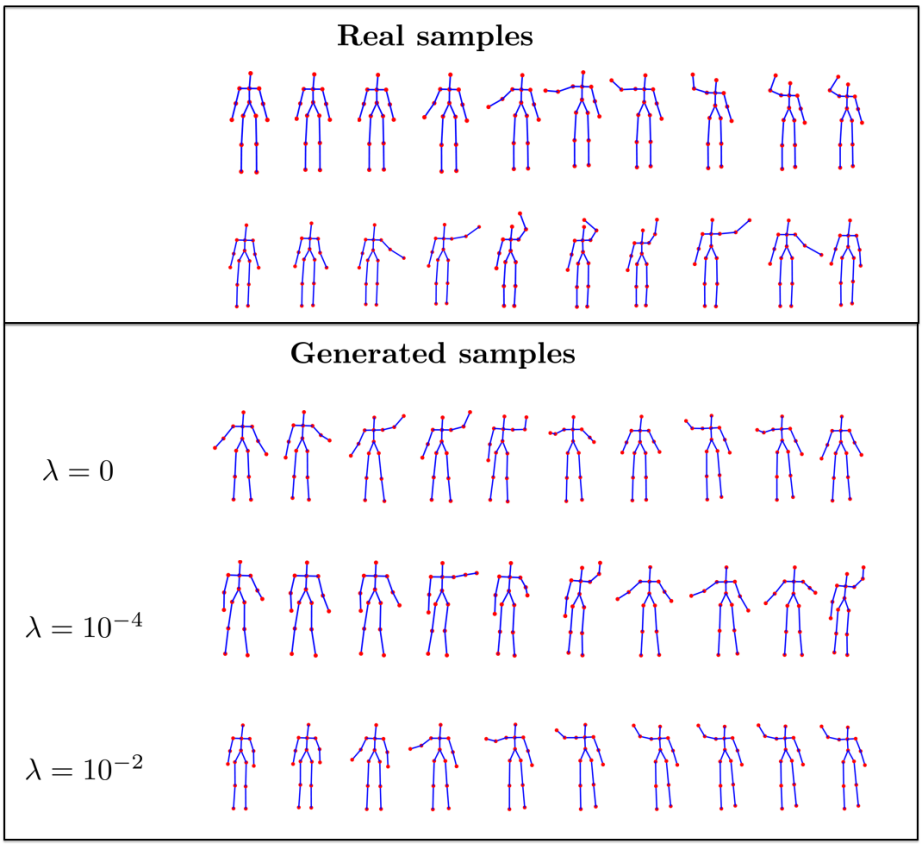


FIGURE 4.9 – Examples of generated samples with shapeVGAN. Different values of the sparsity regularization parameter λ (in the sparse coding) were used in each experiment.

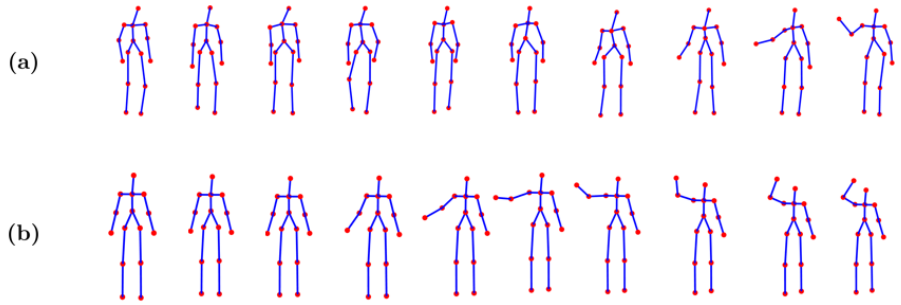


FIGURE 4.10 – (a) Pose sequence generated after training a GAN directly on raw pose sequences. (b) Pose sequence generated with shapeGAN.

Baseline experiment In this experiment, we further evaluate the importance of the coding step in the shapeGAN framework. To this end, we perform a baseline experiment by training a GAN [39] using raw pose sequences taken from the class *hand wave* of the Florence 3D dataset. We show some qualitative results in Figure 4.10. We can observe that the pose sequence in the first raw (*i.e.* obtained with a GAN trained on raw data) presents some temporal interruptions where the landmarks are not stable and the bones lengths (*e.g.* arms or legs) are not constant. Similarly to the other samples generated with the same model, this sequence does not evolve smoothly over time, in contrast to samples generated with shapeGAN as showed in the second raw of Figure 4.10.

4.4.2.2 Quantitative evaluation

To quantitatively evaluate our shapeGAN model, we propose to augment an action recognition dataset, *i.e.* the Florence 3D dataset, and evaluate the classification performance when using: 1) Original training data; and 2) Training data augmented with an increasing number of generated data. This procedure was proposed in [96] as a metric to evaluate a GAN's performance.

We use the same action recognition method that we proposed in Chapter 3. For temporal

TABLE 4.1 – Action recognition accuracy (%) using an increasing number of generated training data.

Data	**Accuracy**
Training data	93.04
Augmented data ×3	93.98
Augmented data ×6	94.56
Augmented data ×9	95.04
Augmented data ×12	95.04

modeling and classification, we choose to use Bi-LSTM rather than the pipeline of DTW-FTP-SVM since it is known that deep learning methods usually require more data in order to increase their generalization capability compared to classical approaches. Table 4.1 summarizes the obtained results. Note that, in the different experiments that we performed, the amount of data was multiplied by 3, 6, 9 and 12, respectively. The obtained results have seen increasing improvements when increasing the amount of training data. Note that the classification performance reaches saturation when multiplying the amount of training samples by 9. These results demonstrate the efficiency of the proposed shapeGAN model which can be useful as a data augmentation tool for different recognition tasks and potentially for other vision tasks which relieves the burden of manual annotations.

4.4.3 Human video generation

After evaluating the performance of the proposed shapeGAN framework which showed promising results in the generation of human pose sequences, we aim to evaluate the final step of video generation. To this end, in what follows, we present some qualitative results achieved with the shapeVGAN framework. Recall that the generated pose sequences from the first step are used to guide the generation of video frames.

In Figure 4.11, we show examples of generated videos with shapeVGAN which belong to the class *wave hands* of the Weizmann dataset. The first row of this figure represents a training video while the two others are taken from the generated set. The latter are generated

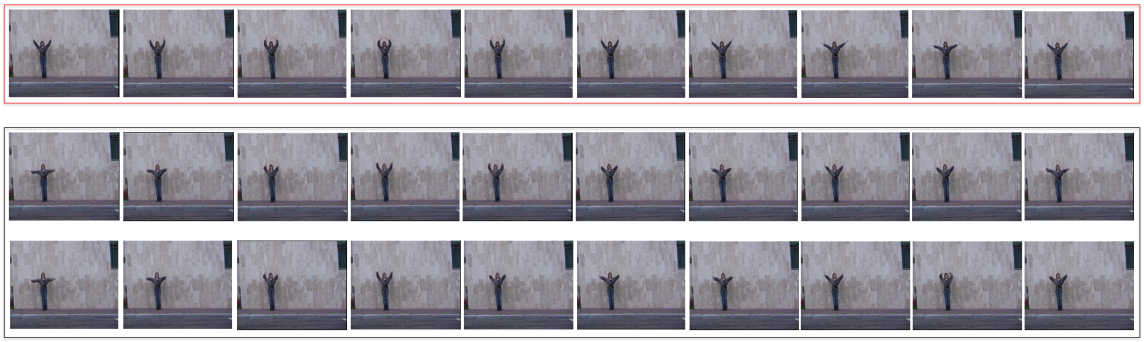


FIGURE 4.11 – Examples of video frames of action wave hands. The first row represents a training video while the two others represent videos generated with our approach.

by taking a pose sequence that is generated using shapeGAN, then transferring each pose to an image to obtain a video. This is achieved by taking a condition image which is the first image of the training video in Figure 4.11. After extracting the pose of the person from this condition image, we transfer it to the poses of the generated sequences to obtain the final videos. Visually, we can clearly observe that our obtained results are realistic in terms of appearance and temporal consistency since the human shape in a generated video evolves smoothly over time. This demonstrates the efficiency of the pose transfer approach as well as the advantage of guiding the video generation process with pose sequences. Besides, if we compare the two generated videos frame by frame, we can notice that the way of performing the same action is different which again demonstrates the variability that has been created in the pose motion thanks to the shapeGAN framework.

To further evaluate our method, we took a generated pose sequence and transferred it to two images of two different actors aiming to obtain two videos of actors with the motion. Figure 4.12 presents the obtained results where we can clearly see that these actors perform the same action with a similar motion. This demonstrates the ability of our framework in preserving the identity of the person (*i.e.,* visual appearance, size of arms, etc.) in the condition image while modifying its motion. Here, it is important to mention that since shapeGAN generates pose sequences in the shape manifold, information regarding scale and

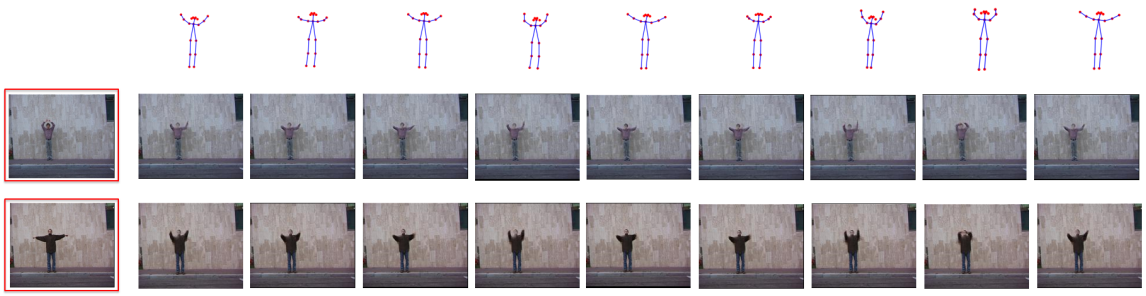


FIGURE 4.12 – Example of a generated pose sequence with shapeGAN (Top row). This sequence is used to guide the generation of the two videos (second and third row) given condition images (in red).

translation of a pose (*i.e.,* location of a person in the image) are missing. As a remedy, before the pose transfer step, we align each generated pose into the pose of the person in the condition image by applying the procrustes algorithm [62].

Comparison to a state-of-the-art method To demonstrate the performance of the presented shapeVGAN framework, we compare it to a state-of-the-art method namely MoCoGAN [104]. This method generates a video by mapping a sequence of random vectors to a sequence of video frames where the motion and content are decomposed in the generation process. In contrast to our approach which guides the video generation with pose sequences, MoCoGAN generates videos directly in pixel space.

We train both models using the same videos collected from the Weizmann database. In Figure 4.13, we show some qualitative results that are randomly taken from generated samples. We can clearly observe that video frames obtained with our approach are more realistic than those obtained with MoCoGAN. In fact, the latter contain some blurry images (*e.g.* images in last raw of Figure 4.13) as well as some content and motion artifacts (noise). In contrast, shapeVGAN is able to keep more details in the video images and the person movements evolve more smoothly and naturally over time. This shows the relevance of the proposed pose-guided generation framework.

4.5 Limitations and ongoing study

Since filtering out translations, rotation and scaling from configurations is a necessary step in our shapeVGAN encoder (to perform SCDL in the shape manifold), these transformations are still important to recover the location of the generated poses in an image, as explained in the previous paragraph. In other words, imagine a video where a person is walking from the left to the right of the scene, its corresponding pose represented in the shape manifold would result in a centered skeleton moving locally in the same position. As a consequence, in addition to generating pose sequences in the shape manifold, it is necessary to generate the filtered transformations (translations, rotation and scaling) then add them to the generated pose sequences as a post-processing step. As an ongoing study, we aim to encode these transformations separately and feed them to a second generator that learns to generate new translations, rotations and scaling of poses. Since these information are correlated with the pose dynamics, the two generators of our model have to be learned jointly to guarantee a certain coherence in the final pose sequences.

On the other hand, we are now capable of generating pose sequences without any assumption on the class labels. Hence, for data augmentation, we have to train a model for each action class. However, it would be more interesting to condition the shapeGAN model on the class label by using a conditional GAN which showed to be efficient in different image generation tasks. This is part of our ongoing work.

4.6 Conclusion

In this chapter, we explored the problem of human motion generation with GANs and proposed a novel framework namely shapeVGAN that allows to generate human videos in a two-stage procedure. In the first stage, we presented the shapeGAN model to generate novel pose sequences which are used in the second stage to guide the video generation. shapeGAN

has an encoder-GAN architecture. The encoder is based on SCDL in the shape manifold which in addition to being a discriminative representation that showed promising results in classification problems, it has been an efficient representation for the human motion generation task. The GAN generates new samples in the encoder latent space that are then transformed to pose sequences. Each pose of a sequence is then mapped to an image given a condition image to obtain a video. The generated videos with shapeVGAN showed to be realistic and temporally consistent in the case of human actions. As a future work, we aim to generalize this framework to the task of facial expression generation.

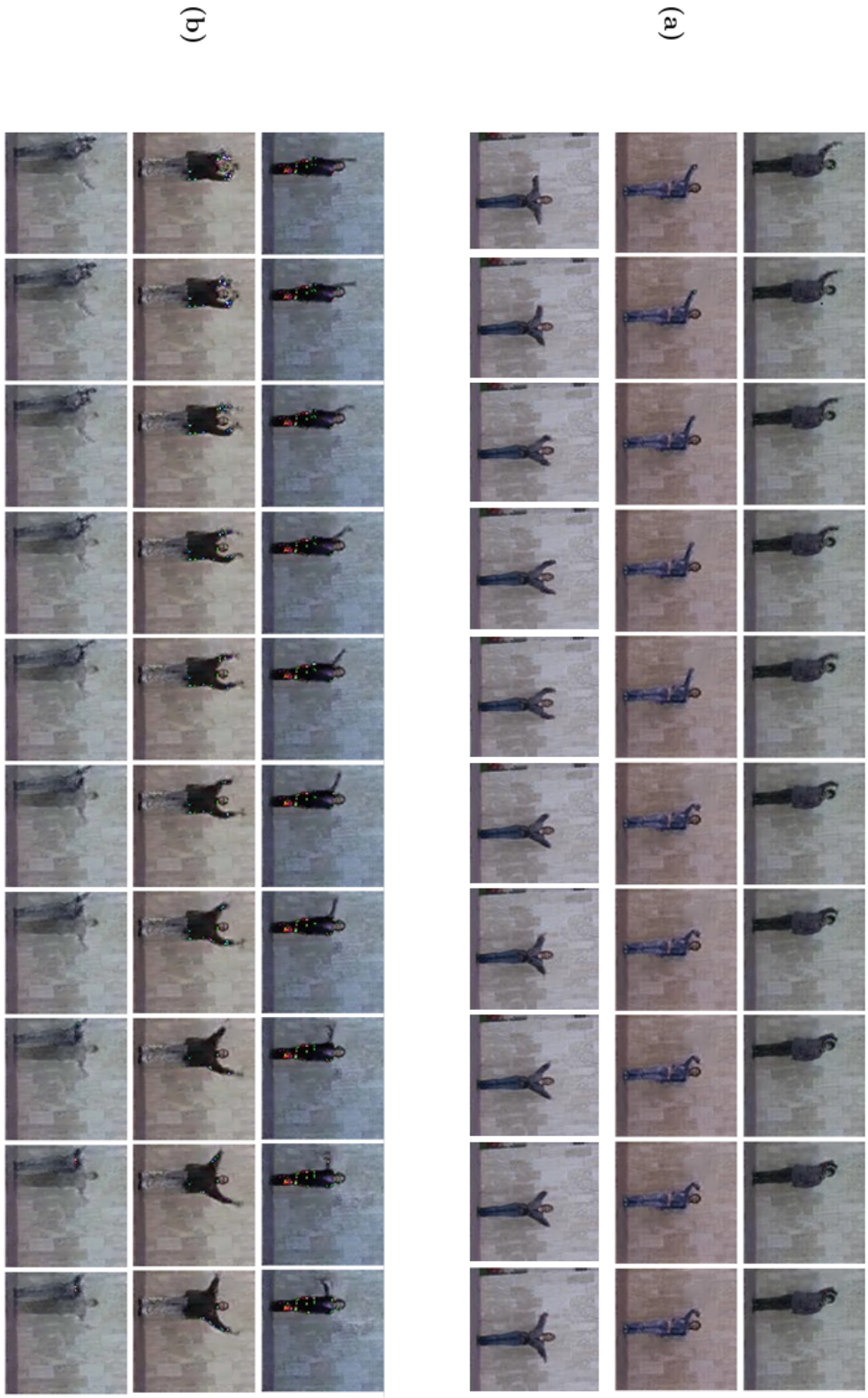


FIGURE 4.13 – Qualitative comparison of (a) Our results obtained with shapeVGAN. (b) Results obtained by training a state-of-the-art approach, *i.e.* MoCoGAN [104].

Chapitre 5

Conclusions and Perspectives

In this dissertation, we have presented a novel framework to model landmark-based human actions and facial expressions. We have demonstrated that the proposed methods are competitive or outperform the state-of-the-art on various and challenging datasets in two recognition tasks: 3D action recognition, 2D micro and macro facial expression recognition. We also demonstrated that the proposed action modeling can be efficiently integrated in a generative model in order to synthesize novel human actions and videos. We conclude our work pointing out the main contributions in Section 5.1 and their limitations in Section 5.2. Finally, we discuss future perspectives in Section 5.3, indicating interesting directions for future work in this field.

5.1 Main contributions

We have proposed sparse representations for landmark-based human actions and facial expressions. We represented sequences of human landmarks as trajectories in the shape manifold to allow for a view-invariant analysis. Trajectories are encoded using Riemannian versions of sparse coding and dictionary learning. This allows to compensate the lack of vector space structure in the shape manifold and yields Euclidean time-series with suitable

computational properties, *e.g.* sparsity and vector space structure.

Application of the presented frameworks to 3D human action recognition and 2D facial expression recognition We have applied the proposed representations in different classification problems, *i.e.* 3D action recognition, 2D micro and macro facial expression recognition. We have adopted two temporal modeling and classification schemes, a deep learning framework based on LSTM and a classical pipeline of methods designed for Euclidean time-series. We have conducted extensive evaluation on seven commonly-used datasets that present different challenges which showed the competitiveness of the proposed frameworks to state-of-the-art.

Evaluation and comparison of extrinsic and intrinsic coding approaches in the shape manifold We have explored two paradigms of sparse coding and dictionary learning in the shape manifold, *intrinsic* and *extrinsic*.

The first allows to perform coding on the manifold tangent spaces which solves for a commonly encountered problem in the literature of Riemannian methods, *i.e.* trajectories distortion when mapped to a reference tangent space. It also enables an easy reconstruction back to the original manifold which allows the visual exploration of the latent variables.

The second performs coding in Hilbert space after mapping the manifold-valued data using a kernel embedding which gives a richer representation of the data and helps identifying complex patterns.

We have evaluated and compared these two paradigms in the context of recognition tasks that we addressed. We showed that the extrinsic approach is more efficient to represent 2D trajectories while the intrinsic one is more suitable to encode 3D trajectories.

A generative framework for human motion synthesis We have proposed a generative model to synthesize human pose sequences (namely shapeGAN). shapeGAN first encodes

training pose sequences using nonlinear SCDL and uses the obtained latent samples to train a generative adversarial network. The generated samples showed to be realistic while representing new styles of actions. We have used this model to augment a state-of-the-art action dataset which improved the action recognition accuracy.

We have used the generated pose sequences with shapeGAN to guide the generation of human videos. Given an input person image, each pose of a generated sequence is transferred to an image using a state-of-the-art pose transfer framework. This allowed us to synthesize new videos.

5.2 Limitations

— The presented techniques assume that human landmarks are available and are sufficiently accurate since all the datasets that we used are collected in controlled lab environments. However, the estimation of landmarks could be more challenging but possible in an uncontrolled setting.

— While the use of an extrinsic approach to represent 2D trajectories in the shape manifold showed to be more efficient for classification than the intrinsic approach, its application in the 3D shape manifold is not usually suitable. In fact, this requires a positive definite kernel to allow mapping manifold-valued data to RKHS. Such valid kernels are only available in the case of 2D shapes.

— The main limitation of the presented human motion generation framework is its disability to generate global shape displacements. In fact, encoding data in the shape manifold discards the global transformations (*i.e.* translations, rotations, scaling of configurations).

5.3 Future work

Several aspects can be investigated in future works in order to improve the proposed approaches and further evaluate their performance.

— The presented extrinsic approach to code shape trajectories is only suitable for the 2D shape manifold where a positive definite kernel exists in the literature. This is not the case for the 3D shape manifold where our extension of the Procrustes Gaussian kernel is not always valid (see Section 3.4.1.4). Therefore, we would like to further study the existence of positive definite kernels in the 3D shape manifold.

— The proposed facial expression framework has been evaluated on datasets that were collected in controlled environments and do not present considerable view-variations (*e.g.* the Cohn-Kanade and the Oulu-Casia datasets). Hence, we would like to evaluate our approach on more challenging datasets such as the AFEW database [24] where the data were collected from movies showing close-to-real-world conditions, which simulates the spontaneous expressions in uncontrolled environment. In addition, the view variations for 2D landmarks yield projective transformations which are more complex to filter out, thereby it would be interesting to investigate this problem.

— The presented generative framework, *i.e.* shapeGAN, is only capable to generate local pose deformations without controlling the skeleton displacement (*i.e.* location of a pose in the video). We would like to use the transformations that are filtered out in the first step (*i.e.* rotation, translation, and scale) as input to a second generator in order to generate new ones. The generated samples could then be added up to the corresponding samples generated from the first generator. The two generators have to be trained jointly in order to learn the relations between the two information (*i.e.* local and global pose deformations).

— We have applied shapeVGAN to generate human actions on the Weizmann dataset only. Hence, we would like to evaluate it on other, more challenging datasets presenting

more action variations such as the UCF-101 dataset [100]. Our next goal is to also apply the shapeVGAN framework on other problems such as the generation of facial expressions. Moreover, we exploited shapeGAN to augment the Florence3D dataset and we would like to augment other human actions and facial expressions datasets and evaluate recognition performances using different state-of-the-art approaches.

Bibliographie

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[2] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

[3] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014.

[4] Rushil Anirudh, Pavan Turaga, Jingyong Su, and Anuj Srivastava. Elastic functional coding of human actions: From vector-fields to latent variables. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3147–3155, 2015.

[5] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223, 2017.

[6] American Psychiatric Association et al. Diagnostic and statistical manual of mental disorders. *BMC Med*, 17:133–137, 2013.

[7] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Incremental face alignment in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1859–1866, June 2014.

[8] Tadas Baltrusaitis, Amir Zadeh, Yao Chong Lim, and Louis-Philippe Morency. Openface 2.0: Facial behavior analysis toolkit. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 59–66. IEEE, 2018.

[9] Boulbaba Ben Amor, Jingyong Su, and Anuj Srivastava. Action recognition using rate-invariant analysis of skeletal shape trajectories. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(1):1–13, 2016.

[10] Amor Ben Tanfous, Hassen Drira, and Boulbaba Amor Ben. Reconnaissance d'actions 3d par codage parcimonieux sur l'espace de kendall.

[11] Amor Ben Tanfous, Hassen Drira, and Boulbaba Ben Amor. Coding kendall's shape trajectories for 3d action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2840–2849, 2018.

[12] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1395–1402. IEEE, 2005.

[13] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European Conference on Computer Vision*, pages 561–578. Springer, 2016.

[14] William M Boothby. *An introduction to differentiable manifolds and Riemannian geometry*, volume 120. Academic press, 1986.

[15] Ran Breuer and Ron Kimmel. A deep learning perspective on the origin of facial expressions. *arXiv preprint arXiv:1705.01842*, 2017.

[16] Darshan Bryner, Eric Klassen, Huiling Le, and Anuj Srivastava. 2d affine and projective shape analysis. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):998–1011, 2014.

[17] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7291–7299, 2017.

[18] H Ertan Cetingül and René Vidal. Sparse riemannian manifold clustering for hardi segmentation. In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pages 1750–1753. IEEE, 2011.

[19] Hasan Ertan Çetingül and René Vidal. Intrinsic mean shift for clustering on stiefel and grassmann manifolds. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 1896–1902. IEEE Computer Society, 2009.

[20] A. Cherian and S. Sra. Riemannian dictionary learning and sparse coding for positive definite matrices. *IEEE Transactions on Neural Networks and Learning Systems*, PP(99):1–13, 2017.

[21] Dong Yoon Choi, Dae Ha Kim, and Byung Cheol Song. Recognizing fine facial micro-expressions using two-dimensional landmark feature. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1962–1966. IEEE, 2018.

[22] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65, 2018.

[23] Fernando De la Torre, Wen-Sheng Chu, Xuehan Xiong, Francisco Vicente, Xiaoyu Ding, and Jeffrey Cohn. Intraface. 05 2015.

[24] Abhinav Dhall, Roland Goecke, Simon Lucey, Tom Gedeon, et al. Collecting large,

richly annotated facial-expression databases from movies. *IEEE multimedia*, 19(3):34–41, 2012.

[25] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[26] Ian L. Dryden and Kanti V. Mardia. *Statistical shape analysis*. Wiley, 1998.

[27] Ian L Dryden and Kanti V Mardia. *Statistical Shape Analysis: With Applications in R*. John Wiley & Sons, 2016.

[28] Yong Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1110–1118, June 2015.

[29] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1110–1118, 2015.

[30] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.

[31] S. Elaiwat, M. Bennamoun, and F. Boussaid. A spatio-temporal rbm-based model for facial expression recognition. *Pattern Recognition*, 49:152 – 161, 2016.

[32] Kjersti Engan, Sven Ole Aase, and J Hakon Husoy. Method of optimal directions for frame design. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, volume 5, pages 2443–2446. IEEE, 1999.

[33] P Thomas Fletcher, Conglin Lu, Stephen M Pizer, and Sarang Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE transactions on medical imaging*, 23(8):995–1005, 2004.

[34] Guillermo Garcia-Hernando and Tae-Kyun Kim. Transition forests: Learning discriminative temporal transitions for action recognition and detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 432–440, 2017.

[35] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[36] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[37] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005.

[38] Tanaya Guha and Rabab K Ward. Learning sparse representations for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8):1576–1588, 2012.

[39] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[40] K. Guo, P. Ishwar, and J. Konrad. Action recognition from video using feature covariance matrices. *IEEE Transactions on Image Processing*, 22(6):2479–2494, June 2013.

[41] M. Harandi and M. Salzmann. Riemannian coding and dictionary learning: Kernels to the rescue. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3926–3935, June 2015.

[42] Mehrtash Harandi, Richard Hartley, Chunhua Shen, Brian Lovell, and Conrad Sanderson. Extrinsic methods for coding and dictionary learning on grassmann manifolds. *International Journal of Computer Vision*, 114(2-3):113–136, 2015.

[43] Mehrtash T Harandi, Richard Hartley, Brian Lovell, and Conrad Sanderson. Sparse coding on symmetric positive definite manifolds using bregman divergences. *IEEE transactions on neural networks and learning systems*, 27(6):1294–1306, 2016.

[44] Mehrtash T. Harandi, Conrad Sanderson, Richard Hartley, and Brian C. Lovell. Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach. In *Proceedings, Part II, of the 12th European Conference on Computer Vision — ECCV 2012 - Volume 7573*, pages 216–229, Berlin, Heidelberg, 2012. Springer-Verlag.

[45] Jeffrey Ho, Yuchen Xie, and Baba Vemuri. On a nonlinear generalization of sparse coding and dictionary learning. In *International conference on machine learning*, pages 1480–1488, 2013.

[46] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[47] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[48] Yongjun Hong, Uiwon Hwang, Jaeyoon Yoo, and Sungroh Yoon. How generative adversarial networks and their variants work: An overview. *ACM Computing Surveys (CSUR)*, 52(1):10, 2019.

[49] Weiming Hu, Tieniu Tan, Liang Wang, and Steve Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 34(3):334–352, 2004.

[50] Xiaohua Huang, Guoying Zhao, Xiaopeng Hong, Wenming Zheng, and Matti Pietikäinen. Spontaneous facial micro-expression analysis using spatiotemporal completed local quantized patterns. *Neurocomputing*, 175:564–578, 2016.

[51] Zhiwu Huang, Chengde Wan, Thomas Probst, and Luc Van Gool. Deep learning on lie groups for skeleton-based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6099–6108, 2017.

[52] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[53] S. Jain, Changbo Hu, and J. K. Aggarwal. Facial expression recognition with temporal modeling of shapes. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1642–1649, nov 2011.

[54] Sadeep Jayasumana, Richard Hartley, Mathieu Salzmann, Hongdong Li, and Mehrtash Harandi. Kernel methods on riemannian manifolds with gaussian rbf kernels. *IEEE transactions on pattern analysis and machine intelligence*, 37(12):2464–2477, 2015.

[55] Sadeep Jayasumana, Mathieu Salzmann, Hongdong Li, and Mehrtash Harandi. A framework for shape analysis via hilbert space embedding. In *IEEE ICCV*, pages 1249–1256, 2013.

[56] Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14(2):201–211, 1973.

[57] H. Jung, S. Lee, J. Yim, S. Park, and J. Kim. Joint fine-tuning in deep neural networks for facial expression recognition. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2983–2991, Dec 2015.

[58] Anis Kacem, Mohamed Daoudi, Boulbaba Ben Amor, Stefano Berretti, and Juan Carlos Alvarez-Paiva. A novel geometric framework on gram matrix trajectories for human behavior understanding. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[59] Anis Kacem, Mohamed Daoudi, Boulbaba Ben Amor, and Juan Carlos Alvarez-Paiva. A novel space-time representation on the positive semidefinite cone for facial expression recognition. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[60] H. Karcher. Riemannian center of mass and mollifier smoothing. *Comm. Pure Appl. Math.*, 30(5):509–541, September 1977.

[61] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. A new representation of skeleton sequences for 3d action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3288–3297, 2017.

[62] David G. Kendall. Shape manifolds, Procrustean metrics, and complex projective spaces. *Bulletin of the London Mathematical Society*, 16(2):81–121, 1984.

[63] Dae Hoe Kim, Wissam J Baddar, and Yong Man Ro. Micro-expression recognition with expression-state constrained spatio-temporal feature representations. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 382–386. ACM, 2016.

[64] Tae Soo Kim and Austin Reiter. Interpretable 3d human action analysis with temporal convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1623–1631. IEEE, 2017.

[65] Yu Kong and Yun Fu. Human action recognition and prediction: A survey. *arXiv preprint arXiv:1806.11230*, 2018.

[66] Piotr Koniusz, Anoop Cherian, and Fatih Porikli. Tensor representations via kernel linearization for action recognition from 3d skeletons. In *European Conference on Computer Vision*, pages 37–53. Springer, 2016.

[67] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang,

et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, volume 2, page 4, 2017.

[68] P. Li, Q. Wang, W. Zuo, and L. Zhang. Log-euclidean kernels for sparse representation and dictionary learning. In *2013 IEEE International Conference on Computer Vision*, pages 1601–1608, Dec 2013.

[69] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. Action recognition based on a bag of 3D points. In *IEEE Inter. Workshop on CVPR for Human Communicative Behavior Analysis (CVPR4HB)*, page 9–14, 2010.

[70] Sze-Teng Liong, John See, Raphael C-W Phan, Yee-Hui Oh, Anh Cat Le Ngo, KokSheik Wong, and Su-Wei Tan. Spontaneous subtle expression detection and recognition based on facial strain. *Signal Processing: Image Communication*, 47:170–182, 2016.

[71] Baiyang Liu, Junzhou Huang, Lin Yang, and Casimir Kulikowsk. Robust tracking using local sparse appearance model and k-selection. In *CVPR 2011*, pages 1313–1320. IEEE, 2011.

[72] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C. Kot. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[73] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *ECCV*, 2016.

[74] M. Liu, S. Shan, R. Wang, and X. Chen. Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1749–1756, June 2014.

[75] Liliana Lo Presti and Marco La Cascia. 3d skeleton-based human action classification. *Pattern Recogn.*, 53(C):130–147, May 2016.

[76] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 94–101, June 2010.

[77] Yui Man Lui. Advances in matrix manifolds for computer vision. *Image Vision Comput.*, 30(6-7):380–388, June 2012.

[78] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Advances in Neural Information Processing Systems*, pages 406–416, 2017.

[79] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013.

[80] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Discriminative learned dictionaries for local image analysis. Technical report, MINNESOTA UNIV MINNEAPOLIS INST FOR MATHEMATICS AND ITS APPLICATIONS, 2008.

[81] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *CoRR*, abs/1611.02163, 2016.

[82] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. *arXiv preprint arXiv:1610.09585*, 2016.

[83] Yee-Hui Oh, John See, Anh Cat Le Ngo, Raphael C-W Phan, and Vishnu Monn Baskaran. A survey of automatic facial micro-expression analysis: Databases, methods and challenges. *Frontiers in psychology*, 9:1128, 2018.

[84] Adeline Paiement, Lili Tao, Sion Hannuna, Massimo Camplani, Dima Damen, and Majid Mirmehdi. Online quality assessment of human movement from skeleton data. In *British Machine Vision Conference*, pages 153–166. BMVA press, 2014.

[85] Mohsen Ramezani and Farzin Yaghmaee. A review on human action analysis in videos for retrieval applications. *Artificial Intelligence Review*, 46(4):485–514, 2016.

[86] Ron Rubinstein, Alfred M Bruckstein, and Michael Elad. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057, 2010.

[87] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2830–2839, 2017.

[88] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[89] Najmeh Samadiani, Guangyan Huang, Borui Cai, Wei Luo, Chi-Hung Chi, Yong Xiang, and Jing He. A review on automatic facial expression recognition systems assisted by multimodal sensor data. *Sensors*, 19(8):1863, 2019.

[90] B. Schölkopf and AJ. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, USA, December 2002.

[91] Bernhard Schölkopf, Ralf Herbrich, and Alexander J. Smola. A generalized representer theorem. In *COLT/EuroCOLT*, 2001.

[92] Lorenzo Seidenari, Vincenzo Varano, Stefano Berretti, Alberto Del Bimbo, and Pietro Pala. Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2013, Portland, OR, USA, June 23-28, 2013*, pages 479–485, 2013.

[93] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang 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*, pages 1010–1019, 2016.

[94] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang 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.

[95] Jianping Shi, Xiang Ren, Guang Dai, Jingdong Wang, and Zhihua Zhang. A non-convex relaxation approach to sparse dictionary learning. In *CVPR 2011*, pages 1809–1816. IEEE, 2011.

[96] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. How good is my gan? In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 213–229, 2018.

[97] 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 *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1297–1304, 2011.

[98] Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuilière, and Nicu Sebe. Deformable gans for pose-based human image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3408–3416, 2018.

[99] Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 129–136, 2011.

[100] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[101] Jingyong Su, Sebastian Kurtek, Eric Klassen, and Anuj Srivastava. Statistical analysis of trajectories on riemannian manifolds: Bird migration, hurricane tracking, and video surveillance. *Annals of Applied Statistics*, 2013.

[102] Sima Taheri, Pavan Turaga, and Rama Chellappa. Towards view-invariant expression analysis using analytic shape manifolds. In *Automatic Face & Gesture Recognition*

and Workshops (FG 2011), 2011 IEEE International Conference on, pages 306–313. IEEE, 2011.

[103] Amor Ben Tanfous, Hassen Drira, and Boulbaba Ben Amor. Sparse coding of shape trajectories for facial expression and action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[104] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018.

[105] Michel Valstar and Maja Pantic. Induced disgust, happiness and surprise: an addition to the mmi facial expression database. In *Proc. 3rd Intern. Workshop on EMOTION (satellite of LREC): Corpora for Research on Emotion and Affect*, page 65, 2010.

[106] Vivek Veeriah, Naifan Zhuang, and Guo-Jun Qi. Differential recurrent neural networks for action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 4041–4049, 2015.

[107] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, 2014.

[108] Raviteja Vemulapalli and Rama Chellapa. Rolling rotations for recognizing human actions from 3d skeletal data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4471–4479, 2016.

[109] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621, 2016.

[110] Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert. The pose knows: Video forecasting by generating pose futures. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3332–3341, 2017.

[111] Chunyu Wang, Yizhou Wang, and Alan L Yuille. Mining 3d key-pose-motifs for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2639–2647, 2016.

[112] Jiang Wang, Zicheng Liu, Jan Chorowski, Zhuoyuan Chen, and Ying Wu. Robust 3D action recognition with random occupancy patterns. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part II*, pages 872–885, 2012.

[113] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1290–1297, 2012.

[114] Pei Wang, Chunfeng Yuan, Weiming Hu, Bing Li, and Yanning Zhang. Graph based skeleton motion representation and similarity measurement for action recognition. In *European conference on computer vision*, pages 370–385. Springer, 2016.

[115] Wei Wang, Xavier Alameda-Pineda, Dan Xu, Pascal Fua, Elisa Ricci, and Nicu Sebe. Every smile is unique: Landmark-guided diverse smile generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7083–7092, 2018.

[116] Ziheng Wang, Shangfei Wang, and Qiang Ji. Capturing complex spatio-temporal relations among facial muscles for facial expression recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3422–3429, 2013.

[117] Lu Xia, Chia-Chih Chen, and Jake K Aggarwal. View invariant human action recognition using histograms of 3d joints. In *Computer vision and pattern recognition workshops (CVPRW), 2012 IEEE computer society conference on*, pages 20–27. IEEE, 2012.

[118] Xuehan Xiong and Fernando De la Torre. Supervised descent method and its applications to face alignment. In *Proceedings of the IEEE conference on computer*

vision and pattern recognition, pages 532–539, 2013.

[119] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[120] Wen-Jing Yan, Xiaobai Li, Su-Jing Wang, Guoying Zhao, Yong-Jin Liu, Yu-Hsin Chen, and Xiaolan Fu. Casme ii: An improved spontaneous micro-expression database and the baseline evaluation. *PloS one*, 9(1):e86041, 2014.

[121] Ceyuan Yang, Zhe Wang, Xinge Zhu, Chen Huang, Jianping Shi, and Dahua Lin. Pose guided human video generation. In *European Conference on Computer Vision*, pages 204–219. Springer, 2018.

[122] Chunfeng Yuan, Weiming Hu, Xi Li, Stephen Maybank, and Guan Luo. *Human Action Recognition under Log-Euclidean Riemannian Metric*, pages 343–353. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[123] Kaihao Zhang, Yongzhen Huang, Yong Du, and Liang Wang. Facial expression recognition based on deep evolutional spatial-temporal networks. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, PP, 03 2017.

[124] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2117–2126, 2017.

[125] Qiang Zhang and Baoxin Li. Discriminative k-svd for dictionary learning in face recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2691–2698. IEEE, 2010.

[126] S. Zhang, X. Liu, and J. Xiao. On geometric features for skeleton-based action recognition using multilayer lstm networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 148–157, March 2017.

[127] Zhengwu Zhang, Debdeep Pati, and Anuj Srivastava. Bayesian clustering of shapes of curves. *Journal of Statistical Planning and Inference*, 166:171 – 186, 2015. Special Issue on Bayesian Nonparametrics.

[128] Hao Zheng, Xin Geng, and Zhongxue Yang. A relaxed k-svd algorithm for spontaneous micro-expression recognition. In *Pacific Rim International Conference on Artificial Intelligence*, pages 692–699. Springer, 2016.

[129] L. Zhong, Q. Liu, P. Yang, B. Liu, J. Huang, and D. N. Metaxas. Learning active facial patches for expression analysis. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2562–2569, June 2012.

[130] Zhen Zhu, Tengteng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. Progressive pose attention transfer for person image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2347–2356, 2019.

[131] H. E. Çetingül, M. J. Wright, P. M. Thompson, and R. Vidal. Segmentation of high angular resolution diffusion mri using sparse riemannian manifold clustering. *IEEE Transactions on Medical Imaging*, 33(2):301–317, Feb 2014.