Analysis and prediction of human behavior temporal sequences in the wild

Benjamin Szczapa

Université de Lille

THÈSE

en cotutelle entre l'Université de Lille et l'Université de Florence

préparée et soutenue publiquement par

Benjamin Szczapa

le 30/09/2022

pour obtenir le grade de Docteur en Informatique

Analyse et Prédiction du Comportement Humain dans des Séquence Temporelles non Controlées

Analysis and Prediction of Human Behavior Temporal Sequences in the Wild

COMPOSITION DU JURY

Mr Mohamed Daoudi	Directeur de la thèse	Professeur, IMT Nord Europe, France
Mr Alberto Del Bimbo	Co-Directeur de la thèse	Professeur, Université de Florence, Italie
Mr Stefano Berretti	Co-Encadrant de la thèse	Professeur, Université de Florence, Italie
Mr Pietro Pala	Co-Encadrant de la thèse	Professeur, Université de Florence, Italie
Mme Djamila Aouadi	Rapporteure	Professeure, Université du Luxembourg, Luxembourg
Mr Denis Hamad	Rapporteur	Président, Professeur, Université du Littoral Côte d'Opale, France
Mme Zakia Hammal	Examinatrice	Professeure Assistant, Université Carnegie-Mellon, États-Unis
Mr Claudio Ferrari	Examinateur	Professeur Assistant, Université de Parme, Italie

Abstract

Human behavior understanding has been an important research topic in the past decades. Indeed, the development of machines that work and help humans in their daily lives has never been more important than it is today. It is important to develop appropriate methods to better understand human behavior. In this sense, recent breakthroughs in computer science and computer vision have made the development of such methods possible. Understanding body and facial movements can be done by detecting 2D or 3D landmarks from different sources like a video or the feed of a camera. Performing this acquisition process over time makes it possible to construct temporal sequences of landmark configurations that can be processed to address different tasks, including the recognition of actions and emotions. However, deformations can be observed during the analysis, due to view variations, inaccurate landmark detection or tracking, especially in uncontrolled situations. In this thesis, we propose a space-time approach of body joint and facial landmark sequences, while tackling different problems in understanding of human behavior. We propose a representation based on trajectories of Gram matrices computed from body joints or facial landmarks. The Gram matrices representation defines positive semi-definite matrices of fixed rank that lay on a non-linear Riemannian manifold, where traditional computations and machine learning techniques could not be applied. To overcome this issue, the trajectories defined by sequences of Gram matrices on the manifold of SemiPositive definite matrices are analyzed by considering metric properties induced by the Riemannian geometry of the manifold. The proposed approach was evaluated in several applications related to body movements and action recognition from skeletons using 2D and 3D body joints as well as facial expression analysis to estimate the level of pain directly from 2D facial

landmarks.

Résumé

La compréhension du comportement humain est sujet de recherche important depuis plusieurs années. En effet, le développement de nouvelles machines qui travaillent et aident les humains dans leur quotidien n'a jamais été aussi important aujourd'hui. Il est alors important de développer des méthodes appropriées pour une meilleure compréhension du comportement humain. Dans ce sens, les récents progrès en informatique et en vision par ordinateur ont permit le développement de ces méthodes. La compréhension des mouvements du corps et du visage peut être effectuée par la détection de points de repères 2D ou 3D à partir de différentes sources comme une vidéo or le flux d'une caméra. Cette acquisition nous permet de construire une séquence temporelle de configurations de points de repères qui peuvent être traitées pour répondre à différents problèmes, comme la reconnaissance d'actions ou d'émotions. Cependant, des déformations peuvent être observées pendant l'analyse, du fait des changements de point de vue, la détection ou le suivi incorrecte des points de repères, particulièrement dans les situations non contrallées. Dans cette thèse, nous proposons une approche spatio-temporelles basées sur les points de repères du corps et du visage. La représentation avec des matrices de Gram définie des matrices définient semi-positives de rang fixe qui vivent sur des variétés Riemannienne non linéaires, sur lesquelles les techniques classiques de calculs et d'apprentissages machine ne peuvent pas être appliquées. Pour surmonter ce problème, les trajectoires définissent par des séquences de matrices de Gram sur la variété des matrices définissent semi-positives sont analysées en considérant une métrique qui respecte la géométrie Riemmanienne sur la variété. L'approche proposée a été évaluée sur différentes applications d'analyse du mouvement du corps et de la reconnaissance d'action à partir de points de

repères sur le corps en 2D et 3D, ainsi que sur l'analyse d'expressions faciales pour estimer le niveau de douleur à partir de points de repères faciaux.

Riassunto

La comprensione del comportamento umano è stata un importante argomento di ricerca negli ultimi decenni. In effetti, lo sviluppo di macchine che funzionano e aiutano gli esseri umani nella loro vita quotidiana non è mai stato così importante come lo è oggi. È importante sviluppare metodi appropriati per comprendere meglio il comportamento umano. In questo senso, le recenti scoperte nell'informatica e nella visione artificiale hanno reso possibile lo sviluppo di tali metodi. È possibile comprendere i movimenti del corpo e del viso rilevando punti di riferimento 2D o 3D da diverse fonti come un video o il feed di una telecamera. L'esecuzione di questo processo di acquisizione nel tempo consente di costruire sequenze temporali di configurazioni di riferimento che possono essere elaborate per affrontare diversi compiti, incluso il riconoscimento di azioni ed emozioni. Tuttavia, durante l'analisi possono essere osservate deformazioni dovute a variazioni della vista, rilevamento o tracciamento imprecisi del punto di riferimento, soprattutto in situazioni non controllate. In questa tesi, proponiamo un approccio spazio-temporale delle sequenze di punti di riferimento articolari e facciali del corpo, affrontando diversi problemi nella comprensione del comportamento umano. Proponiamo una rappresentazione basata su traiettorie di matrici di Gram calcolate da articolazioni del corpo o punti di riferimento facciali. La rappresentazione delle matrici di Gram definisce matrici semidefinite positive di rango fisso che giacciono su una varietà riemanniana non lineare, dove non è stato possibile applicare i calcoli tradizionali e le tecniche di apprendimento automatico. Per ovviare a questo problema, le traiettorie definite da sequenze di matrici di Gram sulla varietà di matrici definite SemiPositive vengono analizzate considerando le proprietà metriche indotte dalla geometria riemanniana della varietà. L'approccio proposto è stato

valutato in diverse applicazioni relative ai movimenti del corpo e al riconoscimento dell'azione degli scheletri utilizzando le articolazioni del corpo 2D e 3D, nonché l'analisi dell'espressione facciale per stimare il livello di dolore direttamente dai punti di riferimento facciali 2D.

Acknowledgements

First of all, I would like to thank my thesis director, Prof. Mohammad Daoudi. He always encouraged me to go further in my research, while being a teacher. Even in the difficult moments of this thesis, he knew how to find the words to motivate me. We had long discussions during which we exchanged on this work, which would not be there today.

I would also like to thank my thesis co-director Prof. Alberto Del Bimbo from the University of Florence, for welcoming me to the MICC laboratory and to the University of Florence for this cotutelle. The discussions we were able to have allowed me to integrate into the laboratory.

I also thank Prof. Stefano Berretti and Prof. Pietro Pala from the University of Florence, my co-supervisors, with whom we had many meetings and discussions to advance this work. Their advice and support allowed me to better understand certain aspects of this thesis.

I would like to thank Dr. Zakia Hammal from Carnegie Mellon University in the United States for the collaboration we had during this thesis. Our exchanges allowed me to better understand certain aspects of this work, in particular on the more medical part.

Special thanks to the members of my PhD committee for their time and evaluation of this work: Prof. Djamila Aouadi, Prof. Denis Hamad, Dr Zakia Hammal and Dr Claudio Ferrari.

This thesis was funded by the Hauts-de-France region and the University of Florence, Italy.

I would also like to thank all my colleagues from the GT-Image group for the exchanges we were able to have, in particular during the presentations of my work during the monthly meetings. I would also like to thank the PhD students of the group with whom I had a great time and for the moral support they represent.

Finally, I would like to thank my parents Véronique and Ludovic, and my sister Marie who have always supported me during this thesis. They always listened to what I told them about this work even if they didn't understand everything, which helped me a lot to externalize what I could feel, especially in the difficult times we experienced. This work would not have been the same without their support.

Table of Contents

List of Figures

List of Tables

Introduction

Human behavior understanding has been an important research topic in the past decades. Indeed, the development of machines that work and help humans in their daily lives has never been more important than it is today. It is important to develop appropriate methods to better understand human behavior. In this sense, recent breakthroughs in computer science and computer vision have made the development of such methods possible. Understanding body and facial movements can be done by detecting 2D or 3D landmarks from different sources like a pre-recorded video or in real-time from the feed of a camera. Performing this acquisition process over time makes it possible to construct temporal sequences of landmark configurations that can be processed to address different tasks, including the recognition of actions and emotions. However, deformations can be observed during the analysis, due to view variations, inaccurate landmark detection or tracking, especially in uncontrolled situations. To overcome these issues, multiple approaches have been adopted by using kernel methods [59, 5], different deep learning approaches [51, 49, 107, 71] or geometric methods based on the Riemannian geometry [13, 69, 41].

The breakthrough of deep learning sees a multiplication of the proposed approaches using different types of architectures. While being powerful tools to overcome the problems of action recognition or emotions recognition, some limitations appeared. The amount of data needed to train complex architectures, like Convolutional Neural Networks (CNN), makes them time consuming to train and powerful hardware is always necessary to reduce these training times. Another issue with these approaches is the lack of interpretation. Deep learning architecture are seen as black box and it can be difficult to fully understand what is treated in the hidden layers. In comparison,

geometric approaches are easy to interpret. The use of the Riemannian geometry allow us to use extracted features that can lie on non-linear manifolds by defining tools to work directly on these manifolds

In this thesis we will explore the advantages of using a geometric framework based on the Riemannian geometry to manage sequences of landmark configurations by proposing a space-time approach to tackle the problems of action recognition and pain estimation by using 2D and 3D landmarks on the human body or face extracted from sequences. The Riemannian geometry has been well study in the literature [7, 39, 89, 90, 62, 64] and some of its properties can be exploited to build a method capable of managing and analyzing landmark sequences. The main contributions of this thesis can be summarized to:

- The use of a spatial-temporal representation of human body joints sequences based on Gram matrix trajectories of landmark configurations for action recognition. The Gram matrices of n body joints lies on the non-linear manifold of Positive Semi-Definite (PSD) matrices of fixed rank d, where $d = 2$ or $d = 3$ for 2D and 3D landmark configurations respectively. As standard computational machine learning tools are not applicable to this kind of representation, we conducted a comprehensive study of the Riemannian geometry of the PSD manifold in order to find the most suitable tools to analyze and classify Gram matrix trajectories. This approach was tested on several benchmarks and we demonstrate its competitiveness compared to other approaches of the state-of-the-art. This work was published in publication P1.

- The proposed framework on Gram matrix trajectories was then applied in a different task, that is pain estimation from 2D tracked facial landmarks. Moving from body joints to facial landmarks may increase the complexity of the problem as the number of tracked landmarks is increasing. Furthermore, the range of motion of this kind of landmarks compared to body joint is different and the representation must be refined. Finally, the process of estimating pain level from sequences differ from a classification problem, therefore, we will use a regression model that is suitable for our representation. We demonstrate this approach on

a widely used benchmarks showing competitive results with respect to the state-of-the-art. This work was published in publication P2.

- The previous was then refined to take into account the localization of pain on the human face by splitting it into different region. Therefore, each region is considered independent and the Gram matrix representation of the landmarks of a specific region lies on different manifolds. A study to merge the results of the different manifolds was conducted and the effectiveness of this approach was demonstrated on several benchmarks. This work was published in publication P3.

The rest of the manuscript is organized as follows: Chapter 1 introduces the concept of Human Behavior Understanding and the use of tracked body joints or facial landmarks sequences to tackle this problem, followed by a review of different state-of-the-art approaches. In Chapter 2, we will present a geometric framework based on Gram matrix trajectories to analyze 2D and 3D tracked body joints and facial landmarks to tackle the problem of action recognition and pain estimation. In Chapter 3 we present the different experiments we conducted with a presentation of the datasets we use as well as the different protocols to test our approach explained in the previously mentioned. Chapter 4 will introduce an extension of the framework presented in Chapter 2 by defining multiple manifolds to analyze sequences of facial landmark configurations from different region of the face to tackle the problem of pain estimation. Finally, Chapter 5 we will conclude this manuscript and expose its limitations and present future work.

List of Publications

P1: **B. Szczapa**, M. Daoudi, S. Berretti, A. Del Bimbo, P. Pala and E. Massart, Fitting, Comparison, and Alignment of Trajectories on Positive Semi-Definite Matrices with Application to Action Recognition, International Workshop on Human Behaviour Understanding at IEEE International Conference on Computer Vision Workshops (ICCVW 2019)

P2: **B. Szczapa**, M. Daoudi, S. Berretti, P. Pala, A. Del Bimbo and Z. Hammal, Automatic Estimation of Self-Reported Pain by Interpretable Representations of Motion Dynamics, IEEE International Conference on Pattern Recognition (ICPR 2020)

P3: **B. Szczapa**, M. Daoudi, S. Berretti, P. Pala, A. Del Bimbo and Z. Hammal, Automatic Estimation of Self-Reported Pain by Trajectory Analysis in the Manifold of Fixed Rank Positive Semi-Definite Matrices, IEEE Transactions on Affective Computing (TAC), paper accepted September 15 2022.

Chapter 1

Related work on Landmark-based Human Behavior Analysis

1.1 Introduction

Detecting and tracking a set of landmarks to use them during the analysis of a sequence is a common task to better understand the human behavior. Two commonly used landmarks are the body joints, to analyze movements of the human body to recognize or predict actions, and the facial landmarks on the human face to analyze patterns for emotion understanding. The problem of analyzing a sequence becomes an analysis of the motion of the tracked points. In this chapter, we will introduce the definition of human behavior understanding and its application in real-world. We will then expose the challenges of using human body or facial landmarks for this specific task and analyze the different approaches presented in the state-of-the-art.

1.2 Human Behavior Definition

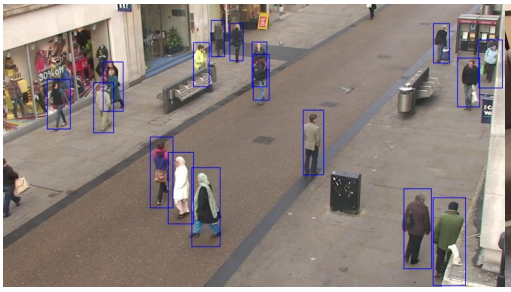
Human behavior is the expression of individuals or groups of human to respond to internal and external stimuli. Theses responses consist of a set of changes in the physiological activity along time. Theses changes can last a few milliseconds like blinking or several minutes like sitting. Behavior can also be driven by affective states (*e.g.* joy, fear or disgust) or manipulations to act on an object in an environment, with someone else like handshaking or with itself like applauding.

In this thesis, we are interested in the development of new methods allowing intelligent systems to understand some of these signals from visual data. We would like to make machines able to automatically recognize the nature of the signal (*e.g. an action or an emotion*) by analyzing a behavioral signals in a video sequence.

Human behavior understanding covers multiple applications if different fields such as:

- Security: The monitoring of a crowd has become usual in different places like train station or airport to prevent accidents or threats. Understand the human behavior can help by analyzing and anticipating dangerous interactions or suspected persons (Figure 1.1a).
- Human-Computer Interaction: The interface between the human and the machine is a crucial points to facilitate the interactions between the two as we interact with a computer in many ways. Conventional interface devices like a keyboard or a display assumed that the human will be attentive during the control flow. Development of better human behavior understanding tools can significantly improve the interfaces between humans and the machines, providing more effective interactions (Figure 1.1b).
- Healthcare: The development of intelligent systems that assist clinicians in their diagnosis has become important during the past decades. By a better understanding of the human behavior, the machines can now automatically measure the pain intensity or the level of depression severity, helping clinicians to effectively apply treatments (Figure 1.1c).

- Psychology: Human behavior understanding is important in the field of psychology as the practitioners can better understand the affective states of a person. Analyzing emotions from facial expressions can help them in their diagnosis.



(a) Security

(b) Human-Computer-Interaction

(c) Healthcare

Figure 1.1: Examples of different fields of application for human behavior understanding

Facial expression and body movements analysis from video sequences are two basics tasks in human behavior understanding. However, other modalities can be used like audio, written expressions or physiological measurements in order to improve the understanding of human behavior.

In this thesis, we will focus our work on recognizing actions based on the human bodies and estimating the pain level of a patient from human faces using landmarks extracted from video sequences.

1.3 The Use of Human Landmarks

A broad range of applications in Computer Vision make use of tracked landmarks from visual data. The source of these visual data can vary, and some examples are given in Figure 1.2.

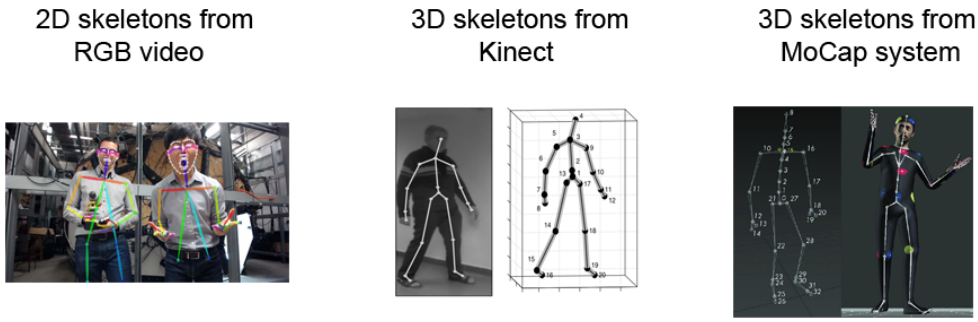


Figure 1.2: Examples of body skeletons from different modalities [10, 75, 12]

The simplest way to capture human body skeleton is with the use of methods to estimate the pose from RGB videos. Recent approaches made the tracking of 2D or 3D skeletons from 2D RBG videos possible, with good performance [10, 52]. For each frame of a sequence, a set of landmarks is detected on some articulation of the human skeleton, forming a sequence of landmark configurations. To better estimate the 3D location of the body joints composing a skeleton, the use of depth sensors, like the Kinect V2, or specific methods [4, 18] is important. Like the use of RGB videos, the analysis of the human body in a depth video turned to detect and extract a set of landmarks for each frame. More sophisticated solutions to detect and track 3D landmarks make use of multiple cameras at the same to recreate the depth [76] or uses infra-red camera to track body markers [12]. This last solution is mainly used in motion capture systems, but they are expensive and requires a lot of computing power, even if they can detect a large amount of joints with accurate estimations.

Another example of human landmarks tracking is represented by the face, where several approaches have been proposed to detect and track facial points from videos. Most of the methods detect a set of 2D facial landmarks, localized at important positions of the human face like around the eyes, eyebrows, nose and mouth. The number of detected landmarks can vary from

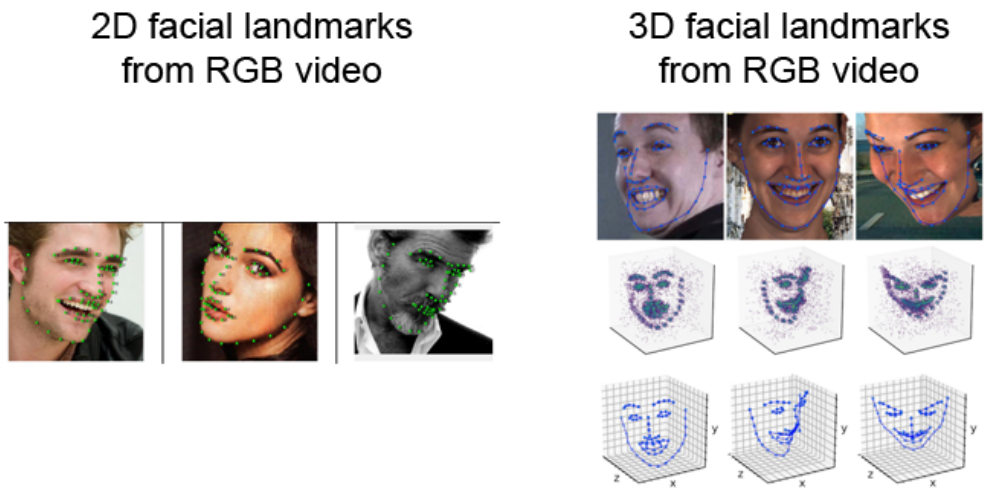


Figure 1.3: Examples of 2D and 3D facial landmarks [47, 105]

one approach to another, for example by adding the landmark detection of the jaw. However, the detection of 2D facial landmarks can lead to distortions due to pose variations. To overcome this problem, some approaches tried to estimate the 3D locations of the landmarks from RGB videos the same way some approaches track 3D skeletons. Finally, to better estimate the 3D locations of facial landmarks, some approaches make use of 3D scanners that will extract a large amount of points to form a mesh of the face [ref coming]. However, these scanners are expensive and can require multiple scans in order to produce a complete model of the scanned face.

By today standards, most of the proposed methods in the state-of-the-art are real-time solution to track body or facial landmarks with impressive performance.

1.4 Motivations and Challenges

In this thesis, we will focus on the use of an effective landmark based solution to tackle some human behavior understanding tasks such as action recognition or pain estimation. We chose to work with human landmarks for multiple reasons:

- The recent advances in human landmark tracking make them reliable to use. Modern body and facial landmarks tracking methods are accurate and robust to illumination changes that can occur in RGB videos. Some of the recent approaches are also robust to occlusions, particularly for the tracking of facial landmarks (*e.g. a person is wearing sunglasses or a mask*).

- The use of human landmarks instead of the full images of a RGB video reduces the complexity of the visual data. A video contains a large amount of pixels in each of its frame, which could make the analysis of such data complex and computationally expensive. Moreover, landmarks bring an overview of the frame by providing a set of relevant 2D or 3D points, making landmark solutions less computationally expensive, more efficient and suitable for real-time applications.

- As the use of landmarks only provide relevant information in each frame and not the full set of features from the pixels, landmarks bring a certain privacy, which can be necessary in some domains like healthcare where confidential data are very important.

While these approaches are powerful and robust, several challenges can emerge from the tracking techniques to generate sequences of landmark configuration:

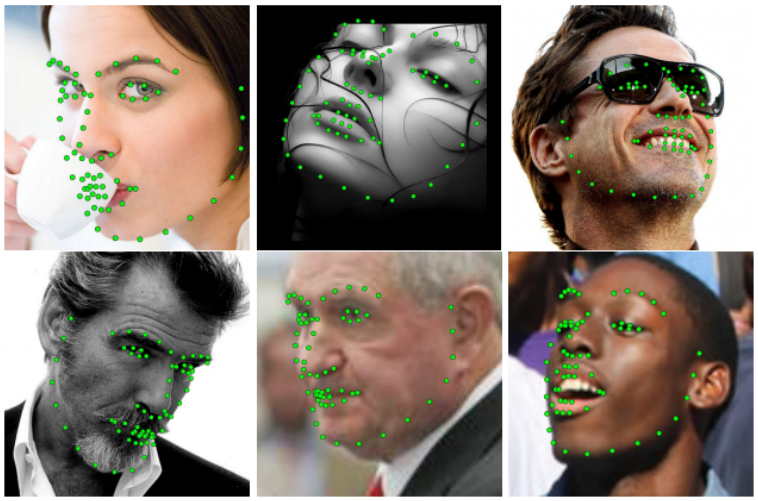


Figure 1.4: Challenging examples of 2D facial landmarks (occluded face, view variation) [45]

- Tracking issue: Despite the advances in the field of tracking human landmarks, inaccurate tracking or missing data can occur. This issue is especially visible in unconstrained environment or challenging conditions like detecting facial landmarks from an occluded face (see first image in Figure 1.4).
- View variations: The coordinates of the 2D or 3D landmarks are given by the position of the camera. However, the signals belonging to the same category can occur un different positions with respect to the camera. These variations prevent us from using the original landmarks locations directly and we must first filter and normalize the landmark coordinates in order to analyze the human behavior signals. These view variations can be seen as rigid transformations affecting the landmarks like translations, rotations, etc (see Figure 1.4).
- Speed variations: The behavioral signals are subject to high temporal variations. Considering the same action, two persons will not perform it at the same speed. Like the view variations, we can not use the original data directly and take into account these temporal variations during the analysis of landmark sequences. Some methods are designed to filter these temporal variations by aligning the frames of two sequences like Dynamic Time Warping (DTW) [34].
- Intra-class variations: Finally, large variations can be observed within the same category as two person will not always perform an action the same way. This is also true for one person not performing the same exact action twice, making the behavioral signals different.

Many efforts have been made to make the analysis of landmark sequences efficient. However, the issues described above are far from being solved, even if the methods are more and more efficient.

1.5 Related Work on Modeling Landmarks Sequences

In this section, we review some recent state-of-the-art methods to analyze sequences of human facial or body landmarks on the task on human behavior understanding. We will focus on works

that use 2D or 3D landmarks from faces or bodies as input data. The state-of-the-art methods presented are organized in three different categories: kernel methods, deep learning approaches and Riemannian methods.

1.5.1 Kernel Methods

Kernel methods have proved to be powerful for many Computer Vision tasks. The concept is based on the definition of similarities between objects and more specifically by predicting properties of new objects based on the one already known [48]. Based on this idea, Lorincz *et al.* [59] proposed a two time series kernels computed from 3D facial landmarks for expression recognition. They considered temporal evolution of normalized 3D facial landmark configurations as a time series by using a kernel based on Dynamic Time Warping (DTW) [34] representing the similarities of all the sequences in a dataset. Dynamic Time Warping is based on dynamic programming to allow temporal alignment of two sequences. However, the kernel derived from DTW is not positive definite and does not satisfy Mercer's theorem. The authors thus considered an approximated version of this kernel. To obtain a positive definite kernel, they also employ a Global Alignment Kernel (GAK) [14]. To tackle the problem of 3D action recognition, Bagheri *et al.* [5] propose to compute a DTW kernel which was not approximated. They also introduced a kernel based on Longest Common Subsequence Similarity (LCSS), consisting of counting the number of pairs that match from two sequences. Differently from [59] the authors used a variant of SVM, called Pairwise Proximinty Function SVM (ppfSVM) [32]. This variant learns a proximity model of the data and the only constraint is to define a proximity function which can be the original DTW or LCSS measures.

1.5.2 Deep Learning Approaches

Over the last decade, Deep Learning approaches became the most powerful and commonly used tools for Computer Vision. Many recent approaches used deep leaning to analyze landmark se-

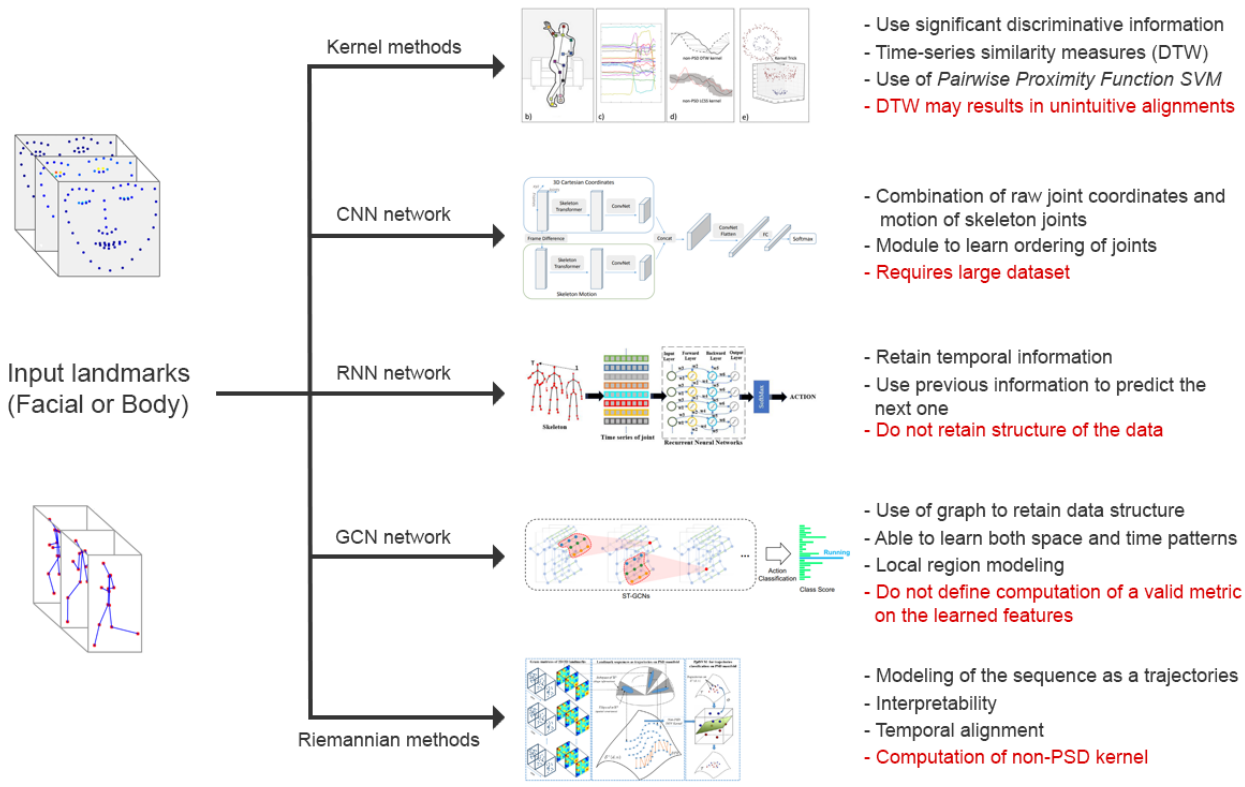


Figure 1.5: Overview of different approaches to process landmark sequences

quences by defining network architectures that model the dynamics of the landmark configurations in sequences to classify them for the task of human behavior understanding. These approaches can be organized in three categories: Feed-Forward Neural Networks based on different architectures such as Convolutional Neural Networks (CNN) or Graph Convolutional Networks (GCN); Recurrent Neural Networks (RNN) that uses recurrent connections to keep the information of previous activations that are propagated over time and Transformer Neural Network that are based on the attention mechanism to handle sequential data.

Feed-Forward Neural Networks (CNN, GCN): For the task of expression recognition, Liu *et al.* [53] proposed a two-stage personalized model, to analyze facial landmark sequences for automatic estimation of the self-reported pain score. This approach is based on the combination of a Neural Network and a Gaussian process regression model, and is used to personalize the estima-

tion of self-reported pain via a set of hand-crafted personal features and multitask learning. As an alternative, some approaches introduce Convolutional Neural Networks (CNNs) [51] and Graph Convolutional Networks (GCNs) [101, 49] in the overall architecture so as to retain the structural information among joints of the skeleton. Although these approaches result in state-of-the-art performance [26] on public action recognition benchmarks, it is not possible to define a formal mathematical framework to compute a valid metric on the internal, learned feature representation so as to perform a statistical analysis of the learned actions.

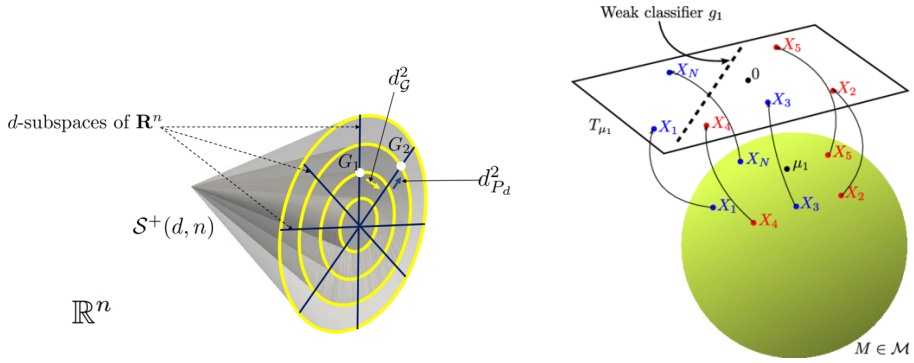
Recurrent Neural Networks: Several solutions have experimented the application of Recurrent Neural Networks (RNNs) and Long Short Term Memory (LSTM) networks to the case of 2D or 3D human landmarks for human behavior understanding. Recurrent neural networks (RNNs) and particularly Long-Short-Term Memory Networks (LSTMs) have been used to perform action recognition by the analysis of sequences of skeleton poses [107]. However, these methods typically lose structural information when converting the skeleton data and joint connectivity into the vector-shaped input of the neural network. For expression recognition and estimation, Martinez *et al.* [61] [61], the authors proposed a two-step learning approach to estimate pain scores from extracted 2D facial landmark sequences. The authors employed a Recurrent Neural Network (RNN) to first estimate the pain scores at frame-level and then fed into a personalized Hidden Conditional Random Fields (HCRF) these scores to derive a pain score for the entire sequence. On the same problem, Erekat *et al.* [22] proposed a spatio-temporal Convolutional Neural Network - Recurrent Neural Network (CNN-RNN) for the automatic measurement of self reported pain and observed pain intensity, respectively, from facial landmarks. The authors proposed a new loss function that explores the added value of combining different self reported pain scales for a reliable assessment of pain intensity from facial expression. Using an automatic spatio-temporal architecture, they proposed a reliable assessment of pain by maximizing the consistency between different pain assessment scales.

Transformer Neural Network: Following the recent trend of Transformer, Plizzari *et al.* [71] propose a spatio-temporal transformer architecture for the task of action recognition. The authors present an architecture that combines both spatial and temporal attention from body joint sequences. More recently, Mazzia *et al.* [66] present a fully self-attentional network exploiting 2D pose representations. These new approaches demonstrated the performance of such architectures, competing with more traditional neural network architectures like CNN, RNN or GCN. However, these approaches need a large amount of data to be effective.

1.5.3 Riemannian Methods

The drawback of the approaches using deep learning methods is that they are not taking into account the geometry of the data. The extracted landmark features may lie on non-linear manifolds where traditional machine learning techniques are difficult to use, although some recent works try to implement Riemannian methods in combination with deep learning approaches [13, 50]. In this case, it is suitable to use methods based on the Riemannian geometry that defines tools to work directly on non-linear manifolds. To overcome this problem, some works used the Riemannian geometry [6, 69, 41]. Different examples of manifolds are presented in Figure 1.6. To effectively use the Riemannian geometry, we need to define a smoothly varying inner product on each tangent space of the manifold as a metric. By defining this Riemannian metric, we can locally use the vector space structure of the tangent space to define geometric notions like the geodesic distance which defines the length of the shortest path connecting two points on the manifold. This distance allows us to measure the proximity of the feature points on the manifold. Exploiting other notions like the logarithm and exponential maps allowing us to map a point of the manifold to a tangent space and back onto the manifold is also important. We present here methods that use the Riemannian geometry in order to study human behavior by taking into account the geometry of the data represented by facial or body landmarks.

One promising idea is to formulate the motion features as trajectories on the underlying manifolds. Indeed, features computed from static landmark configurations often lie on non-linear



(a) Cone Manifold [41] (b) Sphere Manifold [79]

Figure 1.6: Different types of Riemannian manifolds

manifolds [6, 87, 92, 93]. Hence, landmark sequences can be seen as trajectories on this manifold. In contrast to the first family of Riemannian methods, the temporal structure of landmark sequences is preserved allowing desirable operations in the manifold such us interpolation. One of the first approaches to perform action recognition by the analysis of trajectories of tracked body interest points was presented in Matikainen *et al.* [65]. Despite the promising results obtained, the authors did not take into account the geometric information of the trajectories. More recently, in the case of human skeleton in RGB-D images, Devanne *et al.* [17] proposed to formulate the action recognition task as the problem of computing a distance between trajectories generated by the joints moving during the action. An action is then interpreted as a normalized parameterized curve in $\mathbb{R}^N$. However, this approach does not take into account the relationship between the joints. In the same direction, Su *et al.* [83] proposed a metric that considers time-warping on a Riemannian manifold, thus allowing the registration of trajectories and the computation of statistics on the trajectories. Su *et al.* [84] applied this framework to the problem of visual speech recognition. Similar ideas have been developed by Ben Amor *et al.* [6] on the Kendall's shape space with application to action recognition using rate-invariant analysis of skeletal shape trajectories. Anirudh *et al.* [3] started from the framework of Transported Square-Root Velocity Fields (TSRVF), which has desirable properties including a rate-invariant metric and vector space representation. Based on this framework, they proposed to learn an embedding such that each action trajectory is mapped to a single point in a low-dimensional Euclidean space, and the trajectories that differ only in the temporal rate map to the same point.

The TSRVF representation and accompanying statistical summaries of Riemannian trajectories are used to extend existing coding methods such as PCA, KSVD, and Label Consistent KSVD to Riemannian trajectories. In the experiments, it is demonstrated that such coding efficiently captures distinguishing features of the trajectories, enabling action recognition, stroke rehabilitation, visual speech recognition, clustering, and diverse sequence sampling. In [92], Vemulapalli *et al.*proposed a Lie group trajectory representation of the skeletal data on a product space of special Euclidean (SE) groups. For each frame, this representation is obtained by computing the Euclidean transformation matrices encoding rotations and translations between different joint pairs. The temporal evolution of these matrices is seen as a trajectory on $SE(3) \times \cdots \times SE(3)$ and mapped to the tangent space of a reference point. A one-versus-all SVM, combined with Dynamic Time Warping and Fourier Temporal Pyramid (FTP) is used for classification. One limitation of this method is that mapping trajectories to a common tangent space using the logarithm map could result in significant approximation errors. Aware of this limitation, in [93] the same authors proposed a mapping combining the usual logarithm map with a rolling map that guarantees a better flattening of trajectories on Lie groups. More recently, Kacem *et al.* [40] proposed a geometric approach for modeling and classifying dynamic 2D and 3D landmark sequences based on Gramian matrices derived from the static landmarks. This results in an affine-invariant representation of the data. Since Gramian matrices are positive-semidefinite, the authors relies on the geometry of the manifold of fixed-rank positive-semidefinite matrices, and more specifically, to the metric investigated in [7]. However, this metric is parametrized, and the parameter should ideally be learned from the data. In addition, this paper adopts Dynamic Time Warping for sequence alignment. The resulting distance does not generally lead to a positive-definite kernel for classification.

1.6 Related Work on Pain Estimation

One of the problem we tackle in this thesis is the problem of pain estimation. Estimating the pain intensity can help the practitioners to better understand the suffering of the patients. In some situation, it necessary to understand this pain from visual features as the patient is not capable of

expressing it verbally. We describe here recent approaches to tackle this problem by using facial features.

Significant efforts have been made in human behavioral studies to identify reliable and valid facial indicators of pain [74, 23, 46, 73]. In these studies, pain expression and intensity were characterized at the frame level by highly trained human coders that annotated anatomical facial actions using the Facial Action Coding System (FACS) [21]. However, manual FACS based pain assessment requires over a hundred hours of training for FACS certification, and approximately an hour or more to manually annotate a minute of video. The intensive time required to annotate videos using the manual FACS makes it ill suited for daily clinical use. This limitation lead to the emergence of considerable efforts in computer vision and machine learning for automatic pain assessment of self-reported pain (*i.e.*, VAS based measurement) and observed pain (*i.e.*, FACS based), respectively [35]. Since the goal of our work is to automatically assess the self-reported pain, the state-of-the-art on FACS based pain assessment is not reviewed here (see [35] for a detailed review).

Using the UNBC-McMaster Shoulder Pain Archive [60] a few recent efforts have investigated video based measurement of self-reported Visual Analog Scale (VAS) pain intensity scores. The VAS is a self-reported pain score that indicates on a 0 to 10 scale the intensity of pain (where 0 corresponds to no pain, and 10 to the worst possible pain). For instance, Martinez *et al.* [61] proposed a two-step learning approach to estimate pain scores consistent with the self-reported VAS. The authors employed a Recurrent Neural Network (RNN) to first estimate the Prkachin and Solomon Pain Intensity score (PSPI) at frame-level from face images. The estimated PSPI scores were then fed into a personalized Hidden Conditional Random Fields (HCRF) to derive a pain score consistent with the VAS. Liu *et al.* [53] proposed a two-stage personalized model, named DeepFaceLIFT, for automatic estimation of the self-reported VAS pain score. This approach is based on a Neural Network and a Gaussian process regression model, and is used to personalize the estimation of self-reported pain via a set of hand-crafted personal features and multitask learning. Xu *et al.* [99] proposed a three-stage multitask pain model to estimate self-reported pain scores. First, a VGGFace neural network is used to predict frame-level PSPI based pain scores. Second, a fully connected neural network is employed to estimate the VAS at sequence-level from

frame-level PSPI predictions using multitask learning to learn multidimensional pain scales instead of the VAS for the entire sequence. Finally, an optimal linear combination of the multidimensional sequence-level VAS was used to predict the final VAS based pain score. Xu *et al.* [100] further refined the work in [99] by using the four labels available in the dataset (*i.e.*, VAS, AFF, SEN and OPR) to estimate the level of pain from human-labeled Action Units. The authors combined the use of multitask learning neural network to predict pain scores with an ensemble learning model to linearly combine the multi-dimensional pain scores to estimate the VAS. Erekat *et al.* [22] proposed a spatio-temporal Convolutional Neural Network - Recurrent Neural Network (CNN-RNN) for the automatic measurement of self reported pain and observed pain intensity, respectively. The authors proposed a new loss function that explores the added value of combining different self reported pain scales for a reliable assessment of pain intensity from facial expression. Using an automatic spatio-temporal architecture, they proposed a reliable assessment of pain by maximizing the consistency between different pain assessment scales. Their results show that enforcing the consistency between different self-reported pain intensity scores collected using different pain scales enhances self-reported pain estimation.

All the previously mentioned methods make use of (deep-)neural networks or try to estimate pain intensity at frame level first (PSPI scores), and predict the sequence level pain index from this first estimation.

Only few works investigated video or geometric based approaches to estimate self-reported pain using the Biovid Heat Pain dataset [95]. In this dataset, the self-reported pain ranges from 0 to 4 (where 0 corresponds to no pain and 4 to high level of pain). This dataset is composed of different parts that come with several modalities such as long or short video sequences and biomedical signals like ECG, EMG or skin conductance. Much of the work that has been done with this dataset used the Part A, which comes with biomedical signals and short sequences to estimate the pain intensity at sequence level. Skin conductance was used by Pouromran *et al.* [72] or Lopez-Martinez and Picard [57]. Other approaches like the one proposed by Kachele *et al.* [42] tested the combination of different modalities. They also extracted the facial landmarks and computed several statistical geometric features from the raw coordinates. In a different way, Lopez-Martinez *et*

al. [58] also extracted facial landmarks in order to compute statistical geometric features and combine them with the biomedical signals to estimate the levels of pain. However, they do not use the short video sequences available in the Part A of the dataset.

1.7 Conclusion

Motivated by the recent advances in human landmarks detection, we focused our works on landmark based solution to study and understand human behaviors. In order to develop efficient approaches, we need to take into account several challenges like inaccurate tracking of the landmarks, views and rate variations as well as intra-class variations. We presented in this chapter different landmark based solutions from the existing literature, organized in different categories. Deep Learning approaches are powerful, yet still require a large amount of data and a high computational power to achieve good performance. Collecting large datasets for the task of human behavior understanding is difficult, especially when based on landmark sequences, where the landmarks have to be extracted beforehand, using time consuming methods (*i.e. hand crafted or fully automated methods on large scale data*). The drawback of such methods is that they are not taking into account the geometry of the data. The extracted landmark features may lie on non-linear manifolds where traditional machine learning techniques are not applicable. In this case, it is suitable to use methods based on the Riemannian geometry that defines tools to work directly on non-linear manifolds. To preserve the original temporal structure of the landmark sequences, we can represent the data as trajectories lying on a manifold and use the available tools of the Riemannian geometry to work on the manifolds directly. In this thesis, we will focus on Riemannian trajectory based representations of the landmark sequences for the tasks of action recognition and pain estimation.

Chapter 2

Fitting, Comparison, and Alignment of Trajectories in the Manifold of Fixed Rank Positive Semi-Definite Matrices

2.1 Introduction

In the last decades, automatic analysis of human motion has been an active research topic, with applications that have been exploited in a number of different contexts, including video surveillance, semantic annotation of videos, entertainment, human computer interaction and home care rehabilitation, to say a few. Differences in body proportion (size, height, corpulence), body stiffness and training, influence the way different people perform an action. Even one same person is not able to perform the same action twice, exactly replicating the same sequence of body poses in space and time. This variability makes the task of human motion analysis very challenging.

For years, the approaches could be distinguished in two main classes: those operating on pixel values extracted from the RGB stream (either stacking groups of consecutive frames or extracting motion vectors) and those building upon the higher level representation of body skele-

tons. These latter approaches were supported by the diffusion of low-cost RGB-D cameras (such as the Microsoft Kinect) that can operate in real-time, while reliably extracting the 3D coordinates of body joints. More recently, deep CNN architectures have demonstrated real-time and accurate extraction of the coordinates of body joints from RGB streams [11].

These advances make it possible to use a skeleton-based body representation in a much broader range of domains and operative contexts than before, being not limited by the short operative range of RGB-D sensors that typically operate indoor and in the range of a few meters. The design of the recognition/classification module on top of the body skeleton representation makes it possible to describe an action as a sequence of body poses, each one corresponding to a point in a feature space, whose dimension is proportional to the number of body joints. By exploiting the geometric properties of the manifold where these pose descriptors lie, it is possible to define a similarity metric that is invariant under translation, scaling, rotation and also under variations of the speed of execution of the action. Furthermore, the explicit representation of an action as a trajectory, *i.e.*, a sequence of poses, on the manifold makes it possible to extract statistical summaries, such as mean and deviation from the mean, from a group of actions. Through these summaries, one action can be better characterized for the purpose of detecting outliers corresponding to the anomalous execution of an action, that can be of particular relevance for action prediction. In fact, when analyzing the skeleton sequences, there are four main aspects to challenge: (1) A shape representation invariant to undesirable transformations; (2) A temporal modeling of landmark sequences; (3) A suitable rate-invariant distance between arbitrary sequences, and (4) A solution for temporal sequence classification.

Lying on the continuity of recent works, we modeled the comparison and classification of temporal sequences of landmarks on the Riemannian manifold of positive-semidefinite matrices. This formulation has shown promising results in action recognition [17, 6, 16] and in facial expression recognition [41]. Building on the work [40], our approach involves four different steps: 1) We build a trajectory on the Riemannian manifold from the body skeletons; 2) We apply a curve fitting algorithm on the trajectories to denoise the data points; 3) We perform a temporal alignment using a Global Alignment Kernel, defining a positive-semidefinite kernel; 4) Finally, we use this kernel

with a classic SVM to classify the actions. An overview of the full approach is given in Figure 2.1.

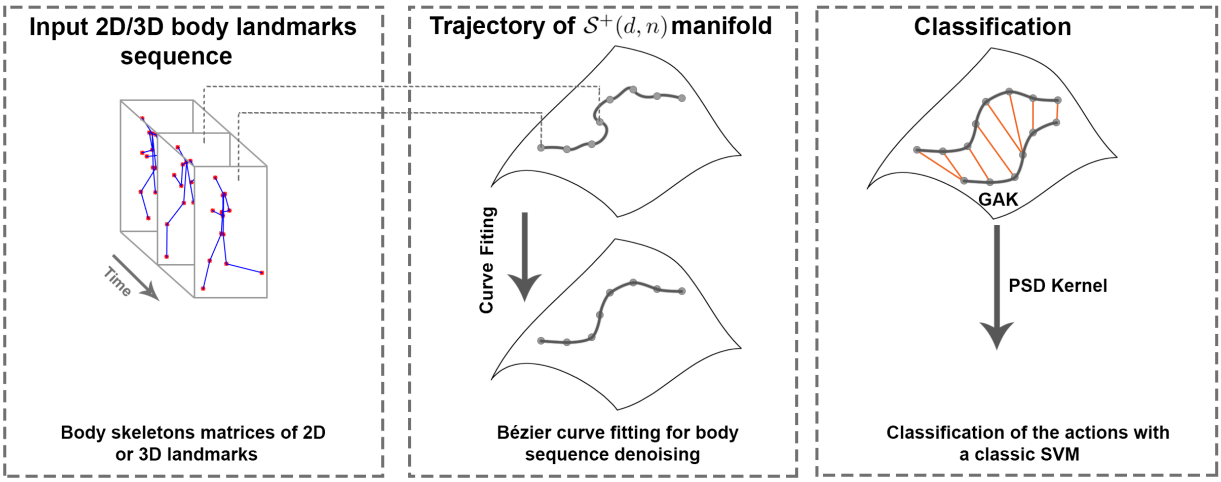


Figure 2.1: Overview of the proposed approach - After automatic body skeletons detection for each frame of a sequence, the Gram matrices are computed to build the trajectory on the $\mathcal{S}^+(d,n)$ manifold. We apply a curve fitting algorithm on the trajectory to smooth the curve and reduce noise. Global Alignment Kernel (GAK) is then used to align the trajectories on the manifold. Finally, we use the kernel generated from GAK with SVM to classify the actions.

The novelties presented in this work with respect to [40] are:

- The manifold of positive-semidefinite matrices is here endowed with a different metric;
- A recent curve fitting method is used to smooth trajectories on the manifold;
- We use Global Alignment Kernel for temporal alignment, instead of Dynamic Time Warping.

Differently, this approach can be applied to different kind of joints such as facial landmarks and used on a different task like expression recognition. We also present here the application of such framework for the task of pain estimation from facial landmarks.

Pain is an unpleasant sensory and emotional experience associated with actual or potential tissue damage caused by illness or injury [67]. Pain assessment is necessary for differential

diagnosis, choosing, monitoring, and evaluating treatment efficiency. The assessment of pain is accomplished primarily through subjective self-reports using different medical scales like the Visual Analog Scale (VAS)—the most commonly used scale in clinical assessment [1, 25, 36, 37]—or the Numerical Rating Scale (NRS) [104]. However, while useful, self-reported pain is difficult to interpret due to subjectivity and personal experiences, and may be impaired or, in some circumstances, not even possible to obtain, such as for children, cognitively impaired patients or patients requiring breathing assistance [35].

To improve assessment of pain and guide treatment, objective measurement of pain from nonverbal behavior (*i.e.*, facial expressions, head and body movements, and vocalizations) is emerging as a powerful option [35, 97].

Extensive behavioral research has documented reliable facial indicators of pain [74, 23, 46, 73]. The core facial movements that have been found to discriminate the presence from the absence of pain are brow lowering, orbit tightening, upper-lip raising, nose wrinkling, and eye closure [74]. Based on these findings, most efforts in automatic assessment of pain have focused on facial expression. Using either the Facial Action Coding System (FACS) [74] or the holistic dynamics of the face, computational models have been trained to learn the association between discriminative facial features and pain occurrence or intensity [35, 97].

Building upon previous efforts, our primary measure for computational pain assessment is the dynamics of pain related facial movements [74]. To capture changes in the dynamics of facial movement relevant to pain expression, we modified our framework to assess the temporal evolution of facial landmarks modeled as a trajectory on a Riemannian manifold. In our case, Gram matrices are computed from facial landmarks at each video frame and their temporal evolution is modeled as a trajectory on the Riemannian manifold of symmetric positive semi-definite (PSD) matrices. With this representation, pain estimation is modeled as the problem of computing similarity between trajectories on the manifold, then using a Support Vector Regression (SVR) [19] model to predict pain scores. The pipeline of the modified framework is presented in Figure 2.2.

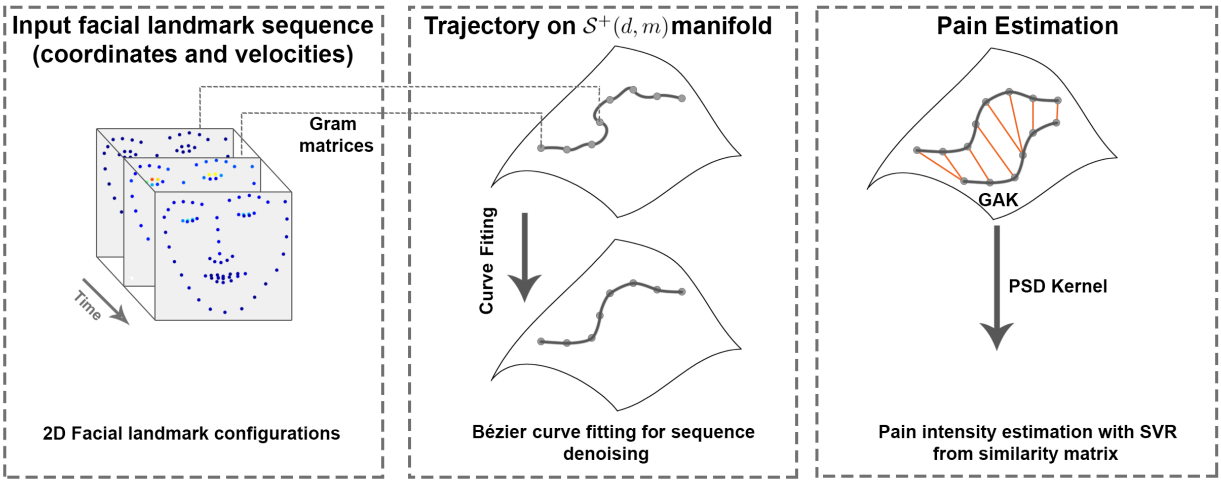


Figure 2.2: Overview of the proposed approach: (left plot) First, facial landmarks are detected using Active Appearance Model (AAM) on each video frame and velocities are computed as the displacement of the coordinates between two consecutive frames. Then Gram matrices are computed from the combination of the landmark coordinates and velocities. These matrices delineate a trajectory on the $\mathcal{S}^+(d, m)$ manifold; (middle plot) We apply a curve fitting algorithm to the trajectory for smoothing and noise reduction; (right plot) The Global Alignment Kernel (GAK) is then used to align the trajectories on the manifold, which results in a similarity score between the trajectories. Finally, we use the kernel generated from GAK with SVR to estimate the pain intensity.

In summary, the main differences of this work compared to the problem of action recognition are:

- The temporal dynamics of facial landmarks are modeled from position and velocity of each landmark as Gram matrix trajectories on the Positive Semi-Definite (PSD) manifold;
- We estimate pain score at sequence-level, rather than at frame-level.
- This pain estimation task relies on the use of a SVR instead of a SVM.

2.2 Presentation of the Approach

2.2.1 Gram Formulation

To represent body movement dynamics, we rely on the time series made of the coordinates of the n tracked body points (*i.e.*, $p_1 = (x_1, y_1), \ldots, p_n = (x_n, y_n)$ in 2D, or $p_1 = (x_1, y_1, z_1), \ldots, p_n = (x_n, y_n, z_n)$ in 3D), during each video sequence. Each video sequence is thus characterized by a set of landmark configurations $\{Z_0, \ldots, Z_\tau\}$, where τ is the number of frames of the video sequence, and where each configuration matrix Z_i $(1 \leq i \leq \tau) \in \mathrm{R}^{n \times d}$ encodes the position of the n landmarks in d dimensions (with $d = 2$ or $d = 3$). We aim to measure the dynamic changes of the curves made of the landmark configurations, remaining invariant to rotation and translation.

Similarly as in [40], this goal is achieved through a Gram matrix representation, where we compute the Gram matrices as:

$$G = ZZ^T \ . \tag{2.1}$$

These Gram matrices are $n \times n$ positive-semidefinite matrices, of rank smaller than or equal to d (always equal to d in the datasets considered). Conveniently for us, the Riemannian geometry of the space $\mathcal{S}^+(d, n)$ of $n \times n$ positive-semidefinite matrices of rank d has been studied in [7, 39, 89, 90, 62, 64], and used in, *e.g.*, [24, 68, 28, 63].

A classical approach in the design of algorithms on manifolds consists in resorting to first order local approximations on the manifold, called tangent spaces. This requires two tools: the Riemannian exponential (that allows us to map tangent vectors from the tangent space to the manifold), and the Riemannian logarithm (mapping points from the manifold to the tangent space).

In [40], the manifold $\mathcal{S}^+(d, n)$ is identified to the quotient manifold $(\mathrm{St}(d, n) \times \mathcal{P}_d)/\mathcal{O}_d$, where $\mathrm{St}(d, n) := \{Y \in \mathbb{R}^{n \times d} | Y^T Y = I_d\}$ is the Stiefel manifold, $\mathcal{P}_d$ is the manifold of $d \times d$ positive-definite matrices, and $\mathcal{O}_d$ is the orthogonal group in dimension d. We consider here another

representation of the manifold $\mathcal{S}^+(d,n)$, that will result in different expressions for the distance between two points, the Riemannian exponential and logarithm.

2.2.2 Gram Matrix Distance

We consider here the identification of $\mathcal{S}^+(d,n)$ to the quotient manifold $\mathbb{R}_*^{n\times d}/\mathcal{O}_d$, where $\mathbb{R}_*^{n\times d}$ is the set of full-rank $n \times d$ matrices. This geometry has been studied in [39, 62, 64].

The identification of $\mathcal{S}^+(d,n)$ with the quotient $\mathbb{R}_*^{n\times d}/\mathcal{O}_d$ comes from the following observation. Any PSD matrix $G \in \mathcal{S}^+(d,n)$ can be factorized as $G = ZZ^T$, with $Z \in \mathbb{R}_*^{n\times d}$. However, this factorization is not unique, as any matrix $\tilde{Z} := ZQ$, with $Q \in \mathcal{O}_d$, satisfies $\tilde{Z}\tilde{Z}^T = ZQQ^TZ^T = G$. The two points Z and $\tilde{Z}$ are thus *equivalent* with respect to this factorization, and the set of equivalent points

$$Z\mathcal{O}_d := \{ZQ | Q \in \mathcal{O}_d\},$$

is called the equivalence class associated to G. The quotient manifold $\mathbb{R}_*^{n\times d}/\mathcal{O}_d$ is defined as the set of equivalence classes. The mapping $\pi : \mathbb{R}_*^{n\times d} \to \mathbb{R}_*^{n\times d}/\mathcal{O}_d$, mapping points to their equivalence class, induces a Riemannian metric on the quotient manifold from the Euclidean metric in $\mathbb{R}_*^{n\times d}$. This metric results in the following distance between PSD matrices:

$$d(G_i, G_j) = \left[\mathrm{tr}(G_i) + \mathrm{tr}(G_j) - 2\mathrm{tr}\left(\left(G_i^{\frac{1}{2}} G_j G_i^{\frac{1}{2}}\right)^{\frac{1}{2}}\right)\right]^{\frac{1}{2}}. \tag{2.2}$$

This distance can be expressed in terms of the landmark variables $Z_i, Z_j \in \mathbb{R}_*^{n\times d}$ as follows:

$$d(G_i, G_j) = \min_{Q\in\mathcal{O}_d} \|Z_jQ - Z_i\|_F. \tag{2.3}$$

The optimal solution is $Q^* := VU^\top$, where $Z_i^\top Z_j = U\Sigma V^\top$ is a singular value decomposition.

As stated by the next theorem, when $d = 2$, the distance can also be formulated as follows:

Theorem. *Let $G_i, G_j \in \mathcal{S}^+(2, n)$ be two Gram matrices, obtained from landmark matrices $Z_i, Z_j \in \mathbb{R}^{n\times 2}$. The Riemannian distance* (2.2) *can be expressed as:*

$$d(G_i, G_j) = \operatorname{tr}(G_i) - 2\sqrt{(a+d)^2 + (c-b)^2} + \operatorname{tr}(G_j), \tag{2.4}$$

where $Z_i^T Z_j = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$.

Proof. of Theorem 2.2.2. We can reformulate our metric introduced in Eq. (2.3) with:

$$\begin{aligned} d^2(G_i, G_j) &= \operatorname{tr}\left[(Z_jQ - Z_i)(Z_jQ - Z_i)^T\right] \\ &= \operatorname{tr}(G_i) - 2\operatorname{tr}(Z_iQ^TZ_j^T) + \operatorname{tr}(G_j). \end{aligned}$$

To minimize our distance, we need to maximize the term $\operatorname{tr}(Z_iQ^TZ_j^T)$. Let $Z_j^TZ_i$ be a 2×2 matrix with four unknown values a, b, c, d and let $Q \in \mathcal{O}_p$, we maximize:

$$\max\ \operatorname{tr}\left[\begin{pmatrix} a\cos\Theta - b\sin\Theta & - \\ - & c\sin\Theta + d\cos\Theta \end{pmatrix}\right]. \tag{2.5}$$

From Eq. (2.5) we now have to find the maximum of $(a+d)\cos\Theta + (c-b)\sin\Theta$, meaning that we have to maximize $\sqrt{(a+d)^2 + (c-b)^2}\cos(O - O')$. As we want to maximize this value, O has to be equal to O', so $\sqrt{(a+d)^2 + (c-b)^2}\cos(O - O') \leqslant \sqrt{(a+d)^2 + (c-b)^2}$. Therefore we can say that:

$$\max \operatorname{tr}(Z_iQ^TZ_j^T) = \sqrt{(a+d)^2 + (c-b)^2}. \tag{2.6}$$

■

Expressions for the Riemannian exponential and logarithm are given in [62]. We used the implementations provided in the Manopt toolbox [9].

2.3 Modeling the Temporal Dynamics of Landmarks

Based on the landmark representation introduced in the previous section, each face in a frame of a video sequence is mapped to a point on the PSD manifold. Thus, it becomes natural to interpret the points mapped from consecutive frames as describing a trajectory on the manifold. However, making these trajectories useful for subsequent processing and comparison requires smoothing, as illustrated in the following.

The dynamic changes of facial landmarks movement originate trajectories on the Riemannian manifold of positive-semidefinite matrices of fixed rank. We fit a curve β_G to a sequence of landmark configurations $\{F_0, \ldots, F_\tau\}$ represented by their corresponding Gram matrices $\{G_0, \ldots, G_\tau\}$ in $\mathcal{S}^+(2, m)$. This curve enables us to model the spatio-temporal evolution of the elements on $\mathcal{S}^+(2, m)$.

Modeling a sequence of landmarks as a piecewise-geodesic curve on $\mathcal{S}^+(2, m)$ showed very promising results when the data are well acquired, *i.e.*, without tracking errors or missing data.

To smooth the data, accounting both for missing data and tracking errors, we propose to use cubic blended curve fitting algorithms [31, 30]. These algorithms only require to compute Riemannian exponential and logarithm, and also represent the curve by means of a number of tangent vectors that grows linearly with the number of data points. In this paper, we use the algorithm defined in [29]. Specifically, given a set of points $\{G_0, \ldots, G_\tau\} \in \mathcal{S}^+(2, m)$ associated to times $\{t_0, \ldots, t_\tau\}$, with $t_i := i$, the curve β_G, defined on the interval $[0, \tau]$, is defined as:

$$\beta_G(t) := \gamma_i(t - i), \qquad t \in [i, i + 1], \tag{2.7}$$

where each curve γ_i is obtained by blending together fitted cubic Bézier curves computed on the tangent spaces of the data points d_i and d_{i+1} (represented by Gram matrices on the manifold forming a trajectory). The De Casteljau algorithm, used during the reconstruction process is fully performed in the tangent spaces of d_i and d_{i+1}, and a weighted mean is done on the two obtained points.

These fitting cubic Bézier curves depend on a parameter λ, allowing us to balance two objectives: *(i)* Proximity to the data points at the associated time instants; *(ii)* Regularity of the curve (measured in terms of mean square acceleration). A high value of λ results in a curve with possibly high acceleration that almost interpolates the data, while taking $\lambda \to 0$ results in a smooth function approximating the original trajectory.

2.4 Classification and Regression Problems

Now that we have defined how to represent a sequence and how to compare two distinctive landmark configurations, we present in this section how we compare two landmark sequences and how to classify the actions performed in these same sequences.

2.4.1 Trajectory Alignment

As we described in Section 2.3, we represent a sequence as a trajectory of Gram matrices in $\mathcal{S}^+(d, n)$. The sequences represented in this manifold can be of different length as the execution rate of the actions can vary from one person to another, meaning that we can not effectively compare them. A common method to do so is to use Dynamic Time Warping (DTW) as proposed in several works [6, 40, 33]. However, DTW does not define a proper metric and can not be used to derive a valid positive-definite kernel for the classification/regression phase. To address the problem of non positive definiteness of the kernel defined by DTW, Cuturi *et al.* [15] proposed the Global Alignment Kernel (GAK), which allows us to derive a valid positive-definite kernel when

aligning two time series.

More recently Otberdout *et al.* [69] have proposed to classifiy deep trajectories in SPD manifold using GAK. The generated kernel can be used directly with Support Vector Machine (SVM) for the classification phase, whereas it is not the case with kernels generated with DTW. In fact, the kernels built with DTW do not show favorable positive definiteness properties as they rely on the computation of an optimum rather than the construction of a feature map. Note that the computation of the kernels with GAK can be done in quadratic complexity, similarly to naive implementation of DTW. The next paragraph describes how to compute the similarity score between two sequences, using this Global Alignment Kernel.

The kernel proposed by the authors in [15] is based on computing multiple paths by finding the optimal path such that $\prod_{i=1}^{|\pi|} k(x_{\pi_1}(i), y_{\pi_2}(i))$ is maximum, with k being a kernel containing measures to compare the two sequences x and y, and by summing up over all the others alignments. Instead of only consider the optimum path, they take advantage of all the score values across all possible alignment. An example is given in Figure 2.3.

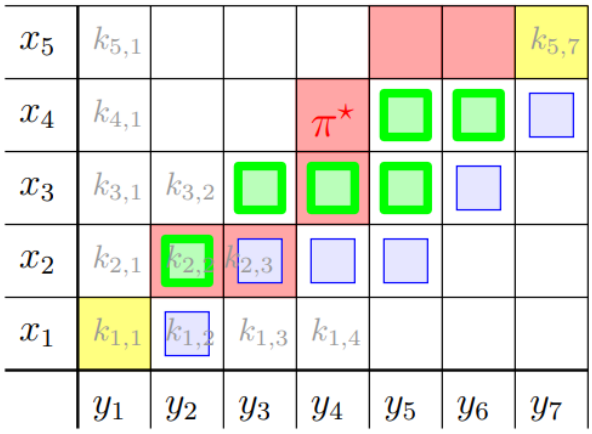


Figure 2.3: Example of the use of multiple paths to find the best alignment between two sequences [15] (the red path denoted $\pi^\star$ is the optimum path, while the green and blue paths are other alignments that will be summed up over the optimal path).

The Global Alignment Kernel is defined as presented in equation 2.8:

$$K(x,y) = \sum_{\pi \in \mathcal{A}(x,y)} \prod_{i=1}^{|\pi|} k(x_{\pi_1}(i), y_{\pi_2}(i)). \tag{2.8}$$

where $\mathcal{A}(x,y)$ defines the set of all possible alignment π between the two sequences x and y.

The Theorem 1 presented in [15] states that if we consider k to be positive definite kernel such that $\frac{k}{1+k}$ is positive definite, then K as defined id 2.8 is positive definite. According to this theorem, the kernel K is positive definite only if $\frac{k}{1+K}$ is also positive definite. The authors states that using the halved Gaussian kernel, presented in equation 2.9, to compute k, then the kernel can be used directly and is very close to the Gaussian kernel. We will used this formula below to ensure that our kernel is positive definite.

$$Gk = \frac{1}{2} e^{-\frac{||x-y||^2}{\sigma^2}}. \tag{2.9}$$

Let us now consider $Z^1 = \{Z_0^1, \cdots, Z_{\tau_1}^1\}$ and $Z^2 = \{Z_0^2, \cdots, Z_{\tau_2}^2\}$, two sequences of landmark configuration matrices. Given a metric to compute the distance between two elements of each sequence, we propose to compute the matrix D of size $\tau_1 \times \tau_2$, where each $D(i,j)$ is the distance between two elements of the sequences, with $1 \leq i \leq \tau_1$ and $1 \leq j \leq \tau_2$.

$$D(i,j) = d(Z_i^1, Z_j^2). \tag{2.10}$$

The kernel $\tilde{k}$ can now be computed using the halved Gaussian kernel, presented in equation 2.9, on this same matrix D. Therefore, the kernel $\tilde{k}$ can be defined as:

$$\tilde{k}(i,j) = \frac{1}{2} * exp\left(-\frac{D(i,j)}{\sigma^2}\right). \tag{2.11}$$

As reported in [15], our final kernel k is positive definite if $\frac{k}{1+K}$ is positive definite, so we can redefine our kernel such as:

$$k(i,j) = \frac{\tilde{k}(i,j)}{(1-\tilde{k}(i,j))}. \tag{2.12}$$

This strategy of using the halved Gaussian kernel assures us that the kernel yields a positive semi-definite matrix in practice and can be used in its own. Finally, using this kernel, we can compute the similarity score between the two sequences Z^1 and Z^2. Remember that this computation is performed in the same complexity as DTW. To do so, we define a new matrix M that will contain the path to the similarity between our two sequences. We define M as a zeros matrix of size $(\tau_1 + 1) \times (\tau_2 + 1)$ and $M_{0,0} = 1$. Computing the terms of M is done using Theorem 2 in [15, §2.3]:

$$M_{i,j} = (M_{i,j-1} + M_{i-1,j-1} + M_{i-1,j}) * k(i,j). \tag{2.13}$$

The similarity score we seek is the value at $M_{(\tau_1+1),(\tau_2+1)}$. Algorithm 1 describes all the steps to get the similarity score. As stated by in [15], this result is equivalent to the DTW algorithm where the max-sum algebra is replaced by the sum-product algebra.

Finally, we build a new matrix K of size $n_{seq} \times n_{seq}$, where n_{seq} is the number of sequences in the dataset we test. This matrix is symmetric and contains all the similarity scores between all the sequences of the dataset and it is used as the kernel for the classification phase with SVM. As this matrix is built with values computed from positive semi-definite kernel, it is a positive semi-definite matrix itself.

2.4.2 Classification with SVM

Our trajectory representation reduces the problem of landmark sequence classification to that of trajectory classification in $\mathcal{S}^+(d,n)$. Given that GAK provides a valid PSD kernel as demonstrated by Cuturi *et al.* [15], and given that our local kernel K satisfies this condition as discussed before, we use the standard SVM with the K kernel that represents the matrix containing the similarity scores

input : Two sequences of landmark configurations $Z^1 = \{Z_0^1, \cdots, Z_{\tau_1}^1\}$, where $Z_{0 \leq i \leq \tau_1}^1$ and $Z^2 = \{Z_0^2, \cdots, Z_{\tau_2}^2\}$, where $Z_{0 \leq j \leq \tau_2}^2$.

output: The similarity score between two sequences Z^1, Z^2

$\tilde{k} \longleftarrow \frac{1}{2} * exp\left(-\frac{D(Z^1, Z^2)}{\sigma^2}\right)$ Equations (2.10) and (2.11)

for $i \leftarrow 0$ **to** τ_1 **do**

  for $j \leftarrow 0$ **to** τ_2 **do**

    $k(i,j) \longleftarrow \frac{\tilde{k}(i,j)}{(1-\tilde{k}(i,j))}$ Equation (2.12)

  end

end

$M \longleftarrow zeros(\tau_1 + 1, \tau_2 + 1)$

$M_{0,0} \longleftarrow 1$

for $i \leftarrow 1$ **to** $\tau_1 + 1$ **do**

  for $j \leftarrow 1$ **to** $\tau_2 + 1$ **do**

    $M_{i,j} \longleftarrow (M_{i,j-1} + M_{i-1,j-1} + M_{i-1,j}) * k(i,j)$ See Equation (2.13)

  end

end

$similarity \longleftarrow M_{\tau_1+1, \tau_2+1}$

return similarity, the similarity score between Z^1 and Z^2

Algorithm 1: Computing the similarity score between two sequences using Global Alignment Kernel [15]

between all the sequences of a dataset to classify the aligned trajectories with global alignment on $\mathcal{S}^+(d,n)$.

By contrast, DTW may define a non positive definite kernel. Hence, we adopt the pairwise proximity function SVM (ppfSVM), which assumes that instead of a valid kernel function, all that is available is a proximity function without any restriction. That is, let us consider $\mathcal{T} = \{\beta_G : [0,1] \to \mathcal{S}^+(d,n)\}$, the set of time-parameterized trajectories of the underlying manifold. Like in [40, §4.1], we define a matrix D_{dtw} containing the similarity measure between two trajectories aligned with DTW.

In that case, given m trajectories in $\mathcal{T}$, the proximity function $\mathcal{P} : \mathcal{T} \times \mathcal{T} \to \mathbb{R}^+$ between two trajectories Z^1 and Z^2 is defined by,

$$\mathcal{P}(Z^1, Z^2) = D_{dtw}(Z^1, Z^2) \ . \tag{2.14}$$

Using this proximity function, the main idea of ppfSVM is to represent each training example Z with a vector $[\mathcal{P}(Z, Z^1), \ldots, \mathcal{P}(Z, Z^m)]^T$. The set of trajectories can be represented by a $m \times m$ matrix P, where $P(i,j) = \mathcal{P}(Z^1, Z^2)$, with $1 \leq i, j \leq m$. From this matrix P we can use a classical linear SVM.

2.4.3 Pain Estimation with Support Vector Regression

We build a new matrix K of size $n_{seq} \times n_{seq}$, where n_{seq} is the number of sequences in the dataset used to test our method. This symmetric matrix contains all the similarity scores between all the sequences of the dataset. This matrix is built with values computed from positive-semidefinite kernel, meaning that it is a positive-semidefinite matrix itself. Now that we have a valid and positive-semidefinite kernel K, as demonstrated by Cuturi *et al.* [15], we can use it directly as a valid kernel for classification. To estimate pain intensity score (i.e., self-reported VAS scores), we use a Support Vector Regression (SVR) model. To train our SVR model, we give as input a training set that is

a part of our kernel K containing the similarity scores between all training trajectories. This part of the kernel, containing the training set, is also positive-semidefinite by definition. We also give a vector containing the labels for the trajectories in our training kernel.

2.5 Conclusion

In this chapter, we have proposed a method for comparing and classifying temporal sequences of 2D/3D landmarks on the manifold of positive semi-definite matrices of fixed rank. This approach involves different steps: 1) Building trajectories on the Riemannian manifold from body skeleton or facial landmark sequences; 2) Applying a curve fitting algorithm on the trajectories to smooth and denoise the data points or interpolate missing data due to bad landmark extraction; 3) Performing temporal alignment using the Global Alignment Kernel instead of Dynamic Time Warping to obtain a positive definite kernel that can be used directly with Support Vector Machine (SVM) or Support Vector Regression (SVR). This framework is capable of working on different task using different kind of landmarks and in the next chapter we will present the results we obtained from multiple experiments on both action recognition and pain estimation tasks. We also present the metrics and protocols used for each specific problems as well as the datasets.

Chapter 3

Experimental Results on Action Recognition and Pain Estimation

3.1 Introduction

In this chapter we present the different experiments conducted on the tasks of action recognition and pain estimation from the analysis of 2D and 3D tracked landmarks from the body or the face. We first describe the problem of both task with an explanation of the measure associated with them. Then we describe the datasets used to conduct the different experiments as well as the protocols. Finally, we present the results obtained using our approach, described in the previous chapter, with different configurations. A comparison with recent state-of-the-art approaches for each dataset is also presented as well as some drawbacks of our approach.

3.1.1 Metrics Used

The action recognition problem consists of associating the sequences with the right class of action. This means that we define here a classification problem. We decided to used the standard accuracy

as our metric to measure the effectiveness of our approach.

In the case of the pain estimation problem, we predict a continuous value representing the predicted pain score associated for each sequence. The evaluation of our approach is obtained by computing two error measures: the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) between the predicted pain scores and the ground-truth.

The MAE is computed as follows:

$$\mathrm{MAE} = \frac{1}{n_{seq}} \sum_{i=1}^{n_{seq}} \left| \hat{y}_i - y_i \right|, \tag{3.1}$$

and the RMSE is given by:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n_{seq}} \left(\hat{y}_i - y_i \right)^2}{n_{seq}}}, \tag{3.2}$$

where n_{seq} is the number of sequences considered, y_i is the ground truth (*i.e.*, self-reported VAS pain score), and $\hat{y}_i$ is the predicted pain score.

3.2 Datasets Presentation

We present in this section the different datasets used to tackle the problem of action recognition and pain estimation. The datasets consist of a mix of 2D and 3D tracked landmarks from the body and the face. Some of the dataset comes with landmarks already extracted and other don't. In the latter case, we extracted the landmarks (body or face) ourselves using the OpenPose framework [10] as mentioned in the different presentations of the datasets.

3.2.1 Action Recognition Datasets

UTKinect-Action3D Dataset The UTKinect-Action3D dataset [98] is a widely used dataset for 3D action recognition. It contains 199 sequences, consisting of 10 actions, namely *walk, sit down,*

stand up, pick up, carry, throw, push, pull, wave hands and *clap hands* performed by 10 different subjects. The videos and the skeletons were captured with a Microsoft Kinect and the skeletons are composed of 20 body joints. In our approach, we use the available skeletal joint locations, where each body joint is defined with its x, y and z coordinates. Figure 3.1 presents the ten actions included in the dataset.

Figure 3.1: Samples From the UTK Dataset

KTH-Action Dataset The KTH-Action dataset [80] is a 2D action recognition dataset. It consists of six actions, namely *boxing, handclapping, handwaving, jogging, running* and *walking* performed by 25 subjects in four different conditions, which are outdoor, outdoor with scale variations, outdoor with different clothes and indoor (examples are presented in Figure 3.2). The sequences were acquired with a static camera at a frame rate of 25 fps and a resolution of 160×120 pixels. The dataset contains a total of 599 clips, with 100 clips per actions (1 clip is missing for one action). As the sequences in the dataset are 2D videos, we have to extract the skeletons of the subjects performing the actions. To do so, we used the OpenPose framework [10] to extract the skeletons in the COCO format, with 18 body joints. Note that we clean the landmark sequences by removing the frames where the body joints where not effectively estimated. Keeping all the frames leads to worst results due to misdetected joints, meaning that we do not need all the frames available to recognize

an action. Figure 3.3 shows the configuration of the body joints that we analyzed.

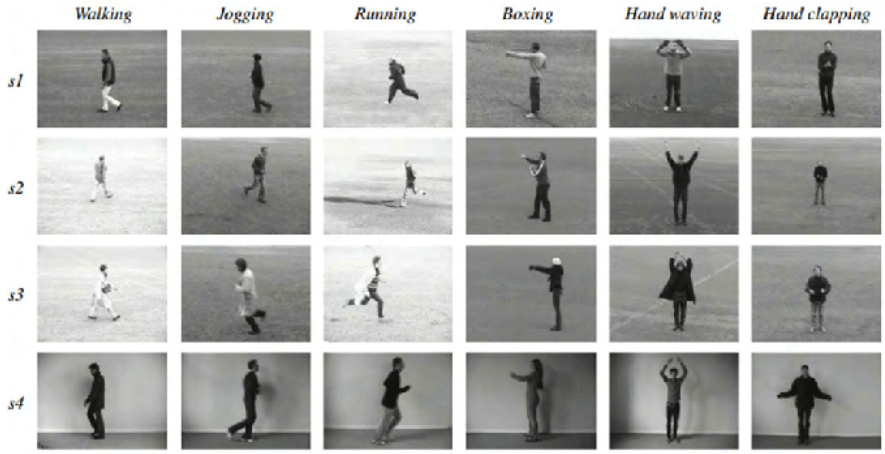


Figure 3.2: Samples From the KTH Dataset

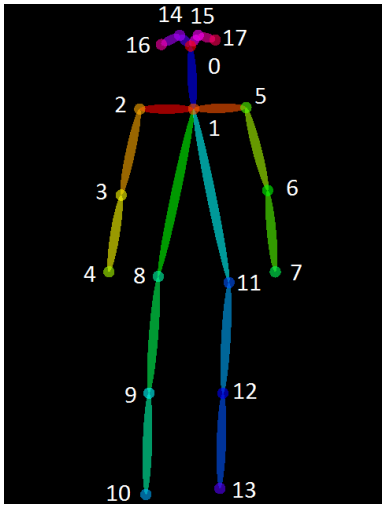


Figure 3.3: Skeleton with the COCO format.

UAV-Gesture Dataset The UAV-Gesture dataset [70] is a 2D videos dataset, consisting of 13 actions corresponding to UAV (*i.e.*, Unmanned Aerial Vehicles) gesture signals. These actions are *All Clear, Have Command, Hover, Land, Landing Direction, Move Ahead, Move Downward, Move To Left, Move To Right, Move Upward, Not Clear, Slow Down* and *Wave Off*. The actions are performed by 11 different subjects in an outdoor scenario with slight camera movements. The

dataset contains 119 high-quality clips consisting of 37151 frames in total. As reported in [70], this dataset is not primarily designed for action recognition, but it can be used for this specific task. The skeletons are available with the dataset and the OpenPose framework was also used to extract them in the COCO format. Figure 3.4 presents samples from the dataset.

Figure 3.4: Samples From the UAV-Gesture Dataset

3.2.2 Pain Estimation Datasets

UNBC-McMaster Shoulder Pain Archive The UNBC-McMaster Shoulder Pain Archive [60] is a widely used dataset for pain expression recognition and intensity estimation. The dataset contains 200 video recordings of 25 subjects performing a series of active and passive range-of-motion of their affected and unaffected shoulders. Each video sequence is annotated for pain intensity score at the sequence-level using three self-reported scales (including the VAS) and an Observer Pain

Rating scale. The video recordings are also annotated at the frame-level using the manual FACS (*i.e.*, Facial Action Coding System). The facial landmarks are available with the dataset and are extracted using an Active Appearance Model (AAM). In total, 66 landmarks are available at the jaw, the mouth, the nose, the eyes and the eyebrows. Figure 3.5 shows two images from a sequence of the dataset with their corresponding facial landmarks, colored by their velocities. Our goal is to estimate the self-reported pain score (VAS).

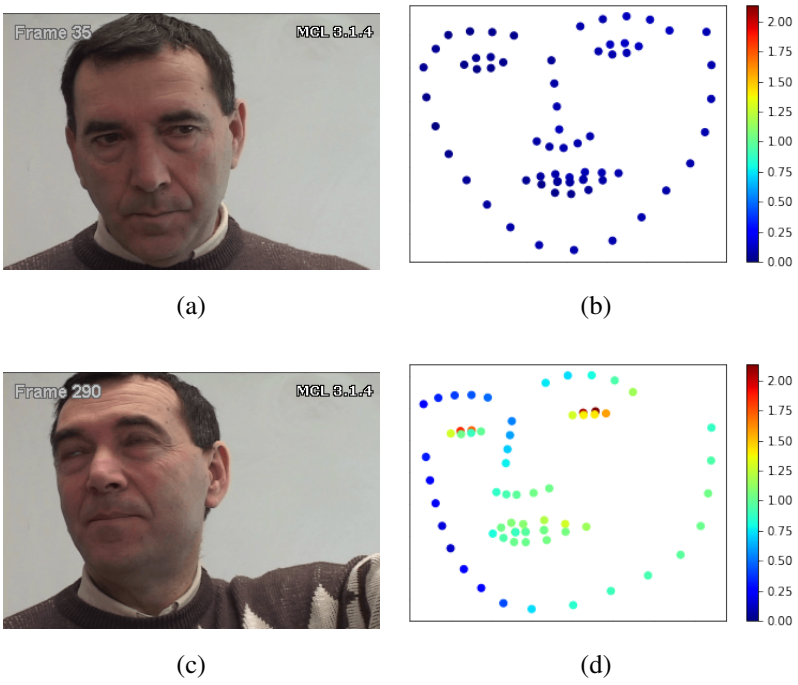


Figure 3.5: UNBC-McMaster Shoulder Pain Archive [60]: (a) and (c) show two example images from a sequence; In (b) and (d) the landmark coordinates for images in (a) and (b) are reported, with velocities evidenced by different colors (best viewed in color).

Figure 3.6 shows the distribution of the VAS score across the dataset. One can observe that the number of available sequences per VAS score is not uniformly distributed: 50% of the sequences have a VAS pain score of $\{0, 1, 2\}$, while only 11% of the sequences have a VAS pain score of $\{8, 9, 10\}$. Also, the number of sequences per subject is not uniform, as shown in Figure 3.7. This bias, both in terms of number of sequences per VAS score and number of sequences per subject, hampers accurate learning and prediction of the VAS score, making the estimation

more challenging.

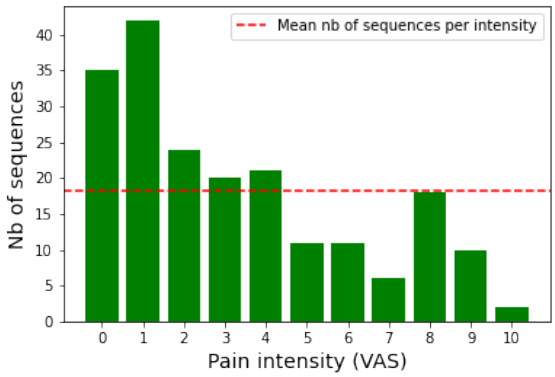


Figure 3.6: Distribution of the VAS Pain Scores for the UNBC-McMaster Shoulder Pain Archive. The red dashed line represents the mean number of sequences per intensity

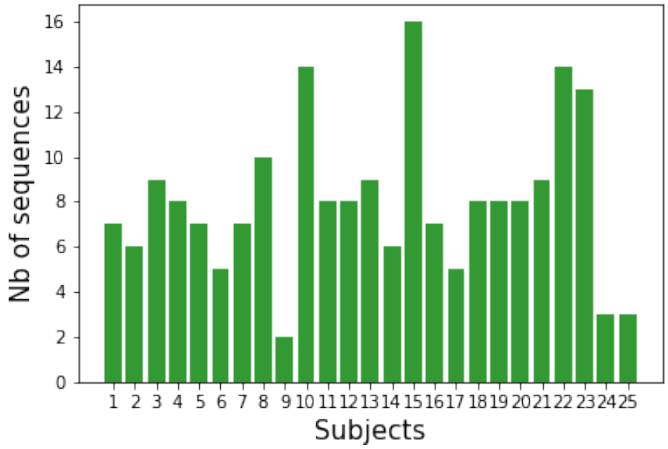


Figure 3.7: Number of sequences per subject for the UNBC-McMaster Shoulder Pain Archive

3.3 Testing Protocols

To evaluate the proposed approach, we used the three protocols, with two of them being subject-independent: Leave-One-Sequence-Out cross validation, Leave-One-Subject-Out cross validation, and k-fold cross validation.

Leave-One-Sequence-Out cross validation protocol In this protocol, training and testing are performed on different sequences. For each round, we use all sequences of the dataset but one for training. and the remaining sequence for testing. That is, data from the same subject can be used during the training and the testing phase as there are at least two sequences per subject in the dataset. Therefore, this protocol is sequence-independent, but not subject-independent. We use this protocol as a baseline for our approach.

Leave-One-Subject-Out cross validation In this protocol, for each round, we use the sequences from all subjects but two for training, and the remaining sequences of one subject for validation and the sequences of the other subject for testing. There is no overlap between the training, validation and testing sets. Accordingly, this is a *subject-independent* evaluation protocol. We perform this operation for all the subjects in the dataset, so that each subject is used once for testing.

k-fold cross validation This protocol is similar to the Leave-One-Subject-Out cross validation one, but instead of taking only the sequences of one subject at a time for validation and testing at each round, we take all the sequences of k subjects for validation and testing, and the remaining sequences for training. The choice of the k subjects for the validation set is done by choosing the k first subjects in the dataset, then the k next subjects for testing and the remaining for training and so on until all the subjects are used for testing (*i.e.*, k rounds). Also this evaluation protocol is subject-independent.

Cross validation has the advantage of preventing from having results that are due to the chance as all data are used to train and test the proposed method. The average across all folds is more representative of the whole dataset.

3.4 Experimental Results

In this section we present the results we obtained on the datasets using the previously mentioned protocols using our approach. The first part of this section is dedicated to the problem of action recognition and the second part to the pain estimation problem.

3.4.1 Action Recognition Results

UTKinect-Action3D Dataset Following the same experimental settings of [88, 55, 43], we performed the Leave-One-Sequence-Out cross validation protocol on the UTKinect-Action3D dataset, meaning that we used one sequence for testing and the rest for training.

Our experimental results are summarized in Table 3.1. In particular, the columns are as follows: *Curve Fitting* indicates if we performed the curve fitting algorithm described in Section 2.3; *Lambda* indicates the value of the lambda parameter in curve fitting; *Alignment Method* indicates if we used the standard DTW to align sequences or GAK as described in Section 2.4.1; *Sigma* indicates the value of the sigma parameter for the Gaussian Kernel when using GAK; and *Results* indicates our scores.

Curve Fitting	**Lambda**	**Alignment Method**	**Sigma**	**Results**
Yes	0.5	DTW	-	97%
No	-	DTW	-	97.49%
Yes	0.5	GAK	0.3	97.49%
No	-	GAK	0.3	**97.99%**
Yes	0.5	GAK	0.5	**97.99%**

Table 3.1: Our results on the UTKinect-Action3D dataset

The best accuracy that we obtained on this dataset is 97.99%. Overall, we can say that the application of curve fitting does not increase our results. Our assumption is that the data in this

dataset are very clean, and we can loose some information with the application of smoothing on clean data. Note that we obtained better results when using the Global Alignment Kernel rather that DTW.

	Protocol	
Methods	**H-H**	**LOOCV**
Trajectory on $\mathcal{S}^+(d,n)$ [40] (2019)	-	96,48%
SCK+DCK [44] (2016)	98.2%	-
Bi-LSTM [88] (2018)*	-	98.49%
LM^{3}TL [102] (2017)	-	98.8%
GCA-LSTM [55] (2018)*	-	99%
MTCNN [43] (2018)*	-	99%
Hankel & Gram matrices [106] (2016)	-	**100%**
Ours	-	97.99%

Table 3.2: Comparison of our approach with state-of-the-art results for the UTKinect-Action3D dataset. *: Deep Learning approach

In Table 3.2, we compare our method with recent state-of-the-art results. Overall, our approach achieves competitive results with respect to most recent approaches. We directly compare our results with [40] as we work on the same geometric space of $\mathcal{S}^+(d,n)$ manifold. The main differences between our method and the method in [40] is the use of a different metric and of the Global Alignment Kernel instead of DTW. Our metric is simpler that the metric in [40], as we do not have to estimate the parameter k used in Eq. (7) in [40] for distance computation. Furthermore, the k parameter in [40] is more of a constraint as they have to determine its best value for each dataset they test. The use of GAK is also an advantage for us as it defines a positive semi-definite kernel, which is not the case for DTW allowing us to use a classic SVM instead of ppfSVM.

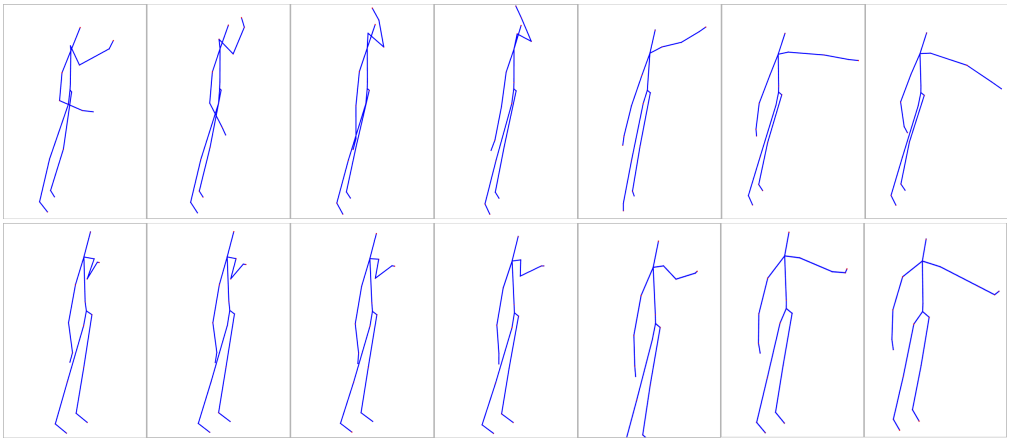


Figure 3.8: Comparison of two sequences that are confused in UTKinect-Action3D dataset (top: *Throw* action, bottom: *Push* action)

The Figure 3.8 presents two sequences that are confused, leading to a misclassification for one of them. In that case, the top action (*i.e.* Throw) is misclassified as the bottom action (*i.e.* Push). One of the reasons can be the position of the arm at the end of the action, which is the same in the two sequences. The *Throw* action is the most confused action in the dataset.

KTH-Action Dataset For this dataset, we followed the Leave-One-Subject-Out cross validation protocol, meaning that we used the sequences of one subject for testing and the rest for training. Table 3.3 summarizes our experimental results on this dataset.

Curve Fitting	Lambda	Alignment Method	Sigma	Results
No	-	DTW	-	94.49%
Yes	10	DTW	-	94.66%
No	-	GAK	0.2	95.16%
Yes	10	GAK	0.2	**96.16%**

Table 3.3: Our experimental results on the KTH-Action dataset

Here, again, we obtained better results when using the GAK, demonstrating superior performance over DTW. The results reported with DTW are the best accuracy over all the configu-

rations we tested. Unlike the data in UTKinect-Action3D dataset, the data in KTH-Action are 2D and low resolution videos, with presence of noise in the background, leading to noisy skeleton data after extraction. In this regard, the application of the curve fitting algorithm improves our results by 1%. We compare our approach with the state-of-the-art in Table 3.4. Overall, our method achieves competitive results with recent approaches, while only using skeletal data.

Methods	**Input data**	**Protocol**	**Accuracy**
Schüldt *et al.* [80] (2004)	RGB	Split	71.7%
Liu *et al.* [54] (2009)	RGB	LOAO	93.8%
Yoon *et al.*[103] (2010)	Skeleton	-	89%
Raptis & Soatto [77] (2010)	RGB	LOAO	94.5%
Wang *et al.* [96] (2011)	RGB	Split	94.2%
Gilbert *et al.* [27] (2011)	RGB	LOAO	95.7%
Jiang *et al.* [38] (2012)	RGB	LOAO	95.77%
Vrigkas *et al.* [94] (2014)	RGB	LOAO	**98.3%**
Veeriah *et al.* [91] (2015)*	RGB	Split	93.96%
Liu *et al.* [56] (2016)	RGB	Split	95%
Almeida *et al.* [2] (2017)	RGB	LOAO	98%
Our	Skeleton	LOAO	96.16%

Table 3.4: Comparison of our approach with state-of-the-art results for the KTH-Action dataset. *: Deep Learning approach

UAV-Gesture Dataset Again, for this dataset we followed the Leave-One-Subject-Out cross validation protocol. Table 3.5 compares our results with the baseline experiment reported in [70].

Method	Curve Fitting	Lambda	Alignment Method	Results
P-CNN [70] (2018)*	-	-	-	91.9%
Ours	No	-	GAK	91.6%
Ours	Yes	10	GAK	**92.44%**

Table 3.5: Comparison of our approach with the baseline on the UAV-Gesture dataset. *: Deep Learning approach

This is a recent dataset and its principal interest does not rely on action recognition, meaning a lack of results to compare our results with. However, the authors have tested their dataset for the case of action recognition based on skeletons with Pose-Based Convolutional Neural Network (P-CNN) descriptors, that gives us a baseline to compare our results.

The baseline achieves an accuracy of 91.9% with a Deep Learning based approach, whereas our approach achieves an accuracy of 92.44%, outperforming the state-of-the-art results when applying curve fitting and the GAK alignment method.

The Figure 3.9 presents two very similar actions in the dataset, that is *All Clear* and *Not Clear*. The only big difference between these two actions is the orientation of the hand on the raised arm. This information is not captured when only retrieving the body skeleton from the OpenPose framework, even if it is possible to retrieve the hand skeleton. Note that with our method, the two actions are only confused three time on a total of 22 sequences, meaning that our method is capable of differentiate minimal changes in the actions.

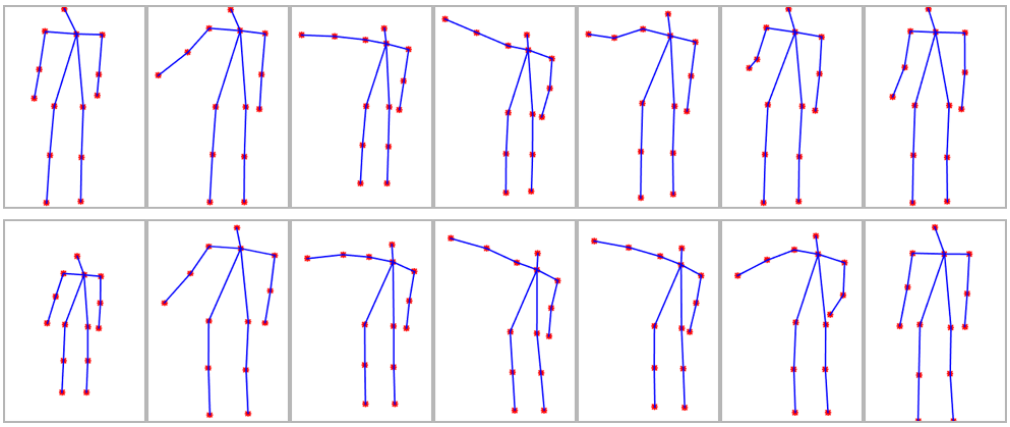


Figure 3.9: Comparison of two sequences that are confused in UAV-Gesture dataset (top: *All Clear* action, bottom: *Not Clear* action)

Computation Time Comparison In this analysis, we have computed the time for each step of our approach. Applying the curve fitting algorithm can be resource demanding on some manifolds, as it requires successive computations of Riemannian exponentials and logarithms (see [29] for more information on the computational cost of the method). The alignment method can also be resource demanding, regarding the size of the trajectory. With all these parameters in mind, we propose to compute the time that our method takes to compute specific tasks from pose extraction to action classification. The tests were conducted on a laptop equipped with an Intel Core i7-8750H CPU, 16G of RAM and a NVidia Quadro P1000 GPU. Table 3.6 and Table 3.7 summarize the execution time of each part of our method for the KTH-Action and UAV-Gesture datasets, respectively. For metric notation, M_1 refers to Eq. (2.3) and M_2 refers to Eq. (2.4).

Pose extraction	**Curve Fitting**	**Alignment method - Metric**	**Alignment**	**Classification**
147	0.069	DTW - M_1	0.034	0.41
147	0.069	DTW - M_2	**0.02**	0.41
147	0.069	GAK - M_1	0.04	0.49
147	0.069	GAK - M_2	**0.019**	0.49

Table 3.6: Execution time (in seconds) obtained on the KTH-Action dataset for the different steps of the method for one sequence.

Pose extraction	Curve Fitting	Alignment method - Metric	Alignment	Classification
-	0.504	DTW - M_1	0.128	0.038
-	0.504	DTW - M_2	**0.072**	0.038
-	0.504	GAK - M_1	0.138	0.53
-	0.504	GAK - M_2	**0.066**	0.053

Table 3.7: Execution time (in seconds) obtained on the UAV-Gesture dataset for the different steps of our method for one sequence.

For the KTH-Action dataset, we consider a sequence of 61 frames and a sequence of 192 frames for UAV-Gesture. First, we can observe that the pose extraction phase takes most of the execution time for the KTH-Action dataset. This is partially due to the fact that our GPU is not powerful enough (we get around 3.5fps running OpenPose with our Quadro P1000). The extraction time is not reported for the UAV-Gesture dataset as the skeletons are available with the dataset. The second thing we can observe is the low difference in computation time for the alignment part when switching from DTW to GAK. We can also note that when we use M_2, the computation time can be reduced by a factor of 2 compared to the use of M_1, showing that the formula (2.4) is in our case cheaper to evaluate than (2.3). If we only consider the execution time for the treatment of the skeletons, it takes around 0.499 seconds to classify an action of the KTH-Action dataset and around 0.614 seconds for an action of the UAV-Gesture dataset in the best case scenario.

3.4.2 Pain Estimation Results

UNBC-McMaster Shoulder Pain Archive Our goal is to estimate the VAS pain score for each sequence of the dataset. We test our method with the three protocols described above and report the results in Table 3.8. For each protocol, we fix the value of the curve fitting parameter lambda to 1000 and the Gaussian kernel in the sequence alignment sigma to 0.8. *Protocol* indicates the protocol used for training and testing our method; *% of frames* indicates the percentage of frames used from each sequence for training and testing; *MAE* indicates the Mean Absolute Error and

RMSE the Root Mean Square Error of our estimation.

Protocol	**% of frames**	**MAE**	**RMSE**
Leave-One-Sequence-Out	25%	**2.31**	3.14
	100%	2.53	3.32
Leave-One-Subject-Out cross validation	25%	**2.52**	3.27
	100%	2.92	3.51
5-fold cross validation	25%	**2.43**	3.14
	100%	2.79	3.51

Table 3.8: Results of our method with the three different protocols.

From Table 3.8, we notice that in every cases, the MAE is lower when we down-sample the sequences by considering 1 frame each 4 frames, leading to 25% of the frames available for pain assessment. This is due to the high amount of non-pain frames that are present in the dataset. We also notice that the best MAE we obtained is 2.31 with the Leave-One-Sequence-Out protocol. This result is expected as this protocol is not subject-independent and sequences of the same subject can be used for both training and testing. The second best MAE we obtained is 2.43, using the 5-fold cross validation protocol. We report the RMSE as a second measure of the error of our estimation. Results show the same trend as the MAE with the best RMSE observed for the Leave-One-Sequence-Out protocol.

Comparison with state-of-the-art We compared our approach to two state-of-the-art methods for VAS pain intensity measurement from video (see Table 3.9). Here, we report the best results for DeepFaceLIFT [53] that only uses the VAS as training labels as the authors also present results while combining VAS and OPR labels. They obtained a MAE of 2.30 using a 5-fold cross validation protocol. Our results are close to theirs, while only using a geometry based formulation of facial landmark dynamics (meaning that our method is less expensive as we do not have to train a neural network). Our results are comparable to RNN-HCRF [61] results, as they obtain a MAE of 2.46, though using a different protocol. In fact, in the results for RNN-HCRF, data have been randomly

split by taking the sequences of 15 subjects for training and the sequences of 10 subjects for testing. It is also important to highlight that in RNN-HCRF the face appearance is also used, while our method only considers the shape of the face.

One of the advantage of our method over the two approaches presented here is the explainability of the results. As our method is based on facial landmarks and modeling of their dynamics as a trajectory on the manifold, it is possible to interpret the predicted VAS score for a new observation based on distances of this observation to train trajectories. This makes it possible to support the explanation of results on a much more solid base than would be by using alternative models for prediction, such as those based on deep neural networks. Interpretability is also very important in a day-to-day use by practitioners as they can better estimate the pain from the different parts of the face.

Method	**Protocol**	**Labels for training**	**MAE**
DeepFaceLift [53]	5-fold cross validation	VAS	2.30
RNN-HCRF [61]	random split	VAS & PSPI	2.46
Ours	5-fold cross validation	VAS	2.43

Table 3.9: Comparison of our method with state-of-the-art results.

3.5 Conclusion and Discussion

In this chapter, we have presented experiments and results obtained using our approach to model and classify 2D/3D landmark sequences of human behavior. Specifically, we focused our experiments on two common tasks that are action recognition from body joints and the estimation of the pain index from facial landmarks. On the task of action recognition, we demonstrated the effectiveness of our approach on three datasets, one consisting of 3D body joints and two consisting of 2D landmarks. We also evaluated the effectiveness of using the Global Alignment Kernel over Dynamic Time Warping for temporal alignment. In the case of 2D landmarks, we demonstrated

that using an optimized metric to measure the distance between Gram matrices composing the trajectories improve our results while being faster than using the standard metric on the manifold.

In the case of pain estimation, using the UNBC-McMaster Shoulder Pain Archive, we demonstrated that our approach is competitive with state-of-the-art approaches, that make use of deep learning framework, while being only based on a geometric method. Estimating the pain score of a sequence with a geometric method allow us to better explain the results. This interpretability is important for practitioners as they can better understand the pain of a patient and thus choose a better solution to heal him.

One limitation of the proposed approach is the time to compute the similarity matrix that will be used by either SVM or SVR, depending of the task. In fact, increasing the number of sequences in a dataset also increase the time to build this kernel as more similarity scores need to be computed. Our approach is effective on datasets composed of less than a thousand sequences but began to show its limits with larger dataset, and therefore is not appropriate for real-time classification or estimation, even if it was not the goal to begin with. Another drawback is the use of landmark configurations instead of the whole images for the sequences. By doing so, we greatly reduce the number of features at each frame and some sequences can be confused with others, resulting in a wrong classification for the action recognition task or a wrong estimation of the pain degree for pain estimation.

Chapter 4

Refinement of the Approach with Application to Pain Estimation

4.1 Introduction

We propose in this chapter a video based measurement of self-reported VAS based pain intensity scores using the dynamics of facial movement. This method is a refinement of the approach presented in Chapter 2. The main differences compared to the previous method are:

- A split of the face in different regions allowing us to highlight the presence of pain is specific region of the face,
- The use of multiple manifolds, one per region to study locally the dynamics of the face,
- A presentation of multiple strategies to combine the prediction of pain intensity on these different regions.

An overview of the proposed approach is reported in Figure 4.1.

First, facial landmarks are detected from each video frame to form a sequence of landmark configurations. The landmark configurations are then split into four regions to form four sequences of facial region landmark configurations. For each region based time series, velocities are then computed as the displacement of the coordinates between two consecutive frames. Gram matrices are computed from the combination of the landmark coordinates and their velocities. These matrices are represented as trajectories on the $\mathcal{S}^+(2, m)$ manifold, which is the set of $m \times m$ symmetric positive semi-definite matrices of rank 2, with one manifold per region. We apply the same curve fitting algorithm to the trajectories as before on each manifold for smoothing and noise reduction. Alignment of the trajectories is obtained by using the Global Alignment Kernel (GAK) [15], which results in a similarity matrix per region containing the similarities between trajectories of homologous regions. Finally, we use the kernels generated by GAK with different strategies to estimate the pain intensity based on each region. These strategies are used to combine the estimated pain scores on each region in order to estimate the pain intensity of the whole sequence.

4.2 Refinement of the Approach in the Case of Facial Landmarks for Pain Estimation

The global specificity of this method is based on our approach presented in Chapter 2. However, some big changes have been made on the construction of the trajectories and the fusion of the different manifolds created (one per region of the face). A new feature is also presented and concatenated to the already known coordinates of each landmark in a configuration: their velocity. The construction of a trajectory on a manifold remains the same as before as well as the alignment of the trajectories. To this end, we only focus our research on the use of the Global Alignment Kernel instead of Dynamic Time Warping as this leads to the generation of positive-definite kernel that can be used directly with Support Vector Regression modules to estimate the pain score of a sequence.

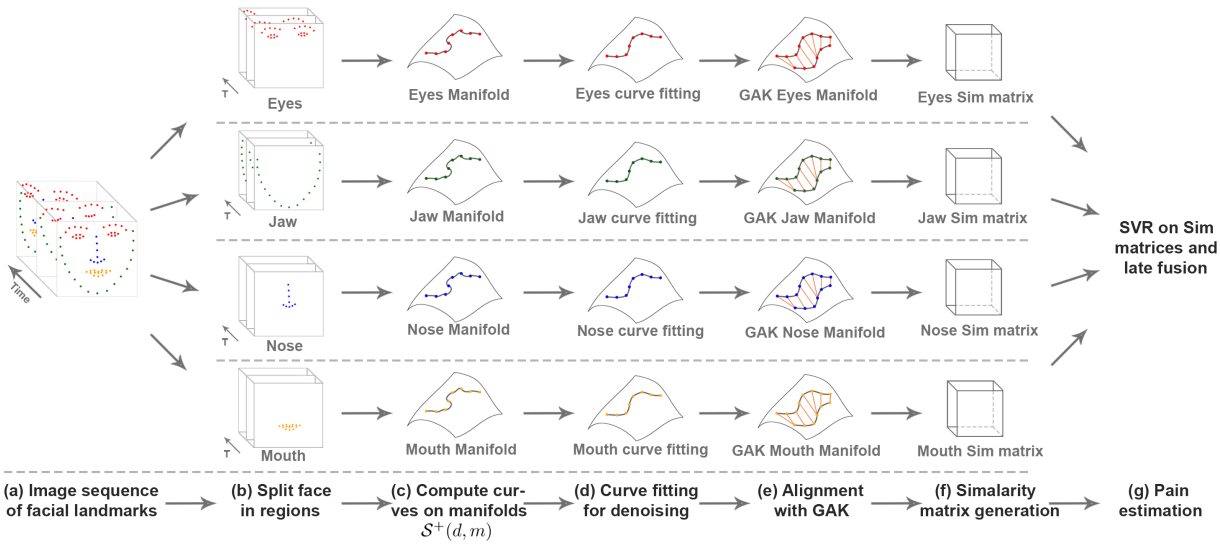


Figure 4.1: Method overview: *(a)* Detection and extraction of facial landmarks; *(b)* Split of the landmark configurations in different regions and computation of their velocities; *(c)* Computation of Gram matrices and modeling of their temporal dynamics as trajectories on the $\mathcal{S}^+(2,m)$ manifold; *(d)* Application of curve fitting for noise reduction and smoothing of the trajectories; *(e)* Alignment of the trajectories with the Global Alignment Kernel (GAK); *(f)* Similarity matrix computation for all the regions; *(g)* Pain estimation for each region and late fusion of the scores for the final pain level.

4.2.1 Gram Formulation

Given an image sequence, we represent the dynamics of facial movement with a time series formed by the coordinates $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ of n tracked facial landmarks and grouped into matrices Z_i. Each Z_i $(0 \leq i \leq \tau)$ being a $n \times 2$ matrix $[(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)]^T$ of rank 2 encoding the positions of the n facial landmarks. For each landmark l_i, its velocity is also measured as the magnitude of the displacement between two consecutive matrices Z_i and Z_{i+1}. We denote the velocity matrix at frame i as $V_i = Z_{i+1} - Z_i \in \mathbb{R}^{n \times 2}$. Since velocity cannot be extracted from the last frame, V_i is computed only for $i \in \{0, \ldots, \tau - 1\}$. However, to simplify the notation, we adopt the same range of the frame indexes $\{1, \tau\}$ for both the landmark position and their velocity. In doing so, the last frame is dropped from the video sequence, and it is only used to estimate the

velocity.

Our objective here is to find a shape representation that is invariant to Euclidean transformations (rotations and translations). To remove the translation, each landmark configuration Z_i is centered by subtracting the landmarks center of mass. The velocity of each landmark is computed after this normalization. Similar to [16, 85, 41], we propose the Gram matrix G as a representation of landmarks and velocities. The Gram matrix is defined by:

$$G = FF^T = \langle p_i, p_j \rangle , \quad 1 \leq i, j \leq 2n, \tag{4.1}$$

where $F = [Z|V]$ is the $2n \times 2$ matrix obtained by concatenating the position Z, and the velocity V of the landmarks. The Gram matrix representation is invariant to rotation and translation. In addition, Gram matrices of the form FF^T, where F is an $m \times 2$ matrix of rank 2 ($m = 2n$), are characterized as $m \times m$ positive semi-definite (PSD) matrices of rank 2, a Riemannian manifold of well-studied geometry and theoretical properties [7]. As an example, Figure 4.2 shows a Gram matrix representation as a trajectory on the manifold of PSD matrices.

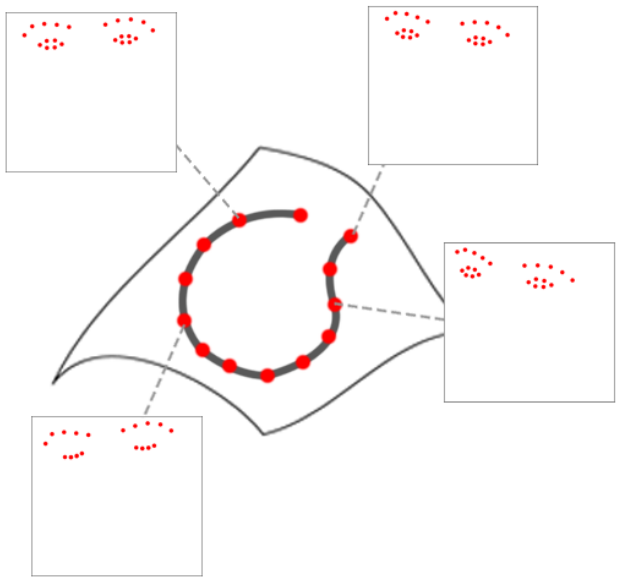


Figure 4.2: Example of a trajectory of Gram matrices for the eyes region.

4.2.2 Gram Matrix Distance

Considering the Riemannian geometry of the space $\mathcal{S}^+(2, m)$ of $m \times m$ positive semi-definite matrices of rank 2, we rely on the use of the same metric to measure the distance between two Gram matrices $G_i = F_i F_i^T$ and $G_j = F_j F_j^T$ as presented in Chapter 2:

$$d(G_i, G_j) = \mathrm{tr}(G_i) + \mathrm{tr}(G_j) - 2\sqrt{(a+d)^2 + (c-b)^2}\ , \tag{4.2}$$

where $F_i^T F_j = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$.

This distance is computed between Gram matrices of the same region, indicating that we compute distances separately depending on the region of the face considered. The modeling and smoothing of the trajectories is also performed separately on the manifolds representing the different regions. The application of the curve fitting algorithm to smooth the trajectories by blending Bézier curves computed on the tangent spaces of data points, represented by Gram matrices on each manifold, is also performed separately on each individual manifold. Remember that a high value of the λ parameter allow us to interpolate the data while taking a λ value that tends to 0 results in smoothing the original trajectory.

The construction of the similarity matrix is performed using the Global Alignment Kernel method, as presented in Section 2.4.1 of Chapter 2. Each region is treated independently and four different similarity matrices are thus computed. The use and fusion of the information contained in each matrix is explained in the next section.

4.3 Pain Estimation Protocols

Having a representation of landmark sequences as a trajectory on the PSD manifold and a similarity measure between them, we are in the position of using these similarities to train a regressor for VAS

pain score estimation at video level.

As described in Section 4.2, the dynamics of the facial landmarks is captured by four trajectories, each one capturing the dynamics of one out of four regions of the face. In order to estimate the pain score, these trajectories can be processed following three different strategies: *(i)* perform a manifold product among the four manifolds, one for each region, to form a new valid manifold and compute the similarity between the trajectories on this new manifold; *(ii)* compute the similarity scores between the trajectories on each manifold independently, then perform an early fusion to estimate the pain score; *(iii)* compute the similarity scores between the trajectories on each manifold independently, then perform a late fusion to estimate the pain score.

Given a dataset composed of n_{seq} videos annotated with pain intensity score, a sequence of facial landmark configuration matrices is extracted from each video. Then, a symmetric matrix K^p of size $n_{seq} \times n_{seq}$ is built to store the similarity scores between all the trajectories for a given region p. These matrices are built with values computed using the positive semi-definite kernel, meaning that each matrix is positive semi-definite. Now that we have a valid and positive semi-definite kernel K^p, as demonstrated by Cuturi *et al.* [15], we can use it directly as a valid kernel for estimation. To estimate the pain intensity score (*i.e.*, self-reported VAS scores), we use a Support Vector Regression (SVR) model [82, 20]. In order to predict the level of pain based on similarity matrices from different regions of the face, three different strategies can be adopted, namely, *manifold product*, *early fusion* and *late fusion*. These three strategies are combined with two evaluation protocols presented in the previous chapter to estimate the pain index of each sequence, *i.e.*, *Leave-One-Subject-Out cross validation* and k*-fold cross validation*. We didn't perform any experiments using the Leave-One-Sequence-Out cross validation protocol in order to only keep subject-independent protocols.

4.3.1 Regions Manifold Product

The idea here to compute pain scores is that of combining the manifolds, one for each region, before using SVR for pain estimation. Indeed, the decomposition of the face into four regions can be seen as the product space of four manifolds $\mathcal{M} = \mathcal{S}^+(2, n_1) \times \mathcal{S}^+(2, n_2) \times \mathcal{S}^+(2, n_3) \times \mathcal{S}^+(2, n_4)$, one manifold per region. Thus, the distance between two elements $G_i, G_j \in \mathcal{M}$ can be modeled as the square root of the sum of the squared distances between these elements in each manifold [8]:

$$d_{\mathcal{M}}(G_i, G_j) = \sqrt{\sum_{k=1}^{4} d(G_{ki}, G_{kj})^2_{\mathcal{S}^+(2,n_k)}}\ , \tag{4.3}$$

where each n_k, $1 \leq k \leq 4$, encodes the landmark coordinates in region k, and $d(.,.)$ is the distance defined in (2.4). The result is a new manifold that preserves the structure of the original manifolds. The result of this formula is the distance between G_i and G_j. This distance is computed between all the Gram matrices composing two trajectories and the alignment algorithm is then performed on the resulting distance matrix as explained in the previous section.

The distances between the trajectories in the manifold $\mathcal{M}$ are computed to form the similarity matrix. As the manifolds are combined into one new manifold, only one similarity matrix is computed and is used as our kernel for estimation. In this case, no weights combination is performed and we only have to train one SVR, like in the early fusion strategy.

4.3.2 Early Fusion

In this strategy, the SVR model is fed with the combination of the four kernels K^p and trained to estimate pain score. By adopting the early fusion approach, the combination of the kernels is done by averaging the similarity scores:

$$K_{i,j} = \frac{\sum_{p=1}^{4} K^p_{i,j}}{4}. \tag{4.4}$$

By doing so, we only need to train one SVR for the whole face using this new kernel that is

computed by fusing the scores of different regions of the face in such a way that all regions are assigned the same weight.

4.3.3 Late Fusion

When adopting the late fusion strategy, the training sets used as inputs to train the models are part of our kernel K^p containing the similarity scores between all the training trajectories for the region p. Taking a subset of the entire kernel K^p for training gives us a new kernel that is also positive semi-definite by construction. A vector containing the ground-truth VAS scores for the trajectories is also given for the training part. Finally, the outputs of region specific models are combined to predict the VAS scores for the whole face. Accordingly, we train one SVR per region independently, using the kernels K^p. Once the VAS scores are predicted for all the regions and for all the sequences in the dataset, we apply a late fusion of the scores to obtain the VAS pain index $\hat{\mathrm{y}}$ for the whole face by taking a weighted combination of the four predictions for each sequence:

$$\hat{\mathrm{y}} = \frac{(w_j \cdot \hat{y}_{jaw} + w_n \cdot \hat{y}_{nose} + w_m \cdot \hat{y}_{mouth} + w_e \cdot \hat{y}_{eyes})}{4}. \tag{4.5}$$

In order to identify the best combination of the weight values, a grid search approach has been adopted, with values in the set $\{0.1, 0.2, \ldots, 0.9, 1.0\}$. The best weight values are determined at each round of the cross validation by taking out the sequences of one (or k) subject that will be used as testing data. Then, a second cross validation loop is included inside the first one, where the sequences of a subject are taken out and used as validation data, while the remaining sequences are used as training data. The weights are estimated at each round of this second cross validation loop, using the validation data, and the best weight combination is used to estimate the pain index of each sequence of the testing data. By this double cross validation loop, the weights are optimized using validation data that are not included in the testing set, reducing the risk of overfitting.

4.4 Datasets Presentation

We present in this section the two datasets used to validate our approach and the refinement performed over our previous method. The goal is to test our approach on the problem of pain estimation, and to do so, we chose the previously used UNBC-McMaster Shoulder Pain Archive and the Biovid Heat Pain Dataset.

UNBC-McMaster Shoulder Pain Archive As mentioned in the previous chapter, this dataset is widely used for pain expression recognition and intensity estimation. This dataset is challenging in the sense of its unbalanced data, with around 80% of the frames labeled as *non pain* frames. The number of sequences per pain intensity and the number of sequences per subject are also unbalanced, making the prediction of a pain intensity difficult for both frame and sequence level estimation.

The Biovid Heatpain Dataset The Biovid Heat Pain dataset [95] is widely used for pain expression recognition and pain intensity estimation. This dataset contains 8,700 videos of 87 different subjects. The dataset is composed of 5 pain classes (pain level from 0 to 4), with 20 samples per class and subject, with a time window of 5.5 seconds. The dataset consists of 5 different parts, containing pain stimulation (parts A, B and C), posed expression (part D) and emotion elicitation (part E). We worked with part A of the dataset, characterized by the absence of electromyography sensors (EMG) on the user face. In the videos, the subjects are asked to put a hand on a heat source, while the heat sensation increases with the time lapse. The thresholds for the minimum and maximum temperature is determined for each subject on a scale of $\{0, \ldots, 4\}$, with 0 meaning no pain and 4 meaning worst possible pain level.

Our goal is to estimate the pain intensity scores consistently with the self-reported pain level over the dataset. This dataset is larger and more balanced than the UNBC-McMaster Shoulder Pain Archive, as every subject has 100 sequences, with 20 sequences per pain class.

This dataset only contains videos and annotations, so we used the OpenPose framework [11] to extract 70 facial landmarks from each frame of each video in the dataset. The main difference between the landmarks extracted with OpenPose and those distributed with the UNBC-McMaster Shoulder Pain Archive dataset is the addition of 4 landmarks (2 at the extremity of the mouth and 2 at the center of the eyes). Moreover, the landmarks that come with the UNBC-McMaster dataset are extracted using an Active Appearance Model (AAM), that is a semi-automatic algorithm with human in the loop annotation, compared to the fully automatic algorithm proposed by OpenPose. Figure 4.3 shows two frames from a sequence of the Biovid Heat Pain dataset with their corresponding extracted facial landmarks.

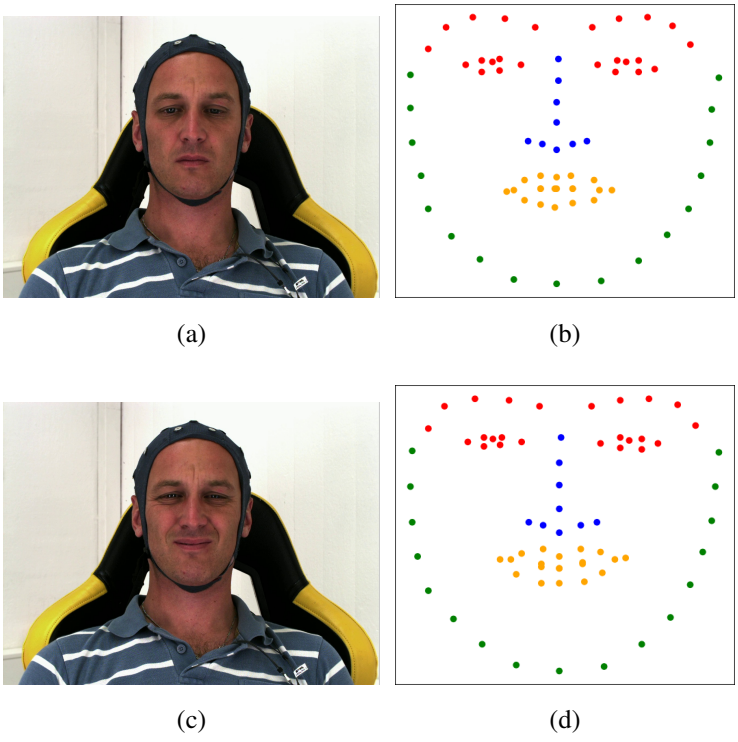


Figure 4.3: Biovid Heat Pain dataset: Sample images are shown in (a) and (c). In (b) and (d) their corresponding landmark coordinates are evidenced using a different color for each region (best viewed in color) [95].

4.5 Experimental Results

In this section we will present the different results obtained on the two datasets presented in the section above. Using the UNBC-McMaster Shoulder Pain Archive, we focused on the search of the best hyper-parameter combination using a grid search approach. This strategy was adopted on the previously mentioned protocols and validated on the validation set of the dataset and finally used on the testing set. A presentation of our results on two different benchmarks is conducted as well as comparison with recent state-of-the-art approaches.

4.5.1 The UNBC-McMaster Shoulder Pain Archive

We present here the different results obtained on the UNBC-McMaster Shoulder Pain Archive. Especially, we will focus on the search of the best hyper-parameters with a grid search strategy on the different protocols. As each protocol used to test our approach are cross validation protocols, the estimation of the best parameters was performed on the validation set and then used on the testing set. The final results are then presented and discussed before comparing them to recent state-of-the-art methods for pain estimation using the same dataset.

4.5.1.1 Ablation Study and Best Configuration Estimation

The process of estimating the pain index by the analysis of face dynamics depends on three main hyper-parameters that determine the amount of smoothing of trajectories on the manifold (*i.e.*lambda value), the sigma value used as a parameter for the application of the Gaussian Kernel during sequence alignment, and the number of frames that are actually used to compute these trajectories. In fact, reducing the number of frames for each sequence allows us to speed up the computation time because we need to compare fewer frames to calculate the similarity score between any two sequences in the dataset. To identify a convenient choice of these hyper-parameters, a grid search strategy is adopted. For this purpose, the value of the parameter lambda (Section 2.3)

is discretized into four reference values $\{Nofitting, 10, 100, 1000\}$, with 10 meaning we apply a fairly strong amount of smoothing of the trajectories and 1000 a soft application of smoothing (this value is closer to no fitting than 10) and 100 as a middle value for smoothing. The σ parameter is discretized into three reference values $\{0.5, 0.7, 0.9\}$. Explanations on the choice of this parameter can be found in Section 2.4.1. As for the number of frames that are used to compute the trajectory, three different frame subsampling rates were explored: 25%, 50% and 100% of the frames, with 25% meaning that we kept only 1 frame out of 4 and 50% meaning that we kept 1 frame out of 2.

The best configuration was identified by computing the prediction accuracy on the validation set of the UNBC-McMaster Shoulder Pain Archive using the LOSO and a 5-fold cross validation protocol as described in previous chapter. The late fusion method was used to estimate the pain scores, as presented in Section 4.3.3. Table 4.1 reports the prediction accuracy in terms of MAE for the different configurations using the LOSO protocol, and Table 4.2 reports the accuracy using the 5-fold cross validation protocol.

Results in Table 4.1 show that the best configuration on the validation set corresponds to λ=100 (*i.e.*, soft smoothing of the trajectories), a σ=0.7 (*i.e.*, trade-off between high and low values that can penalize the similarity scores), with a sub sampling of 50% when using the Leave-One-Subject-Out protocol. Results in Table 4.2 show the same trend using the 5-fold cross validation protocol. The choice of the σ value demonstrates that a value too high (*i.e.*. close to 1) or too low can negatively impact the estimation of the pain index. For the sampling of the sequences, using the total amount of available frames did not improve the results. This can be explained by the fact that 80% of the frames in this dataset are non pain frames. A reduction of the frame sampling rate is also beneficial to the overall computation time as a lower number of frame comparisons is necessary to estimate the similarity between two sequences.

		Sampling		
σ	λ	25%	50%	100%
0.5	No fitting	1.82	1.79	**1.72**
	10	1.89	**1.72**	1.76
	100	1.75	**1.72**	**1.72**
	1000	1.93	1.81	**1.74**
0.7	No fitting	1.72	**1.69**	1.74
	10	1.68	**1.66**	1.72
	100	1.65	**1.63***	1.66
	1000	1.71	1.67	**1.66**
0.9	No fitting	1.74	**1.72**	1.73
	10	1.81	1.72	**1.71**
	100	1.74	1.73	**1.70**
	1000	**1.71**	1.79	1.88

Table 4.1: MAE of our proposed method on a validation set on the UNBC-McMaster Shoulder Pain Archive using the LOSO protocol. Best results for a given configuration at varying sampling rates is given in bold. The best result is marked with *.

4.5.1.2 Results on the Testing Set

Our goal here is to estimate the VAS pain score for each video sequence. We tested our method with the two protocols described above: the Leave-One-Subject-Out, and a 5-fold cross validation. Results are reported in Table 4.3. We chose to use 5-folds for the k-folds cross validation protocol as reported in the state-of-the-art for better comparison.

For each protocol, we fixed the value of the parameters according to the results reported in Section 4.5.1.1: curve fitting parameter lambda (λ in Section 2.3) equal to 100, because the data are well acquired, and we do not need a strong smoothing of the curves and a sub sampling of 50%. Columns in Table 4.3 have the following meaning: *Protocol* indicates the protocol used for

		Sampling		
σ	λ	25%	50%	100%
0.5	No fitting	1.78	**1.69**	1.76
	10	1.70	**1.68**	1.76
	100	1.78	**1.75**	**1.75**
	1000	1.75	**1.73**	1.74
0.7	No fitting	**1.65**	1.69	1.75
	10	1.76	**1.71**	1.77
	100	1.77	**1.72**	1.76
	1000	1.75	**1.68***	1.79
0.9	No fitting	1.82	1.74	**1.72**
	10	**1.82**	1.89	1.89
	100	1.86	**1.77**	1.85
	1000	1.86	**1.78**	1.82

Table 4.2: MAE of our proposed method on a validation set on the UNBC-McMaster Shoulder Pain Archive using the 5-fold cross validation protocol. Best results for a given configuration at varying sampling rates is given in bold. The best result is marked with *.

training and testing our method; *MAE* and *RMSE* are the two error measures of our estimation. Furthermore, we report results for the whole face as baseline for comparison.

From Table 4.3, we notice the best MAE was obtained with the late fusion strategy and the 5-fold cross validation protocol, with an error of 1.59. The best MAE with the Leave-One-Subject-Out cross validation protocol is 1.61, also obtained with the late fusion strategy. The weights for the late fusion were estimated during the cross-validation rounds on the validation set as mentioned in Section 4.3.3. Values of the weights are as follow: 0.39 for the jaw region, 0.56 for the nose region, 0.88 for the mouth region and 0.94 for the eyes region. The relative values of these weights can be regarded as an index of how much relevant is each part of the face for the prediction of the pain level. The relevance of the eyes region is 34%, the mouth region 32%, the nose region

Protocol	Regression Setup	MAE	RMSE
Leave-One-Subject-Out cross validation	Whole Face	2.52	3.27
	Cartesian product	2.12	2.72
	Early fusion	2.39	3.11
	Late fusion	**1.61**	**2.04**
	Late fusion - augmented	<u>1.41</u>	<u>1.87</u>
5-fold cross validation	Whole Face	2.44	3.15
	Cartesian product	2.28	3.02
	Early fusion	2.32	3.10
	Late fusion	**1.59**	**1.98**
	Late fusion - augmented	<u>1.36</u>	<u>1.75</u>

Table 4.3: Prediction accuracy of the proposed method on the test set of the UNBC-McMaster Shoulder Pain Archive dataset. Bold values indicate best results without using augmentation. Underlined values have been obtained using augmentation.

20% and the jaw region 14%. For every tested protocol, we obtained a better MAE using facial decomposition compared to the baseline using the whole face. The late fusion approach gives better results than the early fusion strategy. Therefore, training one SVR for each region is more effective than combining the similarity matrices of the regions in one similarity matrix that represents the whole face and train one SVR on that. This observation is also valid for the Cartesian product of the manifolds, where one SVR is trained after the computation of the similarity matrix between the trajectories in the result of the manifold product. However, the manifold product strategy yields better results than early fusion. We report the RMSE as a second error measure of our estimation. Results show the same trend as for the MAE with the best RMSE observed while applying the late fusion strategy.

To cope with the non-uniform distribution of videos per class of pain level on the prediction accuracy, we augmented the number of videos of the pain classes where the number of sequences is below the mean number of sequences per class, represented by the red dashed line

in Figure 3.6 (the classes concerned are $\{5, 6, 7, 8, 9, 10\}$). Accordingly, data for sequences with VAS label greater than or equal to 5 have been augmented by first flipping the landmark coordinates along the horizontal axis, *i.e.*, x coordinate; Then, each sequence was modeled as a trajectory on the manifold by also applying curve fitting to it. This augmentation allows us to have 58 new sequences with high level of pain (above 5). The new data distribution can be seen in Figure 4.4. Augmenting the data in this way allowed us to improve the prediction accuracy for all protocols used for testing (see underlined scores in Table 4.3). We improved our results of about 15%, leading to a MAE of 1.36 with the 5-fold cross validation protocol and a MAE of 1.41 with the Leave-One-Subject-Out protocol, by only considering augmentation of the sequences with pain level greater than 5.

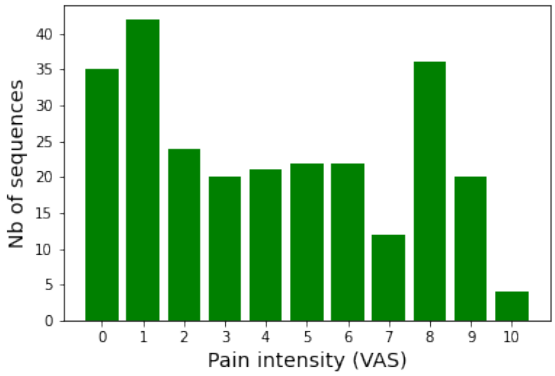


Figure 4.4: New Distribution of the VAS Pain Scores after data augmentation for the UNBC-McMaster Shoulder Pain Archive.

Figure 4.5 and Figure 4.6 show the MAE per intensity with the Leave-One-Subject-Out and 5-folds cross validation protocols, respectively, with and without data augmentation (*i.e.*, green bars show the MAE from original data and orange bars show the MAE from augmented data). From both figures, we can notice that the higher the VAS score is, the higher the MAE is. This can be explained by the fact that there is a limited amount of sequences with high pain scores, as reported in Figure 3.6. It is also worth noticing that augmenting the data in the dataset, as described above, significantly reduces the MAE per intensity for the sequences with a higher VAS, highlighting the fact that having a more balanced dataset can improve the prediction accuracy.

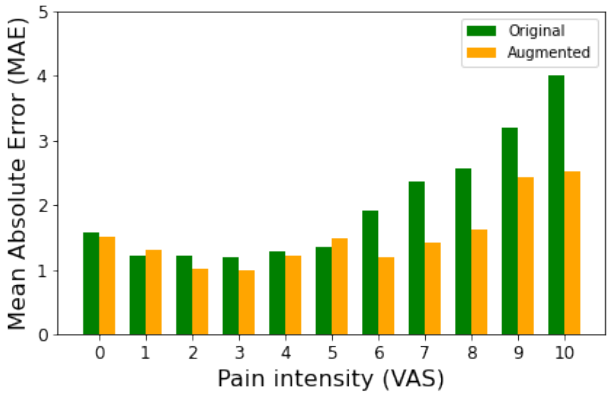


Figure 4.5: MAE per intensity for the Leave-One-Subject-Out protocol. Green bars represents original data, Orange bars represents augmented data.

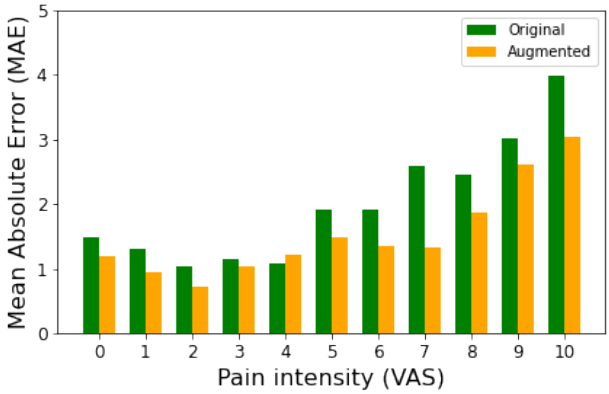


Figure 4.6: MAE per intensity for the 5-fold cross validation protocol. Green bars represents original data, Orange bars represents augmented data.

OpenPose Landmarks: We also tested our approach on the UNBC-McMaster dataset using landmarks extracted with OpenPose [11]. We decided to test two landmark configurations extracted with OpenPose: *(i)* the complete configuration containing 70 facial landmarks, and *(ii)* a reduced configuration with 66 facial landmarks corresponding to the original AAM landmarks available with the dataset. We tested these two configurations using the late fusion approach on the two testing protocols and results are summarized in Table 4.4. Using the complete OpenPose configuration (*i.e.*, 70 landmarks), we obtained a MAE of 1.63 for the LOSO protocol, and a MAE of 1.62 for the 5-fold cross validation protocol. Using the reduced configuration (*i.e.*, 66 landmarks

by excluding the center of the eyes and the corners of the mouth to correspond to the original landmarks available with the dataset), we obtained MAE of 1.62 and 1.60 for the LOSO and the 5-fold cross validation protocols, respectively. These results are close to those obtained with the AAM landmarks, indicating that extracted landmarks from a fully automatic method can lead to good results. However, the results are slightly less accurate than those obtained with AAM landmarks. This can be explained by the fact that AAM landmarks are extracted with human in the loop and therefore can be a little more precise than landmarks extracted from a fully automatic method.

Protocol	**Adopted Landmarks**	**MAE**	**RMSE**
LOSO cross validation	UNBC Original	**1.61**	**2.04**
	OpenPose (complete)	1.63	2.07
	OpenPose (reduced)	1.62	2.05
5-fold cross validation	UNBC Original	**1.59**	**1.98**
	OpenPose (complete)	1.62	2.00
	OpenPose (reduced)	1.60	1.98

Table 4.4: Comparison of prediction accuracy using different landmarks: the 66 landmarks provided with the UNBC dataset (original), the 70 landmarks provided by the OpenPose library (complete), the 66 landmarks provided by the OpenPose library and corresponding to those provided with UNBC (reduced). Bold values indicate best results.

Sampling	Traj. Comp. (fitting)	Traj. Comp. (no fitting)	Similarity Computation	SVR Training	Prediction
25%	≈ 11.2	≈ 0.83	≈ 734	≈ 0.625	≈ 0.009
50%	≈ 42.7	≈ 1.56	≈ 3211	≈ 0.625	≈ 0.009
100%	≈ 207	≈ 2.97	≈ 12422	≈ 0.625	≈ 0.009

Table 4.5: Computation time of each step of the proposed method on the UNBC-McMaster Shoulder Pain Archive. Time is in seconds.

Computation Time: In Table 4.5, we also summarize the computation time for each step of our approach, with the different sub-samplings and with or without the application of the curve fitting

algorithm. Testing is performed on the entire dataset with the LOSO protocol and the late fusion pain estimation, after the estimation of the best combination of weights for each region. Tests were conducted on a laptop equipped with a 6 cores CPU, 16GB RAM, running MatLab 2020b. Table columns have the following meaning: *Sampling* indicates the number of frames that are kept in each sequence of the dataset; *Trajectory Computation* corresponds to the computation of the Gram matrices, trajectory modeling (separate columns are used to report data corresponding to the adoption or not of the curve fitting algorithm); *Similarity Computation* indicates the time to compute the similarity scores between all the sequences in the dataset, including the computation of the distance matrix between all frames of two sequences and the application of GAK; *SVR Training* corresponds to the time to train the four SVR models, one per region, from the similarity matrix, and *Prediction* is the time to predict the self-reported pain score. From Table 4.5, we can see the impact of reducing the number of frames for each sequence, especially to build the trajectories on the manifolds and on the computation of the similarity matrix. In fact, each trajectory contains less points as we reduce the number of frames, so a lower number of distance computations is required to measure the similarity between two sequences. We can also note that the application of the curve fitting algorithm can have a strong impact on the computation time. This impact is more significant when we use all the available frames, further demonstrating that processing the video sequences at a reduced frame rate yields computational savings without affecting the prediction accuracy. However, applying the fitting algorithm or the sub-sampling of the sequences does not impact the computation time for the SVR training or the prediction of the pain scores. This behavior is desired, as the size of the similarity matrix used for SVR training remains the same (*i.e.*, a square matrix of size $n_{seq} \times n_{seq}$, with n_{seq} the number of sequences in the dataset).

4.5.1.3 Comparison with state-of-the-art

We compared our approach to several state-of-the-art methods for VAS pain intensity measurement from videos (see Table 4.6). We focused our comparison with other approaches that estimated the pain index at sequence level, but we also reported some results of methods estimating pain index

at frame level. The main difference between the two strategies is the use of a different label for training (VAS for sequence level, and PSPI for frame level estimation) and the amount of data used. In order to estimate pain at sequence level, we have to rely on 200 annotated sequences, whereas pain estimation at frame level can leverage on the use of 48,398 annotated frames. Here, we report the best results for DeepFaceLift [53] for the case where only the VAS scores were used as training labels (in that work authors also presented results, while combining VAS and OPI labels). They obtained a MAE of 2.3 using a 5-fold cross validation protocol. Our best result for MAE with the same protocol is 1.59, while only using a geometry based formulation of the dynamics of facial landmarks. We also compare our results to the RNN-HCRF method [61]. In that work, authors used a different protocol for testing as data have been randomly split by taking the sequences of 15 subjects for training and the remaining 10 sequences for testing. They also used two different labels, the VAS and the PSPI (frame-level label), to train their network to estimate pain at sequence-level. They obtained a MAE of 2.46 with this configuration. It is important to highlight that in [61] the authors used the face appearance, while our method only considers the shape of the face through facial landmarks. The manifold trajectories proposed in [86] allow the authors to obtain a MAE of 2.44 when they performed the 5-fold cross validation protocol and a MAE of 2.52 using the Leave-One-Sequence-Out protocol. Our approach is based on the same structure, but we estimate the self-reported pain level by decomposing the face, whereas in [86] the estimation was performed on the whole face, demonstrating the effectiveness of our proposed facial decomposition. Recently, Xu *et al.* [99] obtained a MAE of 1.95 using the 5-fold cross validation protocol and this result was further refined in [100] with a MAE of 1.73, using the same protocol. In the first work, the authors estimated the frame-level label before estimating the sequence-level pain. In the second work, they used the different labels available in the UNBC-McMaster Shoulder Pain Archive to estimate the VAS at sequence-level. Finally, we report the best results for CNN-RNN [22], when the authors combined different labels for training. A MAE of 2.34 was obtained using a two-level 5-fold cross validation scheme.

Pain Estimation	Method	Protocol	Modalities	Training labels	MAE (VAS)	MAE (PSPI)
Frame Level	Deep Pain[78]*	LOSO	Images	PSPI	-	0.50
	Compact CNN[81]*	LOSO	Images & Landmarks	PSPI	-	0.20
Sequence Level	RNN-HCRF [61]*	Random split	Facial landmarks	VAS & PSPI	2.46	-
	CNN-RNN [22]*	5-fold CV	Images	VAS & OPI & AFF & SEN	2.34	-
	DeepFaceLift [53]*	5-fold CV	Facial landmarks	VAS	2.30	-
	Extended MTL from pixel [99]*	5-fold CV	Images	VAS	1.95	-
	Extended MTL with AU [100]*	5-fold CV	Action Units sequences	VAS	1.73	-
	Manifold trajectories [86]	5-fold CV	Facial landmarks	VAS	2.44	-
	Proposed	**5-fold CV**	Facial landmarks	**VAS**	**1.59**	-

Table 4.6: Comparison of Our Method With State-of-the-Art Approaches on the UNBC-McMaster Shoulder Pain Archive. (* Indicates Methods That Use a Neural Network)

4.5.2 The Biovid Heatpain Dataset

4.5.2.1 Results

The goal here is to estimate the self-reported pain level for each sequence of the dataset. The results of our method are obtained using the same two protocols described in the previous section: the Leave-One-Subject-Out protocol and a 3-fold cross validation. The results are summarized in Table 4.7. For each of these protocols, the curve fitting parameter λ and the sampling of each sequence are the same we used on the UNBC-McMaster Shoulder Pain Archive. This means that lambda is equal to 100 and the sampling rate is set to 50%, by taking out one frame every two consecutive frames. Since we observed that face decomposition leads to better results, we only report here our results using the early and late fusion methods, described in Section 4.3.

Using the Leave-One-Subject-Out cross validation protocol, we obtained a MAE of 1.13, while we got a MAE of 1.06 using the 3-fold cross validation protocol with the late fusion strategy. In the same way as with the UNBC-McMaster dataset, we observe an improvement of the results using the late fusion strategy over the early fusion, showing the effectiveness facial decom-

Protocol	Regression Setup	MAE	RMSE
Leave-One-Subject-Out cross validation	Late fusion	1.13	1.47
	Early fusion	1.51	1.89
3-fold cross validation	Late fusion	1.06	1.36
	Early fusion	1.88	2.27

Table 4.7: Biovid Heat Pain Dataset: Comparison of Results of the Proposed Method.

position and training of one SVR per region. The overall MAE is lower for the Biovid dataset as there are only 5 different levels of pain, compared to 11 for the UNBC-McMaster dataset and the dataset is larger, meaning that at each round of the cross validation, there are more training data. Figure 4.7 shows the MAE per intensity obtained using the late fusion strategy and both protocols (*i.e.*, blue bars correspond to the Leave-One-Subject-Out protocol and yellow bars to the 3-folds cross validation protocol).

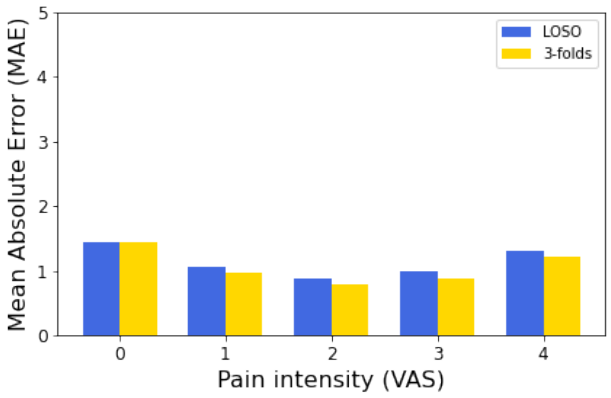


Figure 4.7: MAE per intensity on the Biovid Heat Pain dataset.

4.5.2.2 Comparison with state-of-the-art

As mentioned, the goal of our proposed method is to estimate the pain scores at sequence level for each video of the Biovid Heat Pain dataset. From [97], most of previous works using this dataset considered the pain estimation as a binary classification problem (presence of pain vs. different

intensities of pain) or classified pain intensity in binary pairs. However, some of the approaches focused on continuous pain estimation at sequence level and we report their results in Table 4.8.

Method	**Protocol**	**Modalities**	**Training Labels**	**MAE**	**RMSE**
Pouromran *et al.* [72]	LOSO	Skin Conductance	VAS	0.93	1.16
Lopze-Martinez and Picard [57]	LOSO	Skin Conductance	VAS	1.05	1.29
Kachele *et al.* [42]	LOSO	Bio-signals and videos	VAS	0.99	1.16
Kachele *et al.* [42]	LOSO	Statistical geometric features	VAS	1.16	1.35
Proposed	**LOSO**	**Facial landmark coordinates**	**VAS**	**1.13**	**1.47**
Proposed	**3-fold CV**	**Facial landmark coordinates**	**VAS**	**1.06**	**1.36**

Table 4.8: Comparison of Our Method With State-of-the-Art Approaches on the Biovid Heat Pain Dataset.

Kachele *et al.* [42] reported multiple results using different modalities to estimate pain indexes. First, the result using early fusion of multiple physiological signals (skin conductance, ECG and EMG) with video features was reported, with a MAE of 0.99. They also reported a result using statistical geometric features computed after extracting facial landmarks with OpenFace and obtained a MAE of 1.16. Both these results were obtained by applying the Leave-One-Subject-Out cross validation protocol. For a fair analysis, we compared our results with their second result, as it is not using physiological signals. However, their statistical features were extracted from the landmark coordinates, whereas we only used the landmark coordinates and their velocities. Despite of this, we obtained competitive results with a MAE of 1.13. Pouromran *et al.* [72] obtained a MAE of 0.93 with the LOSO protocol, but using skin conductance as input features. The advantage of the methods using physiological signals can be observed in the different results. However, they required the adoption of intrusive instruments like sensors on the head or on the hand to record bio-signals. Landmark coordinates can be obtained using a simple camera, with no impact on the privacy of each subject.

Table 4.8 shows that our approach achieves state of the art results in terms of MAE among approaches using only visual features. If RMSE is considered, the measured accuracy of our approach decreases more than what is observed for the other approaches. Considering that a

characterizing trait of the RMSE compared to the MAE is that it gives more relevance to large error values, a plausible interpretation of this pattern is that with our approach there is a residual number of predictions with a large error, yet this error being very low in most of the cases. These predictions with large errors are less frequent in [42] although in most cases the error is higher compared to our approach.

4.6 Discussion and Conclusions

We proposed a model for predicting the level of pain based on the dynamics of facial landmarks. The model is based on the decomposition of facial landmarks in different regions of the face and representation of the motion dynamics of these landmarks as trajectories on the Riemannian manifold of fixed rank symmetric positive semi-definite matrices. We have demonstrated the effectiveness of our approach through extensive experiments on the UNBC-McMaster Shoulder Pain Archive dataset and the Biovid dataset. Our approach is competitive with the state-of-the-art on the UNBC-McMaster Shoulder Pain Archive dataset among the approaches that predict the VAS pain score based only on the shape of the face at sequence level.

The main issue with the proposed method is the time required to compute the kernel of the SVR model. As the size of the dataset increases, the time to compute the similarity matrix used kernel increases as well. Future work will investigate solutions to speed-up this computation, for example by clustering the training sequences so as to reduce the number of sequences used to build the kernel. One solution could be based on computing a mean trajectory to represent each pain level index, thus reducing the size of the similarity matrix to compute.

Finally, we also plan to learn the weights for the late fusion strategy, allowing us to better understand the contribution of each region of the face for pain assessment as this remains an open question. This could be addressed through the adoption of a more effective strategy than the grid search approach currently adopted.

Chapter 5

Conclusion and Future Work

5.1 Conclusions and Limitations

We proposed in this thesis a novel geometric framework to better understand human behavior based on the analysis of landmark sequences. Our approach is based on the use of Gram matrix to represent a landmark configuration at each frame of a sequence and a trajectory representation of the sequences on the manifold of positive semi-definite matrices of fixed rank. To overcome the non-linear nature of the shape of the trajectories, we used the Riemannian geometry to define tools in order to analyze the Gram matrices composing them. We defined an optimized metric to compute the distances between two Gram matrices for the specific case of 2D landmarks which greatly simplifies the computation of the distances by not computing Singular Value Decomposition (SVD) in order to find the best angle between data. The trajectory representation of the sequences allow us to smooth them by applying a curve fitting algorithm that will smooth or denoise them. This parameterized algorithm is also able to interpolate data between already known points, making it suitable in the case of bad landmark extraction or missing data. We also present the use of the Global Alignment Kernel (GAK) instead of Dynamic Time Warping (DTW) for temporal alignment. Compared to DTW, the use of GAK yields a positive definite kernel that can be used directly with Support

Vector Machine (SVM) or Support Vector Regression (SVR) instead of using pairwise proximity function SVM (ppfSVM), which assumes that instead of a valid kernel function, all that is available is a proximity function without restrictions. We demonstrated the effectiveness of this framework on multiple datasets over two specific tasks that are action recognition and pain estimation from both 2D and 3D human landmarks.

Finally, we present a refinement of this framework with a split of the landmarks in different region. This split can contribute to better understand local changes in the dynamics of the landmarks and have more control on the prediction in the case of pain estimation. It was demonstrated that not all region contribute the same to define the level of pain and splitting the face in multiple regions allow us to have a better granularity and to focus our estimation on specific parts like the eyes or the mouth. Multiple fusions methods were also presented and tested in order to fuze the information of these regions to predict the pain score of the whole face.

While being a powerful method, the use of human landmarks rely on the performance of landmark detectors. This can be somehow attenuated with the use of the curve fitting algorithm to interpolate missing data. However, if multiple data are missing from the extraction, the algorithm can show its limits and interpolated data can be far from the real data if they were extracted correctly. The smoothing of the trajectories using this algorithm can allow have a great impact on the performance of our approach. It is crucial to find the best parameters to not over smooth the trajectories and thus loosing information.

The lack of information given by only using human landmarks can also be an issue. This can be important in the case of action recognition when multiple actions are very close to each other. This is the case for actions like *all clear* and *not clear* in the UAV-Gesture dataset. The main difference between these two actions is the orientation of the hand. Using a simple body representation to define each skeleton does not allow us to distinguish this information. To overcome this issue, we need to use other landmark detectors in order to extract the landmarks on both hands.

Moreover, our approach is dependent of the number of sequences in each dataset. The computation time to compute the similarity matrix that will be used as our kernel with SVM or SVR depends on the number of samples. Increasing the number of sequences increase the number of similarity scores to compute and thus the time to compute the kernel. Therefore, our method is not appropriate for a real-time use because each new sample needs to be compared to each samples in the training set. To overcome this problem, we can adapt a sliding window over the sequences to only compute the similarity score of a part of each new sequence. Finding a representation of each trajectory by a single point on the manifold can also be used. In this case, we do not need to compute the distances between all the elements composing two trajectories but only the distance between two point on the manifold, which greatly simplify the construction of the similarity matrix.

5.2 Future Work

As future works, we would like to investigate the following points:

- As presented in the discussion at the end of Chapter 3, the computation time of such framework can be an issue when using large dataset. The number of operations to compute the similarity matrix greatly increase when we add new data, as a new distance matrix needs to be computed by measuring the distances between all Gram matrices of the trajectories before applying the temporal alignment with the Global Alignment Kernel or Dynamic Time Warping in order to get the similarity scores. One way to reduce this computation time is by clustering the training sequences to reduce the number of sequences needed to build the kernel. One solution could be by considering a mean trajectory for each class or value (in the case of continuous values like pain levels). This will reduce the number of trajectories considered on the manifold and therefore reduce the size of the similarity matrix to compute. This can also speed-up the process of classification or regression by having a smaller kernel.
- Extracted human landmarks my lie on non-linear manifolds and the use of traditional machine learning techniques can be difficult. Some recent approaches tried to combine deep

learning methods with Riemannian geometry to take in into account the geometry of the data. Based on these approaches, it could be interesting to consider neural network architecture that accept embedding of data extracted from the manifold. Applying deep learning techniques directly on the tangent space of the data can also be a possibility, however, we need to define the right tangent space and exploiting logarithm and exponential maps as well as parallel transport to preserve the geometry of the original data mapped on the tangent space from the manifold and vice versa.

– With the recent breakthrough of Transformer architectures in Computer Vision, it becomes possible to consider such architecture to work with human landmarks. The attention mechanism employed in theses architecture is powerful and has demonstrated its effectiveness in multiple applications. Recent approaches already use Transformers combined with Graph Neural Network that consider as input sequences of body joints for the task of action recognition. Hence, it becomes possible to use a Transformer neural network to estimate the most important facial landmarks to characterize the pain expression. The network could learn the relation between the landmarks by computing spatial attention and highlight the regions of the face in which the pain is present or not. A temporal attention could also be employed to determined the important frames in a sequence on which the network should focus its computation to determine the pain level of the sequence.

Bibliography

[1] B. Aicher, H. Peil, B. Peil, and H.-C. Diener. Pain measurement: Visual analogue scale (vas) and verbal rating scale (vrs) in clinical trials with otc analgesics in headache. *Cephalalgia*, 32(3):185–197, 2012.

[2] Raquel Almeida, Zenilton Kleber Gonçalves do Patrocínio Jr., and Silvio Jamil Ferzoli Guimarães. Exploring quantization error to improve human action classification. In *2017 International Joint Conference on Neural Networks, IJCNN 2017, Anchorage, AK, USA, May 14-19, 2017*, pages 1354–1360, 2017.

[3] Rushil Anirudh, Pavan K. Turaga, Jingyong Su, and Anuj Srivastava. Elastic functional coding of riemannian trajectories. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 39(5):922–936, 2017.

[4] Dieudonné Fabrice Atrevi, Damien Vivet, Florent Duculty, and Bruno Emile. A very simple framework for 3d human poses estimation using a single 2d image: Comparison of geometric moments descriptors. *Pattern Recognition*, 71:389–401, 2017.

[5] Mohammad Ali Bagheri, Qigang Gao, and Sergio Escalera. Support vector machines with time series distance kernels for action classification. In *IEEE Winter Conf. on Applications of Computer Vision (WACV)*, pages 1–7, 2016.

[6] Boulbaba Ben Amor, Jingyong Su, and Anuj Srivastava. Action recognition using rate-invariant analysis of skeletal shape trajectories. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 38(1):1–13, 2016.

[7] Silvere Bonnabel and Rodolphe Sepulchre. Riemannian metric and geometric mean for positive semidefinite matrices of fixed rank. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1055–1070, 2009.

[8] William Munger Boothby. *An introduction to differentiable manifolds and Riemannian geometry; 2nd ed.* Pure and applied mathematics (Elsevier). Academic Press, Orlando, FL, 1986.

[9] Nicolas Boumal, Bamdev Mishra, P.-A. Absil, and Rodolphe Sepulchre. Manopt, a matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15:1455–1459, 2014.

[10] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[11] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[12] Anargyros Chatzitofis, Leonidas Saroglou, Prodromos Boutis, Petros Drakoulis, Nikolaos Zioulis, Shishir Subramanyam, Bart Kevelham, Caecilia Charbonnier, Pablo Cesar, Dimitrios Zarpalas, Stefanos Kollias, and Petros Daras. Human4d: A human-centric multimodal dataset for motions and immersive media. *IEEE Access*, 8:176241–176262, 01 2020.

[13] Xin Chen, Jian Weng, Wei Lu, Jiaming Xu, and Jiasi Weng. Deep manifold learning combined with convolutional neural networks for action recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 29(9):3938–3952, 2018.

[14] Marco Cuturi. Fast global alignment kernels. In *Int. Conf. on Machine Learning (ICML)*, pages 929–936, 2011.

[15] Marco Cuturi, Jean-Philippe Vert, Øystein Birkenes, and Tomoko Matsui. A kernel for time series based on global alignments. In *Proceedings of the IEEE International Conference*

on Acoustics, Speech, and Signal Processing, ICASSP 2007, Honolulu, Hawaii, USA, April 15-20, 2007, pages 413–416, 2007.

[16] M. Daoudi, Z. Hammal, A. Kacem, and J. F. Cohn. Gram matrices formulation of body shape motion: An application for depression severity assessment. In *Int. Conf. on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pages 258–263, 2019.

[17] Maxime Devanne, Hazem Wannous, Stefano Berretti, Pietro Pala, Mohamed Daoudi, and Alberto Del Bimbo. 3-D human action recognition by shape analysis of motion trajectories on Riemannian manifold. *IEEE Trans. on Cybernetics*, 45(7):1340–1352, 2015.

[18] Dylan Drover, Rohith M. V, Ching-Hang Chen, Amit Agrawal, Ambrish Tyagi, and Cong Phuoc Huynh. Can 3d pose be learned from 2d projections alone? In Laura Leal-Taixé and Stefan Roth, editors, *Computer Vision – ECCV 2018 Workshops*, pages 78–94, Cham, 2019. Springer International Publishing.

[19] Harris Drucker, Christopher J. C. Burges, Linda Kaufman, Alexander J., and Vladimir Vapnik. Support vector regression machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 155–161, 1996.

[20] Harris Drucker, Christopher J. C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. In M.C. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9. MIT Press, 1996.

[21] Paul Ekman, WV Friesen, and JC Hager. *Facial Action Coding System: The Manual on CD ROM*. 2002.

[22] Diyala Erekat, Zakia Hammal, Maimoon Siddiqui, and Hamdi Dibeklioğlu. Enforcing multilabel consistency for automatic spatio-temporal assessment of shoulder pain intensity. In *Companion Publication of the Int. Conf. on Multimodal Interaction (ICMI Companion)*. Association for Computing Machinery, 2020.

[23] K. D. Craig et al. *The facial expression of pain*. Guilford Press, 2011.

[24] Masoud Faraki, Mehrtash T Harandi, and Fatih Porikli. Image set classification by symmetric positive semi-definite matrices. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–8. IEEE, 2016.

[25] John T. Farrar, Russell K. Portenoy, Jesse A. Berlin, Judith L. Kinman, and Brian L. Strom. Defining the clinically important difference in pain outcome measures. *Pain*, 88(3):287–294, 2000.

[26] Xiang Gao, Wei Hu, Jiaxiang Tang, Jiaying Liu, and Zongming Guo. Optimized skeleton-based action recognition via sparsified graph regression. *arXiv:1811.12013*, 2019.

[27] Andrew Gilbert, John Illingworth, and Richard Bowden. Action recognition using mined hierarchical compound features. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):883–897, 2011.

[28] P.-Y. Gousenbourger, E. Massart, A. Musolas, P.-A. Absil, L. Jacques, J. M. Hendrickx, and Y. Marzouk. Piecewise-Bézier C^1 smoothing on manifolds with application to wind field estimation. *Proceedings of the 25^{th} European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 305–310, 2017.

[29] Pierre-Yves Gousenbourger, Estelle Massart, and P.-A. Absil. Data fitting on manifolds with composite Bézier-like curves and blended cubic splines. *Journal of Mathematical Imaging and Vision*, 61(5):645–671, 2018.

[30] Pierre-Yves Gousenbourger, Estelle M. Massart, and Pierre-Antoine Absil. Data fitting on manifolds with composite bézier-like curves and blended cubic splines. *J. Math. Imaging Vis.*, 61(5):645–671, 2019.

[31] Pierre-Yves Gousenbourger, Estelle M. Massart, Antoni Musolas, Pierre-Antoine Absil, Julien M. Hendrickx, Laurent Jacques, and Youssef Marzouk. Piecewise-bézier C1 smoothing on manifolds with application to wind field estimation. In *25th European Symposium on Artificial Neural Networks, ESANN 2017, Bruges, Belgium, April 26-28, 2017*, 2017.

[32] T. Graepel, R. Herbrich, P. Bollmann-Sdorra, and Obermayer. Classification on pairwise proximity data. In *Advances in Neural Information Processing Systems (NIPS)*, 1999.

[33] Alexei Gritai, Yaser Sheikh, Cen Rao, and Mubarak Shah. Matching trajectories of anatomical landmarks under viewpoint, anthropometric and temporal transforms. *Int. Journal of Computer Vision*, 84(3):325–343, 2009.

[34] Steinn Gudmundsson, Thomas Philip Runarsson, and Sven Sigurdsson. Support vector machines and dynamic time warping for time series. In *IEEE World Congress on Computational Intelligence*, pages 2772–2776, 2008.

[35] Zakia Hammal and Jeffrey F. Cohn. Automatic, objective, and efficient measurement of pain using automated face analysis. *Ken Prkachin, Zina Trost and Kai Karos (EDs.), Handbook of Social and interpersonal processes in pain: We don't suffer alone*, page 121–146, 2018.

[36] M. P. Jensen, C. Chen, and A. M. Brugger. Interpretation of visual analog scale ratings and change scores: a reanalysis of two clinical trials of postoperative pain. *The Journal of Pain*, 4(7):407–414, 2003.

[37] M. P. Jensen, S. A. Martin, and R. Cheung. The meaning of pain relief in a clinical trial. *The Journal of Pain*, 6(6):400–406, 2005.

[38] Zhuolin Jiang, Zhe Lin, and Larry S. Davis. Recognizing human actions by learning and matching shape-motion prototype trees. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(3):533–547, 2012.

[39] M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 20(5):2327–2351, 2010.

[40] A. Kacem, M. Daoudi, B. Ben Amor, S. Berretti, and J. C. Alvarez-Paiva. A novel geometric framework on gram matrix trajectories for human behavior understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[41] 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 Trans. on Pattern Analysis and Machine Intelligence*, 42(1):1–14, 2020.

[42] Markus Kächele, Mohammadreza Amirian, Patrick Thiam, Philipp Werner, Steffen Walter, Günther Palm, and Friedhelm Schwenker. Adaptive confidence learning for the personalization of pain intensity estimation systems. *Evol. Syst.*, 8(1):71–83, 2017.

[43] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Ahmed Sohel, and Farid Boussaïd. Learning clip representations for skeleton-based 3d action recognition. *IEEE Trans. Image Processing*, 27(6):2842–2855, 2018.

[44] Piotr Koniusz, Anoop Cherian, and Fatih Porikli. Tensor representations via kernel linearization for action recognition from 3d skeletons. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, pages 37–53, 2016.

[45] M. Kowalski, J. Naruniec, and T. Trzcinski. Deep alignment network: A convolutional neural network for robust face alignment. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2034–2043. IEEE Computer Society, jul 2017.

[46] M. Kunz, S. Scharmann, U. Hemmeter, K. Schepelmann, and S. Lautenbacher. The facial expression of pain in patients with dementia. *Pain*, 133(1), 2007.

[47] Hanjiang Lai, Shengtao Xiao, Yan Pan, Zhen Cui, Jiashi Feng, Chunyan Xu, Jian Yin, and Shuicheng Yan. Deep recurrent regression for facial landmark detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(5):1144–1157, 2018.

[48] Christoph H. Lampert. Kernel methods in computer vision. *Found. Trends Comput. Graph. Vis.*, 4(3):193–285, 2009.

[49] Bin Li, Xi Li, Zhongfei Zhang, and Fei Wu. Spatio-temporal graph routing for skeleton-based action recognition. In *AAAI Conference on Artificial Intelligence*, 2019.

[50] Ce Li, C. Chen, Baochang Zhang, Qixiang Ye, J. Han, and R. Ji. Deep spatio-temporal manifold network for action recognition. *ArXiv*, abs/1705.03148, 2017.

[51] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Skeleton-based action recognition with convolutional neural networks. In *IEEE International Conference on Multimedia & Expo Workshops*, page 597–600, 2017.

[52] Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Hao-Shu Fang, and Cewu Lu. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10855–10864, 2019.

[53] Dianbo Liu, Fengjiao Peng, Ognjen (Oggi) Rudovic, and Rosalind W. Picard. Deepfacelift: Interpretable personalized models for automatic estimation of self-reported pain. In *AffComp@IJCAI*, volume 66 of *Proceedings of Machine Learning Research*, pages 1–16. PMLR, 2017.

[54] Jingen Liu, Jiebo Luo, and Mubarak Shah. Recognizing realistic actions from videos. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 1996–2003, 2009.

[55] Jun Liu, Gang Wang, Ling-Yu Duan, Kamila Abdiyeva, and Alex C. Kot. Skeleton-based human action recognition with global context-aware attention LSTM networks. *IEEE Trans. Image Processing*, 27(4):1586–1599, 2018.

[56] Li Liu, Ling Shao, Xuelong Li, and Ke Lu. Learning spatio-temporal representations for action recognition: A genetic programming approach. *IEEE Trans. Cybernetics*, 46(1):158–170, 2016.

[57] Daniel Lopez-Martinez and Rosalind Picard. Continuous pain intensity estimation from autonomic signals with recurrent neural networks. In *2018 40th Annual International Confer-*

ence of the IEEE Engineering in Medicine and Biology Society (EMBC), pages 5624–5627, 2018.

[58] Daniel Lopez-Martinez, Ognjen Rudovic, and Rosalind Picard. Physiological and behavioral profiling for nociceptive pain estimation using personalized multitask learning. In *Neural Information Processing Systems (NIPS) Workshop on Machine Learning for Health*, Long Beach, USA, 2017.

[59] Andras Lorincz, Laszlo Jeni, Zoltan Szabo, Jeffrey Cohn, and Takeo Kanade. Emotional expression classification using time-series kernels. In *IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pages 889–895, 2013.

[60] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, and I. A. Matthews. Painful data: The unbc-mcmaster shoulder pain expression archive database. In *IEEE Int. Conf. on Automatic Face and Gesture Recognition (FG)*, pages 57–64, 2011.

[61] Daniel Lopez Martinez, Ognjen Rudovic, and Rosalind W. Picard. Personalized automatic estimation of self-reported pain intensity from facial expressions. In *IEEE Conf. on Computer Vision and Pattern Recognition Workshops CVPR*, pages 2318–2327, 2017.

[62] E. Massart and P.-A. Absil. Quotient geometry with simple geodesics for the manifold of fixed-rank positive-semidefinite matrices. In *Preprint*, 2018.

[63] Estelle Massart, Pierre-Yves Gousenbourger, Nguyen Thanh Son, Tatjana Stykel, and P.-A. Absil. Interpolation on the manifold of fixed-rank positive-semidefinite matrices for parametric model order reduction: preliminary results. In *Proceedings of the 27th European Symposium on Artifical Neural Networks, Computational Intelligence and Machine Learning (ESANN2019)*, pages 281–286, 2019.

[64] Estelle Massart, Julien M. Hendrickx, and P.-A. Absil. Curvature of the manifold of fixed-rank positive-semidefinite matrices endowed with the Bures-Wasserstein metric. In *Proceedings of the 4th conference on Geometric Sciences of Information (GSI 2019)*, pages 739–748, 2019.

[65] Pyry Matikainen, Martial Hebert, and Rahul Sukthankar. Trajectons: Action recognition through the motion analysis of tracked features. In *Workshop on Video-Oriented Object and Event Classification (ICCV)*, September 2009.

[66] Vittorio Mazzia, Simone Angarano, Francesco Salvetti, Federico Angelini, and Marcello Chiaberge. Action transformer: A self-attention model for short-time pose-based human action recognition. *Pattern Recognition*, 124:108487, 2022.

[67] H. Merskey and et al. Pain terms: a list with definitions and notes on usage. *Pain*, 6(3), 1979.

[68] Gilles Meyer, Silvère Bonnabel, and Rodolphe Sepulchre. Regression on fixed-rank positive semidefinite matrices: a Riemannian approach. *Journal of Machine Learning Research*, 12(Feb):593–625, 2011.

[69] Naima Otberdout, Anis Kacem, Mohamed Daoudi, Lahoucine Ballihi, and Stefano Berretti. Automatic analysis of facial expressions based on deep covariance trajectories. *CoRR*, abs/1810.11392, 2018.

[70] Asanka G. Perera, Yee Wei Law, and Javaan Singh Chahl. UAV-GESTURE: A dataset for UAV control and gesture recognition. In *Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part II*, pages 117–128, 2018.

[71] Chiara Plizzari, Marco Cannici, and Matteo Matteucci. Skeleton-based action recognition via spatial and temporal transformer networks. *Computer Vision and Image Understanding*, 208-209:103219, 2021.

[72] Fatemeh Pouromran, Srinivasan Radhakrishnan, and Sagar Kamarthi. Exploration of physiological sensors, features, and machine learning models for pain intensity estimation. *PLOS ONE*, 16(7):1–17, 07 2021.

[73] K. M. Prkachin. The consistency of facial expressions of pain: a comparison across modalities. *Pain*, 51(3), 1992.

[74] K. M. Prkachin and P. E. Solomon. The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain. *Pain*, 139(2):267–274, 2008.

[75] Aleš Procházka, Oldřich Vyšata, Martin Vališ, Ondřej Ťupa, Martin Schätz, and Vladimír Mařík. Bayesian classification and analysis of gait disorders using image and depth sensors of microsoft kinect. *Digital Signal Processing*, 47:169–177, 2015. Special Issue in Honour of William J. (Bill) Fitzgerald.

[76] Haibo Qiu, Chunyu Wang, Jingdong Wang, Naiyan Wang, and Wenjun Zeng. Cross view fusion for 3d human pose estimation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4341–4350, 2019.

[77] Michalis Raptis and Stefano Soatto. Tracklet descriptors for action modeling and video analysis. In *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I*, pages 577–590, 2010.

[78] Pau Rodriguez, Guillem Cucurull, Jordi Gonzàlez, Josep M. Gonfaus, Kamal Nasrollahi, Thomas B. Moeslund, and F. Xavier Roca. Deep pain: Exploiting long short-term memory networks for facial expression classification. *IEEE Transactions on Cybernetics*, pages 1–11, 2017.

[79] Andrés Romero Mier y Teran. *Real-time multi-target tracking : a study on color-texture covariance matrices and descriptor/operator switching*. Theses, Université Paris Sud - Paris XI, December 2013.

[80] Christian Schüldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: A local SVM approach. In *17th International Conference on Pattern Recognition, ICPR 2004, Cambridge, UK, August 23-26, 2004.*, pages 32–36, 2004.

[81] Ashish Semwal and Narendra D. Londhe. Computer aided pain detection and intensity estimation using compact cnn based fusion network. *Applied Soft Computing*, 112:107780, 2021.

[82] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14:199–222, 2004.

[83] J. Su, S. Kurtek, E. Klassen, and A. Srivastava. Statistical analysis of trajectories on riemannian manifolds: Bird migration, hurricane tracking and video surveillance. *Annals of Applied Statistics*, 8(1), 2014.

[84] Jingyong Su, Anuj Srivastava, Fillipe D. M. de Souza, and Sudeep Sarkar. Rate-invariant analysis of trajectories on riemannian manifolds with application in visual speech recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[85] Benjamin Szczapa, Mohamed Daoudi, Stefano Berretti, Alberto Del Bimbo, Pietro Pala, and Estelle Massart. Fitting, comparison, and alignment of trajectories on positive semi-definite matrices with application to action recognition. In *IEEE Int. Conf. on Computer Vision (ICCV) Workshops*, Oct 2019.

[86] Benjamin Szczapa, Mohamed Daoudi, Stefano Berretti, Pietro Pala, Alberto Del Bimbo, and Zakia Hammal. Automatic estimation of self-reported pain by interpretable representations of motion dynamics. In *25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021*, pages 2544–2550. IEEE, 2020.

[87] Sima Taheri, Pavan Turaga, and Rama Chellappa. Towards view-invariant expression analysis using analytic shape manifolds. In *2011 IEEE International Conference on Automatic Face Gesture Recognition (FG)*, pages 306–313, 2011.

[88] Amor Ben Tanfous, Hassen Drira, and Boulbaba Ben Amor. Coding kendall's shape trajectories for 3d action recognition. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 2840–2849, 2018.

[89] Bart Vandereycken, P-A Absil, and Stefan Vandewalle. Embedded geometry of the set of symmetric positive semidefinite matrices of fixed rank. In *Statistical Signal Processing, 2009. SSP'09. IEEE/SP 15th Workshop on*, pages 389–392. IEEE, 2009.

[90] Bart Vandereycken, P.-A. Absil, and Stefan Vandewalle. A Riemannian geometry with complete geodesics for the set of positive semidefinite matrices of fixed rank. *IMA Journal of Numerical Analysis*, 33(2):481–514, 2013.

[91] Vivek Veeriah, Naifan Zhuang, and Guo-Jun Qi. Differential recurrent neural networks for action recognition. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 4041–4049, 2015.

[92] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3D skeletons as points in a Lie group. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 588–595, 2014.

[93] Raviteja Vemulapalli and Rama Chellapa. Rolling rotations for recognizing human actions from 3d skeletal data. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4471–4479, 2016.

[94] Michalis Vrigkas, Vasileios Karavasilis, Christophoros Nikou, and Ioannis A. Kakadiaris. Matching mixtures of curves for human action recognition. *Computer Vision and Image Understanding*, 119:27–40, 2014.

[95] S. Walter, S. Gruss, H. Ehleiter, Junwen Tan, H. C. Traue, P. Werner, A. Al-Hamadi, S. Crawcour, A. O. Andrade, and G. Moreira da Silva. The biovid heat pain database data for the advancement and systematic validation of an automated pain recognition system. In *IEEE Int. Conf. on Cybernetics (CYBCO)*, pages 128–131, 2013.

[96] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 3169–3176, 2011.

[97] P. Werner, D. Lopez-Martinez, S. Walter, A. Al-Hamadi, S. Gruss, and R. Picard. Automatic recognition methods supporting pain assessment: A survey. *IEEE Trans. on Affective Computing*, pages 1–1, to appear 2019.

[98] L. Xia, C.C. Chen, and JK 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.

[99] Xiaojing Xu, Jeannie S Huang, and Virginia R De Sa. Pain Evaluation in Video using Extended Multitask Learning from Multidimensional Measurements. In Adrian V. Dalca, Matthew B.A. McDermott, Emily Alsentzer, Samuel G. Finlayson, Michael Oberst, Fabian Falck, and Brett Beaulieu-Jones, editors, *Proceedings of the Machine Learning for Health NeurIPS Workshop*, volume 116 of *Proceedings of Machine Learning Research*, pages 141–154. PMLR, 13 Dec 2020.

[100] Xiaojing Xu and V. R. D. Sa. Exploring multidimensional measurements for pain evaluation using facial action units. In *IEEE Int. Conf. on Automatic Face and Gesture Recognition (FG)*. IEEE Computer Society, 2020.

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

[102] Yanhua Yang, Cheng Deng, Dapeng Tao, Shaoting Zhang, Wei Liu, and Xinbo Gao. Latent max-margin multitask learning with skelets for 3-d action recognition. *IEEE Trans. Cybernetics*, 47(2):439–448, 2017.

[103] Sang Min Yoon and Arjan Kuijper. Human action recognition using segmented skeletal features. In *20th International Conference on Pattern Recognition, ICPR 2010, Istanbul, Turkey, 23-26 August 2010*, pages 3740–3743, 2010.

[104] Jarred Younger, Rebecca McCue, and Sean Mackey. Pain outcomes: A brief review of instruments and techniques. *Current Pain and Headache Reports*, 13:39–43, 2009.

[105] Hongwen Zhang, Qi Li, and Zhenan Sun. Joint voxel and coordinate regression for accurate 3d facial landmark localization. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2202–2208, 2018.

[106] Xikang Zhang, Yin Wang, Mengran Gou, Mario Sznaier, and Octavia I. Camps. Efficient temporal sequence comparison and classification using gram matrix embeddings on a riemannian manifold. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4498–4507, 2016.

[107] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *AAAI Conference on Artificial Intelligence*, 2016.