Learning algorithms on Grassmann manifolds

March 2021

Lincon Sales de Souza

Learning algorithms on Grassmann manifolds

Graduate School of Systems and Information Engineering
University of Tsukuba

March 2021

Lincon Sales de Souza

Abstract

Subspace representation has been successfully applied to the classification of image sets and acoustic signals. Subspaces can model compactly gaussian-like noise caused by various factors, such as an object's appearance, effectively representing the variations such as the change in pose and illumination condition. Subspaces exist on a Riemannian manifold called Grassmannian, and as such, subspace processing is performed on this space. Various conventional classification methods have been constructed on the Grassmannian, ranging from simple 1-nearest neighbor to more discriminant ones using a kernel trick. However, there is still potential for improvement of the discriminant ability of subspace representation. This thesis develops learning algorithms to classify subspace-valued data, with applications in signal processing and computer vision. First, we introduce general-purpose, straightforward methods that generalize the Fisher discriminant analysis: enhanced Grassmann discriminant analysis (eGDA) to enhance the discriminant ability in motion recognition; and Grassmann singular spectrum analysis (GSSA) and tangent SSA (TSSA) for acoustic signal classification. Then, we introduce the Grassmann log model, an end-to-end learnable method, integrating subspace representation with deep learning (DL) to enhance the discriminant ability of subspace representation in several computer vision tasks. We demonstrate the validity of the proposed method and its extensions through comprehensive experiments on face identification, face expression, static and dynamic hand gesture classification, and human body action recognition.

Acknowledgements

This thesis is the culmination of efforts from many people around me. Therefore, I would like to use this opportunity to express my sincere gratitude to those who supported me during this Ph.D. journey.

First, I would like to acknowledge the institutions that enabled me to pursue a doctoral degree. For its scholarship, I thank the Japanese Ministry of Education, Culture, Sports, Science, and Technology (MEXT). Then, I would like to acknowledge the University of Tsukuba for offering academic and technical support to perform my research.

I would like to express my sincere gratitude to my advisor Prof. Kazuhiro Fukui for his support and guidance of my research. It was a privilege to learn about the subspace-based methods from him. Besides sharing his knowledge with me, he offered encouragement and was always willing and enthusiastic to assist me. His cheerful attitude and dedication to research will continue to be a reference for my future career.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Hideitsu Hino, Prof. Keisuke Kameyama, Prof. Jun Sakuma, and Prof. Shoji Makino, for their insightful comments and for the questions which led me to broaden my research from various perspectives.

Throughout my research journey, I had the pleasure of working with many other knowledgeable researchers. My sincere thanks go to Dr. Takumi Kobayashi and Prof. Yutaka Satoh for allowing me to join their team at AIST as an intern and using their facilities. The weekly discussions with Dr. Kobayashi were essential to expand my research towards new exciting directions. Without his precious support, it would not be possible to conduct this research. I am also indebted to Prof. Jing-Hao Xue for collaborating with me in writing my first journal paper. Special thanks go to Prof. Eulanda M. dos Santos, Prof. Waldir S. S. Júnior, Prof. Kenny V. Santos and Prof. Juan Colonna, who co-authored papers with me.

I also would like to thank my friend and colleague, Dr. Bernardo Gatto. Throughout the years, we collaborated on many projects and had many stimulating discussions about challenging research problems. He has always encouraged me to keep walking and inspired me to explore new directions.

I thank my fellow actual and former labmates in the Computer Vision Lab.: Ms. Erica Kido, Mr. Naoya Sogi, for research collaborations. I also thank them together with Mr. Bojan Batalo and all other members for the countless paper proofreadings and for having interesting and pleasant discussions, not only about research but about many facets of life. Thank you for making this journey an enjoyable experience.

Thank you to Mrs. Hiroko Sawabe, the Computer Vision laboratory's administrative staff, for supporting me with paperwork throughout six years since I was a research student.

A warm thank you goes to my friends. They put up with my distractions, listened to me unburden, and did not mind my absence. I am forever grateful for their patience and understanding. I hope to have time now to reconnect with each of them. I extend my gratitude to my mother for her love and moral support. Her unwavering faith, honesty, and effort took me to where I stand today.

Finally, I thank my fiancee Lisa for her love and understanding. Without her believing in me, I would not be able to overcome all the dark times I have faced. We walked together in this endeavor, and she earned this degree right along with me.

Contents

List of Figures

Chapter 1

Introduction

This thesis proposes subspace-based methods for applications in computer vision and signal processing. The *linear subspace*, meaning a subset of vector space closed under linear combinations, is an essential concept in mathematics and physics. Here, we use the subspace as an approach to represent data in engineering problems, specifically the fields of pattern recognition and machine learning.

We seek to exploit subspaces' useful properties for classifying data accurately. To reach this goal, we propose four methods to classify image sets and acoustic signals represented as subspaces. To establish the motivation and position this work in the field, we first overview the subspace-based methods.

1.1 Overview of subspace-based methods

Subspaces have an extensive history in pattern recognition and statistics. We briefly describe subspace methods used for data representation and classification while referring to the diagram in Figure 1.1. The methods illustrated by gray boxes indicate conventional methods, while red boxes indicate methods proposed in this thesis. The arrows show that some method is an extension/ generalization of a previous method.

Subspaces were arguably first used in statistics for representing a data distribution in Principal component analysis (PCA), invented by Hotelling [51]. It can be seen as a discrete form of the Karhunen-Loève expansion (KL expansion), which was independently invented by Karhunen [62] and Loève [75]. KL expansion was also independently introduced by Iijima [58] and Watanabe (Watanabe, 1965) [117] in pattern recognition.

The subspace-based methods started from the central problem of pattern recognition: matching two patterns. A *similarity score* to compare objects can be defined in numerous ways, but a reasonable similarity should capture the regularities in data and ignore the "noise". This noise can come from measurement devices, quantization processes, but most important, from the fact that no two objects are identical in the real world, even if they belong to the same category [79].

One way to define a similarity is by the correlation between two patterns, i.e., the angle between them as vectors. This similarity applied to pattern matching became known as the correlation method [60], indicated at the top of Figure 1.1. However, when applied to image patterns, such a

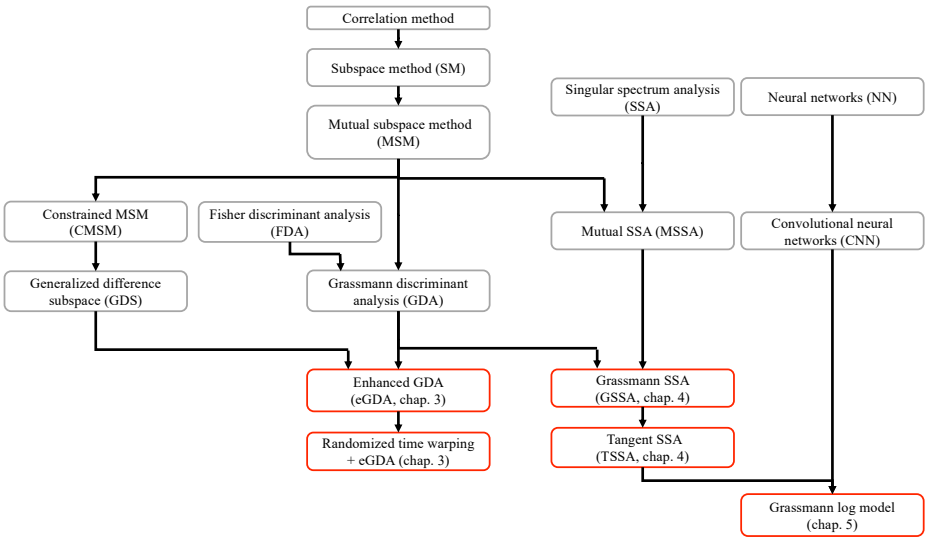


Figure 1.1: Diagram of the algorithms proposed in this thesis (in red boxes) and the relationship with several conventional algorithms (in gray boxes).

simple matching method has a problem caused by a change of shape or position of the patterns. Even though the amount of change is small, the correlation can change drastically. The subspace-based methods were invented to allow similarities that were more stable against pattern variations and more discriminative against similar classes [79].

The subspace method (SM) is a pattern matching method where a matching more sophisticated than correlation is employed. SM is the second method in Figure 1.1. The key idea is that the dictionary or the reference pattern is represented not with a single pattern but a subspace. The similarity is then defined as the angle from an input pattern to its projection onto the reference subspace. SM employs a variant of principal component analysis (PCA) without centering to generate the subspace.

Iijima's method was originally called the multiple similarity method [56, 57], and corresponding Watanabe's method was called CLAFIC [118]. They are slightly different in dealing with the variances of data, but the process is almost the same: given a set of class training patterns, apply PCA on them to make a class subspace. Then, calculate similarities for all classes to classify an unknown pattern.

The subspace methods have also been investigated by many researchers, such as Kittler and Young [67], Kohonen [68] and Fu and Yu [25]. Several subspace methods have been proposed, e.g., the learning subspace methods [69, 85] and orthogonal subspace method [63]. For a detailed treatment, refer to a textbook by Oja [84].

An important generalization of SM is the mutual subspace method (MSM), the third method in Figure 1.1. SM improved the stability of similarity against the changes of pattern positions or shapes. MSM was created from the idea that if both the input and reference patterns are represented with subspaces, we may expect even more stability against the changes. This new idea required similarity between two subspaces, constructed from the mathematical concept of canonical angles between subspaces. The original MSM used the minimum angle (i.e., the first canonical angle) between

subspaces as a similarity. MSM was employed successfully in hand-printed Chinese character (Kanji) recognition, where there are many classes, and each class has countless variations. In a paper by Maeda et al. [80], MSM was employed for 3D face recognition, and multiple canonical angles were used to compute similarity. The reason for introducing more angles is that as faces have far more complicated 3D shape than that of 2D characters, more canonical angles are needed to capture information to differentiate two faces' representative landmarks.

A limitation of MSM is that it does not exploit any discriminative mechanism, i.e., it does not try to separate subspaces from different classes. Various methods were proposed to add discrimination, such as the Grassmann discriminant analysis (GDA) [41], constrained MSM (CMSM) [122], orthogonal mutual subspace method (OMSM) [29], projective metric learning (PML) [54] and discriminative canonical correlation (DCC) [64]. Other extensions of MSM include the kernel MSM (KMSM) [93] and Grassmann Dictionary Learning (GDL) [43]. In the following paragraphs, we discuss CMSM and GDA, two methods marked in Figure 1.1 that are foundational for this thesis.

CMSM (leftmost in Figure 1.1) corresponds to MSM conducted on a generalized difference subspace (GDS) [27]. GDS was invented as a straightforward and elegant solution to tackle pattern-set related problems such as face and hand shape recognition. A GDS is a discriminative subspace containing the peculiar components that differentiate one class from another. It is obtained by computing a common subspace from all classes' patterns and then removing the first components, containing landmarks typical to all classes. All reference and input subspaces are projected onto the GDS, producing discriminative subspaces. The constraint subspace's effect depends on the application and may have interesting properties, such as nearly orthogonalizing subspaces of different classes and removing common features between them.

For most of its history, subspace-based methods were formulated mostly by using linear algebra and numerical analysis. However, the set of all subspaces forms a space with a natural Riemannian structure. This space is named Grassmann manifold (or Grassmannian) and has been well studied in mathematics and physics [1, 12, 26]. Therefore, subspace methods for engineering can also be developed from the perspective of Riemannian geometry. From this viewpoint, (dis)similarities can be thought of as geodesic distances in the Grassmannian, and MSM can be regarded as the nearest neighbor (1NN) between subspaces on this manifold.

Grassmann discriminant analysis (GDA) [41] (center in Figure 1.1) is one popular discriminative extension of MSM that exploits the Grassmannian structure. GDA is a generalization of the Fisher discriminant analysis (FDA) [23, 95] to the Grassmann manifold. It computes a discriminant analysis through a kernel trick from the Grassmannian to a Euclidean space.

In computer vision, subspace-based methods have been employed in several tasks, e.g., face [63, 93, 127], hand gesture recognition [83, 123, 38, 32, 112], and robot vision [28, 13]. It has been the most popular with the task of image set recognition, where multiple images are used to model an object, rather than a single image.

While so far we have focused on pattern recognition, subspaces also have been used in signal processing. Singular spectrum analysis (SSA) [47] has been used for tasks such as signal denoising and source separation [39]. It has recently been combined with MSM for signal classification, resulting in a method called mutual SSA (MSSA) [31]. MSSA is an elegant method for comparing signals in terms of their frequency information, offering low storage, consistent compactness ratio selection, and invariance to signal length.

Finally, subspaces have recently appeared as tools in the field of machine learning. Recent

advances in deep neural networks (DNN) [98, 72, 16] made them into powerful feature extractors. Some researchers started to explore the direction of combining subspaces and DNNs for achieving better stability in classification of image sets. Among the works using subspaces and DNNs, we can cite Gatto et al.s work [35, 37, 33], which uses subspaces as filters of convolution in shallow networks, and Sogi et al. [100], which performs subspace classification with subspaces of fine-tuned DNN features. However, it is not easy to integrate subspace representation and DNNs to exploit subspace's invariance and DNN's expressivity. Therefore, subspace representation in the paradigm of end-to-end gradient-based learning DNNs is mostly unexplored.

1.2 Motivations

The motivation to use subspace representation is that 1) it arises naturally in many types of data, and 2) it is both practical, robust to noise, and invariant to some types of change.

Image data (e.g., MRI tractography, movie clips), signal data (e.g., machine vibrations, bioacoustic records), text data (e.g., medical reports, tweets), biological data (e.g., gene expression levels, metabolomic profile) often come in the form of a set of feature vectors. We can arrange these vectors conveniently as columns of a matrix (e.g., frame matrices of grayscale values, signal trajectory matrices, gene-microarray matrices of gene expression levels, term-document matrices of term-frequency inverse document-frequencies). In modern applications, one will often encounter a prohibitively large sample size and a massive amount of independent variables (high-dimensional problem).

Such raw data in this matrix form is not so informative as it is massive and contains considerable noise. Instead, its eigenspace, i.e., the columns' subspace, is much more interesting. Moreover, the raw data matrix can usually be well approximated by a low-dimensional subspace with basis vectors corresponding to the matrix's largest eigenvalues. We call data in such form by the term Grassmann-valued data.

Grassmann-valued data are robust to noise and invariant to changes in some data factors often seen as noise in applications. Let us consider, for example, an image application such as face recognition. Images with changes in illumination settings can be modeled into an illumination subspace. This illumination subspace can be invariant to the illumination changes under a fixed pose of an object. Another situation is to have images containing changes in pattern position and shape of an object. A subspace can model the appearance of such an object compactly containing some pose variations due to subspaces' invariance to linear transformations. Note that a subspace cannot be invariant to all shape and position changes. Furthermore, it can be generated from a small amount of data while keeping this invariance, using principal component analysis (PCA). PCA captures regularities in data, getting as much variance as possible in a small number of dimensions, represented by principal components. By this process of PCA, the illumination variations are captured as linear combinations of the principal components.

Similarly, background sounds are seen as noise in an acoustic signal. Besides, a signal's volume or length should not affect classification. A subspace computed through singular spectrum analysis (SSA) [39] can model a signal's frequency information compactly while maintaining invariance to the signal's volume and length. Since SSA works as a low-pass filter, a subspace can also filter out high-frequency background noise.

Grassmann-valued data emerge in a wide range of applications: computer vision, signal processing, natural language processing, bioinformatics, machine learning, communication, coding theory, statistical classification, and system identification [124]. With such a considerable potential for applications, it is essential to develop Grassmann-valued data processing algorithms, such as feature extraction, classification, and regression.

The open problem in Grassmann-valued data that is the central motivation of this thesis is: the tools to process Grassmann-valued data are still underdeveloped despite their useful and practical nature. First, the Grassmann manifold is not a Euclidean space, so that standard machine learning methods cannot be promptly utilized to process Grassmann-valued data. Most methods specialized in subspaces are extensions of linear methods in vector space, with nonlinearity added only through a kernel with little learning feedback. As such, their evolution has not yet reached the final stage; they can still be extended to obtain better classification performance in their current applications or enable their use in more complicated unconstrained applications. For example, GDA maps data to a Euclidean space through a kernel trick, a limited representation of the manifold that may not generalize its distance structure well when not enough data is given. Another example is MSSA, which has no discriminant mechanism and works just as the nearest neighbor algorithm for signals. Finally, tools for processing Grassmann-valued data within deep neural networks are lacking.

1.3 Objectives

This thesis aims to provide learning algorithms to classify Grassmann-valued data that can be employed in signal processing and computer vision applications. We propose learning methods to represent the image sets and acoustic signals discriminatively. We also propose different approaches to adequately classify these representations. The goal is to develop general-purpose and straightforward algorithms that achieve high classification performance.

1.4 Contributions

Based on the open problems discussed, we enumerate the contributions of this thesis as follows:

1. We propose an algorithm to enhance the discrimination ability of GDA, named enhanced Grassmann discriminant analysis (eGDA) (Chapter 3). The proposed algorithm can be regarded as a generalization of FDA from Euclidean space to the Grassmann manifold. Our eGDA projects subspaces onto a GDS, which works as a feature extraction on the Grassmannian. It then uses a kernel trick to reproduce the manifold and compute a discriminant. By combining GDS and GDA ideas, we constructed a method that achieves better discrimination ability than both separately. We further combined eGDA with a feature extraction called randomized time warping (RTW) [105] to classify ordered image data. We demonstrate the discriminative ability of this method to the classification of motion images.

2. We propose two algorithms to classify signals, which are extensions of MSSA and Fisher discriminant analysis from Euclidean to Grassmannian (Chapter 4). First, we propose Grassmann singular spectrum analysis (GSSA), a method that uses singular spectrum analysis to

represent signals by subspaces, and then a Grassmann kernel trick just as GDA and eGDA, to map the subspaces onto a discriminant space. Then, we propose the tangent singular spectrum analysis (TSSA). In this method, we exploit the mathematical structure of the Grassmannian as a Riemannian manifold to compute a discriminant directly without a kernel trick, leading to a different mechanism from that of GDA. We compute the discriminant in the tangent space of the data Karcher mean. We demonstrate the application of these methods to the task of bioacoustic signal classification.

3. We attempt to integrate the paradigm of DNNs and subspace representation by introducing a new interface between them, which is called the Grassmann log model, an end-to-end learnable neural network layer (Chapter 5). We aim to integrate subspace representation with deep learning (DL) techniques for more powerful learned representations of Grassmann-valued data. The Grassmann log model maps Grassmann-valued data to vectors by learning a tangent space. This layer can be seamlessly connected to conventional neural network layers. We employ the proposed log model to image set recognition tasks, such as face identification, facial expression recognition, and hand shape recognition.

In summary, this thesis contributes to the line of research of subspace-based methods by introducing new ideas from a differential geometric perspective, focusing on the Grassmannian. Case studies in a wide range of applications are offered, particularly advancing the use of subspace-based methods in signal processing, where its use for classification is relatively unexplored. And finally, we evolve subspace representation by integrating it as a layer of a deep learning framework.

1.5 Thesis organization

The rest of this thesis is organized as follows.

- Chapter 2 provides the mathematical background of the subspace representation and the Grassmannian.
- Chapter 3 introduces enhanced Grassmann discriminant analysis (eGDA).
- Chapter 4 introduces Grassmann singular spectrum analysis (GSSA) and tangent singular spectrum analysis (TSSA).
- Chapter 5 describes the Grassmann log model.
- Chapter 6 concludes the thesis by providing summaries and future works.
- Appendix A demonstrates that a subspace computed through PCA without centering minimizes reconstruction error of a set of patterns.
- Appendix B provides the gradient computation of the Grassmann log map in Chapter 6.

Chapter 2

Theoretical background

In this section, we review the basics of subspace representation, the mathematical structure of the Grassmann manifold and operations we can perform on it. The following notation is used: plain letters for scalars and functions, lowercase bold for vectors, uppercase bold for matrices, lowercase bold greek for Grassmann manifold points and blackboard bold typefaces for sets/spaces such as manifolds and number fields.

2.1 Assumption of subspace representation

First, we discuss the basic idea of representing a set of patterns/ feature vectors by a subspace. The primary assumption of subspace representation is that a low-dimensional subspace can approximate the underlying pattern distribution in a high-dimensional vector space. This assumption's intuition is that the distribution should be more "elliptical" rather than "spherical". Note that the data does not need to be entirely contained by the subspace; one needs only the variance to be considerably larger in a few directions so that we have a unique low-dimensional subspace that fits the data well enough.

2.2 Constructing a subspace

Now we consider the problem of how to compute the m-dimensional subspace $\boldsymbol{\beta}$, i.e. how to find the orthonormal basis $\{\boldsymbol{b}_j \in \mathbb{R}^d\}_{j=1}^m$ of the subspace $\boldsymbol{\beta}$ given sample vectors from the distribution. Let $\{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^n$ be a set of sample patterns from the underlying data distribution, with a sample size n and number of variables (dimension) d. Note that usually m is selected as a hyperparameter, often much smaller than d.

Throughout this thesis, we utilize the principal component analysis (PCA) without centering to compute a subspace. The objective of this procedure is minimizing the *reconstruction error* between the sample patterns $\{\boldsymbol{x}_i\}_{i=1}^n$ and a subspace $\boldsymbol{\beta}$ spanned by the *principal component vectors* $\{\boldsymbol{b}_j\}_{j=1}^m$. This objective can be defined as follows:

$$\min_{\substack{\boldsymbol{\beta}\subset\mathbb{R}^d \\ \dim\boldsymbol{\beta}=m}} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{P}\boldsymbol{x}_i\|_2^2, \tag{2.1}$$

where $\boldsymbol{P} : \mathbb{R}^d \to \mathbb{R}^d$ is the *projection operator* onto $\boldsymbol{\beta}$.

We compute the solution to the problem above as follows. Let $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n] \in \mathbb{R}^{d \times n}$ be the matrix where each column is a sample pattern and $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_m] \in \mathbb{R}^{d \times m}$ be the matrix of basis vectors of $\boldsymbol{\beta}$. We refer to $\boldsymbol{B}$ as *basis matrix*. We can write:

$$\boldsymbol{P} = \boldsymbol{B}\boldsymbol{B}^\top. \tag{2.2}$$

The choice of low-dimensional subspace $\boldsymbol{\beta}$ that minimizes reconstruction error between all patterns $\{\boldsymbol{x}_i\}_{i=1}^n$ is the $\boldsymbol{\beta} = \text{span}(\boldsymbol{B})$ such that the columns of $\boldsymbol{B}$ correspond to the m leading eigenvectors of the *autocorrelation matrix*:

$$\boldsymbol{A} = \boldsymbol{X}\boldsymbol{X}^\top = \sum_{i=1}^{n} \boldsymbol{x}\boldsymbol{x}^\top. \tag{2.3}$$

Note that different from the conventional PCA, we do not utilize the *covariance matrix* $\boldsymbol{C} = \sum_{i=1}^n (\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^\top$, i.e., we do not remove the mean μ from each sample pattern. We also do not assume that the mean is zero.

The autocorrelation matrix $\boldsymbol{A}$ is symmetric positive semidefinite and hence has eigenvectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r$, and corresponding real eigenvalues $\lambda_1, \ldots, \lambda_r$, where $r = \text{rank}\, \boldsymbol{A}$ and $m \leq r$. Without loss of generality we can assume $\lambda_1 \geq \lambda_2 \cdots \geq \lambda_r$. Then, let $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_r] \in \mathbb{S}(d, r) \subseteq \mathbb{R}^{d \times r}$ be the matrix of eigenvectors, where $\mathbb{S}(d, r)$ denotes the manifold of tall orthogonal matrices, called compact Stiefel manifold. Also let $\boldsymbol{\Lambda} \in \mathbb{R}^{r \times r}$ be a diagonal matrix where each k-th diagonal entry is an eigenvalue λ_k. Then, the matrix $\boldsymbol{U}$ diagonalizes $\boldsymbol{A}$ so that:

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{-1} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top. \tag{2.4}$$

We obtain the basis matrix $\boldsymbol{B}$ as the m leading eigenvectors, which we write as:

$$\boldsymbol{B} = \boldsymbol{U}_{1:m}, \tag{2.5}$$

where $\boldsymbol{U}_{1:m}$ denotes the leftmost m columns of $\boldsymbol{U}$. We show the details of the derivation of a subspace that minimizes the reconstruction error in the Appendix A.

2.3 Canonical angles and similarity

In this subsection, we explain the idea of canonical angles between subspaces. Let $\boldsymbol{\chi}$ and $\boldsymbol{\upsilon}$ be subspaces spanned by basis matrices $\boldsymbol{X} \in \mathbb{S}(d, m)$ and $\boldsymbol{Y} \in \mathbb{S}(d, p)$ respectively. In this thesis, these subspace bases are usually computed from PCA as described in the previous section. The canonical angles $\{0 \leq \theta_1, \cdots, \theta_r \leq \frac{\pi}{2}\}$ between $\boldsymbol{\chi}$ and $\boldsymbol{\upsilon}$, where $r = \min(p, m)$, are recursively defined as follows [52, 4]:

$$\begin{aligned} \cos\theta_i = &\max_{\mathbf{u} \in \boldsymbol{\chi}} \max_{\mathbf{v} \in \boldsymbol{\upsilon}} \mathbf{u}^\mathrm{T}\mathbf{v} = \mathbf{u}_i^\mathrm{T}\mathbf{v}_i \\ &s.t.\ \|\mathbf{u}_i\|_2 = \|\mathbf{v}_i\|_2 = 1, \mathbf{u}_i^\mathrm{T}\mathbf{u}_j = \mathbf{v}_i^\mathrm{T}\mathbf{v}_j = 0, i \neq j, \end{aligned} \tag{2.6}$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$ are the canonical vectors forming the i-th smallest canonical angle θ_i between $\boldsymbol{\chi}$ and $\boldsymbol{\upsilon}$. The j-th canonical angle θ_j is the smallest angle in the direction orthogonal to the canonical angles $\{\theta_k\}_{k=1}^{j-1}$.

This optimization problem can be solved from the orthogonal basis matrices of subspaces $\boldsymbol{\chi}$ and $\boldsymbol{\upsilon}$, i.e., $\cos^2\theta_i$ can be obtained as the i-th largest singular value of $\boldsymbol{X}^\top\boldsymbol{Y}$ [52, 4]:

$$\boldsymbol{X}^\top\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top. \tag{2.7}$$

Here, the columns of $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{S}(d, r)$ are called canonical vectors, a basis for $\boldsymbol{\chi}$ and $\boldsymbol{\upsilon}$ that is the closest to the opposing subspace. The matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{r\times r}$ is a diagonal matrix where each element corresponds to a singular value $\sigma_i = \cos^2\theta_i$.

The similarity between two subspaces used throughout this thesis is defined as the sum of the squared cosines of the canonical angles θ_i between $\boldsymbol{\chi}$ and $\boldsymbol{\upsilon}$:

$$s(\boldsymbol{\chi}, \boldsymbol{\upsilon}) = \sum_{i=1}^{r} \cos^2\theta_i = \sum_{i=1}^{r} \sigma_i = \operatorname{tr}\boldsymbol{\Sigma}. \tag{2.8}$$

2.4 Grassmann manifold

The *Grassmann manifold* $\mathbb{G}(d, m)$ is defined as the set of m-dimensional linear subspaces of $\mathbb{R}^d$. It is an $m(d-m)$-dimensional compact manifold and can be written as a quotient space of orthogonal groups $\mathbb{G}(d, m) = \mathbb{O}(d)/\mathbb{O}(m)\times\mathbb{O}(d-m)$, where $\mathbb{O}(m)$ is the group of $m\times m$ orthonormal matrices.

A point $\boldsymbol{\chi} \in \mathbb{G}(d, m)$, i.e., a m-dimensional subspace of $\mathbb{R}^d$, can be represented extrinsically by an orthogonal basis matrix $\boldsymbol{X} \in \mathbb{S}(d, m)$, where the columns of the matrix form a basis such that $\boldsymbol{\chi}$ is the set of all their linear combinations.

2.5 Tangent spaces

A *tangent space* $T_{\boldsymbol{\kappa}}\mathbb{G}$ at $\boldsymbol{\kappa} \in \mathbb{G}(d, m)$ can be seen intuitively as a subspace of $\mathbb{R}^{d\times m}$, since $T_{\boldsymbol{\kappa}}\mathbb{G}$ can be derived as a horizontal space. Given a point $\boldsymbol{K} \in \mathbb{S}(d, m)$, such that $\operatorname{span}(\boldsymbol{K}) = \boldsymbol{\kappa}$, a *horizontal space* $\mathbb{H}$ is a subspace of the tangent space $T_{\boldsymbol{K}}\mathbb{S}$ (a copy of $\mathbb{R}^{d\times m}$) which contains the directions of infinitesimal variation of $\boldsymbol{K}$ that modify its span. In this work we always work with orthogonal bases for subspaces, so it is intuitive to imagine a tangent space from the derivative of the orthogonal property $\boldsymbol{K}^\top\boldsymbol{K} = \boldsymbol{I}$, given by $\boldsymbol{K}^\top\dot{\boldsymbol{K}} + \dot{\boldsymbol{K}}^\top\boldsymbol{K} = \boldsymbol{0}$. Any tangent vector $\dot{\boldsymbol{K}}$ to a subspace $\boldsymbol{\kappa}$ must satisfy this constraint for some $\boldsymbol{K}$ that spans $\boldsymbol{\kappa}$. The *tangent bundle* $T\mathbb{G} = \{(\boldsymbol{\chi}, \boldsymbol{V}) | \boldsymbol{\chi} \in \mathbb{G}, \boldsymbol{V} \in T_{\boldsymbol{\chi}}\mathbb{G}\}$ is the set of all tangent vectors of $\mathbb{G}$, paired with their respective points.

2.6 Riemannian metric

The Grassmann manifold is Riemannian when endowed with a Riemannian metric g. $\mathbb{G}(d, m)$ is a quotient manifold of $\mathbb{S}(d, m)$ (an embedded submanifold of $\mathbb{R}^{d\times m}$), so the Euclidean metric induces the Grassmannian canonical metric $g : \boldsymbol{U}_1 \times \boldsymbol{U}_2 \;\rightarrow\; \operatorname{tr}\boldsymbol{U}_1^\top\boldsymbol{U}_2$, $\boldsymbol{U}_1, \boldsymbol{U}_2 \in T_{\boldsymbol{\kappa}}\mathbb{G}$. g can be intuitively seen as the restriction of the Euclidean metric to a Grassmann tangent space (a horizontal space $\mathbb{H}$).

2.7 Exponential map

The *exponential map* $\text{Exp} : T\mathbb{G} \times \mathbb{R} \to \mathbb{G}$ can be used to calculate an specific point on a geodesic $\gamma(t)$, parameterized by arc length, given a point $\gamma(0)$, a normalized direction $\dot{\gamma}(t)$ and a length t. It is denoted by $\gamma(t) = \text{Exp}_{\boldsymbol{\kappa}} \boldsymbol{H}$, meaning the point $\gamma(t)$ in the geodesic emanating from $\boldsymbol{\kappa} = \gamma(0)$ in the direction of $\boldsymbol{H} = t\dot{\gamma}(0) \in T_{\boldsymbol{\kappa}}\mathbb{G}$. In this thesis, we utilize the following extrinsic function derived by [1], written in terms of orthonormal matrix representation. Given the basis matrix $\boldsymbol{K} \in \mathbb{R}^{d \times m}$, and given a tangent vector $\boldsymbol{H} \in \mathbb{R}^{d \times m}$:

$$\text{Exp}_{\boldsymbol{K}}\, \lambda \boldsymbol{H} = \text{orth}(\boldsymbol{K}\boldsymbol{Q}(\cos \boldsymbol{\Sigma}\lambda)\boldsymbol{Q}^\top + \boldsymbol{J}(\sin \boldsymbol{\Sigma}\lambda)\boldsymbol{Q}^\top), \tag{2.9}$$

where $\boldsymbol{J}\boldsymbol{\Sigma}\boldsymbol{Q}^\top = \boldsymbol{H}$ is the compact singular value decomposition (SVD) of the tangent vector $\boldsymbol{H}$. Note that $\boldsymbol{H}$ is written in upper case as it is a matrix; yet it is still is a vector in the sense that is a member of a tangent vector space. Here, $\boldsymbol{J}, \boldsymbol{K}, \boldsymbol{Q}$ and $\text{Exp}_{\boldsymbol{K}}\, \lambda \boldsymbol{H}$ are orthogonal matrices, and $\boldsymbol{\Sigma}$ is a diagonal matrix. λ is the geodesic parameter, and can be seen as a step value to control the magnitude of the movement towards the direction $\boldsymbol{H}$.

2.8 Logarithmic map

The inverse of the exponential map is the *logarithmic map* (or log map) $\text{Log} : \mathbb{G} \times \mathbb{G} \to T\mathbb{G}$, denoted by $\boldsymbol{H} = \text{Log}_{\boldsymbol{\kappa}} \boldsymbol{\chi}$. Given two points on the manifold $\boldsymbol{\chi}$ and $\boldsymbol{\kappa}$, one wants to find the tangent vector $\boldsymbol{H}$ at $\boldsymbol{\kappa}$ pointing towards $\boldsymbol{\chi}$. In other words, the log outputs a tangent vector to the shortest path curve between $\boldsymbol{\kappa}$ and $\boldsymbol{\chi}$. Note that $\text{Log}_{\boldsymbol{\chi}} \boldsymbol{\kappa} \neq \text{Log}_{\boldsymbol{\kappa}} \boldsymbol{\chi}$. In this thesis, given two basis matrices $\boldsymbol{X}$ and $\boldsymbol{K}$ for input subspace $\boldsymbol{\chi}$ and anchor $\boldsymbol{\kappa}$, we utilize the following three equations to calculate the log map [2]:

$$\boldsymbol{B} = (\boldsymbol{K}^\top \boldsymbol{X})^{-1}(\boldsymbol{K}^\top - \boldsymbol{K}^\top \boldsymbol{X}\boldsymbol{X}^\top), \tag{2.10}$$

$$\boldsymbol{W}\boldsymbol{\Theta}\boldsymbol{Z}^\top = \boldsymbol{B}^\top, \tag{2.11}$$

$$\text{Log}_{\boldsymbol{K}}\, \boldsymbol{X} = \boldsymbol{H} = \boldsymbol{W}^* \arctan(\boldsymbol{\Theta}^*)\boldsymbol{Z}^{*\top}, \tag{2.12}$$

where $\boldsymbol{W}^*, \boldsymbol{\Theta}^*, \boldsymbol{Z}^*$ represent the matrices with the first m columns of $\boldsymbol{W}, \boldsymbol{\Theta}$ and $\boldsymbol{Z}$ respectively.

2.9 Sample mean and variance in the Grassmann manifold

Since a manifold is not necessarily a vector space, additive mean may not be valid. Instead, inspired by the approach to statistics on manifolds of previous works [24, 110], the Karcher mean is utilized, which can be computed through algorithm 1. Likewise, *sample variance* is defined as the expected value of the squared Riemannian distance from the mean, i.e. $E[\| \text{Log}_{\boldsymbol{\kappa}} \boldsymbol{\chi}\|^2]$, where $\boldsymbol{\kappa}$ is the Karcher mean, $\boldsymbol{\chi}$ is a random point, and $\| \cdot \|$ is the norm for the Riemannian metric.

2.10 Orthogonal projection in matrix space

We use the notation $P_{\boldsymbol{A}}(\boldsymbol{X}) : \boldsymbol{X} \to \sum_{j=1}^{n} \boldsymbol{A}_j \,\text{tr}\, \boldsymbol{A}_j^\top \boldsymbol{X}$ to denote the projection of $\boldsymbol{X}$ onto a n-dimensional subspace spanned by the matrix space basis $\{\boldsymbol{A}_j\}, (j = 1, \ldots, n)$, with the canonical

Algorithm 1: Computation of the Karcher mean

input : subspaces $\{\boldsymbol{x}_i\} \in \mathbb{G}$, where $i = 1, \ldots, N$; a step τ; and a patience ϵ.

$\boldsymbol{k}_0 = \boldsymbol{x}_1$ `// 1: initialise the mean with a sample`

do

  $\Delta\boldsymbol{k} = \frac{\tau}{N} \sum_{i=1}^{N} \mathrm{Log}_{\boldsymbol{k}_j} \boldsymbol{x}_i$ `// 2: log-map points to` $T_{\boldsymbol{k}_j}\mathbb{M}$

  $\boldsymbol{k}_{j+1} = \mathrm{Exp}_{\boldsymbol{k}_j} \Delta\boldsymbol{k}$ `// 3: exp-map velocities to the manifold`

while $\|\boldsymbol{k}\| > \epsilon$ `// 4: stop when the candidate become small`

output $\boldsymbol{k} \in \mathbb{G}$, the Karcher mean
:

metric. Note that this does not refer to subspaces of the Grassmannian of data, but refers to projection onto subspaces of the tangent space.

Chapter 3

Enhanced Grassmann discriminant analysis

In this chapter, we propose a method that generalizes the classical Fisher discriminant analysis (FDA) from Euclidean space to the Grassmann manifold, named *enhanced Grassmann discriminant analysis* (eGDA). This chapter is organized as follows. The background of the proposed method is discussed in Section 3.1. The basic idea of the proposed method is elaborated in Section 3.2. The details of the proposed method are then described in Section 3.3. Experimental results are presented in Section 3.4. Finally, the summary is given in Section 3.5.

3.1 Background

The main goal of the proposed method is to characterize and classify motion image sequences, focusing on hand gestures and human actions. We extend the framework of Grassmann discriminant analysis (GDA) [41] to work more effectively in the application of motion recognition. The problem of GDA that we address in this chapter is that GDA's discriminant space is not necessarily optimal. This limitation becomes even more prominent when representing motion sequences by the randomized time warping (RTW) [105] subspace representation.

Randomized time warping (RTW) is an effective generalization of dynamic time warping (DTW) [19], which is one of the most widely used methods for motion analysis. The core idea of DTW is to compare two sequences by searching for the best alignment of their sequential patterns; this is performed by optimizing a warping function with dynamic programming. In contrast to DTW, RTW has a compact representation and does not need dynamic programming, thus providing a fast and light algorithm. It converts the problem of comparing two sequences to comparing two low-dimensional subspaces, called sequence hypothesis (hypo) subspaces.

This problem can in turn be solved by measuring the canonical angles between them. The mutual subspace method (MSM) [121] is well known as a fundamental classification method using canonical angles, which has been used along with RTW. In this framework, a subspace-based method is regarded as a simple classification method on a Grassmann manifold, where each single subspace is treated as a point, and thereby, each motion video is represented by a point in the manifold.

Other types of classification methods have been constructed on a Grassmann manifold, such as

Grassmann discriminant analysis (GDA) [41], Bayesian classifier on the Grassmann manifold [111], or learning on the manifold [99]. Among them, in particular, RTW formulation has been used along with GDA [105], which has been known as one of the useful tools for image set classification [5, 108]. GDA can be easily conducted as a kernel discriminant analysis (KDA) through the kernel trick with a Grassmann kernel [40, 46].

Although it has been useful to combine RTW with GDA, some issues arise from this representation:

- same-class actions may have vary large variations, while semantically different actions may have similar movements, making different action subspaces closer to each other, causing overlap among them in the worst case;
- although GDA is capable of finding the most discriminant directions in a reproducing kernel Hilbert space (RKHS), it cannot operate the corresponding original subspaces in the Grassmann manifold. Hence, if subspaces were not well separated in the manifold, the corresponding data points on the induced RKHS are also not adequately separated, in such a way that GDA may not be able to separate them.

To address those problems, the key idea proposed in this chapter is to project hypo subspaces onto a generalized difference subspace (GDS) [27], before mapping each subspace onto the RKHS. GDS is a general concept that represents difference among multiple class subspaces, which forms a discriminative space. GDS projection works effectively as a powerful feature extraction for subspace-based methods such as subspace method [84] and mutual subspace method (MSM) [121], as it can enlarge the angles among class subspaces toward the orthogonal status. These subspace methods conduct the classification by using the canonical angles between an input vector/subspace and each reference class subspace. A MSM with GDS projection is called constrained MSM (CMSM) [28]. GDS has also been extended recently, such as by adding regularization [107], working with tensor data [34] and for different applications, i.e. high-dimensional spectral data [126].

It is worth mentioning that other methods have extended the MSM formulation to produce discriminative features. A remarkable example is the discriminative canonical correlation (DCC) [65]. The main motivation of DCC is that the structural similarity between class subspaces is measured by the canonical angles between them. Different from CMSM, DCC iteratively computes a discriminative subspace using the Fisher discriminant analysis (FDA) as an objective function to further improve its class separability. Although its exceptional results, DCC's computational time is usually costly. GDS, on the other hand, requires only an SVD computation, which is very efficient in modern implementations.

Figure 3.1 shows the conceptual diagram of the proposed method. A set of RTW's time elastic (TE) features is extracted by randomly sampling images from an image sequence. Next, a hypo subspace is generated by applying PCA to the set. For each image sequence, a hypo subspace is generated in this way. Finally, the relationship among hypo subspaces comes close to the orthogonal status by projecting them onto the GDS, and then the projected subspaces are mapped onto the RKHS. The reason for performing GDS projections before mapping each subspace with the Grassmann kernel is that GDS can operate the hypo subspaces directly in the vector space. Concretely, when some data overlaps among multiple classes, GDA's vector representation cannot necessarily distinguish these data, even if they are projected onto the optimal discriminant space

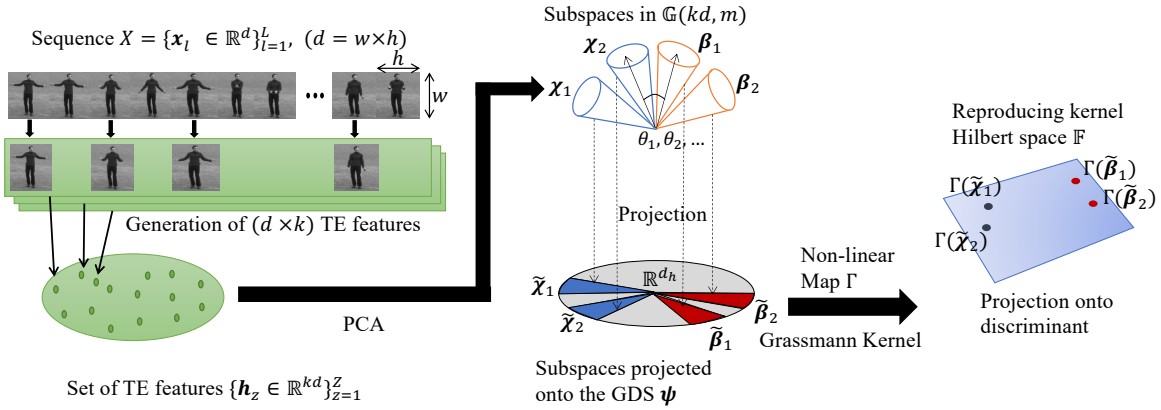


Figure 3.1: Conceptual diagram of the proposed method. A set of TE features is extracted by randomly sampling images from an image sequence. Next, a subspace is generated by applying PCA to the set. For each image sequence, a subspace is generated in this way. Finally, the subspaces are orthogonalized by projecting them onto the GDS, and then are projected onto the RKHS using the Grassmann kernel trick.

found by GDA. In contrast, GDS can remove overlapping components of the subspaces in the vector space, nearly orthogonalizing them, and as a result creating more discriminant data points for GDA.

As GDS has the function of removing common features among class subspaces, providing more discriminative sample for GDA, it is expected that GDS projection can solve the overlap problem and further enhance the representation of the RTW hypo subspaces on the Grassmann manifold. The validity of our proposed method is demonstrated through experiments with the Cambridge gesture [64], KTH action [96] and UCF sports [91, 102] datasets.

In summary, the main contribution of our method is to provide a simple and practical means for further enhancing the performance of GDA, which has been widely used in various applications. In particular, we introduce GDS projection to the GDA formulation to enhance RTW+GDA by alleviating the problems regarding TE feature generation for RTW.

3.2 Basic Idea for enhancing GDA

Our key idea for enhancing Grassmann discriminant analysis (GDA) with hypo subspaces is to project hypo subspaces onto a generalized difference subspace (GDS) before applying GDA to them. In the following, we describe more deeply the problem mentioned in Section 3.1 and the mechanism which induces the effective function to address it.

First, we discuss the intuition of RTW and the advantage of utilizing hypo subspaces as a representation for sequences. The core idea of RTW is to generate a set of time warped patterns, called time elastic (TE) features, through repeated random subsampling, while preserving the original temporal order. This mechanism can be regarded as a simultaneous search for the most similar warped patterns from a number of randomly obtained candidates, in contrast to the inefficient search performed by DTW. As the cost of comparing two sets of TE features increases dramatically as the number of features increase, the comparison is conducted using a subspace based method, in which

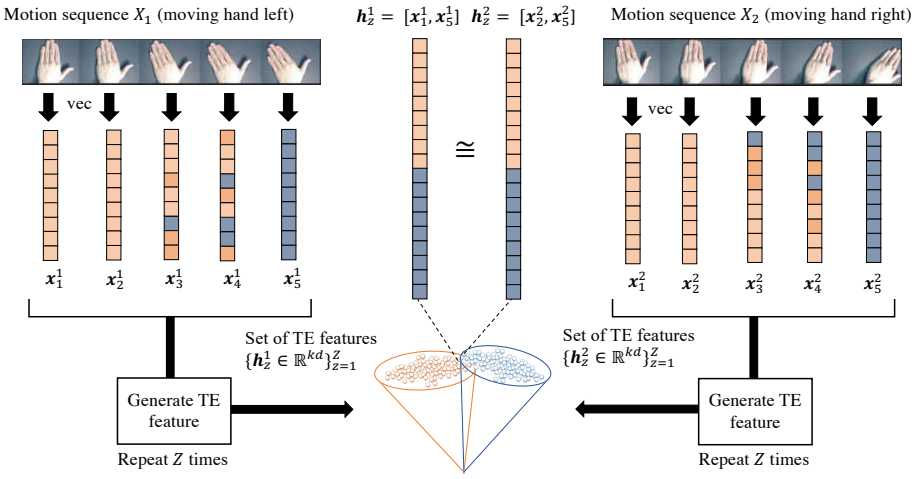


Figure 3.2: Conceptual diagram of the detailed intuition behind RTW. Two sequences of images of different hand gesture classes (e.g. moving hand left and right) are processed through RTW: the frames are vectorized, and a set of TE features is generated by randomly sampling frames and concatenating. In the middle, two examples of TE feature are examined, one vector from each set, which are very close to each other.

each set of TE features is represented as a hypo subspace. In summary, a hypo subspace is a compact representation for sequences as it is independent of sequence length, while the canonical angles offer a simple tool to calculate the most similar warped patterns between two sequences.

In the following we elaborate on the problem of overlapping hypo subspaces. The reason for proximity in the TE feature space is that some frames lead to similar variables. In practice this may have various reasons: 1) some sampled frames may be motionless and composed of similar texture; 2) or their movement is similar in direction; 3) or important moving parts are occluded. Figure 3.2 shows a conceptual diagram of the detailed intuition behind RTW. Two sequences of images of different hand gesture classes (e.g. moving hand left and right) are processed through RTW: each frame $\boldsymbol{x}_l$ corresponds to a vectorized image. In this simple example, a set of TE features is generated by repeating Z times the process of sampling 2 frames from 5 frames and concatenating the 2 frames. A TE feature is denoted as $\boldsymbol{h}_z$, where $z = 1, \ldots, Z$. The center of Figure 3.2 shows the case 1), where two examples of TE features are very close to each other. As a result, when the hypo subspaces of the two sets are generated by applying PCA to each set, they are more likely to overlap.

In applications with real unconstrained data, the probability that concatenated frames present a significative amount of correlation becomes high given various conditions, such as: slow motion speed, specially when the action contains moments of idleness or interruptions; and small appearance changes, specially when the moving target object is far from camera, or some parts are occluded. In many cases, more than one of these factors cause TE features to be close to each other.

To solve this problem, one could think a naive approach of calculating the similarities between frames and then removing similar frames between sets; however, the random sampling of RTW is by itself a statistical technique to avoid the need to compare individual frames, as this is not a scalable operation in terms of complexity. In this sense, a desirable solution needs to consider a subspace

representation, rather than analyzing the individual TE features or their frames.

Now, we explain the definition and mechanism of GDS and how GDS projection can be harnessed for solving the aforementioned problem. GDS is defined as a subspace, which represents a "difference" among multiple class subspaces [27]. GDS is a further extension of difference subspace (DS) for two class subspaces, which is a natural generalization of a difference vector of two vectors.

Given $C(\geq 2)$ m-dimensional *class subspaces*, $\{\boldsymbol{\omega}_c\}_{c=1}^{C}$, a *generalized difference subspace* (GDS) $\boldsymbol{\psi}$, can be defined as the subspace produced by removing the *principal component subspace* (PCS) of all the class subspaces from the sum subspace of those subspaces. This definition of GDS leads GDS projection to the function of automatically removing overlap among class subspaces, which can alleviate the problem. It is worth noting here that we consider removing the overlapping components of the data, not the data themselves. The details of the process of GDS projection will be explained in Sec. 3.3.3. On the other hand, we should note that GDA cannot necessarily distinguish data belonging to overlap region, even by projecting them onto its optimal discriminant space.

Besides, GDS projection has the function of orthogonalizing class subspaces by enlarging the canonical angles among class subspaces. Although GDA also has a similar function, the mechanisms of both are quite different. GDA works on a RKHS, while GDS projection works in the original high dimensional vector space. Based on this difference, we expect different effects from GDS and GDA to learn a discriminant space where the classes are as separated as possible.

3.3 Algorithm of the proposed method

We first describe the representation by RTW to generate a hypo subspace; then we explain how to generate a GDS and use its projection to enhance GDA. The step-by-step training and testing algorithms of the proposed method are shown in Algorithms 2 and 3, respectively.

3.3.1 Motion sequence representation by RTW

In our method, an image with the size $w \times h$ is represented by a d-dimensional vector $\boldsymbol{x} \in \mathbb{R}^d$, either by using the pixels themselves, in which case $d = w \times h$, or by extracting handcrafted or convolutional neural network features. A video is then an ordered sequence of L input vectors $X = \{\boldsymbol{x}_l\}_{l=1}^{L}$, where l denotes frame number. For example, a sequence represents a body motion or hand gesture captured by video. Then, a training set $\{(X_i, y_i)\}_{i=1}^{N}$ consists of a video sample X_i coupled with a respective label y_i.

An $d \times k$ dimensional TE feature vector $\boldsymbol{h} = [\boldsymbol{x}_1^\top \boldsymbol{x}_2^\top \ldots \boldsymbol{x}_k^\top]^\top$ is created by randomly selecting k images from a sequence X, such that $t(\boldsymbol{x}_1) < \ldots < t(\boldsymbol{x}_k)$, where $t(\cdot)$ denotes the original order of the image.

Let this procedure of random selection be repeated Z times, such that we obtain $\boldsymbol{h}_1, \ldots, \boldsymbol{h}_Z$. Subsequently, an auto-correlation matrix $\boldsymbol{R}$, which corresponds to the set of the TE feature vectors of X, can be computed as:

$$\boldsymbol{R} = \frac{1}{Z} \sum_{z=1}^{Z} \boldsymbol{h}_z \boldsymbol{h}_z^\top. \tag{3.1}$$

This procedure corresponds to steps 1 and 2 in Algorithms 2 and 3.

Algorithm 2: Learning algorithm of the proposed method

input training ordered sequences and their labels $\{(X_i, y_i)\}_{i=1}^{N}$
:
for $c = 1, \ldots, C$ **do**
  for $i = 1, \ldots, N_c$ **do**
    $\{\boldsymbol{h}_z\}_{z=1}^{Z} \leftarrow \mathrm{TE}(X_i)$ `// 1: obtain TE features`
    $\boldsymbol{R}_i \leftarrow \frac{1}{Z}\sum_{z=1}^{Z} \boldsymbol{h}_z \boldsymbol{h}_z^{\top}$ `// 2: calculate set covariance matrix`
    $\boldsymbol{X}_i \leftarrow \mathrm{EVD}(\boldsymbol{R}_i)$ `// 3: apply eigendecomposition`
  end
  $\boldsymbol{R}_c \leftarrow \frac{1}{N_c}\sum_{i|y_i=c} \boldsymbol{R}_i$ `// 4: calculate class covariance matrix`
  $\boldsymbol{M}_c \leftarrow \mathrm{EVD}(\boldsymbol{R}_c)$ `// 5: apply eigendecomposition`
end
$\boldsymbol{G} \leftarrow \mathrm{EVD}(\sum_{c=1}^{C} \boldsymbol{M}_c \boldsymbol{M}_c^{\top})$ `// 6: obtain GDS`
foreach i **do** $\tilde{\boldsymbol{X}}_i \leftarrow \boldsymbol{G}^{\top}\boldsymbol{X}_i$ `// 7: project all subspaces onto the GDS`
for $i = 1, \ldots, N$ **do**
  for $j = 1, \ldots, N$ **do**
    $[\boldsymbol{K}]_j^i \leftarrow k_p(\tilde{\boldsymbol{X}}_i, \tilde{\boldsymbol{X}}_j)$ `// 8: generate similarity matrix`
  end
end
$\{\boldsymbol{u}^*\} \leftarrow \max_{\boldsymbol{u}} J(\boldsymbol{u}; \boldsymbol{K})$ `// 9: solve KDA problem`
$\boldsymbol{F}_{\mathrm{tr}} \leftarrow \boldsymbol{u}^{*\top}\boldsymbol{K}$ `// 10: compute training coefficients`
return $\boldsymbol{F}_{tr}, \boldsymbol{G}, \boldsymbol{u}^*$ `// return dictionary, GDS and GDA projection operators`

3.3.2 Subspace representation

We utilize the principal component analysis (PCA) by computing the eigenvectors of a matrix $\boldsymbol{R}_i$ to construct a m-dimensional subspace $\boldsymbol{\chi}_i$. The orthonormal basis of $\boldsymbol{\chi}_i$ is obtained as the eigenvectors corresponding to the m largest eigenvalues. In the following, each m-dimensional subspace $\boldsymbol{\chi}_i$ is represented by the matrix $\boldsymbol{X}_i \in \mathbb{R}^{kd \times m}$, which has the corresponding orthonormal basis as its column vectors. A set of TE features generated from a sequence contains various possible warped patterns in time, each of which corresponds to one hypothesis. In this sense, the subspace generated from a set of TE features is called a sequence hypothesis (hypo) subspace. In Algorithms 2 and 3, the generation of hypo subspaces corresponds to step 3.

3.3.3 Projection onto generalized difference subspace

In order to utilize the feature extraction function of GDS effectively, we introduce the *global class subspaces* $\boldsymbol{\omega}_c$, which is spanned by a matrix $\boldsymbol{M}_c \in \mathbb{R}^{kd \times d_m}$. The subspace $\boldsymbol{\omega}_c$ represents compactly all the subspaces belonging to the same class c. The orthogonal basis of $\boldsymbol{\omega}_c$ can be obtained as the

Algorithm 3: Input evaluation algorithm of the proposed method

input ordered sequences X_{in}
:
$\{\boldsymbol{h}_z\}_{z=1}^{Z} \leftarrow \text{TE}(X_{\text{in}})$ // 1: obtain TE features
$\boldsymbol{R}_{in} \leftarrow \frac{1}{Z}\sum_{z=1}^{Z} \boldsymbol{h}_z \boldsymbol{h}_z^{\top}$ // 2: calculate set covariance matrix
$\boldsymbol{X}_{\text{in}} \leftarrow \text{EVD}(\boldsymbol{R}_{\text{in}})$ // 3: apply eigendecomposition
$\tilde{\boldsymbol{X}}_{\text{in}} \leftarrow \boldsymbol{G}^{\top}\boldsymbol{X}_{\text{in}}$ // 4: project subspace onto the GDS
for $i = 1, \ldots, N$ **do**
  $[\boldsymbol{K}_{\text{in}}]_i \leftarrow k_p(\tilde{\boldsymbol{X}}_i, \tilde{\boldsymbol{X}}_{\text{in}})$ // 5: generate similarity matrix
end
$\boldsymbol{F}_{\text{in}} \leftarrow \boldsymbol{u}^{*\top}\boldsymbol{K}_{\text{in}}$ // 6: compute test coefficients
$\text{pred}(X_{\text{in}}) \leftarrow \text{NN}(\boldsymbol{F}_{\text{tr}}, \boldsymbol{F}_{\text{in}})$ // 7: perform 1-NN classification
return $\text{pred}(X_{in})$ // return a class prediction

eigenvectors corresponding to the d_m largest eigenvalues of the auto-correlation matrix:

$$\boldsymbol{R}_c = \frac{1}{N_c}\sum_{i|y_i=c} \boldsymbol{R}_i. \tag{3.2}$$

Here N_c denotes the number of samples labeled as class c. Note that d_m is a hyperparameter in our framework.

Next, to generate a GDS $\boldsymbol{\psi}$, we calculate the total sum matrix, $\boldsymbol{S}$, which is defined as:

$$\boldsymbol{S} = \sum_{c=1}^{C} \boldsymbol{M}_c \boldsymbol{M}_c{}^{\top}. \tag{3.3}$$

The orthogonal basis of the GDS $\boldsymbol{\psi}$ can be obtained as d_h eigenvectors, $\{\boldsymbol{d}_i\}_{i=1}^{d_h}$ corresponding to the d_h smallest eigenvalues of the sum matrix $\boldsymbol{S}$. We set these basis vectors as columns of a matrix $\boldsymbol{G} \in \mathbb{R}^{d \times d_h}$. A subspace $\boldsymbol{\chi}_i$ is projected onto the GDS by $\tilde{\boldsymbol{X}}_i = \boldsymbol{G}^{\top}\boldsymbol{X}_i \in \mathbb{R}^{d_h \times m}$ and the projected subspace is denoted by $\tilde{\boldsymbol{\chi}} = \text{span}(\tilde{\boldsymbol{X}})$. An input subspace $\boldsymbol{\chi}_{\text{in}}$ is also projected onto the GDS and its projection is denoted by $\tilde{\boldsymbol{\chi}}_{\text{in}}$. In Algorithm 2 the generation of class subspaces and the GDS corresponds to steps 4 to 6, while projection of subspaces is step 7. In Algorithm 3, only projection is performed, corresponding to step 4.

3.3.4 Enhancing Grassmann discriminant analysis

Then, we embed the Grassmann manifold of the projected subspaces, $\mathbb{G}(d_h, m)$, in a reproducing kernel Hilbert space by the use of a Grassmann kernel. In our framework, we use the projection kernel k_p, which can be defined from the subspace canonical angles as:

$$k_p(\tilde{\boldsymbol{\chi}}_1, \tilde{\boldsymbol{\chi}}_2) = \sum_{i=1}^{m} \cos^2\theta_i. \tag{3.4}$$

Here, θ_i refers to the i-th canonical angle between two subspaces $\tilde{\chi}_1$ and $\tilde{\chi}_2$. We can measure the distance between two points on a Grassmann manifold by using this projection kernel [40].

Basically, GDA is conducted as kernel discriminant analysis (KDA) with a Grassmann kernel. KDA [95, 9] can be formulated by using the kernel trick as follows. Let $\Gamma : \mathbb{G} \to \mathbb{F}$ be a non-linear map from the Grassmannian $\mathbb{G}$ to a Hilbert space $\mathbb{F}$, and $\boldsymbol{\Gamma} = [\Gamma(\tilde{\chi}_1), \ldots, \Gamma(\tilde{\chi}_N)]$ be the feature matrix of the mapped training points. Assuming a discriminant direction $\boldsymbol{w} \in \mathbb{F}$ is a linear combination of those feature vectors, $\boldsymbol{w} = \boldsymbol{\Gamma u}$, we can use the kernel trick and write the KDA problem in terms of $\boldsymbol{u}$, without accessing vectors on $\mathbb{F}$:

$$\max_{\boldsymbol{u}} J(\boldsymbol{u}; \boldsymbol{K}) = \max_{\boldsymbol{u}} \frac{\boldsymbol{u}^\top \boldsymbol{\Gamma}^\top \boldsymbol{S}_b \boldsymbol{\Gamma u}}{\boldsymbol{u}^\top \boldsymbol{\Gamma}^\top \boldsymbol{S}_w \boldsymbol{\Gamma u}} = \max_{\boldsymbol{u}} \frac{\boldsymbol{u}^\top \boldsymbol{K}(\boldsymbol{V} - \boldsymbol{e}_N \boldsymbol{e}_N^\top / N)\boldsymbol{K u}}{\boldsymbol{u}^\top (\boldsymbol{K}(\boldsymbol{I}_N - \boldsymbol{V})\boldsymbol{K} + \sigma^2 \boldsymbol{I}_N)\boldsymbol{u}} = \max_{\boldsymbol{u}} \frac{\boldsymbol{u}^\top \boldsymbol{\Sigma}_b \boldsymbol{u}}{\boldsymbol{u}^\top (\boldsymbol{\Sigma}_w + \sigma^2 \boldsymbol{I}_N)\boldsymbol{u}}, \tag{3.5}$$

where $\boldsymbol{S}_b$ is a between-class scatter matrix, $\boldsymbol{S}_w$ is a within-class scatter matrix, $\boldsymbol{K}$ is the kernel matrix, $\boldsymbol{e}_N$ is a vector of ones that has length N, $\boldsymbol{V}$ is a block-diagonal matrix whose c-th block is the matrix $\boldsymbol{e}_{N_c} \boldsymbol{e}_{N_c}^\top / N_c$, and $\boldsymbol{\Sigma}_b = \boldsymbol{K}(\boldsymbol{V} - \boldsymbol{e}_N \boldsymbol{e}_N^\top / N)\boldsymbol{K}$.

In our framework, the kernel matrix, $\boldsymbol{K}$, is calculated as the similarity matrix between training subspaces, where each element can be written in terms of the projection kernel as $k_p(\tilde{\chi}_q, \tilde{\chi}_w)$, where q is a row and w is a column of $\boldsymbol{K}$. The term $\sigma^2 \boldsymbol{I}_N$ is used for regularizing the covariance matrix $\boldsymbol{\Sigma}_w = \boldsymbol{K}(\boldsymbol{I}_N - \boldsymbol{V})\boldsymbol{K}$. It is composed of the covariance shrinkage factor $\sigma^2 > 0$, and the identity matrix $\boldsymbol{I}_N$ of size N. The set of optimal vectors $\{\boldsymbol{u}^*\}$ are obtained as the first $C - 1$ eigenvectors of $(\boldsymbol{\Sigma}_w + \sigma^2 \boldsymbol{I}_N)^{-1} \boldsymbol{\Sigma}_b$. We apply the GDA algorithm to the projected subspaces $\tilde{\chi}_i$. The GDA corresponds to steps 8 to 10 in Algorithm 2, and steps 5 to 7 in Algorithm 3.

3.4 Experiments

In this section, we discuss the validity of the proposed method through hand gesture and human action recognition tasks.

3.4.1 Experiment with Cambridge hand dataset

We conducted two types of experiments with the Cambridge hand gesture dataset [64]. This dataset contains 9 classes of hand gesture videos, each in 5 illumination scenarios, and 20 sample videos for each of the scenarios and classes. The number of frames of each video ranges from 37 to 119. In addition, in the experiments, all the images were resized to 12×16 pixels, and the grayscale pixel values compose the image features. As a result, an original feature vector $\boldsymbol{x}_l^{i,c}$ had dimension 12×16 ($d = 192$). The number of selected frames k to build one TE feature is fixed at $k = 15$, and as a result the dimension of a TE feature vector $\boldsymbol{s}_z^{i,c}$ is $d \times k = 192 \times 15 = 2880$. The number of TE features for each set is fixed to be $Z = 100$. Figure 3.3 shows examples of this dataset.

In the first experiment, we performed a qualitative experiment to aid in the visualization of the proposed method mechanism. We utilized three classes of hand gestures from the Cambridge dataset: flat/contract (C), spread/right (E) and spread/contract (F). The normal illumination setting (set 5) was used for training, and the illumination setting of set 1 was used for testing. The parameters were set up in the following manner: dimension of hypo subspaces m was set to 7; dimension of

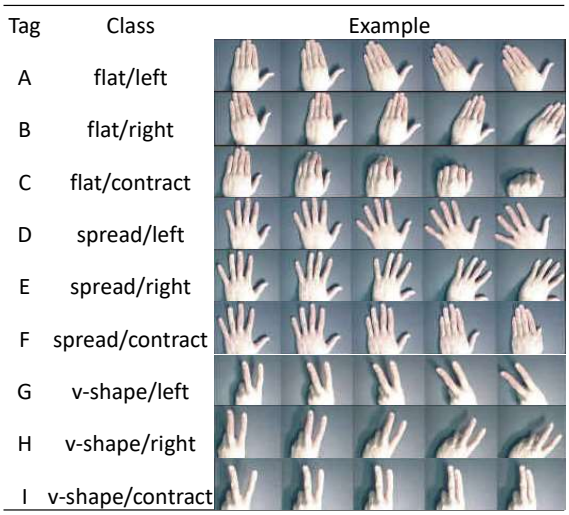


Figure 3.3: Examples of Cambridge hand gestures.

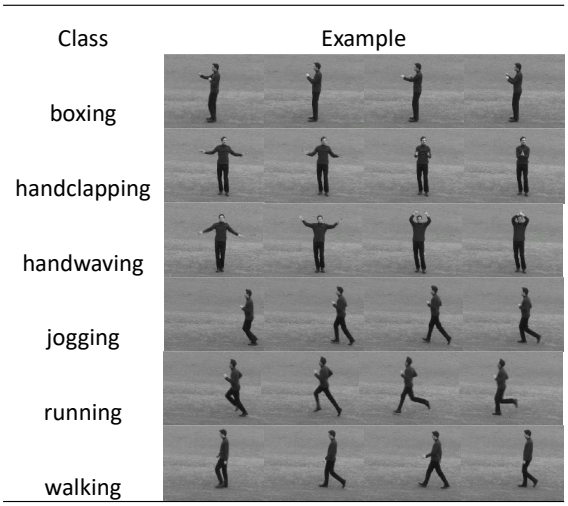


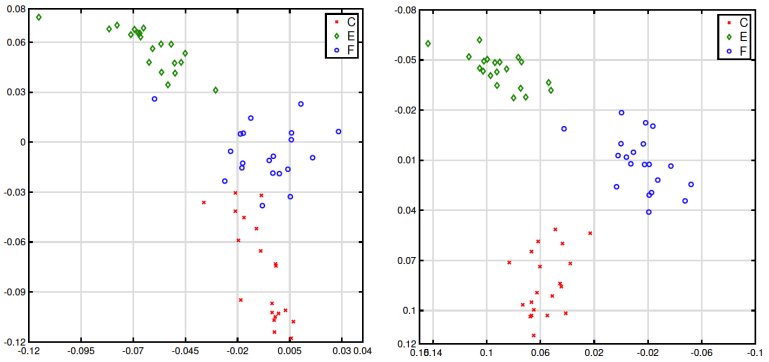
Figure 3.4: Examples of KTH actions.

class subspaces d_m was set to 50 ; and dimension of principal subspace d_p to 5.

In the second experiment, following the same setup as [105], we quantitatively compared our RTW+eGDA with the conventional methods: RTW+GDA, Kim and Cipolla [65], Lui [77] and Hankel [71]. These methods were selected as baselines due to their applications in motion representation and recognition. In Kim and Cipolla [65], the image sets of motions are described as linear subspaces, where a discriminative subspace is created in order to improve the feature extraction ability of the method. Lui [77] represents the image-sets of motions as a factorized tensor, where the geometry of the tensor space is extracted and compactly represented. Finally, in Hankel [71], image-sets of motions are described as autocorrelation matrices computed from Hankel trajectory matrices. In this approach, a discriminative subspace similar to [65] is employed to extract more useful features.

We used the 20 sequences in the normal illumination setting (Set 5) for training, and the remaining sequences in other illumination settings (Sets 1 to 4) for testing. The parameters were varied in the following manner: dimension of hypo subspaces m was varied from 5 to 7; dimension of class subspaces d_m was varied from 30 to 90 in increments of 20; and dimension of principal subspace d_p was varied from 5 to 30 in increments of 5. The results reported here are the best among the parameter settings.

Figure 3.5 shows the qualitative results: scatter plots of the generated points corresponding to the test subspaces, which were generated from the 20 test sequences in each class. In this figure, (a) depicts the result of the combination of RTW and conventional GDA (RTW+GDA), and (b) shows the proposed method, RTW and the enhanced GDA (RTW+eGDA). The figure suggests that by using the proposed method, reduction of the distance between subspaces of the same class can be achieved. The scatter plots given by the Cambridge gesture dataset classes C (flat/contract), E (spread/right) and F (spread/contract) reveals visually that RTW+eGDA is able to produce higher discriminative features than RTW+GDA.



(a) RTW+GDA (b) RTW+eGDA

Figure 3.5: Scatter points of three Cambridge hand gesture classes by using RTW combined with (a) conventional GDA and (b) eGDA.

The results of the quantitative evaluation can be seen in Figure 3.6. The vertical axis refers to the average accuracy of all sets. The error bars represent the method's standard deviation. We also conducted a t-test between RTW+eGDA and RTW+GDA with 4 samples and significance level $\alpha = 0.05$. From the test results, we can conclude with more than 95% confidence ($p = 0.0377$) that the proposed method performed better than the conventional method by using GDA.

3.4.2 Experiment with KTH action dataset

We also conducted experiments using the KTH action dataset [96]. Figure 3.4 shows examples of this database's 6 classes of actions, namely: boxing, hand clapping, hand waving, running, jogging, and walking. The dataset contains actions performed by 25 subjects in videos, filmed under 4 different shooting conditions: outdoors, outdoors with variation of zooming, outdoors with different clothes, and indoors.There are 4 sample videos for each of the conditions and classes. The number of frames of each video ranges from 37 to 119. In addition, in the experiments, all the images were resized to 16×16 pixels. In total there are 2391 sequences of actions.

In the first experiment, we performed a qualitative assessment. For each of the 6 classes, 10 subjects were randomly selected for training, and 15 for testing. The parameters were set up in the following manner: dimension of hypo subspaces m was set to 19; dimension of class subspaces d_m was set to 50 ; dimension of principal subspace d_p to 20; the number of selected frames k to build one TE feature is fixed at $k = 5$, and the number of TE features for each set is fixed to be $Z = 500$.

Figure 3.7 shows the results of the KTH Action Dataset Experiment. To quantitatively confirm the effectiveness of the orthogonalization of RTW class subspaces by GDS projection, we measured the Fisher criterion (class separability degree) among all the classes. The performance is higher as the separability degree approaches 1.0. Table 3.1 shows the experimental results. We can see that

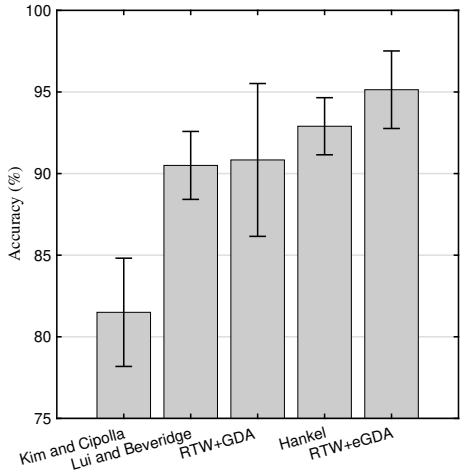


Figure 3.6: Results of the Cambridge Hand Dataset Experiment. The vertical axis refers to the average accuracy of all sets, for each of the methods in the horizontal axis. An error bar represents a method's standard deviation.

the separability degree of GDA is further improved by GDS projection.

Table 3.1: Separability of RTW+GDA and RTW+eGDA in the first experiment using the KTH dataset.

Method	Separability
RTW+GDA	0.23
RTW+eGDA	0.45

Figure 3.8 shows scatter plots of the generated points corresponding to the test subspaces of two classes: handwaving and running, as an example where two classes are comparatively well separated even by a two-dimensional discriminant space. In this figure, (a) depicts the result of the combination of RTW and conventional GDA (RTW+GDA), and (b) shows the proposed method, RTW and the enhanced GDA (RTW+eGDA).

In KTH dataset, the chosen classes for this plot have high overlap. We can observe that both RTW+GDA and RTW+eGDA produced very similar patterns. One observation regarding both investigated datasets is that in the Cambridge dataset temporal information plays an important role. On the other hand, in KTH dataset, temporal information seems to play a weaker role, since some classes (e.g. boxing, handclapping and hand waving) consist of iterations of the same action unit.

We graphically demonstrate that RTW+GDA and RTW+eGDA are well-suited for motion representation, even when dealing with complicated datasets containing cluttered and non-uniform backgrounds. These results encourage us to make another experiment and show the importance of combining RTW and eGDA for motion representation.

Figure 3.9 shows the confusion matrix of RTW+GDA (a) and RTW+eGDA (c). The vertical classes refer to predictions, while the horizontal classes refer to the ground truth. Each number

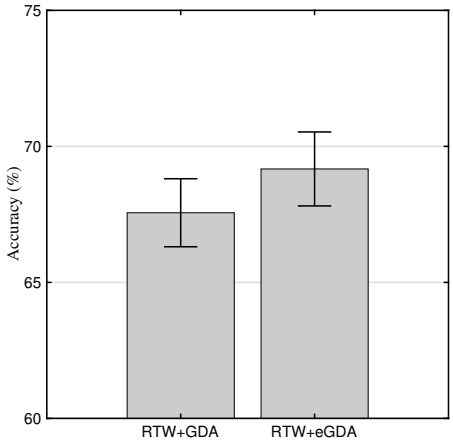


Figure 3.7: Results of the KTH Action dataset experiment. The vertical axis refers to the average accuracy of all 10 folds, for each of the methods in the horizontal axis. An error bar represents a method's standard deviation.

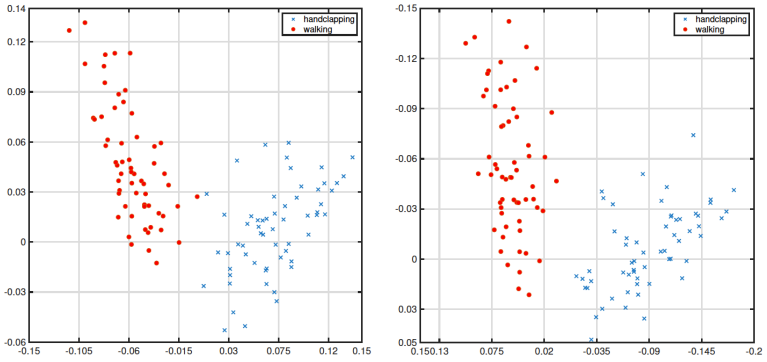
represents the percentage of predictions attributed to a class in relation to their true class. The percentages between parenthesis in each matrix label refer to the average accuracy of the method.

Both RTW+GDA and RTW+eGDA provided efficient results at separating difficult classes in Cambridge dataset. For instance, the methods demonstrated high accuracy in overlapping classes such as A (flat/left), B (flat/right), D (spread/left) and E (spread/right). In the case of employing linear subspaces, where the intrinsic low dimension representation has high level of similarity, such results may weaken. When employing RTW, the temporal coherence between the ordered patterns is efficiently attained, producing very competitive results.

The class with the worst result was F (spread/contract). One of the reasons that can cause such a class to have a low accuracy compared to other classes is that this gesture has a very large number of structures in common with the other classes. Besides, when GDS is employed, such common structures could be removed automatically.

In the second experiment, we quantitatively compared the methods by a 10-fold cross validation scheme, where in each fold, 10 subjects from the 25 were randomly selected for training. For each sequence, we used the bounding box from [61] to do segmentation between actions and resize each original frame to a 16×16 pixels grayscale image. We used the raw pixel values with additional information of the height and width of the bounding box of the subject, resulting in a 258-dimensional vector for each frame.

We compared the combination of RTW and conventional GDA (RTW+GDA) with RTW and the enhanced GDA (RTW+eGDA). The number of selected frames k to build one TE feature is fixed at $k = 5$, and the number of TE features for each set is fixed to be $Z = 500$. The parameters were varied in the following manner: dimension of hypo subspaces m was varied from 5 to 20 in increments of 2; dimension of class subspaces d_m was varied from 30 to 90 in increments of 20; and dimension of principal subspace d_p was varied from 5 to 30 in increments of 5. Figure 3.10 shows the parameter



(a) RTW+GDA (b) RTW+eGDA

Figure 3.8: Scatter points of two KTH action classes by using RTW combined with (a) conventional GDA and (b) eGDA.

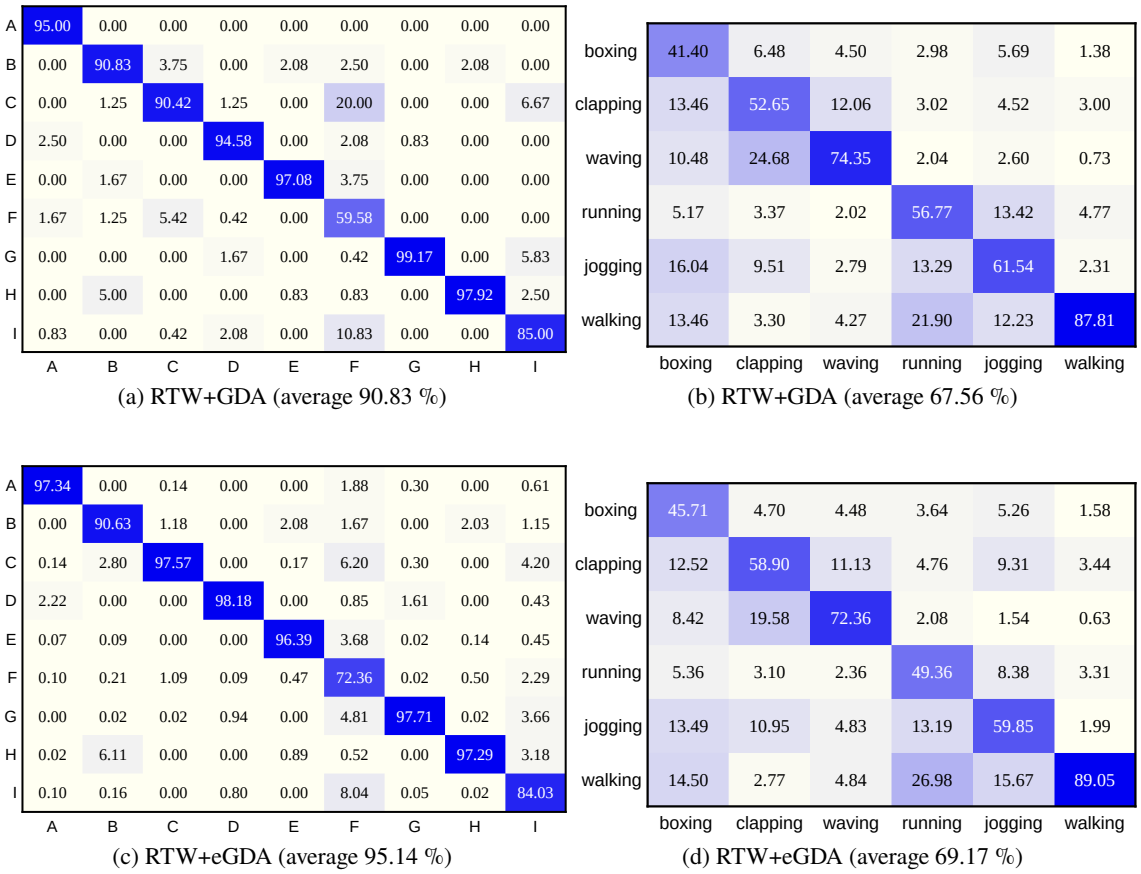
behavior of RTW+eGDA on the KTH action dataset.

The results can be seen in Figure 3.7. The vertical axis refers to the average accuracy of all 10 folds, for each method in the horizontal axis. The error bars represent the method's standard deviation. We again conducted a t-test between RTW+eGDA and RTW+GDA, this time with 10 samples and significance level $\alpha = 0.05$. From the test results, we can conclude with more than 95% confidence ($p = 0.007$) that the proposed method performed better than the conventional method by using GDA.

Figure 3.9 shows the confusion matrix of RTW+GDA (b), and RTW+eGDA (d). The vertical classes refer to predictions, while the horizontal classes refer to the ground truth. Each number represents the percentage of predictions attributed to a class in relation to their true class. The percentages between parenthesis in each matrix label refer to the average accuracy of the method.

From the confusion matrix of KTH dataset, we can observe that RTW+GDA and RTW+eGDA achieved competitive results, where RTW+eGDA substantially outperforms RTW+GDA in boxing, clapping and walking classes and achieved similar results in the remaining classes. In overall, RTW+eGDA outperforms RTW+GDA, justifying the applications of the proposed method.

The results obtained by RTW+eGDA for the classes running and jogging did not outperform RTW+GDA. This may be due to the limitations of representation based on linear subspaces. Specifically, the cause could be that the overlap rate between the two classes' distributions is very large, as linear subspaces are insufficient to represent the complicated boundary between these classes. In this particular case, the principal subspace may contain a substantial amount of data from both classes and when it is removed from the sum subspace, most of the representative information from both classes is also removed, leading to a degradation of both subspace classes, where its projections no longer can represent its semantics. We expect that the introduction of nonlinear kernels composited with Grassmann kernels would largely suppress misclassification due to this severe model overlap.



(a) RTW+GDA (average 90.83 %)

(b) RTW+GDA (average 67.56 %)

(c) RTW+eGDA (average 95.14 %)

(d) RTW+eGDA (average 69.17 %)

Figure 3.9: Confusion matrices of the (a) RTW+GDA in the Cambridge hand gesture experiment and (b) RTW+GDA in the KTH action experiment; (c) RTW+eGDA in the Cambridge hand gesture experiment and (d) RTW+eGDA in the KTH action experiment. The percentage in parentheses refers to the average accuracy of each method.

To elucidate on the effect of the introduced parameters on the proposed method's performance, we have performed an additional evaluation. Figure 3.10 shows a bar plot of accuracy of the proposed RTW+eGDA, when fixing the dimension of principal subspace d_p to 5, and varying both the dimension of hypo subspaces m and dimension of class subspaces d_m. In this dataset, 11 dimensional models have performed better, and small dimension m tends to induce a performance degradation due to a poor model representation. According to Figure 3.10, the class subspace dimension d_m seems to depend on the dimension of hypo subspaces m. However, when $m = 11$, d_m is more invariant to small changes, so that no much emphasis needs to be put in searching an optimal value. Regarding dimension of the principal subspace, small values are usually best, to avoid losing substantial information that may be contained on the removed principal components.

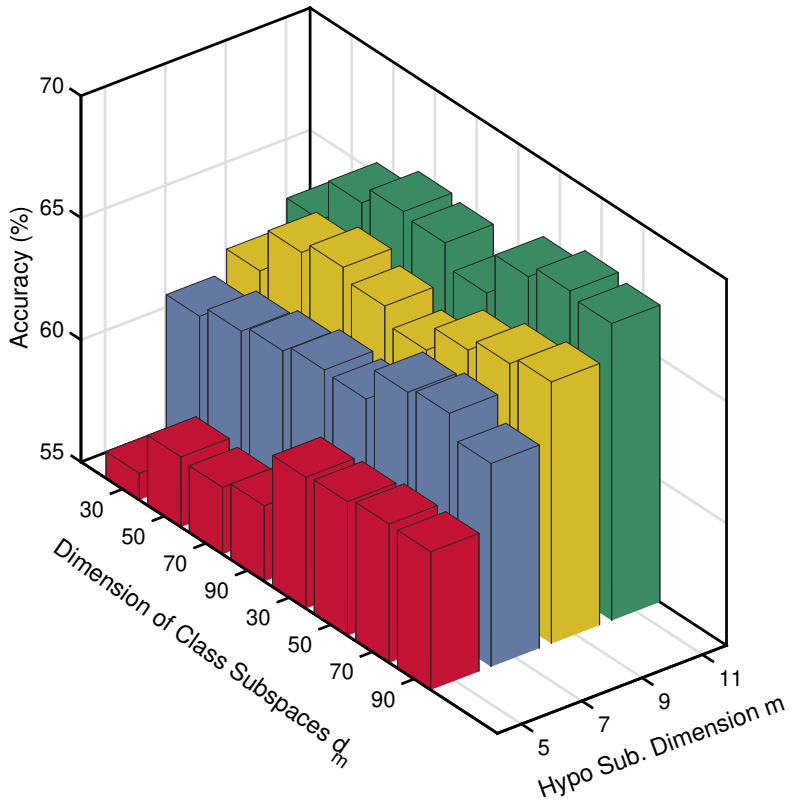


Figure 3.10: Parameter behavior of RTW+eGDA on the KTH action dataset.

3.4.3 Experiment with UCF sports dataset

We also conducted an experiment using the UCF sports dataset [91, 102]. The purpose of this experiment is twofold: to shed light on RTW+eGDA's potential with more challenging data; and to anticipate its behavior when using more sophisticated features. This experiment contrasts with the previous ones, which focused on assessing RTW+eGDA under the simplest scenario, by using raw images, elucidating its usefulness as a simple and practical means for further enhancing RTW+GDA.

The UCF sports dataset contains a total of 150 sequences of subjects performing sports, with 10 classes, namely: diving, golf swing, kicking, lifting, riding horse, running, skateboarding, swing-bench, swing-side, and walking. The number of frames of each video ranges from 50 to 70. The action bounding box has been extracted, using annotations provided. Then, each cropped image was resized to a 38×24 pixels grayscale image, resulting in a 912-dimensional vector for each frame.

We quantitatively compared the methods by a leave-one-out cross-validation scheme (LOOCV), a standard experiment setting for this data. That means 150 repetitions of learning, with one video as query and the remaining 149 videos as reference data.

To anticipate RTW+eGDA's potential when combined with feature extractors, we utilized pre-

processing of each video frame by two features: a histogram of gradients (HOG), the combinations with which are named as HOG+RTW+GDA and HOG+RTW+eGDA; and convolutional neural network (CNN) features extracted from the last fully-connected layer of the AlexNet [70], the combinations with which we refer to as AlexNet+RTW+GDA/eGDA. The AlexNet was pre-trained on more than a million images from the ImageNet database [20], and has not been fine-tuned or equipped with mechanisms to represent the time components of the video data. We compared the above frameworks with RTW+GDA and RTW+eGDA.

In addition, we compare RTW+eGDA to a number of conventional methods that are relevant to the approach taken in the proposed method: 1) Motion extraction methods: robust non-linear knowledge transfer model (R-NKTM) and dense trajectory based method (DT), that have been recently proposed [90]. Namely, we compare 4 variants: trajectory DT, trajectory R-NKTM, HOG+HOF+MBH+Traj. DT and HOG+HOF+MBH+Traj. R-NKTM. Note that these methods are more elaborate in their feature extraction, especially the latter two that use an intricate combination of video descriptors HOG+HOF+MBH+Traj [114]. In contrast, our proposed method here uses only HOG and no feature fusion. 2) Methods for classification of image sets: Grassmann/subspace learning methods include discriminative canonical correlations (DCC) [65], constrained mutual subspace method (CMSM) [28], Grassmannian graph-Embedding discriminant analysis (GGDA) [45] and projection metric learning (PML) [54]. We also compare a covariance-based method named covariance discriminant learning (CDL) [115]. We evaluate the performance of each method utilizing raw images, HOG and AlexNet features.

Regarding RTW parameters, the number of selected frames k to build one TE feature is fixed at $k = 3$, and the number of TE features for each set is fixed to be $Z = 60$. The HOG parameters were set as follows: number of bins is fixed at 9, the cell size at 5, and the block size at 3. The GDA and eGDA parameters were varied in the following manner: dimension of hypo subspaces m was varied from 8 to 14 in increments of 2; dimension of class subspaces d_m was varied either 40 or 50; and dimension of principal subspace d_p was varied from 1 to 8.

The results can be seen in Table 3.2. Both GDA and eGDA perform better with HOG and CNN features, indicating that using more discriminative features instead of raw images can improve them. It also can be noted that the performance gap between RTW+eGDA and RTW+GDA increased when using HOG. When compared to the conventional methods in this experiment, our method is competitive, with HOG+RTW+eGDA and AlexNet+RTW+eGDA achieving better results than methods using trajectory features and the baseline subspace-based based methods. These results demonstrate a potential for extensions and broad utility of the proposed method independent on the type of feature. In addition, our method would benefit from pre-trained deep neural networks, such as DenseNet and ResNet50. This potential is corroborated by the results shown by [100], which indicate that a subspace of deep features can be useful to represent image sets. Although the two methods based on a complicated combination of four types of features (HOG, HOF, MBH and trajectories) over-performed the proposed method, this result was expected to some extent, since these methods combine various types of motion analysis features intricately to obtain high performances.

RTW+eGDA overperforms CDL when using HOG features while the reverse happens when using AlexNet features. To verify if either method is overperforming the other meaningfully, we conducted a paired two-sample t-test between the results of RTW+eGDA and CDL with significance level 0.05 . From the test results ($p = 0.6187$), we cannot conclude with more than 95% confidence that both methods perform statistically significantly differently in this experiment. The reason for

their comparable level may be that, as said in the CDL original paper, the covariance matrix is able to capture ordering in a set of patterns, which may work as a mechanism for representing the time structure like RTW. It also utilizes an LDA based classifier to predict categories. In the following we would like to discuss their differences further.

To demonstrate a situation where our proposed method should be considered as a good choice, we performed another experiment comparing the two best performing image-set based methods CDL and RTW+eGDA. Utilizing the AlexNet deep features we evaluate both frameworks in a situation of small sample size (SSS). Concretely, we have purposefully limited the number of frames available in an image sequence by a percentage of the total number of frames, from 20% to 50%. That is, for 20%, in a video with 100 frames, 20 frames would be used, selected by keeping one frame, skipping the next few frames, and then repeating that process until the sequence ends. The results can be seen in Table 3.3. As shown, the performance of CDL falls significantly as fewer frames are available. The reason is most likely that the rank of the covariance matrix tends to decrease causing instabilities when measuring distances, while subspaces in that context may offer a more robust model. Therefore, RTW+eGDA may be the method of choice in systems where the incoming stream of image data may have the drop of capture speed, or simply data is corrupted or few frames are available.

Another aspect of RTW+eGDA and CDL that is worth discussing is the necessary memory requirements to run the methods. RTW+eGDA requires memory of $dk \times m$ elements to store a subspace corresponding to one motion sequence. CDL utilizes $d \times d$ covariance matrices, which can be orders of magnitude higher than the memory required by the proposed method. For example, in the current experiment with HOG features, $dkm = 1458 \times 3 \times 8 = 34992$, while $d^2 = 1458^2 = 2125764$, meaning that RTW+eGDA uses only 1.6% of the memory used by CDL. Even if one exploits the symmetry of the matrices and reconstructs them from their upper/lower triangles at each time they are accessed in memory, one would still need $d(d+1)/2 = 1063611$ elements. Still RTW+eGDA uses only 3.3% of the memory used by CDL in this case, without the need to reconstruct the representation for storage. Therefore, RTW+eGDA can be one effective choice in mobile phones, car navigation, and embedded systems where memory is a limited resource.

3.5 Summary

In this chapter we have proposed a combination of randomized time warping and eGDA, to address more effectively the classification of motion sequences. Our method may be used for various types of applications with continuous sequences, but we focused on the applications of hand gestures and human action classification. The key idea of our enhanced Grassmann manifold is to project class subspaces onto a generalized difference subspace before mapping them onto a RKHS through a Grassmann kernel. The GDS projection can extract the differences between classes and generate data points with optimized between-class separability on the manifold, which are more desirable for GDA. The validity of our enhanced Grassmann discriminant analysis was demonstrated through classification experiments with the Cambridge hand gesture, KTH action, and UCF sports datasets, where it outperformed its GDA counterpart and showed competitiveness with state-of-art methods.

From the experiments we have also demonstrated that the proposed method can be a good choice in applications with small sample problems and those which require lower memory.

Table 3.2: Results of the UCF sports experiment. Results from DT and R-NKTM are reported in [90].

Methods	Accuracy (%)		
trajectory DT [90]	75.20		
trajectory R-NKTM [90]	76.70		
HOG+HOF+MBH+Traj. DT [90]	88.20		
HOG+HOF+MBH+Traj. R-NKTM [90]	88.20		
	Raw	HOG	AlexNet
DCC [65]	59.33	68.97	72.00
CMSM [28]	68.97	65.33	82.67
GGDA [45]	47.33	49.33	57.33
PML [54]	72.67	73.33	76.67
CDL [115]	70.00	76.00	86.00
RTW+GDA [105]	63.33	70.00	80.00
RTW+eGDA (proposed method)	70.00	78.00	84.67

Table 3.3: Performances in UCF sports when a small sample size (SSS) of videos frames is available.

Frames remaining (%)	20%	30%	40%	50%	100%
AlexNet+RTW+eGDA	76.67	78.00	80.67	80.00	84.67
AlexNet+CDL	57.33	67.33	76.67	82.67	86

Chapter 4

Singular spectrum-based methods

This chapter introduces two generalizations of the classical Fisher discriminant analysis to the Grassmannian, called Grassmann singular spectrum analysis (GSSA) and Tangent singular spectrum analysis (TSSA). The background of the proposed method is discussed in Section 4.1. The proposed methods are elaborated in Section 4.2. Experimental results are presented in Section 4.3. Finally, the summary is given in Section 4.4.

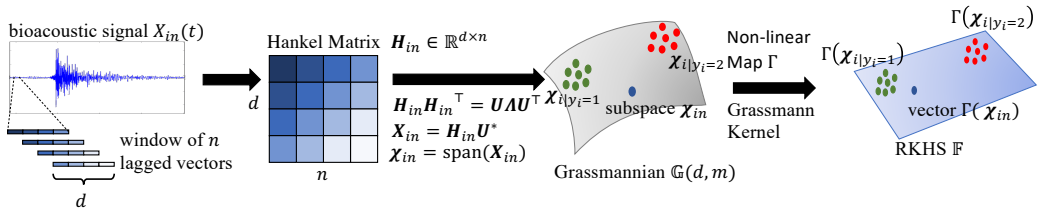


Figure 4.1: Conceptual diagram of the proposed GSSA.

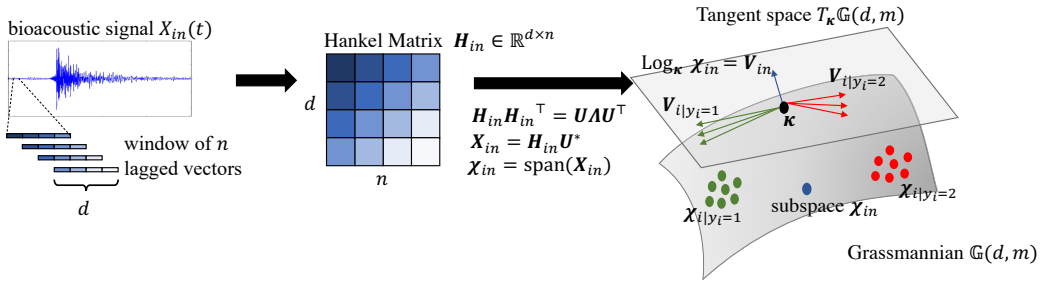


Figure 4.2: Conceptual diagram of the proposed TSSA.

4.1 Background

Bioacoustic signal classification (e.g., animal calls) plays an important role in environmental monitoring as it gives experts the means to efficiently acquire information from areas where it is hard to

access and collect data. It also assists by providing clues about the evolution and categorization of animals from the perspective of similarity of their bioacoustic mechanisms, which may be useful in new species discovery.

Remote monitoring of areas of difficult access, e.g. dense forests or depths of the sea, allows the gathering of critical information in order to predict changes in the population of species and correlated phenomena, e.g. the quality of the sea, air, soil and other natural elements.

An efficient bioacoustic signal classification framework needs to work reliably and within some practical restrictions. Such frameworks need to run on low-cost computational resources, not only to avoid using prohibitively expensive hardware, but to use small-scale devices, deployable on the environment. Likewise, such an algorithm needs to produce reliable predictions from a small dictionary of data, mainly for two reasons: first, gathering and labeling data for an initial dictionary can be expensive and time-consuming for biologists. Second, some species are extremely rare or under risk of extinction, and only a few samples will be available in a practical setting.

Various of the recently proposed bioacoustic signal classification frameworks [7, 18, 17, 120] are comprised of multiples elements, including noise [6] and dimensionality reduction [92], feature selection [94], model training [88] and classification [89].

Motivated by the constraint on computational cost, mutual singular spectrum analysis (MSSA) [31] was introduced as a novel bioacoustic signal representation based on subspaces. Its model is compact and requires no cost-intensive preprocessing techniques common in bioacoustic frameworks such as segmentation, noise reduction, or syllable extraction. The input signals are represented within the singular spectrum analysis (SSA) paradigm, by a set of leftmost eigenvectors extracted based on the signals' accumulated energy. This set forms an orthonormal basis for a subspace in the functional space [27, 37]. Then, canonical angles [50, 3] between the different bioacoustic subspaces are used to measure their similarity, and 1-NN classification is performed. The 1-NN approach to classification of subspaces corresponds to MSM [27, 121].

Although MSSA has efficiently addressed these bioacoustics challenges, its representation has issues that decrease its reliability as a monitoring instrument. One problem is that it has no discriminant mechanism to separate classes (e.g. species of animals). Another is that it assumes a class is composed of linear combinations of the reference signals, which in practice is unlikely, and impairs study of the individuals' signals among the same species.

To tackle these issues, we propose two new algorithms. First, we propose Grassmann singular spectrum analysis (GSSA), which is depicted as a conceptual illustration in the Figure 4.1. We introduce a Grassmann manifold formulation, which simplifies the complicated procedure of the subspace based method using canonical angles. In this framework, a subspace-based method is regarded as a simple classification method on a reproducing kernel Hilbert space (RKHS) of a Grassmann manifold, where each single subspace is treated as a point, and thereby, each bioacoustic signal is represented by a point in the RKHS. Various types of classification methods have been constructed on a Grassmann manifold [41, 111, 103, 104]. In particular, as explained in Chapter 3, GDA has been one of the popular tools for image set classification, being conducted as a kernel discriminant analysis (KDA) through the kernel trick with a Grassmann kernel. In GSSA, we are introducing a straightforward but effective discriminatory mechanism to the bioacoustic signal classification by utilizing GDA.

We also propose an algorithm named Tangent singular spectrum analysis (TSSA). The motivation for this second algorithm is to offer a different discrimination mechanism than that of GSSA. More

specifically, GSSA applies a kernel trick with a Grassmann kernel; it does not perform discriminant analysis on extrinsic matrix variables, but instead it creates a low-dimensional parameterization of the Grassmannian as a RKHS. Although this can work well in many classification tasks, a possible problem with this premise in the setting of bioacoustic signal classification is that we have a constraint on available data as discussed above, and the Grassmann kernel's parameterization is intrinsically dependent on the amount of training data. Using such a kernel with a small data may impair the classification ability, as the kernel may not be able to capture the signals' complexity.

To tackle the problem of bioacoustic signal classification with a constraint on the number of available samples, we formulate TSSA, which is depicted as a conceptual illustration in the Figure 4.2. Instead of relying on a kernel parameterization of the Grassmannian, we utilize the extrinsic matrix structure of the manifold; the subspaces are mapped through the logarithmic map onto a tangent space at the sample mean of the data, called Karcher mean; then discriminant analysis in matrix space is performed.

We summarize our contributions as follows: (1) We propose two bioacoustic classification algorithms assuming that a class may be composed of a set of subspaces, and we consider the bioacoustic signal representation from the perspective of each signal being a point in the Grassmann manifold; (2) We introduce a discriminant algorithm to discriminate signals by the use of GDA; (3) We propose another discriminant bioacoustics classification algorithm called TSSA which performs discriminant analysis on the manifold extrinsic variables. (3) Our methods are extensions of MSSA, and preserve its advantages, namely: less storage requirements than most conventional methods; has a straightforward formulation to select compactness ratio; can handle arbitrary signal lengths; and does not need costly preprocessing techniques, such as segmentation or syllable extraction.

4.2 Proposed methods

In this section, we formulate the classification problem based on the bioacoustic signals. Then, we describe the representation by SSA subspaces, and the algorithms of GSSA and TSSA.

4.2.1 Problem formulation

Let $X(t)$ be a bioacoustic signal. The index $t = 1, \ldots, L$ indicates the signal's ordering in time up to its length L. For a given unknown input bioacoustic signal $X_{\text{in}}(t)$ of length L_{in}, the task is to predict its corresponding bioacoustic class (e.g. species). Consider N_c reference bioacoustic signals for each c-th class ($c = 1, \ldots, C$), adding to a total of N reference signals $\{(X_i(t), y_i)\}$, where $i = 1, \ldots, N$.

This framework assumes that each signal $X(t)$ can be represented by a linear mapping in terms of its autocorrelation. Note that this differs from MSSA in that the assumption has been reduced to the autocorrelation of each single signal, rather than the original assumption of a linear mapping in terms of the average correlation of a whole class of signals. As such, each signal can be represented by the first m orthonormal vectors ordered by the signal's accumulated energy, where $m \ll L_i$. These vectors span a subspace χ_i; thus a class of signals is effectively represented by a set of subspaces $\{\chi_i\}_{i|y_i=c}$, a relaxation similar to a Gaussian mixture rather than a single Gaussian distribution.

This representation provides more flexibility for a class with non-gaussian noise, and also scales

well with an increasing number of signals per class. That implies mainly two points: a signal can be modeled independently and added to the class as another subspace, without needing to recalculate a single model for the whole class; and the trade-off problem of compactness ratio and representativity is slightly alleviated. The dimension of subspaces m, corresponding to the signal's compactness ratio, is selected empirically during the training. For a given unknown input bioacoustic signal $X_{\text{in}}(t)$ of length L_{in}, the task is to compute a subspace χ_{in} and predict its corresponding bioacoustic class (e.g. species) based on the nearest subspace.

4.2.2 Representation by SSA subspaces

Now we review the representation by an SSA subspace; for that, consider a single signal $X(t)$. First, we apply a sliding window over $X(t)$ to turn the 1−dimensional signal of length L in a sequence of n lagged vectors of length d arranged in a Hankel matrix $\boldsymbol{H} \in \mathbb{R}^{d\times n}$, i.e. the vectors form blocks of a matrix: $(\boldsymbol{H})_{i,j} = X(i+j-1)$,

$$\boldsymbol{H} = \begin{bmatrix} X(1) & X(2) & \cdots & X(n) \\ X(2) & X(3) & \cdots & X(n+1) \\ \vdots & & \ddots & \vdots \\ X(d) & X(d+1) & \cdots & X(d+n-1) \end{bmatrix}, \tag{4.1}$$

where the number of columns n is the number of desired oscillatory components (or principal components), which has a direct correspondence with the maximum delay of the autocorrelation. Therefore, the total number of columns in matrix $\boldsymbol{H}$ is given by relationship $n = L - d + 1$. Note that the anti-diagonal elements in $\boldsymbol{H}$ are equal.

The eigenvalue decomposition of the autocorrelation matrix $\boldsymbol{H}^\top\boldsymbol{H}$ is performed to find the directions of maximum variance of the Hankel matrix $\boldsymbol{H}$; that is, for this case:

$$\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top = \boldsymbol{H}^\top\boldsymbol{H}, \tag{4.2}$$

where the columns of $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ corresponds to the eigenvectors of $\boldsymbol{H}^\top\boldsymbol{H}$ and the diagonal of $\boldsymbol{\Lambda}$ corresponds to the singular values λ_j $(j = 1, \ldots, n)$, i.e. $\text{diag}(\boldsymbol{\Lambda}) = \lambda_1, \lambda_2, \ldots, \lambda_n$ in decreasing order. Given the compactness ratio m, the m-leftmost matrix $\boldsymbol{U}^* \in \mathbb{R}^{n\times m}$ from $\boldsymbol{U}$ is selected, containing the first m eigenvectors corresponding to the highest eigenvalues. The Hankel matrix $\boldsymbol{H}$ is projected onto the subspace spanned by $\boldsymbol{U}^*$ to obtain $\boldsymbol{X} = \boldsymbol{H}\boldsymbol{U}^* \in \mathbb{R}^{d\times m}$, which is the best m-rank approximation of $\boldsymbol{H}$ and spans the subspace $\boldsymbol{y}$ that characterizes our signal. In our classification framework, we utilize the previous procedure to compute a basis matrix $\boldsymbol{X}_i \in \mathbb{R}^{d\times m}$ for a SSA subspace χ_i corresponding to each signal $X_i(t)$.

4.2.3 Grassmann singular spectrum analysis

In this subsection, we introduce the algorithm of GSSA. Given the set of training SSA subspaces $\{(\chi_i, y_i)\}$, GSSA proceeds in the same manner as GDA, that is, GSSA is conducted as kernel discriminant analysis with a Grassmann kernel.

We embed the Grassmann manifold $\mathbb{G}(d, m)$ in a RKHS by using the projection kernel:

$$k_p(\boldsymbol{\chi}_1, \boldsymbol{\chi}_2) = \sum_{j=1}^{m} \cos^2 \theta_j. \tag{4.3}$$

Here, θ_j refers to the j-th canonical angle between two subspaces. By using the kernel, a subspace $\boldsymbol{\chi}_i \in \mathbb{G}(d, m)$ is mapped onto a kernel vector $\boldsymbol{k}_i \in \mathbb{R}^N$ as follows:

$$\boldsymbol{k}_i = k_p(\boldsymbol{\chi}_i, \boldsymbol{\chi}_k), \tag{4.4}$$

where $k = 1, \ldots, N$ indexes all N reference signals.

Let $\Gamma : \mathbb{G} \to \mathbb{F}$ be a non-linear map from the Grassmannian $\mathbb{G}$ to a Hilbert space $\mathbb{F}$, and $\boldsymbol{\Gamma} = [\Gamma(\boldsymbol{\chi}_1), \ldots, \Gamma(\boldsymbol{\chi}_N)]$ be the feature matrix of the mapped training points. Assuming a discriminant direction $\boldsymbol{w} \in \mathbb{F}$ is a linear combination of those feature vectors, $\boldsymbol{w} = \boldsymbol{\Gamma}\boldsymbol{u}$, we use the kernel trick to optimize the following discriminant problem:

$$\max_{\boldsymbol{u}} J(\boldsymbol{u}; \boldsymbol{K}) = \max_{\boldsymbol{u}} \frac{\boldsymbol{u}^\top \boldsymbol{\Gamma}^\top \boldsymbol{S}_b \boldsymbol{\Gamma} \boldsymbol{u}}{\boldsymbol{u}^\top \boldsymbol{\Gamma}^\top \boldsymbol{S}_w \boldsymbol{\Gamma} \boldsymbol{u}} = \max_{\boldsymbol{u}} \frac{\boldsymbol{u}^\top \boldsymbol{K}(\boldsymbol{V} - \boldsymbol{e}_N \boldsymbol{e}_N^\top / N)\boldsymbol{K}\boldsymbol{u}}{\boldsymbol{u}^\top (\boldsymbol{K}(\boldsymbol{I}_N - \boldsymbol{V})\boldsymbol{K} + \sigma^2 \boldsymbol{I}_N)\boldsymbol{u}} = \max_{\boldsymbol{u}} \frac{\boldsymbol{u}^\top \boldsymbol{\Sigma}_b \boldsymbol{u}}{\boldsymbol{u}^\top (\boldsymbol{\Sigma}_w + \sigma^2 \boldsymbol{I}_N)\boldsymbol{u}}, \tag{4.5}$$

where $\boldsymbol{S}_b$ is a between-class scatter matrix, $\boldsymbol{S}_w$ is a within-class scatter matrix, $\boldsymbol{K}$ is the kernel matrix, $\boldsymbol{e}_N$ is a vector of ones that has length N, $\boldsymbol{V}$ is a block-diagonal matrix whose c-th block is the matrix $\boldsymbol{e}_{N_c} \boldsymbol{e}_{N_c}^\top / N_c$, and $\boldsymbol{\Sigma}_b = \boldsymbol{K}(\boldsymbol{V} - \boldsymbol{e}_N \boldsymbol{e}_N^\top / N)\boldsymbol{K}$.

In our framework, the kernel matrix, $\boldsymbol{K}$, is calculated as the similarity matrix between training subspaces, where each element can be written in terms of the projection kernel as $k_p(\boldsymbol{\chi}_q, \boldsymbol{\chi}_w)$, where q is a row and w is a column of $\boldsymbol{K}$. The term $\sigma^2 \boldsymbol{I}_N$ is used for regularizing the covariance matrix $\boldsymbol{\Sigma}_w = \boldsymbol{K}(\boldsymbol{I}_N - \boldsymbol{V})\boldsymbol{K}$. It is composed of the covariance shrinkage factor $\sigma^2 > 0$, and the identity matrix $\boldsymbol{I}_N$ of size N. The set of optimal vectors $\{\boldsymbol{u}^*\}$ are obtained as the first $C - 1$ eigenvectors of $(\boldsymbol{\Sigma}_w + \sigma^2 \boldsymbol{I}_N)^{-1} \boldsymbol{\Sigma}_b$. We apply this algorithm to map all reference subspaces, keeping in memory each $\boldsymbol{k}_i$ corresponding to a $\Gamma(\boldsymbol{\chi}_i)$.

After the learning phase, when a novel signal $X_{\text{in}}(t)$ is input, it is first represented by a SSA subspace $\boldsymbol{\chi}_{\text{in}}$, then mapped onto a kernel vector $\boldsymbol{k}_{\text{in}}$ and finally projected onto the discriminant space spanned by $\{\boldsymbol{u}^*\}$. Classification is performed by an 1-NN approach with the reference kernel vectors $\boldsymbol{k}_i$.

4.2.4 Tangent singular spectrum analysis

In this subsection we introduce the algorithm of TSSA. In TSSA, the algorithm of discriminant analysis (LDA) [30] is realized in a tangent space of the Grassmann manifold, using extrinsic matrix coordinates, as follows.

Using the training data $\{(\boldsymbol{\chi}_i, y_i)\}$, we calculate the Karcher mean $\boldsymbol{\kappa}$ of all subspaces. The Karcher mean is computed using Algorithm 1 (Chapter 2). Let N_c be the number of training signals in class c, and let the sample mean of class c be:

$$\bar{\boldsymbol{X}}_c = \frac{1}{N_c} \sum_{i \,|\, y_i = c} \operatorname{Log}_{\boldsymbol{\kappa}} \boldsymbol{\chi}_i. \tag{4.6}$$

Note that the overall mean can be written as:

$$\bar{\boldsymbol{X}} = \frac{1}{N}\sum_{i=1}^{N} \operatorname{Log}_{\boldsymbol{\kappa}} \boldsymbol{\chi}_i = \operatorname{Log}_{\boldsymbol{\kappa}} \boldsymbol{\kappa} = \mathbf{0}. \tag{4.7}$$

In the vector discriminant analysis one seeks the discriminant direction $\boldsymbol{W}$ which maximizes the Rayleigh quotient $R(\boldsymbol{w}) = \boldsymbol{w}^\top \boldsymbol{B}\boldsymbol{w}/\boldsymbol{w}^\top \boldsymbol{S}\boldsymbol{w}$ where $\boldsymbol{B}$ and $\boldsymbol{S}$ are the between-class and within-class covariance matrices respectively. This vector-valued function can be rewritten for matrices as:

$$R(\boldsymbol{W}) = \frac{\operatorname{tr} \boldsymbol{W}^\top P_{\boldsymbol{B}}(\boldsymbol{W})}{\operatorname{tr} \boldsymbol{W}^\top P_{\boldsymbol{S}}(\boldsymbol{W})}, \tag{4.8}$$

where $\boldsymbol{B}, \boldsymbol{S} \in \mathbb{R}^{dm \times dm}$ are the between-class and within-class covariance matrices.

First, the class covariance matrices are defined for each class c as:

$$\boldsymbol{B}_c = N_c(\operatorname{vec} \bar{\boldsymbol{X}}_c \otimes \operatorname{vec} \bar{\boldsymbol{X}}_c^\top), \tag{4.9}$$

$$\boldsymbol{S}_c = \sum_{i|y_i=c} \operatorname{vec}(\operatorname{Log}_{\boldsymbol{\kappa}} \boldsymbol{\chi}_i - \bar{\boldsymbol{X}}_c) \otimes \operatorname{vec}(\operatorname{Log}_{\boldsymbol{\kappa}} \boldsymbol{\chi}_i - \bar{\boldsymbol{X}}_c)^\top. \tag{4.10}$$

Then, the regularized class covariance matrices are given by:

$$\tilde{\boldsymbol{B}}_c(\delta) = (1-\delta)\boldsymbol{B}_c + \delta \sum_{c=1}^{C} \boldsymbol{B}_c \tag{4.11}$$

$$\tilde{\boldsymbol{S}}_c(\delta) = (1-\delta)\boldsymbol{S}_c + \delta \sum_{c=1}^{C} \boldsymbol{S}_c, \tag{4.12}$$

$$\tilde{\boldsymbol{S}}_c(\delta, \gamma) = (1-\gamma)\tilde{\boldsymbol{S}}_c(\delta) + \gamma \operatorname{tr}(\tilde{\boldsymbol{S}}_c(\delta))\boldsymbol{I}, \tag{4.13}$$

where $\boldsymbol{I}$ is the identity matrix, the parameter δ controls the degree of shrinkage of the individual class covariance matrix estimates towards the pooled estimate, and γ controls shrinkage towards a multiple of the identity matrix, in order to alleviate the problem of matrix degeneration when taking the inverse. Both parameters can be set in a range between 0 and 1. Finally, we have the covariance matrices defined as.

$$\boldsymbol{B} = \frac{1}{N}\sum_{c=1}^{C} \tilde{\boldsymbol{B}}_c(\delta), \tag{4.14}$$

$$\boldsymbol{S} = \frac{1}{N}\sum_{c=1}^{C} \tilde{\boldsymbol{S}}_c(\delta, \gamma). \tag{4.15}$$

The optimal $\boldsymbol{W}$ is obtained from reshaping the largest eigenvectors of $\boldsymbol{S}^{-1}\boldsymbol{B}$ back into $d \times m$ matrices. Since $\boldsymbol{S}^{-1}\boldsymbol{B}$ has rank $C-1$, there are $C-1$ optima $\boldsymbol{W}_1, \ldots, \boldsymbol{W}_{C-1}$. Data can be projected onto the discriminant space spanned by $\{\boldsymbol{W}_j\}$ by $P_{\boldsymbol{W}}(\operatorname{Log}_{\boldsymbol{\kappa}} \boldsymbol{\chi}_i)$. This corresponds to feature extraction of data onto the most discriminant subspace.

We apply this projection operation to the reference subspace matrices $\boldsymbol{X}_i$ to generate reference vectors $\boldsymbol{V}_i = P_{\boldsymbol{W}}(\operatorname{Log}_{\boldsymbol{\kappa}} \boldsymbol{\chi}_i)$. When given an unknown bioacoustic signal $X_{in}(t)$, we compute its SSA subspace basis $\boldsymbol{X}_{in}$ and map it onto the discriminant tangent space to generate a vector $\boldsymbol{V}_{in}$; then we predict its corresponding bioacoustic class (e.g. species) based on the nearest reference vector (1-NN) using the canonical metric $\operatorname{tr} \boldsymbol{V}_{in}^\top \boldsymbol{V}_i$.

Table 4.1: Anuran records dataset. The columns min and max stand for the minimum and maximum time duration in seconds of each record in the set. This parameter gives an idea on how different the length of the signals can be.

			Time length (s)	
ID	Species	n^o Records	min	max
(a)	*Adenomera a.*	8	12	360
(b)	*Adenomera h.*	11	40	187
(c)	*Ameerega t.*	5	11	52
(d)	*Hyla m.*	11	8	56
(e)	*Hypsiboas cin.*	4	6	238
(f)	*Hypsiboas cor.*	4	67	346
(g)	*Leptodactylus f.*	4	13	118
(h)	*Osteocephalus o.*	3	3	72
(i)	*Rhinella g.*	5	13	210
(j)	*Scinax ruber*	5	7	62
	Summary	60	$94 \pm 113.26(s)$	

4.3 Experimental results

In this section, we evaluate the validity of GSSA and TSSA through an experiment using the Anuran records dataset, that was also used in [36]. This dataset is described in detail in Table 4.1. It consists of 60 bioacoustic signals with different duration recorded in Amazon rainforest containing anuran's croaks and ribbits and various real background noises from the surrounding nature. Anura is the name of an order of animals in the class Amphibia that includes frogs and toads, so the diversity of signals is complex. The classes of this dataset consists of 10 species of anurans.

We evaluate GSSA and TSSA against 2 variants of MSSA. The variant MSSA-I refers to the conventional method, where the class of signals is assumed to be modeled as a linear mapping in terms of the average correlation of its signals. That means that 1 reference subspace is computed to represent each species, regardless of the amount of records the class contains. The variant MSSA-II computes 1 subspace for each bioacoustic signal. for example, the specie Rhinella g. has 5 records, then we produce 5 subspaces. In this scheme we achieve 60 subspaces. However, MSSA-II does not have a discriminant mechanism.

We performed a 10-fold cross-validation by dividing the signals randomly, with 30 for training and 30 for test, and always ensuring that at least one signal of each anuran species is present in both groups. The parameters were varied in the following manner: the number of lagged vectors n was varied from 10 to 50 and the dimension of SSA subspaces were varied from 3 to 9.

The experimental results can be seen in Table 4.2. The accuracy refers to the average among all folds, and the standard deviation is calculated from the folds assuming a Gaussian distribution. First, we can see that MSSA-II outperforms MSSA-I by approximately 3%, which shows that relaxing the linearity assumption is effective to improve performance. But the main point in this experiment is that although MSSA-I has recently been shown to be the state-of-art in bioacoustics classification [36], we can note that GSSA outperforms MSSA-I in average by 11.5%, while TSSA outperforms MSSA-I

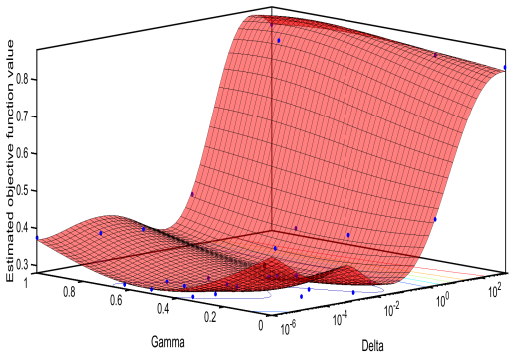


Figure 4.3: Plot of error as a function of the shrinkage parameters.

in average by 15.83%. This result is a compelling evidence that the discriminant mechanisms of the proposed methods are able to enhance the performance of bioacoustic classification in this subspace representation paradigm. We can also see that TSSA overperformed GSSA in this experiment. The reason is perhaps that the GSSA was limited by its Grassmann kernel assumption that the discriminant space lies within a linear combination of data points in the RKHS. Since the amount of training data in this experiment is limited to 30 samples, the parameterization space can be regarded as small to contain the complicated distribution of signals, generating overlap. Meanwhile, TSSA's discriminant analysis operates on the Grassmann manifold Karcher mean parameterization, for which dimension is independent of the amount of data, depending only on the hyperparameters compactness ratio and window size.

We have also investigated the variation of performance of TSSA due to changes on the shrinkage parameters γ and δ. The subspace dimension was fixed at $m = 4$ and the length of the lagged vectors was fixed at $d = 34$, both of which showed good results at the previous experiment. The plot on Figure 4.3 shows the error rate variation when these parameter changed. The blue dots indicate observed results, while the surface is a graphic completion for visualization. From the plot, we can see that in general a low value of δ yields better results, which is closer to a quadratic discriminant model. In addition γ should not be too high or too low, and results around 0.4 yielded the best results.

Table 4.2: Results for the experiment with the Anuran data. The accuracy refers to the average among all folds.

Methods	Average (%)	Std. Dev.
MSSA-I	60.17	4.11
MSSA-II	62.88	4.08
GSSA	71.67	3.60
TSSA	76.00	4.39

4.4 Summary

In this chapter we have proposed Grassmann singular spectrum analysis (GSSA) and tangent singular spectrum analysis (TSSA), extensions of MSSA, to address more effectively the classification of

bioacoustic signals.

The key idea of our proposed methods is 1) relaxing the assumption that a class is composed of linear combinations of the reference signals to that it consists of a set of subspaces and 2) to add a discriminant mechanism to MSSA. GSSA performs discrimination by first mapping the subspaces onto a RKHS, to then introduce a discriminatory mechanism for class separation by using Grassmann discriminant analysis (GDA).

On the other hand, TSSA utilizes the extrinsic matrix structure of the manifold instead of relying on a kernel parameterization of the Grassmannian. The subspaces are mapped through the logarithmic map onto a tangent space at the sample mean of the data. Then discriminant analysis in matrix space is performed.

Our methods also inherit various advantages of MSSA, such as low storage, consistent compactness ratio selection, signal length free formulation, and no costly preprocessing techniques. The validity of both methods was demonstrated through a classification experiment with the Anuran data where it outperformed the state-of-the-art method MSSA.

Chapter 5

Grassmann log model

This chapter discusses a method that learns representations for subspaces in an end-to-end manner, named *Grassmann log model*. The background of the proposed method is discussed in Section 5.1. The proposed method is elaborated in Section 5.2. Experimental results are presented in Section 5.3. Finally, the summary is given in Section 5.4.

5.1 Background

As discussed in Chapter 1, the Grassmann manifold represents the set of subspaces of a vector space, and as such, is a significant foundation for various types of machine learning tools using subspace representation. It has been well known as a practical and robust representation, especially for image set recognition. Despite its usefulness, most standard machine learning methods cannot be promptly utilized on the Grassmann manifold, since they are constructed on Euclidean space. To fill this serious gap and exploit both the compact representation of Grassmann manifold and the handiness of Euclidean space, we propose a method named Grassmann log model to connect those two representations. The proposed method's objective is to map data from a Grassmann manifold to a vector space while maximizing discrimination capability for classification.

The key idea of our method is to formulate the manifold logarithmic map (log) as an end-to-end learnable model working as an interface between the Grassmann manifold and Euclidean space. It can be seamlessly followed by discriminative tools defined on Euclidean space, to provide a discriminative representation for subspaces so that Euclidean methods can perform classification well. Also, the proposed model can be learned in an end-to-end manner, being embedded as a single module in larger network systems.

The motivation of proposing this method is three-fold: 1) a subspace is a robust representation and has become a central research topic in computer vision, being applied to numerous problems such as image set recognition [14, 107, 55, 104, 27, 110], fine-grained classification [119] and action recognition [101, 106, 77]. 2) The image set recognition, where the goal is to model a set of input images and classify it, has been shown to provide significant recognition stability, but the set representation in the deep learning framework remains largely unexplored. On the other hand, 3) there is an abundance of well-established network layers and other end-to-end processing, which work in Euclidean space, e.g. fully-connected, batch normalization dropout layers, activation

functions. We would like to connect the useful subspace representation to these Euclidean tools.

As seen in previous chapters, Grassmann discriminant analysis (GDA) is one of the most popular methods that works by mapping manifold data to a vector space. Concretely, GDA employs a Grassmann kernel to map subspaces onto vectors in a reproducing kernel Hilbert space (RKHS), where then kernel discriminant analysis is performed. However, GDA is limited as it uses vectors in RKHS defined by the kernel function, without learning a manifold-aware discriminant mechanism. Additionally, the Grassmann kernel dimension is directly dependent on the number of samples in the dictionary, which could lead to an insufficiently small space to represent data.

Considering end-to-end methods, one could develop a simple method of concatenating a subspace basis vectors and then using Euclidean layers such as a fully-connected layer directly. However, such an approach cannot learn stably, since this maneuver would break the inherent subspace structure, cancelling the interesting subspace properties. Therefore, an interface is necessary to learn the Euclidean layers properly.

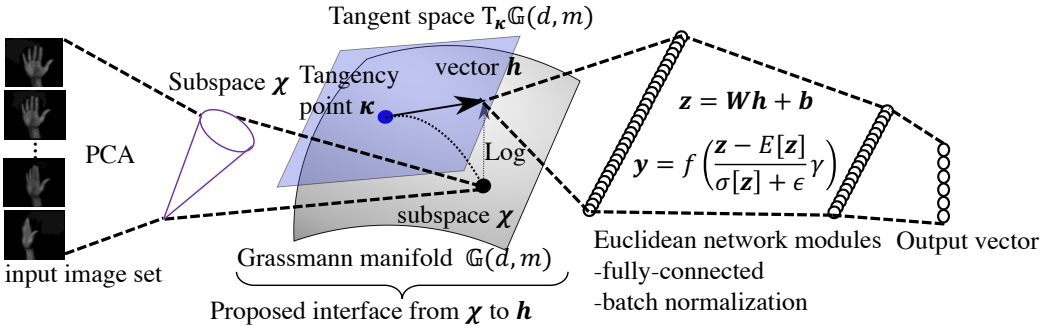


Figure 5.1: Conceptual diagram of the proposed Grassmann log model. A subspace $\boldsymbol{\chi}$ is computed by PCA, represented by an orthogonal basis matrix. Our proposed interface is to log map $\boldsymbol{\chi}$ into a tangent vector $\boldsymbol{h}$; then Euclidean network modules are applied. The first equation indicates a fully-connected layer, where $\boldsymbol{W}$ and $\boldsymbol{b}$ denote the weights and bias, respectively. The second equation indicates a batch normalization where E and σ denote expectation and variance, ϵ, γ are batch normalization hyperparameters, and f is a non-linear activation function.

Our proposed Grassmann log model then contains two main stages, which can be seen in Figure 5.1: 1) a mapping from manifold to vector space, and then 2) Euclidean network modules such as fully connected layers. More concretely, given a manifold data point $\boldsymbol{\chi}$ as input data, the log maps a manifold data point into a tangent vector $\boldsymbol{h}$ in a tangent space parameterized by a tangency point $\boldsymbol{\kappa}$. Then $\boldsymbol{h}$ is transformed through Euclidean layers and the cross-entropy loss function is applied. To obtain a discriminant interface between manifold and vector data, we learn both the tangent space representation and the discriminant Euclidean layers in an end-to-end manner. The log model is a general framework that can be learned with Riemannian stochastic gradient descent [10] on any Riemannian manifold with a defined closed-form log map. In the Grassmannian case, the parameter $\boldsymbol{\kappa}$ is learned as a point constrained to the Grassmann manifold, and the Euclidean layers are learned in conventional Euclidean space. In the following, we refer to tangency point as "anchor point" with emphasis on this point.

Most classification problems have a complicated distribution with a wide variance where a single

tangent vector may yield a suboptimal representation. Therefore, we extend the log model to learn more than one anchor point, obtaining a set of tangent vectors that are concatenated to output a single feature vector. From a differential geometric viewpoint, this idea can be interpreted as a wider atlas of tangent spaces, covering a broader neighborhood of the manifold and diminishing distortion. From the perspective of computer vision, this idea is similar to that of popular image descriptors that combine multiple residual vectors, such as Fisher vectors [87, 86], VLAD [8], and super-vector coding [125], and the log model can be seen as a kind of extension of these ideas to represent an image set rather than just a single image.

Through experiments, we demonstrate that learning a tangent space is important for finding a discriminative map, by making a comprehensive comparison of a learned log model against a fixed tangent space at the manifold data's Karcher mean. We demonstrate the flexibility and scalability of our Grassmann log model as an interface between deep network layers involving subspace data, by also evaluating the log model as a middle stage within larger networks containing convolution layers, and Resnet blocks along with PCA. We show the effectiveness of our model in artificial subspace data and in real data for the applications of face identification, facial expression and hand shape recognition.

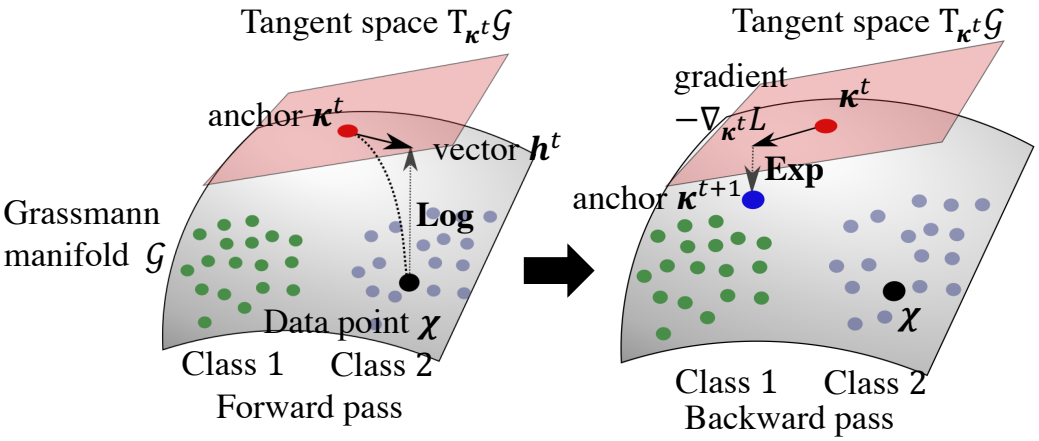


Figure 5.2: Conceptual diagram of the log model learning a tangent space for a binary class problem. The log is used to map point χ to vector $\boldsymbol{h}^t$ in the tangent space (in red). Then, a loss function can be applied and the gradient with respect to the anchor point (tangency point) $\boldsymbol{\kappa}^t$ can be used to move towards a more optimal position $\boldsymbol{\kappa}^{t+1}$ which defines a new tangent space (in blue).

5.2 Proposed Grassmann log model

In this section, we describe the algorithm of the proposed Grassmann Log model. First, we define the Log layer more generally and then introduce its numerical algorithm. Next, we extend it to work with multiple tangent spaces.

5.2.1 Basic idea

Our learning problem is defined as follows: given training subspaces $\{\chi_i\}_{i=1}^N \in \mathbb{G}$ paired with respective labels y_i, we want to learn a model that maps them to vectors $\boldsymbol{h}_i$ in Euclidean space, so that the class distributions are separable. In the following discussion, we consider a minibatch $\{\chi_i\}_{i=1}^n$, as is common in neural networks, but the whole process can be repeated iteratively to use all N training subspaces. When convenient, we omit the i index, and explain the process for a single instance i.e. χ.

Our key idea is to cast the log map $\text{Log}_{\boldsymbol{\kappa}}\, \chi$ as a learnable model by 1) defining the anchor point $\boldsymbol{\kappa}$ as a parameter and χ as input, and 2) seeking an anchor point $\boldsymbol{\kappa}$ such that a tangent vector $\text{Log}_{\boldsymbol{\kappa}}\, \chi$ of the manifold data point χ can be as discriminant as possible in the tangent space.

5.2.2 Learning the log layer

The learning process of the log module is given as follows. Figure 5.2 shows the conceptual diagram of learning the tangent space with the anchor $\boldsymbol{\kappa}^t$ on a binary classification, where t is indexing the update iteration. The whole learning process consists of forward and backward passes. At the iteration $t = 0$, we initialize our anchor parameter as a random manifold point.

Forward pass

The forward pass is processed as follows: we map a point χ into a vector $\boldsymbol{h}^t$ by:

$$\boldsymbol{h}^t = \text{vec}(\text{Log}_{\boldsymbol{\kappa}^t}\, \chi). \tag{5.1}$$

As the Grassmann manifold is a matrix manifold, we utilize the vectorization of matrices vec to turn the tangent vectors from matrix to simple vectors, such that $\boldsymbol{h}^t \in \mathbb{R}^{dm}$. This does not affect the distance structure. The equation above corresponds to the log map projecting a point χ on the manifold to vector $\boldsymbol{h}^t$ in the red tangent space as shown in the first figure in Fig.5.2. After the proposed log layer, Euclidean modules and a discriminative loss function, here abstracted as a loss function $L(\boldsymbol{h}_i, y_i) : \mathbb{R}^{dm} \rightarrow \mathbb{R}$ can be applied to the tangent vectors.

Backward pass

Then, the backward pass is processed as follows: the Euclidean gradient $\nabla_{\boldsymbol{h}^t} L$ with respect to the tangent vectors is computed, and then the log layer gradient is computed through the backpropagation algorithm. We write the chain gradient including the log as:

$$\nabla_{\boldsymbol{\kappa}^t} L(\chi_i, y_i, \boldsymbol{\kappa}^t) = \nabla_{\boldsymbol{h}^t} L \frac{d}{d\boldsymbol{\kappa}^t}(\text{Log}_{\boldsymbol{\kappa}^t}\, \chi_i), \tag{5.2}$$

where $\frac{d}{d\boldsymbol{\kappa}^t}(\text{Log}_{\boldsymbol{\kappa}^t}\, \chi_i)$ represents the derivative of the log map.

Given the gradient $\nabla_{\boldsymbol{\kappa}^t} L$, we perform the update of the anchor $\boldsymbol{\kappa}^t$ by Riemannian stochastic gradient descent (RSGD) [11, 10]. This manifold aware update enforces the updated point $\boldsymbol{\kappa}^{t+1}$ to be a member of the Grassmann manifold, i.e., avoiding the parameter to leave the manifold. The RSGD update consists of two steps: 1) transforming the Euclidean gradient $\nabla_{\boldsymbol{\kappa}^t} L$ to a Riemannian

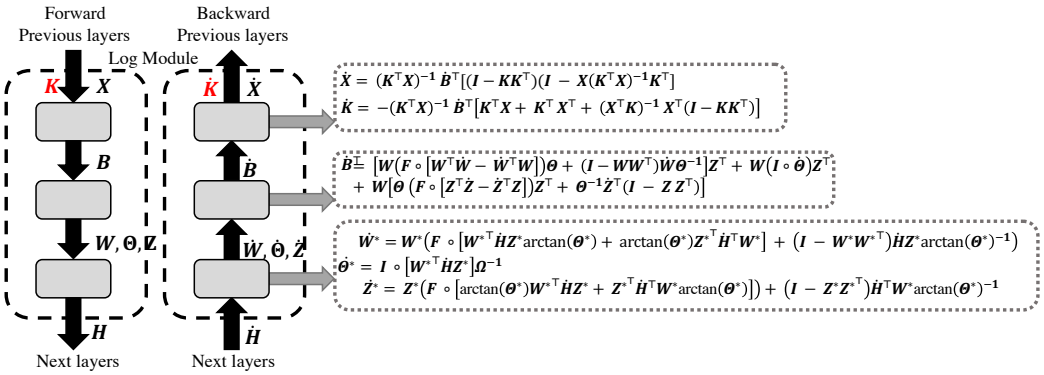


Figure 5.3: Update equations of the Grassmann log module backward phase. The objective is to compute the gradient $\dot{\boldsymbol{K}}$ of the anchor parameter $\boldsymbol{K}$ (in red) and the gradient $\dot{\boldsymbol{X}}$ of input subspace $\boldsymbol{X}$. $\dot{\boldsymbol{K}}$ is used to update the anchor according to the RSGD update, and $\dot{\boldsymbol{X}}$ is used if there are layers previous to the log layer, to compute their respective backward steps.

gradient $\text{grad}_{\boldsymbol{\kappa}^t} L$, that is, the closest vector to $\nabla_{\boldsymbol{\kappa}^t} L$ that is also tangent to the manifold at $\boldsymbol{\kappa}^t$. Then 2) updating the anchor point by:

$$\boldsymbol{\kappa}^{t+1} = \text{Exp}_{\boldsymbol{\kappa}^t} -\lambda \nabla_{\boldsymbol{\kappa}^t} L. \tag{5.3}$$

Here, λ is a learning rate.

The above update equation is illustrated in the second part of Figure 5.2. The exponential map can be seen as a line walk in the opposite direction of the Riemannian gradient (direction of descent), landing on a more optimal point $\boldsymbol{\kappa}^{t+1}$ (in blue). This new anchor point defines a new tangent space, which should be more optimal in the sense that the corresponding log map yields tangent vectors with higher class separability as shown in the last figure. This iterative process is repeated until either a maximum number of iterations is achieved, or the gradient becomes too small, that is, the separability cannot be improved much further.

5.2.3 Numerical algorithm

In the previous section, we have established the framework in general while abstracting the computations. In this section, we describe the equations used in this framework to compute the exp and log maps in matrix form. Then we derive the log backward updates.

To compute the Grassmann exponential map, we utilize the following extrinsic function derived by [1], written in terms of orthonormal matrix representation. Recall our anchor parameter $\boldsymbol{\kappa}$ is a subspace, so given its basis matrix $\boldsymbol{K} \in \mathbb{R}^{d \times m}$, and given a tangent vector $\boldsymbol{H} \in \mathbb{R}^{d \times m}$:

$$\text{Exp}_{\boldsymbol{K}}\, \lambda \boldsymbol{H} = \text{orth}(\boldsymbol{K}\boldsymbol{Q}(\cos \boldsymbol{\Sigma}\lambda)\boldsymbol{Q}^\top + \boldsymbol{J}(\sin \boldsymbol{\Sigma}\lambda)\boldsymbol{Q}^\top), \tag{5.4}$$

where $\boldsymbol{J}\boldsymbol{\Sigma}\boldsymbol{Q}^\top = \boldsymbol{H}$ is the compact singular value decomposition (SVD) of the tangent vector $\boldsymbol{H}$. Note that $\boldsymbol{H}$ is written in upper case as it is a matrix; yet it is still is a vector in the sense that is a member of a tangent vector space. Here, $\boldsymbol{J}, \boldsymbol{K}, \boldsymbol{Q}$ and $\text{Exp}_{\boldsymbol{K}}\, \lambda \boldsymbol{H}$ are orthogonal matrices, and $\boldsymbol{\Sigma}$ is a diagonal matrix. λ is the geodesic parameter, and can be seen as a learning rate to control the magnitude of the movement towards the direction $\boldsymbol{H}$.

As for the log map, in this work, given two basis matrices $\boldsymbol{X}$ and $\boldsymbol{K}$ for input subspace $\boldsymbol{\chi}$ and anchor $\boldsymbol{\kappa}$, we utilize the following three equations to calculate the log map [2]:

$$\boldsymbol{B} = (\boldsymbol{K}^\top \boldsymbol{X})^{-1}(\boldsymbol{K}^\top - \boldsymbol{K}^\top \boldsymbol{X}\boldsymbol{X}^\top), \tag{5.5}$$

$$\boldsymbol{W}\boldsymbol{\Theta}\boldsymbol{Z}^\top = \boldsymbol{B}^\top, \tag{5.6}$$

$$\mathrm{Log}_{\boldsymbol{K}}\, \boldsymbol{X} = \boldsymbol{H} = \boldsymbol{W}^* \arctan(\boldsymbol{\Theta}^*)\boldsymbol{Z}^{*\top}, \tag{5.7}$$

where $\boldsymbol{W}^*, \boldsymbol{\Theta}^*, \boldsymbol{Z}^*$ represent the matrices with the first m columns of $\boldsymbol{W}, \boldsymbol{\Theta}$ and $\boldsymbol{Z}^*$ respectively.

To perform learning by RSGD, we derived the expressions of the derivative for the log on the Grassmann manifold, using conventional techniques to operate differential forms [81] and based on the derivative of SVD [109]. For the detailed procedures, see the Appendix B. We omit the iteration t for simplicity.

Given the gradient of all next layers after the log layer, up to the loss $\dot{\boldsymbol{H}} = \nabla_{\boldsymbol{H}} L$, we compute the gradients with respect to the anchor $\dot{\boldsymbol{K}} = \nabla_{\boldsymbol{\kappa}} L$ and input subspace $\dot{\boldsymbol{X}} = \nabla_{\boldsymbol{\chi}} L$. $\dot{\boldsymbol{K}}$ is used to update the anchor according to equation 5.3, and $\dot{\boldsymbol{X}}$ is used if there are layers previous to the log layer, to compute their respective backward steps.

We provide the update formulation for the Grassmann log equations 5.5, 5.6 and 5.7 in Figure 5.3. Since the log is determined from the composition of three functions, the chain rule can be automatically used to compute the final gradients.

In Figure 5.3, $\boldsymbol{\Omega}$ is defined as a diagonal matrix where the diagonal elements are $\boldsymbol{\Omega}_i = 1/(1+\boldsymbol{\Theta}_i^*)$. $\circ$ represents the Hadamard product, and $\boldsymbol{I}$ represents the identity matrix. $\boldsymbol{F}$ is a matrix of the form:

$$\boldsymbol{F}_{ij} = \begin{cases} 1/(\arctan^2(\boldsymbol{\Theta}_j) - \arctan^2(\boldsymbol{\Theta}_i)), & i \neq j \\ 0, & i = j. \end{cases} \tag{5.8}$$

For the gradient of equation 5.7, the derivatives $\dot{\boldsymbol{W}}^*$, $\dot{\boldsymbol{\Theta}}^*$ and $\dot{\boldsymbol{Z}}^*$ are m-leftmost matrices, so to continue back to the square matrix gradients $\dot{\boldsymbol{W}}$, $\dot{\boldsymbol{\Theta}}$ and $\dot{\boldsymbol{Z}}$, we can fill in columns of zeros (no gradient update in these variables) until the matrices become square. In the gradient of equation 5.6, $\boldsymbol{F}$ is similar to equation 5.8, but the non-diagonals are instead defined as $1/(\boldsymbol{\Theta}_j^2 - \boldsymbol{\Theta}_i^2)$. The most important part is in the gradient of 5.5 (upper box with equations), we obtain two update rules, one to backpropagate $\boldsymbol{X}$ in case a gradient-based pre-processing needs it, and one to update the anchor point $\boldsymbol{K}$.

5.2.4 Extension to multiple tangent spaces

We further extend the proposed log layer to learn multiple tangent spaces at the same time, by defining the layer parameters as a set of anchors $\{\boldsymbol{\kappa}_q\}_{q=1}^{Q}$. We map the point $\boldsymbol{\chi}$ into a vector $\boldsymbol{h}$ in a space $\mathbb{T}$ as follows:

$$\boldsymbol{h}_q = \mathrm{vec}(\mathrm{Log}_{\boldsymbol{\kappa}_q}\, \boldsymbol{\chi}) \tag{5.9}$$

$$\boldsymbol{h} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_q, \ldots, \boldsymbol{h}_Q]. \tag{5.10}$$

Here, the brackets denote the concatenation of vectors such that the output feature vector $\boldsymbol{h}$ is in a $\mathbb{T} \subseteq \mathbb{R}^{dmQ}$. Optimizing for a discriminative space $\mathbb{T}$ can be seen as a search in a product space of

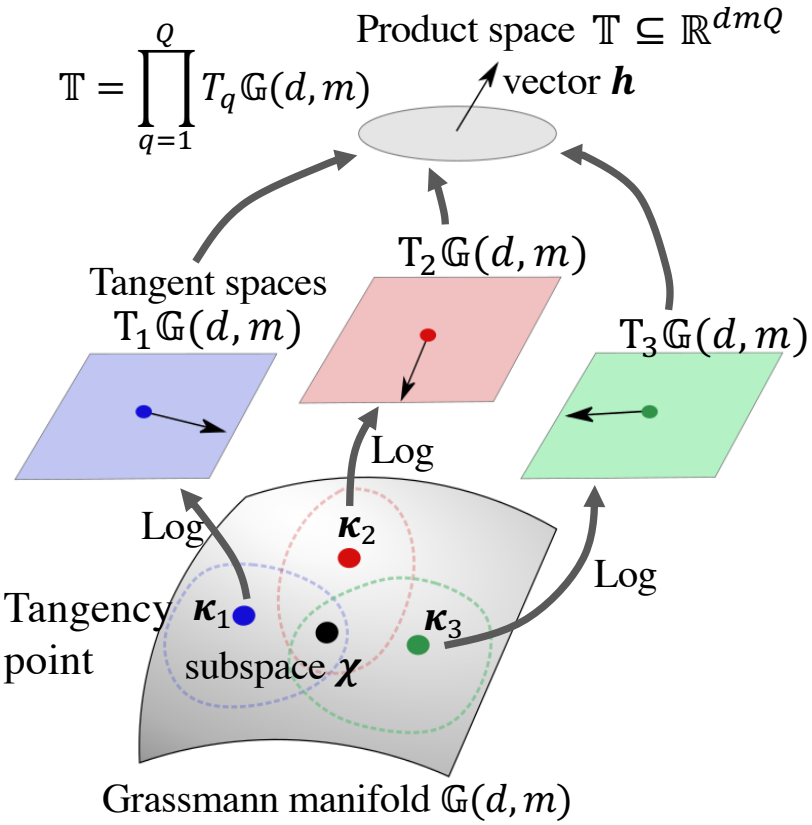


Figure 5.4: Diagram of the log module consisting of multiple tangent spaces.

the Grassmann tangent bundle $T\mathbb{G}(d,m)$, i.e. $\mathbb{T} = \prod_{q=1}^{Q} T_q\mathbb{G}(d,m)$. An example diagram for the case of $Q = 3$ is shown in Figure 5.4.

The basic intuition about introducing multiple tangent spaces is to cover a larger neighborhood of the manifold. As it maps a non-flat manifold onto a Euclidean space, there cannot be a single map that is perfectly distance preserving between any two points of the manifold. In general, the log map is a good approximation of the manifold in a local neighborhood of $\boldsymbol{\kappa}_q$ and its representation power decreases to points too far from it. However, we desire a good representation for the data distribution rather than for all points in the manifold. The core idea here is that each anchor point $\boldsymbol{\kappa}_q$ is treated as an independent learnable parameter. By having several $\boldsymbol{\kappa}_q$ and learning them from a random initialization without assumptions, we may find the tangent spaces that produce discriminant vectors for the given data samples. In the case $Q = 1$, the equations above reduce to conventionally using the log once.

As described in Sec. 5.1, this extension of the log model is similar to that of popular image descriptors such as Fisher vectors [87, 86], VLAD [8], and super-vector coding [125]. A Fisher vector consists of the log-likelihood gradients of data descriptors with respect to a Gaussian mixture. The log model can be seen as a kind of extension of this idea of using multiple tangent vectors, but to represent an image set rather than just a single image.

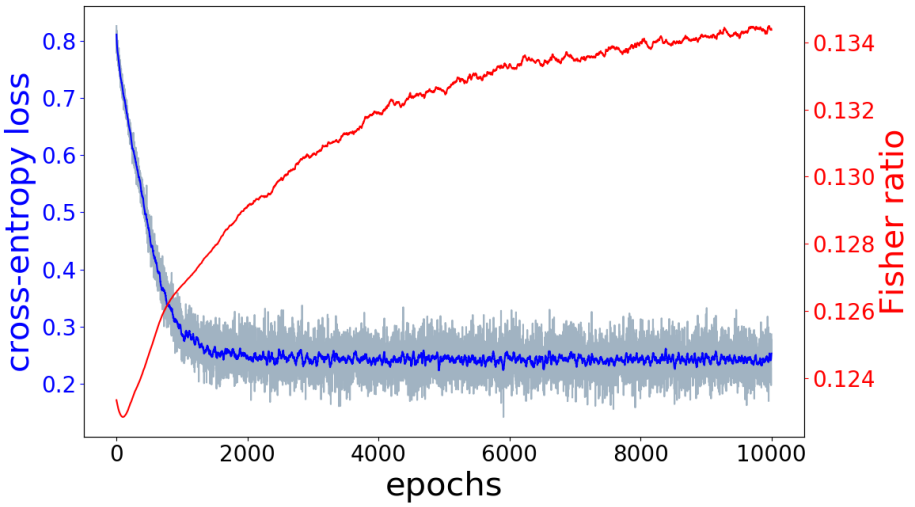


Figure 5.5: Plots of the loss and Fisher ratio of the log model tangent vectors after training for 10 thousand epochs in artificial data. The cross-entropy loss is drawn in blue, while the Fisher ratio between the tangent vectors is denoted in red. Note that the Fisher ratio is from before they have been projected on a discriminant space and shows only the effect of the tangent space representation.

5.3 Experiments

In this section, we discuss the validity of the proposed method through experiments in hand gesture, facial expression recognition and face identification tasks.

5.3.1 Experiments on artificial data

We have trained a Grassmann log model using 3D artificial data to visualize its mapped data and verify its effectiveness in a very simple case, while obtaining some intuition about its mechanism. The artificial data contains two classes of points on the $\mathbb{G}(3, 1)$, that is, the lines on 3-space crossing the origin. The data is represented by 3D vectors, and each class is generated by a tangent Gaussian distribution as developed in [111]. The model we utilized is composed of the log with 1 anchor, and two fully-connected layers, one 3×3 and another 3×2.

We have trained this model for 1000 epochs and plotted the anchor's behavior. Figure 5.5 shows the cross-entropy loss of the network and the Fisher ratio of the tangent vectors of the log map, for each epoch during training.

Here, one can observe that while the loss is minimized, the Fisher ratio of the tangent vectors raises, suggesting the advantageous effects of using a learnable anchor point. Therefore, the choice of a tangent space appears to contribute to the data separability. Moreover, Figure 5.6 shows the similarity between the data's Karcher mean and the anchor point for every epoch.

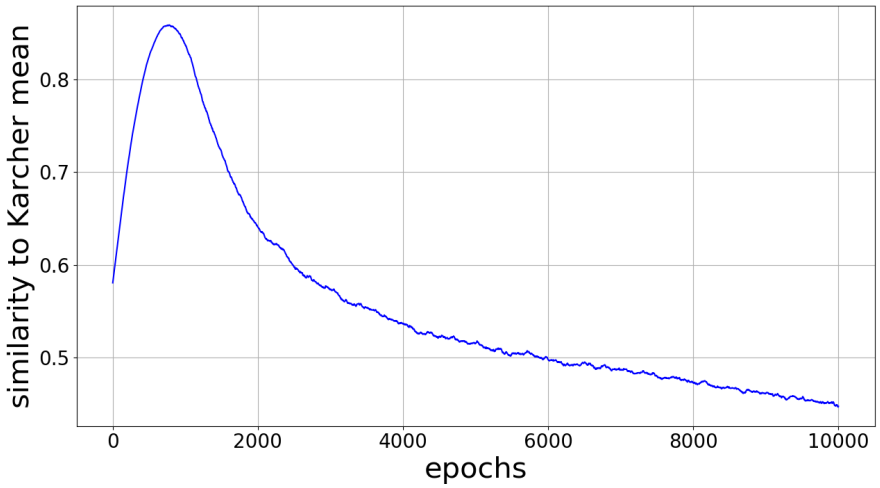


Figure 5.6: Plot of the similarity between the anchor point and the Karcher mean at each epoch. Note that the Karcher mean is fixed while the anchor point moved.

5.3.2 Experiments on hand shape recognition

We conducted an experiment with the Tsukuba hand shape dataset. This dataset contains 30 hand classes $\times$ 100 subjects, each of which contains 210 hand shape images, consisting of 30 frames $\times$ 7 different viewpoints. For each subject, we randomly created 6 sets with 5 frames from each viewpoint, so that each subject has 6 image sets of 35 images. In summary, there are a total of 18000 image sets in this dataset, each set containing image information from 7 camera viewpoints. In the experiments, all the images were resized to 24×24 pixels. We compute the subspaces by PCA and extract 6 components, so for the Log model the input subspaces are on the $\mathbb{G}(576, 6)$.

The Grassmann log model used in this section is formed by a log map followed by 4 fully-connected (F.C) layers, with batch normalization, dropout and number of anchor points is 6.

Effectiveness of learning tangent spaces

First, to verify the effectiveness of learning a tangent space, we trained two log models in the classification problem of 30 classes of hand shapes: one was trained normally, with random initialization and iterative updating of the tangent space according to the gradient. The other model was trained while freezing the tangent space at the Grassmann Karcher mean of all training subspaces, a version we refer to as Karcher log model.

After training, we have extracted the log map tangent vectors, and measured their Fisher ratio (FR), i.e. the ratio between the average inter-class and intra-class Euclidean distances of the tangent vectors. The FR of the log model was 0.0171, while the FR at the Karcher was 0.0159. The FR

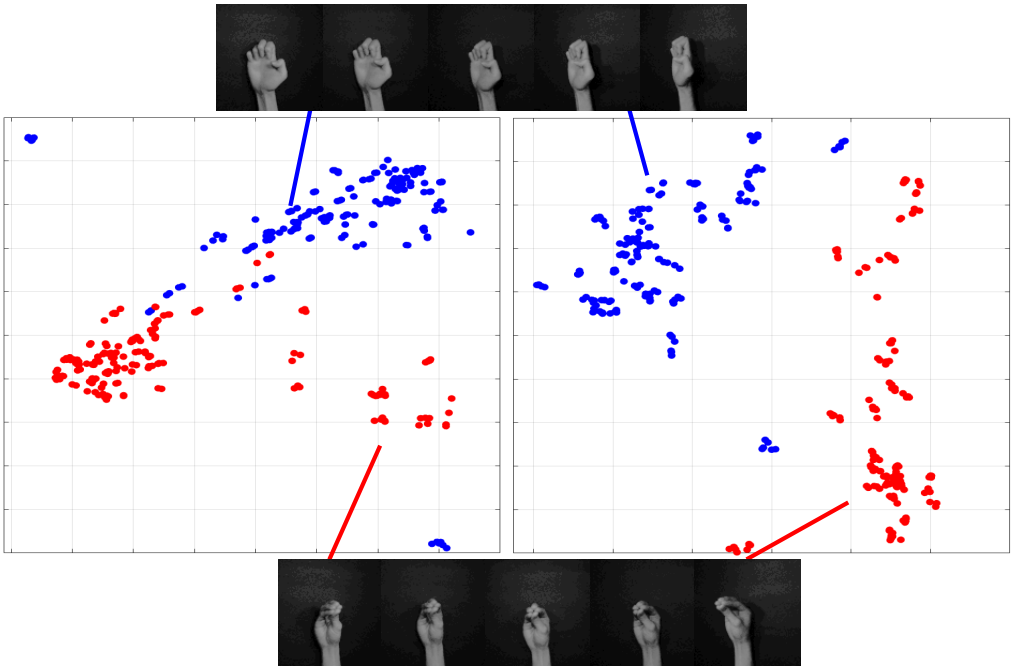


Figure 5.7: tSNE visualizations of 2 classes of hand shapes. The left plot denotes the tangent vectors at the Karcher mean, while the right plot shows the tangent vectors at the log model learned tangent space. It shows a representation of the vectors as 2D points based on their euclidean distances. It can be seen that the tangent vectors at the learned tangent space provide a more discriminative representation.

is higher when we allow the model to learn the anchor point, which indicates the choice of tangent space works towards increasing separation capability of the subspace data. To further exemplify this property, we have selected two classes at random and plotted their distance structure using tSNE [78], which can be seen in Figure 5.7. The plot shows a representation where each feature vector is shown as a point. The left plot denotes the tangent vectors at the Karcher mean, while the right plot shows the tangent vectors at the log model learned tangent space. The log model mapping seems to have made the samples more discriminant simply by finding suitable tangent spaces to project the data.

	Accuracy (%)
Karcher log model	70.65
Log model	81.90
Conv+log model	91.90
Resnet18+log model	99.40

Table 5.1: Results on the Tsukuba hand shape dataset.

As a second experiment, we evaluated the performance of the proposed log model in the classification problem of 30 types of hand shapes and compared it to the Karcher log model. We used the image sets of 70 subjects as training sets, holding a subset of 15 subjects (out of 70) for validation. The remaining 30 subjects were used as testing sets.

Table 5.1 shows the results. The proposed Log model equipped with several anchor points provided to be efficient in modeling the complexity of the hand shapes, while the Karcher log model achieved an inferior performance.

Scalability of the log model

We evaluated the scalability in performance of the proposed log model in the same hand shape classification problem, extending the log model with two variants: the Conv+log and Resnet18+log models.

Conv+log model refers to an architecture with 3 convolutions layers with batch normalization and pooling, where the convolutional filter size is 3, the number of filters of the first layer is 8, while the second and third are set to 2. Each layer has a zero-padding of 1 pixel and a pooling mask size of 2. After the convolutions, we have PCA and a log map, where the number of anchors of the log map is 2. After that, we utilize 4 F.C blocks.The Conv+log architecture was learned end-to-end including PCA, and entirely from scratch, using only randomly initialized weights. The method named Resnet18+log model uses a Resnet18 network pre-trained on ImageNet and fine-tuned on the hand shape data, similar to the conventional methods. We replaced the final F.C. layer of Resnet by a log model with PCA, log, and 4 F.C blocks, and trained this architecture while freezing the weights of the fine-tuned Resnet.

The results for these architectures can be seen in Table 5.1. When the learning framework is equipped with a convolutional layer, the log model achieves considerable accuracy improvement, strengthening the concept that the proposed interface is flexible enough to be incorporated in general neural network architectures. Following this pattern, the resnet18 architecture also benefits from the proposed interface.

Method	Accuracy (%)
DCC [65]	88.89 ± 2.45
MMD [116]	92.50 ± 2.87
CHISD [15]	96.52 ± 1.18
MMDML [76]	97.8 ± 1.0
ADNT [48]	97.92 ± 0.73
PLRC [22]	93.74 ± 4.3
DRM [97]	98.33 ± 1.27
Resnet vote	98.61 ± 1.52
Log model	98.19 ± 1.31

Table 5.2: Results of the experiment on the CMU MoBo dataset.

Method	Accuracy (%)
STM-ExpLet [73]	31.73
RSR-SPDML [44]	30.12
DCC [65]	25.78
GDA [41]	29.11
GGDA [42]	29.45
PML [54]	28.98
DeepO2P [59]	28.54
SPDNet [53]	34.23
GrNet-1 [55]	32.08
GrNet-2 [55]	34.23
Log model	32.61

Table 5.3: Results of the experiment on the AFEW dataset.

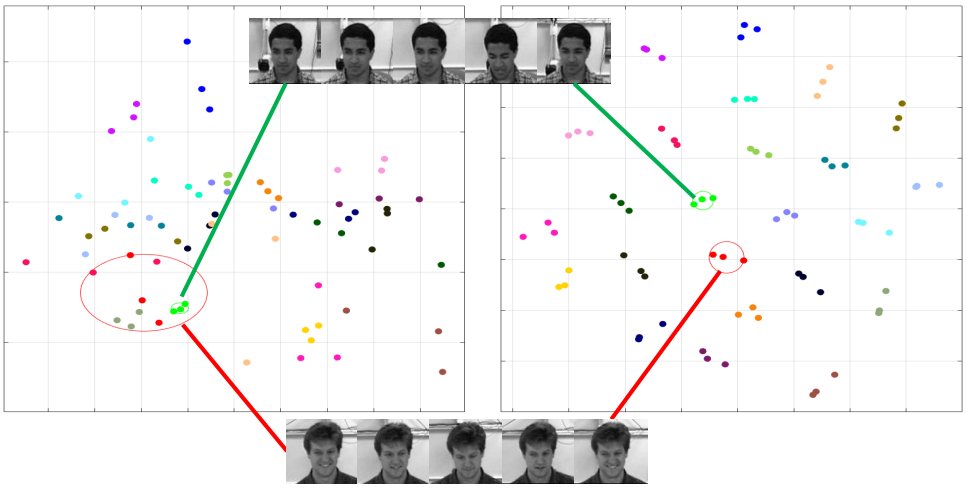


Figure 5.8: tSNE visualizations of 24 individuals of face image sets of the CMU Mobo dataset. The left plot denotes face image sets processed as subspaces. The plot shows a representation of the subspaces relative distances based on their similarities. The right plot shows mapped tangent vectors (output of the log layer). The colors represent each individual (each class). Visually, the tangent vectors provide a more discriminative representation.

5.3.3 Experiment on face identification

We conducted an experiment on the CMU Mobo dataset, consisting of videos of 25 people walking on a treadmill. This dataset was originally utilized for research on human gait analysis, but recently it has been used to compare the performance of image set based face classification methods [97, 48, 15].

We first detected face region from each video frame by the Viola and Jones detection algorithm [113]. A set of face images extracted from one video was considered as an image set. This dataset has four walking patterns (videos) of each person, except for one person. We evaluated the classification performance 10 times with the videos of 24 people with all walking patterns. For the evaluation, one video randomly selected from each person was used for training, while the remaining three videos were used for testing.

In this experiment, the subspaces are generated from CNN feature vectors. As the feature extractor, we used the ResNet-50 [49], which was fine-tuned to classify each face image of training data. For the fine-tuning, we added two fully connected (FC) layers after the last global average pooling layer in the network. The first FC layer outputs a 1024 dimension vector through the ReLU [82] function, and the second layer outputs a 24 (the number of classes) dimension vector through softmax function. We used the cross-entropy loss and Adam optimizer [66]. Hyperparameters of the optimizer were used as suggested by the original paper. We repeated the training to 100 epochs. Then, we extracted 1024 feature vectors by the first FC layer and for each set we computed a subspace with PCA by extracting 10 components. Therefore, a subspace input to the log module is on the $\mathbb{G}(1024, 10)$. We trained a Grassmann log model consisting of a log map followed by one fully-connected (F.C) layer, with number of anchor points 2.

First, we visualize the subspaces obtained through PCA in Figure 5.8 by using tSNE. The left plot shows a representation where each subspace is a point, and subspace similarities correspond to Euclidean distances on the plane. We also visualized the log tangent vectors of our trained model in the right plot. Since there are 2 anchor points, the dimension of the Euclidean feature space (product of the tangent spaces) is $2 \times 1024 \times 10$. Each point corresponds to one tangent vector, and their distances correspond to Euclidean distances on the tangent spaces. The output vectors of the log model exhibits compact clusters, while the ones on the Grassmann manifold are visually more dispersed. Additionally, the log model seems to discriminate data from different classes more appropriately, suggesting that the tangent spaces learned by the model provide a more suitable space for classification.

We have also evaluated the log model against various manifold methods from previous works. Resnet vote [49] is a baseline consisting of a Resnet50 fine-tuned to this dataset, where each image of a set is classified independently and a majority voting strategy is used to select a single class prediction. Table 5.2 shows the results. The proposed model overperforms the classic methods, and achieves a result on par with various deep methods.

5.3.4 Experiments on emotion recognition

We utilize the Acted Facial Expression in Wild (AFEW) [21] dataset. The dataset contains $1,345$ sequences of facial expressions acted by 330 actors in close to real world setting. We follow the experiment protocol established by [73, 53] to present the results on the validation set. The training videos are split into $1,747$ small subvideos augmenting the numbers of sets. For the evaluation, each facial frame is normalized to an image of size 20×20. For representation, following various works [74, 73, 55], we express the facial expression sequences with a set of linear subspaces of dimension 10, which exist on a Grassmann manifold $\mathbb{G}(400, 10)$. The Grassmann log model used in this section was composed of a log map followed by 3 fully-connected (F.C) layers, with batch normalization and dropout, in addition to a final F.C. with softmax, with 4 anchor points.

We compare the proposed log model with a number of methods for classification of manifold-valued data. Grassmann net (GrNet) [55], that proposes a block of manifold layers for subspace data, is denoted by GrNet-1 for the architecture with 1 block and GrNet-2 for the one with 2 blocks.

The results can be seen in Table 5.3. The proposed method achieved competitive results, even though it uses simple Euclidean operations such as fully-connected layers and a cross-entropy loss. First, it overperforms popular methods such as GDA and GGDA, that have a similar purpose of mapping Grassmann manifold data into a vector representation. This may be likely attributed to the fact that the log model learns both the representation and the Euclidean discriminant plane in an end-to-end manner, and the use of multiple F.C. layers allow a higher level of non-linearity for separation. In contrast, GDA uses a fixed kernel function and learns the discriminant independently. Methods such as SPDNet and GrNet are composed of many complex layers involving SVD, QR decompositions, and GramSchmidt orthogonalization and its derivatives are utilized as well. They increase in complexity as the number of layers increase by repeating these operations, which are not easily scalable in GPUs. In contrast, the proposed method offers competitive results employing a smaller set of parameters, benefiting a broader range of applications. By exploiting the tangent space properties, several practical advantages arise. For instance, the proposed model is naturally parallelizable. Also, it presents greater interpretability, providing a tool to understand its decisions

by using the tangent space.

5.4 Summary

This chapter proposes the Grassmann log model to map data from a Riemannian manifold to a vector space while maximizing discrimination capability for classification. The key idea is to formulate the Grassmann log map as a learnable model in such a way that it approximates well the manifold around the neighborhood of the data distribution. The proposed log model can be learned with Riemannian stochastic gradient descent; therefore it can be learned together with other powerful features such as cascaded convolutional layers. We performed classification experiments on multi-view hand shape recognition, face identification and and facial expression classification.

Chapter 6

Concluding remarks

This chapter summarizes the work presented in this thesis concerning the goals and the research direction in the future.

6.1 Summary

This thesis introduced learning algorithms to classify Grassmann-valued data, i.e., subspaces living in a Grassmann manifold. The motivation to use subspace representation is that 1) it arises naturally in many types of data, and 2) it is both practical, robust to noise, and invariant to various types of data transformations. Most importantly, this thesis's central motivation is the fact that the tools to process Grassmann-valued data are still underdeveloped despite their useful and practical nature.

Our algorithms have applications in signal processing and computer vision, particularly representing image sets and acoustic signals discriminatively. The goal was to develop general-purpose and straightforward algorithms that achieve high classification performance.

Chapter 3 proposed the Enhanced Grassmann discriminant analysis (eGDA), a generalization of Fisher discriminant analysis from Euclidean space to the Grassmann manifold. It is more specifically a generalization of GDA, which maps data to a Euclidean space through a kernel trick, a limited representation of the manifold that may not generalize its distance structure well when not enough data is given. In eGDA, more discriminant features can be achieved by projecting the subspaces onto a generalized difference subspace and then using a kernel trick to reproduce the manifold of projected subspaces. We demonstrate the application of this method to the classification of motion images.

Chapter 4 introduces two algorithms to classify signals that are generalizations of Fisher discriminant analysis from Euclidean to Grassmannian. Both methods extend MSSA, a method without a discriminant mechanism, operating just as the nearest neighbor algorithm for signals. The proposed methods are Grassmann singular spectrum analysis (GSSA), which uses a Grassmann kernel to add discrimination, and Tangent singular spectrum analysis (TSSA), which computes a discriminant space in a tangent space to classify Grassmann-valued data in a neighborhood of the data Karcher mean. We demonstrate the application of these methods to the task of bioacoustic signal classification.

Chapter 5 introduces the Grassmann log model, an end-to-end learnable neural network layer.

In this chapter, the problem we addressed is the fact that the tools for processing Grassmann-valued data within deep neural networks are lacking. The proposed log model maps Grassmann-valued data to vectors by learning a tangent space, such that this layer can be seamlessly connected to conventional neural network layers. We employ the log model to image set recognition tasks, such as face identification, facial expression recognition, and hand shape recognition.

6.2 Future work

Although the experimental results validate the proposed methods' effectiveness in various applications, many challenges remain to be done in the future.

In Chapter 3, we utilized a handcrafted feature called histogram of gradients (HOG), and a CNN called AlexNet to preprocess the frames of videos. In the future, we will consider the introduction of multiple sophisticated features such as dense trajectories and investigate their combination in terms of subspace representation. Our framework would also benefit from even more powerful learned representations. Therefore, an attractive research line would investigate DNN architectures combining the proposed eGDA with large models such as DenseNet and ResNet50. For that model to learn in an end-to-end manner, we will seek to reformulate the GDS operator as a neural network layer, where the gradient can be computed in a stable manner.

In Chapter 4, we decomposed a signal into oscillatory components using the simplest form of SSA. In the future, we will consider using other decompositions based on subspaces for various applications. For example, as SSA acts as a low-pass filter, it works well for discriminating bioacoustic data, with considerable environmental background noise. However, in other applications such as factory anomaly detection, we may lose important information hidden in the higher-frequency band of machine sounds. For that, we will consider employing a band-pass filter to construct subspaces of different frequency ranges. Another issue is that subspaces are not robust to outliers, since they can alter the oscillatory components. We will tackle this problem by considering more robust signal transforms. It could also be interesting to apply the proposed methods to other acoustic signals, such as signals with multiple channels.

In Chapter 5, the log operation is performed using a single compact SVD. In the future, we will consider creating an end-to-end model that does not need any matrix decomposition, as this is a costly operation with a theoretically unstable gradient in the paradigm of backpropagation.

Although all algorithms were proposed for Grassmann-valued data, the core ideas are quite general and can be extended to a much larger class of Riemann-valued data; that is, data represented as a point on a Riemannian manifold. Future works also include generalizing this thesis' learning algorithms to other Riemannian manifolds, such as Lie groups (e.g., rotation group, general linear group), structured tensors (e.g., symmetric positive definite, fixed-rank, stochastic matrices), statistical manifolds (e.g., the family of all normal distributions with Fisher metric), and other objects (e.g., hypersphere, hyperbolic space, real phases of Fourier transform). The value in these generalizations is that they would allow our learning algorithms to be applied in many applications even when both vector and subspace representations might not be the most suitable representation choice. We hope that these basic algorithms serve as a basis for data analysis tools involving manifold-valued data.

Appendix A

Derivation of the error-minimizing subspace

In this appendix, we demonstrate that the subspaces obtained through PCA without centering minimize the reconstruction error with respect to a set of patterns.

Our goal is to find the m-dimensional subspace $\boldsymbol{\beta}$ that minimizes the reconstruction error. More concretely, we seek the orthonormal basis $\{\boldsymbol{b}_j \in \mathbb{R}^d\}_{j=1}^m$ of the subspace $\boldsymbol{\beta}$ given sample vectors from the distribution. We also call the basis vectors $\{\boldsymbol{b}_j \in \mathbb{R}^d\}_{j=1}^m$ principal components. Let $\{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^n$ be a set of sample patterns from the underlying data distribution, with a sample size n and number of variables (dimension) d. Note that usually m is selected as a hyperparameter, often much smaller than d.

Let $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n] \in \mathbb{R}^{d\times n}$ be the matrix where each column is a sample pattern and $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_m] \in \mathbb{R}^{d\times m}$ be the matrix of basis vectors of $\boldsymbol{\beta}$. We refer to $\boldsymbol{B}$ as *basis matrix*.

The objective of this procedure is minimizing the error function:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{P}\boldsymbol{x}_i\|_2^2, \tag{A.1}$$

where $\boldsymbol{P} : \mathbb{R}^d \to \mathbb{R}^d$ is the *projection operator* onto $\boldsymbol{\beta}$.

Now we demonstrate the solution for minimizing the error function. First, define the *error matrix* as follows:

$$\boldsymbol{E} = \boldsymbol{X} - \boldsymbol{P}\boldsymbol{X} = [\boldsymbol{x}_1 - \boldsymbol{P}\boldsymbol{x}_1, \boldsymbol{x}_2 - \boldsymbol{P}\boldsymbol{x}_2, \cdots, \boldsymbol{x}_n - \boldsymbol{P}\boldsymbol{x}_n]. \tag{A.2}$$

We rewrite our optimization as:

$$\sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{P}\boldsymbol{x}_i\|_2^2 = \|\boldsymbol{E}\|_F^2, \tag{A.3}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The idea is that minimizing the sum of quadratic errors corresponds to minimizing the squared Frobenius norm of the error matrix. We now manipulate this

form using matrix algebra:

$$\begin{aligned}
\|\boldsymbol{E}\|_F^2 &= \|\boldsymbol{X} - \boldsymbol{P}\boldsymbol{X}\|_F^2 && \text{(A.4)}\\
&= \operatorname{tr}\left((\boldsymbol{X} - \boldsymbol{P}\boldsymbol{X})\,(\boldsymbol{X} - \boldsymbol{P}\boldsymbol{X})^\top\right) && \text{(A.5)}\\
&= \operatorname{tr}\left((\boldsymbol{X} - \boldsymbol{P}\boldsymbol{X})\,\left(\boldsymbol{X}^\top - \boldsymbol{X}^\top\boldsymbol{P}^\top\right)\right) && \text{(A.6)}\\
&= \operatorname{tr}\left(\boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{P}^\top - \boldsymbol{P}\boldsymbol{X}\boldsymbol{X}^\top + \boldsymbol{P}\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{P}^\top\right). && \text{(A.7)}
\end{aligned}$$

Now let $\boldsymbol{A} = \boldsymbol{X}\boldsymbol{X}^\top$, where $\boldsymbol{A}$ is called *autocorrelation matrix*. We replace it in our problem and perform a few more manipulations:

$$\begin{aligned}
\|\boldsymbol{E}\|_F^2 &= \operatorname{tr}\left(\boldsymbol{A} - \boldsymbol{A}\boldsymbol{P}^\top - \boldsymbol{P}\boldsymbol{A} + \boldsymbol{P}\boldsymbol{A}\boldsymbol{P}^\top\right) && \text{(A.8)}\\
&= \operatorname{tr}\left(\boldsymbol{A} - \boldsymbol{A}\boldsymbol{P} - \boldsymbol{P}\boldsymbol{A} + \boldsymbol{P}\boldsymbol{A}\boldsymbol{P}\right) && \text{(A.9)}\\
&= \operatorname{tr}(\boldsymbol{A}) - \operatorname{tr}\left(\boldsymbol{A}\boldsymbol{P}\right) - \operatorname{tr}\left(\boldsymbol{P}\boldsymbol{A}\right) + \operatorname{tr}\left(\boldsymbol{P}\boldsymbol{A}\boldsymbol{P}\right) && \text{(A.10)}\\
&= \operatorname{tr}(\boldsymbol{A}) - \operatorname{tr}\left(\boldsymbol{P}\boldsymbol{A}\right) - \operatorname{tr}\left(\boldsymbol{P}\boldsymbol{A}\right) + \operatorname{tr}\left(\boldsymbol{P}\boldsymbol{A}\right) && \text{(A.11)}\\
&= \operatorname{tr}(\boldsymbol{A}) - \operatorname{tr}\left(\boldsymbol{P}\boldsymbol{A}\right). && \text{(A.12)}
\end{aligned}$$

In eq. A.12 the scalar $\operatorname{tr}(\boldsymbol{A})$ is constant, so minimizing the squared norm of the error matrix is the same as maximizing the trace of the projected autocorrelation matrix. In addition, our goal is to find the basis matrix $\boldsymbol{B}$, so we replace $\boldsymbol{P} = \boldsymbol{B}\boldsymbol{B}^\top$. Now we rewrite the optimization:

$$\begin{aligned}
\min_{\boldsymbol{\beta}} \|\boldsymbol{E}\|_F^2 &= \max_{\boldsymbol{\beta}} \operatorname{tr}\left(\boldsymbol{P}\boldsymbol{A}\right) && \text{(A.13)}\\
&= \max_{\boldsymbol{\beta}} \operatorname{tr}\left(\boldsymbol{B}\boldsymbol{B}^\top\boldsymbol{A}\right) && \text{(A.14)}\\
&= \max_{\boldsymbol{\beta}} \operatorname{tr}\left(\boldsymbol{B}^\top\boldsymbol{A}\boldsymbol{B}\right). && \text{(A.15)}
\end{aligned}$$

Finally, we replace the abstract variable $\boldsymbol{\beta}$ by its basis matrix $\boldsymbol{B}$, and insert constraints of orthonormality:

$$\begin{aligned}
\max_{\boldsymbol{B}} \quad & \operatorname{tr}\left(\boldsymbol{B}^\top\boldsymbol{A}\boldsymbol{B}\right)\\
\text{s.t.} \quad & \boldsymbol{B}^\top\boldsymbol{B} = \boldsymbol{I}.
\end{aligned} \qquad \text{(A.16)}$$

We solve this problem by induction on m, i.e., considering each basis vector, hoping it will be clearer and more instructive for readers. For that, we can rewrite our problem in vector form as:

$$\begin{aligned}
\max_{\{\boldsymbol{b}_j\}} \quad & \sum_{j=1}^{m} \boldsymbol{b}_j{}^\top\boldsymbol{A}\boldsymbol{b}_j\\
\text{s.t.} \quad & <\boldsymbol{b}_j, \boldsymbol{b}_j>= 1, \quad <\boldsymbol{b}_j, \boldsymbol{b}_k>= 0 \quad \forall k = 1, \ldots, j-1.
\end{aligned} \qquad \text{(A.17)}$$

We start with the case $m = 1$, i.e., there is only one principal component. We employ the method of Lagrange multipliers as follows:

$$\max_{\boldsymbol{b}_1} \quad J(\boldsymbol{b}_1, \lambda_j) = \boldsymbol{b}_1{}^\top\boldsymbol{A}\boldsymbol{b}_1 - \lambda_1\left(\boldsymbol{b}_1{}^\top\boldsymbol{b}_1 - 1\right). \qquad \text{(A.18)}$$

The value λ_1 is the multiplier between the two terms. Now, we seek the stationary points:

$$\frac{\partial J}{\partial \boldsymbol{b}_1} = 2\boldsymbol{A}\boldsymbol{b}_1 - 2\lambda_1 \boldsymbol{b}_1 = 0, \tag{A.19}$$

$$\frac{\partial J}{\partial \lambda_1} = \boldsymbol{b}_1{}^\top \boldsymbol{b}_1 - 1 = 0. \tag{A.20}$$

Eq. A.20 simply indicates that the basis vector $\boldsymbol{b}_1$ has norm 1, while eq. A.19 leads to the following equation:

$$\boldsymbol{A}\boldsymbol{b}_1 = \lambda_1 \boldsymbol{b}_1. \tag{A.21}$$

The solution of this equation is well known: $\boldsymbol{b}_1$ must be an eigenvector of $\boldsymbol{A}$, while λ_1 must be its corresponding eigenvalue. In addition, by multiplying both sides by $\boldsymbol{b}_1{}^\top$ we can rewrite our objective as $\boldsymbol{b}_1{}^\top \boldsymbol{A}\boldsymbol{b}_1 = \lambda_1$. Thus, the maximum value of λ_1 is the solution, which leads to conclusion that λ_1 must be the highest eigenvalue of $\boldsymbol{A}$.

Now that we have shown the solution for the base case $m = 1$, we go back to look at the general case. We apply the method of Lagrange multipliers to eq. A.17:

$$\max_{\{\boldsymbol{b}_j\}} \quad \sum_{j=1}^{m} J(\boldsymbol{b}_j, \lambda_j) = \sum_{j=1}^{m} \boldsymbol{b}_j{}^\top \boldsymbol{A}\boldsymbol{b}_j - \sum_{j=1}^{m} \lambda_j \left(\boldsymbol{b}_j{}^\top \boldsymbol{b}_j - 1\right) - \sum_{j=1}^{m} \sum_{k=1}^{j-1} \alpha_k \boldsymbol{b}_j{}^\top \boldsymbol{b}_k. \tag{A.22}$$

The value λ_j is the multiplier for each normality constraint, while α_k is the multiplier for each orthogonality constraint. Now, we compute the stationary points for each term:

$$\frac{\partial J}{\partial \boldsymbol{b}_j} = \boldsymbol{A}\boldsymbol{b}_j - \lambda_j \boldsymbol{b}_j - \sum_{k=1}^{j-1} \alpha_k \boldsymbol{b}_k = 0, \tag{A.23}$$

$$\frac{\partial J}{\partial \lambda_j} = \boldsymbol{b}_j{}^\top \boldsymbol{b}_j - 1 = 0 \tag{A.24}$$

$$\frac{\partial J}{\partial \alpha_k} = \sum_{k=1}^{j-1} \boldsymbol{b}_j{}^\top \boldsymbol{b}_k = 0. \tag{A.25}$$

We multiply eq. A.23 by $\boldsymbol{b}_p{}^\top$ $(p = 1, \ldots, j-1)$ from the left to obtain:

$$\boldsymbol{b}_p{}^\top \boldsymbol{A}\boldsymbol{b}_j - \lambda_j \boldsymbol{b}_p{}^\top \boldsymbol{b}_j - \sum_{k=1}^{j-1} \alpha_k \boldsymbol{b}_p{}^\top \boldsymbol{b}_k = 0. \tag{A.26}$$

Here, since $\boldsymbol{b}_p{}^\top \boldsymbol{b}_j$ is always zero, the second term cancels. The last term cancels except when $p = k$, which leads to:

$$\boldsymbol{b}_p{}^\top \boldsymbol{A}\boldsymbol{b}_j - \alpha_p = 0. \tag{A.27}$$

However, we know that $\boldsymbol{b}_p{}^\top \boldsymbol{A}\boldsymbol{b}_j = \boldsymbol{b}_j{}^\top \boldsymbol{A}\boldsymbol{b}_p = \lambda_p \boldsymbol{b}_j{}^\top \boldsymbol{b}_p = 0$. Therefore, $\alpha_p = 0$.

Replacing this new result into eq. A.23 leads us to:

$$\boldsymbol{A}\boldsymbol{b}_j - \lambda_j \boldsymbol{b}_j = 0 \tag{A.28}$$

$$\boldsymbol{A}\boldsymbol{b}_j = \lambda_j \boldsymbol{b}_j. \tag{A.29}$$

Thus, the solution must among the eigenvalues and eigenvectors of $\boldsymbol{A}$. In addition, by the same argument used for the first component $\boldsymbol{b}_1$, it is clear that the maximizers correspond to the m highest eigenvalues.

In summary, the subspace that minimizes the reconstruction error (eq. A.3) is spanned by the leading m eigenvectors of $\boldsymbol{A}$. We write the basis matrix as:

$$\boldsymbol{B} = \boldsymbol{U}_{1:m}, \tag{A.30}$$

where $\boldsymbol{U}_{1:m}$ denotes the leftmost m columns of $\boldsymbol{U}$.

Appendix B

Gradient computation of the Grassmann log map

In this appendix, we present a derivation of the gradient of the log map for the Grassmann manifold $\mathbb{G}(d, m)$. For that, we utilise various conventional techniques to operate differential forms [81].

Given the data and the loss gradient $\nabla_{\boldsymbol{H}} L = \dot{\boldsymbol{H}}$, we compute the gradients with respect to the tangency point and data.

The Grassmann log consists of the three equations below:

$$\boldsymbol{B} = (\boldsymbol{K}^\top \boldsymbol{X})^{-1}(\boldsymbol{K}^\top - \boldsymbol{K}^\top \boldsymbol{X}\boldsymbol{X}^\top), \tag{B.1}$$

$$\boldsymbol{W}\boldsymbol{\Theta}\boldsymbol{Z}^\top = \boldsymbol{B}^\top, \tag{B.2}$$

$$\mathrm{Log}_{\boldsymbol{K}}\, \boldsymbol{X} = \boldsymbol{H} = \boldsymbol{W}^* \arctan(\boldsymbol{\Theta}^*)\boldsymbol{Z}^{*\top}, \tag{B.3}$$

where $\boldsymbol{W}^*, \boldsymbol{\Theta}^*, \boldsymbol{Z}^*$ represent the matrices with the first m columns of $\boldsymbol{W}, \boldsymbol{\Theta}$ and $\boldsymbol{Z}^*$ respectively. It should be noted that equation B.2 is the transposed SVD of $\boldsymbol{B}$. The differential of the expression B.3 may be written as:

$$\boldsymbol{H} = \boldsymbol{W}^* \arctan(\boldsymbol{\Theta}^*)\boldsymbol{Z}^{*\top} = \boldsymbol{W}^*\boldsymbol{S}\boldsymbol{Z}^{*\top}, \tag{B.4}$$

$$d\boldsymbol{H} = d\boldsymbol{W}^*\boldsymbol{S}\boldsymbol{Z}^{*\top} + \boldsymbol{W}^* d\boldsymbol{S}\boldsymbol{Z}^{*\top} + \boldsymbol{W}^*\boldsymbol{S}d\boldsymbol{Z}^{*\top}. \tag{B.5}$$

Above, we performed the change of variables $\boldsymbol{S} = \arctan(\boldsymbol{\Theta}^*)$. This contains an element-wise differential, simple to compute:

$$d\boldsymbol{S} = \boldsymbol{\Omega} d\boldsymbol{\Theta}, \tag{B.6}$$

where $\boldsymbol{\Omega}_i = 1/(1 + \boldsymbol{\Theta}_i^2)$ is a diagonal matrix and $i = 1, \ldots, d$ iterates over the diagonal. Since $\boldsymbol{W}^*$ is an orthogonal matrix, $\boldsymbol{W}^{*\top} d\boldsymbol{W}^*$ is skew-symmetric. This constraint leads to Townsend's solution [109] for equation B.5. We use this result and reverse the change of variables to obtain the update equations for each variable:

$$\dot{\boldsymbol{W}}^* = \boldsymbol{W}^*(\boldsymbol{F} \circ [\boldsymbol{W}^{*\top}\dot{\boldsymbol{H}}\boldsymbol{Z}^* \arctan(\boldsymbol{\Theta}^*) + \arctan(\boldsymbol{\Theta}^*)\boldsymbol{Z}^{*\top}\dot{\boldsymbol{H}}^\top \boldsymbol{W}^*] + (\boldsymbol{I} - \boldsymbol{W}^*\boldsymbol{W}^{*\top})\dot{\boldsymbol{H}}\boldsymbol{Z}^* \arctan(\boldsymbol{\Theta}^*)^{-1}). \tag{B.7}$$

$$\dot{\boldsymbol{\Theta}}^* = \boldsymbol{I} \circ [\boldsymbol{W}^{*\top} \dot{\boldsymbol{H}} \boldsymbol{Z}^*] \boldsymbol{\Omega}^{-1} \tag{B.8}$$

$$\dot{\boldsymbol{Z}}^* = \boldsymbol{Z}^* (\boldsymbol{F} \circ [\arctan(\boldsymbol{\Theta}^*) \boldsymbol{W}^{*\top} \dot{\boldsymbol{H}} \boldsymbol{Z}^* + \boldsymbol{Z}^{*\top} \dot{\boldsymbol{H}}^\top \boldsymbol{W}^* \arctan(\boldsymbol{\Theta}^*)]) + (\boldsymbol{I} - \boldsymbol{Z}^* \boldsymbol{Z}^{*\top}) \dot{\boldsymbol{H}}^\top \boldsymbol{W}^* \arctan(\boldsymbol{\Theta}^*)^{-1}. \tag{B.9}$$

$\circ$ represents the Hadamard product, and $\boldsymbol{I}$ represents the identity matrix. $\boldsymbol{F}$ is a matrix of the form:

$$\boldsymbol{F}_{ij} = \begin{cases} 1/(\arctan^2(\boldsymbol{\Theta}_j) - \arctan^2(\boldsymbol{\Theta}_i)), & i \neq j \\ 0, & i = j. \end{cases} \tag{B.10}$$

The results $\dot{\boldsymbol{W}}^*$, $\dot{\boldsymbol{\Theta}}^*$ and $\dot{\boldsymbol{Z}}^*$ are m-leftmost matrices, so to continue back to the full matrix gradients we can fill in columns of zeros until the matrices become square, where then we write then as $\dot{\boldsymbol{W}}$, $\dot{\boldsymbol{\Theta}}$ and $\dot{\boldsymbol{Z}}$. Then, the next step is to consider the equation $\boldsymbol{W}\boldsymbol{\Theta}\boldsymbol{Z}^\top = \boldsymbol{B}^\top$. Since it is a reconstruction rather than a decomposition, its update equation can be obtained as:

$$\begin{aligned} \dot{\boldsymbol{B}}^\top = [\boldsymbol{W}(\boldsymbol{F} \circ [\boldsymbol{W}^\top \dot{\boldsymbol{W}} - \dot{\boldsymbol{W}}^\top \boldsymbol{W}])\boldsymbol{\Theta} + (\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^\top)\dot{\boldsymbol{W}}\boldsymbol{\Theta}^{-1}]\boldsymbol{Z}^\top \\ + \boldsymbol{W}(\boldsymbol{I} \circ \dot{\boldsymbol{\Theta}})\boldsymbol{Z}^\top + \boldsymbol{W}[\boldsymbol{\Theta}(\boldsymbol{F} \circ [\boldsymbol{Z}^\top \dot{\boldsymbol{Z}} - \dot{\boldsymbol{Z}}^\top \boldsymbol{Z}])\boldsymbol{Z}^\top + \boldsymbol{\Theta}^{-1}\dot{\boldsymbol{Z}}^\top(\boldsymbol{I} - \boldsymbol{Z}\boldsymbol{Z}^\top)], \end{aligned} \tag{B.11}$$

where $\boldsymbol{F}$ follows equation B.10, but the non-diagonals are defined as $1/(\boldsymbol{\Theta}_j^2 - \boldsymbol{\Theta}_i^2)$.

Finally we consider the equation $\boldsymbol{B}^\top = (\boldsymbol{X}^\top \boldsymbol{K})^{-1}(\boldsymbol{X}^\top - \boldsymbol{X}^\top \boldsymbol{K}\boldsymbol{K}^\top)$. We transpose it and call $\boldsymbol{A} = (\boldsymbol{X}^\top - \boldsymbol{X}^\top \boldsymbol{K}\boldsymbol{K}^\top)$ and $\boldsymbol{C} = (\boldsymbol{X}^\top \boldsymbol{K})^{-1}$, then derivate it by the product rule, massaging the equation to obtain a general form:

$$d\boldsymbol{B} = d\boldsymbol{A}\boldsymbol{C} + \boldsymbol{A}d\boldsymbol{C}, \tag{B.12}$$

$$d\boldsymbol{A} = d\boldsymbol{X} - (d\boldsymbol{K}\boldsymbol{K}^\top \boldsymbol{X} + \boldsymbol{K}d\boldsymbol{K}^\top \boldsymbol{X} + \boldsymbol{K}\boldsymbol{K}^\top d\boldsymbol{X}), \tag{B.13}$$

$$d\boldsymbol{C} = -(\boldsymbol{K}^\top \boldsymbol{X})^{-1}(d\boldsymbol{K}^\top \boldsymbol{X} + \boldsymbol{K}^\top d\boldsymbol{X})(\boldsymbol{K}^\top \boldsymbol{X})^{-1} \tag{B.14}$$

We obtain two update rules, one to from $d\boldsymbol{B}$ with respect to $\boldsymbol{X}$ in case a gradient-based pre-processing needs it, and one with respect to $\boldsymbol{K}$ to update it as a parameter. The derivative with respect to $\boldsymbol{X}$ is calculated as follows. First we consider $d\boldsymbol{K} = 0$:

$$\begin{aligned} d\boldsymbol{B} = (\boldsymbol{X} - (d\boldsymbol{K}\boldsymbol{K}^\top \boldsymbol{X} + \boldsymbol{K}d\boldsymbol{K}^\top \boldsymbol{X} + \boldsymbol{K}\boldsymbol{K}^\top d\boldsymbol{X}))\boldsymbol{C} + \boldsymbol{A}(-(\boldsymbol{K}^\top \boldsymbol{X})^{-1}(d\boldsymbol{K}^\top \boldsymbol{X} \\ + \boldsymbol{K}^\top d\boldsymbol{X} + \boldsymbol{K}^\top d\boldsymbol{X})(\boldsymbol{K}^\top \boldsymbol{X})^{-1}), \end{aligned} \tag{B.15}$$

$$d\boldsymbol{B} = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{K}^\top)d\boldsymbol{X}(\boldsymbol{K}^\top \boldsymbol{X})^{-1} - \boldsymbol{A}(\boldsymbol{K}^\top \boldsymbol{X})^{-1}\boldsymbol{K}^\top d\boldsymbol{X}(\boldsymbol{K}^\top \boldsymbol{X})^{-1}, \tag{B.16}$$

$$d\boldsymbol{B} = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{K}^\top - \boldsymbol{A}(\boldsymbol{K}^\top \boldsymbol{X})^{-1}\boldsymbol{K}^\top)d\boldsymbol{X}(\boldsymbol{K}^\top \boldsymbol{X})^{-1}. \tag{B.17}$$

Then, since our loss function outputs a single real value, we can massage the equations to a canonical form $dL = \operatorname{tr} \dot{\boldsymbol{B}}^\top d\boldsymbol{B}$:

$$dL = \operatorname{tr} \dot{\boldsymbol{B}}^\top ((\boldsymbol{I} - \boldsymbol{K}\boldsymbol{K}^\top - \boldsymbol{A}(\boldsymbol{K}^\top \boldsymbol{X})^{-1}\boldsymbol{K}^\top)d\boldsymbol{X}(\boldsymbol{K}^\top \boldsymbol{X})^{-1}), \tag{B.18}$$

which leads to the final update equation for the gradient of $\boldsymbol{X}$:

$$\dot{\boldsymbol{X}} = (\boldsymbol{K}^\top \boldsymbol{X})^{-1}\dot{\boldsymbol{B}}^\top[(\boldsymbol{I} - \boldsymbol{K}\boldsymbol{K}^\top)(\boldsymbol{I} - \boldsymbol{X}(\boldsymbol{K}^\top \boldsymbol{X})^{-1}\boldsymbol{K}^\top]. \tag{B.19}$$

Repeating the same technique, the update equation for $\boldsymbol{K}$ can be derived as:

$$\dot{\boldsymbol{K}} = -(\boldsymbol{K}^\top \boldsymbol{X})^{-1}\dot{\boldsymbol{B}}^\top[\boldsymbol{K}^\top \boldsymbol{X} + \boldsymbol{K}^\top \boldsymbol{X}^\top + (\boldsymbol{X}^\top \boldsymbol{K})^{-1}\boldsymbol{X}^\top(\boldsymbol{I} - \boldsymbol{K}\boldsymbol{K}^\top)]. \tag{B.20}$$

Bibliography

[1] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. Riemannian geometry of grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematica*, 80(2):199–220, 2004.

[2] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[3] SN Afriat. Orthogonal and oblique projectors and the characteristics of pairs of vector spaces. *Proc. Cambridge Philos. Soc.*, 53:800–816, 1957.

[4] Sydney N Afriat. Orthogonal and oblique projectors and the characteristics of pairs of vector spaces. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 53, pages 800–816, 1957.

[5] Taleb Alashkar, Boulbaba Ben Amor, Mohamed Daoudi, and Stefano Berretti. A grassmann framework for 4d facial shape analysis. *Pattern Recognition*, 57:21–30, 2016.

[6] Jesús B Alonso, Josué Cabrera, Rohit Shyamnani, Carlos M Travieso, Federico Bolaños, Adrián García, Alexander Villegas, and Mark Wainwright. Automatic anuran identification using noise removal and audio activity detection. *Expert Systems with Applications*, 72:83–92, 2017.

[7] Ana Alves, Ricardo Antunes, Anna Bird, Peter L Tyack, Patrick JO Miller, Frans-Peter A Lam, and Petter H Kvadsheim. Vocal matching of naval sonar signals by long-finned pilot whales (globicephala melas). *Marine Mammal Science*, 30(3):1248–1257, 2014.

[8] Relja Arandjelovic and Andrew Zisserman. All about vlad. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1578–1585, 2013.

[9] Gaston Baudat and Fatiha Anouar. Generalized discriminant analysis using a kernel approach. *Neural computation*, 12(10):2385–2404, 2000.

[10] Gary Becigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. In *International Conference on Learning Representations*, 2019.

[11] Silvere Bonnabel. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

[12] Alexander Andreevich Borisenko and Yu A Nikolaevskii. Grassmann manifolds and the grassmann image of submanifolds. *Russian mathematical surveys*, 46(2):45, 1991.

[13] Benjamin Burchfiel and George Konidaris. Hybrid bayesian eigenobjects: Combining linear subspace and deep network methods for 3d robot vision. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6843–6850. IEEE, 2018.

[14] L. E. Carvalho and A. von Wangenheim. 3d object recognition and classification: a systematic literature review. *Pattern Analysis and Applications*, Feb 2019.

[15] Hakan Cevikalp and Bill Triggs. Face recognition based on image sets. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2567–2573. IEEE, 2010.

[16] Jun-Cheng Chen, Vishal M Patel, and Rama Chellappa. Unconstrained face verification using deep cnn features. In *2016 IEEE winter conference on applications of computer vision (WACV)*, pages 1–9. IEEE, 2016.

[17] Juan G Colonna, João Gama, and Eduardo F Nakamura. A comparison of hierarchical multi-output recognition approaches for anuran classification. *Machine Learning*, 107(11):1651–1671, 2018.

[18] Juan Gabriel Colonna, Marco Cristo, Mario Salvatierra, and Eduardo Freire Nakamura. An incremental technique for real-time bioacoustic signal segmentation. *Expert Systems with Applications*, 42(21):7367–7374, 2015.

[19] Trevor Darrell and Alex Pentland. Space-time gestures. In *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93., 1993 IEEE Computer Society Conference on*, pages 335–340. IEEE, 1993.

[20] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[21] Abhinav Dhall, Roland Goecke, Jyoti Joshi, Karan Sikka, and Tom Gedeon. Emotion recognition in the wild challenge 2014: Baseline, data and protocol. In *Proceedings of the 16th international conference on multimodal interaction*, pages 461–466. ACM, 2014.

[22] Qingxiang Feng, Yicong Zhou, and Rushi Lan. Pairwise linear regression classification for image set retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4865–4872, 2016.

[23] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[24] P Thomas Fletcher and Sarang Joshi. Principal geodesic analysis on symmetric spaces: Statistics of diffusion tensors. In *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis*, pages 87–98. Springer, 2004.

[25] King Sun Fu and TS Yu. *Statistical pattern classification using contextual information*, volume 1. Research Studies Press Ltd, 1980.

[26] Kazuyuki Fujii. Introduction to grassmann manifolds and quantum computation. *Journal of Applied Mathematics*, 2, 2002.

[27] Kazuhiro Fukui and Atsuto Maki. Difference subspace and its generalization for subspace-based methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(11):2164–2177, 2015.

[28] Kazuhiro Fukui and Osamu Yamaguchi. Face recognition using multi-viewpoint patterns for robot vision. In *Robotics Research. The Eleventh International Symposium*, pages 192–201. Springer, 2005.

[29] Kazuhiro Fukui and Osamu Yamaguchi. The kernel orthogonal mutual subspace method and its application to 3d object recognition. In *Asian Conference on Computer Vision*, pages 467–476. Springer, 2007.

[30] Reinosuke Fukunaga. Statistical pattern recognition. 1990.

[31] B. B. Gatto, J. G. Colonna, E. M. dos Santos, and E. F. Nakamura. Mutual singular spectrum analysis for bioacoustics classification. In *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2017.

[32] Bernardo B Gatto, Anna Bogdanova, Lincon S Souza, and Eulanda M dos Santos. Hankel subspace method for efficient gesture representation. In *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2017.

[33] Bernardo B Gatto, Eulanda M dos Santos, Kazuhiro Fukui, Waldir SS Júnior, and Kenny V dos Santos. Fukunaga–koontz convolutional network with applications on character classification. *Neural Processing Letters*, 52:443–465, 2020.

[34] Bernardo B Gatto, Eulanda M dos Santos, Alessandro L Koerich, Kazuhiro Fukui, and Waldir SS Junior. Tensor analysis with n-mode generalized difference subspace. *arXiv preprint arXiv:1909.01954*, 2019.

[35] Bernardo B Gatto, Lincon S Souza, Eulanda M dos Santos, Kazuhiro Fukui, Waldir SS Júnior, and Kenny V dos Santos. A semi-supervised convolutional neural network based on subspace representation for image classification. *EURASIP Journal on Image and Video Processing*, 2020(1):1–21, 2020.

[36] Bernardo Bentes Gatto, Juan Gabriel Colonna, Eulanda Miranda dos Santos, and Eduardo Freire Nakamura. Mutual singular spectrum analysis for bioacoustics classification. In *Machine Learning for Signal Processing, 2017 IEEE International Workshop on*. IEEE, 2017.

[37] Bernardo Bentes Gatto, Lincon Sales de Souza, and Eulanda M dos Santos. A deep network model based on subspaces: A novel approach for image classification. In *Machine Vision*

Applications (MVA), 2017 Fifteenth IAPR International Conference on, pages 436–439. IEEE, 2017.

[38] Bernardo Bentes Gatto, Eulanda Miranda dos Santos, and Waldir Sabino Da Silva. Orthogonal hankel subspaces for applications in gesture recognition. In *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 429–435. IEEE, 2017.

[39] Nina Golyandina and Anatoly Zhigljavsky. *Singular Spectrum Analysis for time series*. Springer Science & Business Media, 2013.

[40] Jihun Hamm. Subspace-based learning with grassmann kernels. 2008.

[41] Jihun Hamm and Daniel D Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Proceedings of the 25th international conference on Machine learning*, pages 376–383. ACM, 2008.

[42] Jihun Hamm and Daniel D Lee. Extended grassmann kernels for subspace-based learning. In *Advances in neural information processing systems*, pages 601–608, 2009.

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

[44] Mehrtash T Harandi, Mathieu Salzmann, and Richard Hartley. From manifold to manifold: Geometry-aware dimensionality reduction for spd matrices. In *European conference on computer vision*, pages 17–32. Springer, 2014.

[45] Mehrtash T Harandi, Conrad Sanderson, Sareh Shirazi, and Brian C Lovell. Graph embedding discriminant analysis on Grassmannian manifolds for improved image set matching. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 2705–2712. IEEE, 2011.

[46] Mehrtash T Harandi, Conrad Sanderson, Sareh Shirazi, and Brian C Lovell. Kernel analysis on grassmann manifolds for action recognition. *Pattern Recognition Letters*, 34(15):1906–1915, 2013.

[47] Hossein Hassani and Dimitrios Thomakos. A review on singular spectrum analysis for economic and financial time series. *Statistics and its Interface*, 3(3):377–397, 2010.

[48] Munawar Hayat, Mohammed Bennamoun, and Senjian An. Deep reconstruction models for image set classification. *IEEE transactions on pattern analysis and machine intelligence*, 37(4):713–727, 2014.

[49] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[50] H. Hotelling. Relation between two sets of variables. *Biometrica*, 28:322–377, 1936.

[51] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[52] Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

[53] Zhiwu Huang and Luc Van Gool. A riemannian network for spd matrix learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[54] Zhiwu Huang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Projection metric learning on grassmann manifold with application to video based face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 140–149, 2015.

[55] Zhiwu Huang, Jiqing Wu, and Luc Van Gool. Building deep networks on grassmann manifolds. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[56] T Iijima. A theoretical study of pattern recognition by matching method. In *Proc. of First USA-Japan Computer Conf., 1972*, pages 42–48, 1972.

[57] T Iijima, H Genchi, and K Mori. A theory of character recognition by matching method. In *Proc. of 1st Int. Conf. on Pattern Recognition*, pages 50–56, 1973.

[58] Taizo Iijima. Basic theory on feature extraction for visual pattern. *J. IEICE*, 46:1714, 1963.

[59] Catalin Ionescu, Orestis Vantzos, and Cristian Sminchisescu. Training deep networks with structured layers by matrix backpropagation. *arXiv preprint arXiv:1509.07838*, 2015.

[60] Y Iwai, Lao S., O Yamaguchi, and T Hirayama. A survey on face detection and face recognition. *Computer vision and image media conference of the Japanese information processing society (CVIM)*, 2005(38 (2005-CVIM-149)):343–368, 2005.

[61] Zhuolin Jiang, Zhe Lin, and Larry Davis. Recognizing human actions by learning and matching shape-motion prototype trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):533–547, 2012.

[62] Kari Karhunen. Zur spektraltheorie stochastischer prozesse. *Ann. Acad. Sci. Fennicae, AI*, 34, 1946.

[63] Tomokazu Kawahara, Masashi Nishiyama, Tatsuo Kozakaya, and Osamu Yamaguchi. Face recognition based on whitening transformation of distribution of subspaces. In *Proc. ACCV07 Workshop Subspace*, pages 97–103, 2007.

[64] Tae-Kyun Kim and Roberto Cipolla. Canonical correlation analysis of video volume tensors for action categorization and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8):1415–1428, 2009.

[65] Tae-Kyun Kim, Josef Kittler, and Roberto Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1005–1018, 2007.

[66] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015.

[67] Josef Kittler and Peter C Young. A new approach to feature selection based on the karhunen-loeve expansion. *Pattern recognition*, 5(4):335–352, 1973.

[68] Teuvo Kohonen. *Associative memory: A system-theoretical approach*, volume 17. Springer Science & Business Media, 2012.

[69] Teuvo Kohonen, Gábor Németh, K-J Bry, Matti Jalanko, and Heikki Riittinen. Spectral classification of phonemes by learning subspaces. In *ICASSP'79. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 97–100. IEEE, 1979.

[70] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[71] Binlong Li, Mustafa Ayazoglu, Teresa Mao, Octavia I Camps, and Mario Sznaier. Activity recognition using dynamic subspace angles. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3193–3200. IEEE, 2011.

[72] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5455–5463, 2015.

[73] Mengyi Liu, Shiguang Shan, Ruiping Wang, and Xilin Chen. Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1749–1756, 2014.

[74] Mengyi Liu, Ruiping Wang, Zhiwu Huang, Shiguang Shan, and Xilin Chen. Partial least squares regression on grassmannian manifold for emotion recognition. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 525–530. ACM, 2013.

[75] Michel Loeve. Functions aleatoires du second ordre. *Processus stochastique et mouvement Brownien*, pages 366–420, 1948.

[76] Jiwen Lu, Gang Wang, Weihong Deng, Pierre Moulin, and Jie Zhou. Multi-manifold deep metric learning for image set classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1137–1145, 2015.

[77] Yui Man Lui, J Ross Beveridge, and Michael Kirby. Action classification on product manifolds. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 833–839. IEEE, 2010.

[78] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[79] Ken-ichi Maeda. From the subspace methods to the mutual subspace method. In *Computer Vision*, pages 135–156. Springer, 2010.

[80] Ken-ichi Maeda, Osamu Yamaguchi, and Kazuhiro Fukui. Towards 3-dimensional pattern recognition. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 1061–1068. Springer, 2004.

[81] Thomas P Minka. Old and new matrix algebra useful for statistics. *See www. stat. cmu. edu/minka/papers/matrix. html*, 2000.

[82] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*, pages 807–814, 2010.

[83] Yasuhiro Ohkawa and Kazuhiro Fukui. Hand-shape recognition using the distributions of multi-viewpoint image sets. *IEICE transactions on information and systems*, 95(6):1619–1627, 2012.

[84] Erkki Oja. *Subspace methods of pattern recognition*, volume 6. Research Studies Press, 1983.

[85] Erkki Oja and Maija Kuusela. The alsm algorithman improved subspace method of classification. *pattern Recognition*, 16(4):421–427, 1983.

[86] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007.

[87] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *European conference on computer vision*, pages 143–156. Springer, 2010.

[88] Ilyas Potamitis, Stavros Ntalampiras, Olaf Jahn, and Klaus Riede. Automatic bird sound detection in long real-field recordings: Applications and tools. *Applied Acoustics*, 80:1–9, 2014.

[89] Mohammad Pourhomayoun, Peter Dugan, Marian Popescu, and Christopher Clark. Bioacoustic signal classification based on continuous region processing, grid masking and artificial neural network. *arXiv preprint arXiv:1305.3635*, 2013.

[90] Hossein Rahmani, Ajmal Mian, and Mubarak Shah. Learning a deep model for human action recognition from novel viewpoints. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 40(3):667–681, 2018.

[91] Mikel D Rodriguez, Javed Ahmed, and Mubarak Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[92] JF Ruiz-Muñoz, Zeyu You, Raviv Raich, and Xiaoli Z Fern. Dictionary learning for bioacoustics monitoring with applications to species classification. *Journal of Signal Processing Systems*, pages 1–15, 2016.

[93] Hitoshi Sakano and Naoki Mukawa. Kernel mutual subspace method for robust facial image recognition. In *KES'2000. Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies. Proceedings (Cat. No. 00TH8516)*, volume 1, pages 245–248. IEEE, 2000.

[94] Farook Sattar, Sarika Cullis-Suzuki, and Feng Jin. Identification of fish vocalizations from ocean acoustic data. *Applied Acoustics*, 110:248–255, 2016.

[95] Bernhard Scholkopft and Klaus-Robert Mullert. Fisher discriminant analysis with kernels. *Neural networks for signal processing IX*, 1(1):1, 1999.

[96] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36. IEEE, 2004.

[97] Syed AA Shah, Uzair Nadeem, Mohammed Bennamoun, Ferdous Sohel, and Roberto Togneri. Efficient image set classification using linear regression based image reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 99–108, 2017.

[98] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

[99] Rim Slama, Hazem Wannous, Mohamed Daoudi, and Anuj Srivastava. Accurate 3d action recognition using learning on the grassmann manifold. *Pattern Recognition*, 48(2):556–567, 2015.

[100] N. Sogi, T. Nakayama, and K. Fukui. A method based on convex cone model for image-set classification with cnn features. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, July 2018.

[101] Naoya Sogi and Kazuhiro Fukui. Action recognition method based on sets of time warped arma models. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 1773–1778. IEEE, 2018.

[102] Khurram Soomro and Amir R Zamir. Action recognition in realistic sports videos. In *Computer Vision in Sports*, pages 181–208. Springer, 2014.

[103] Lincon Souza, Hideitsu Hino, and Kazuhiro Fukui. 3d object recognition with enhanced grassmann discriminant analysis. In *ACCV 2016 Workshop (HIS 2016)*, 2016.

[104] Lincon S. Souza, Bernardo Bentes Gatto, and Kazuhiro Fukui. Enhancing discriminability of randomized time warping for motion recognition. In *Machine Vision Applications (MVA), 2017 Fifteenth IAPR International Conference on*, pages 77–80. IEEE, 2017.

[105] Chendra Hadi Suryanto, Jing-Hao Xue, and Kazuhiro Fukui. Randomized time warping for motion recognition. *Image and Vision Computing*, 54:1–11, 2016.

[106] Chendra Hadi Suryanto, Jing-Hao Xue, and Kazuhiro Fukui. Randomized time warping for motion recognition. *Image and Vision Computing*, 54:1–11, 2016.

[107] Hengliang Tan, Ying Gao, and Zhengming Ma. Regularized constraint subspace based method for image set classification. *Pattern Recognition*, 76:434–448, 2018.

[108] Hengliang Tan, Zhengming Ma, Sumin Zhang, Zengrong Zhan, Beibei Zhang, and Chenggong Zhang. Grassmann manifold for nearest points image set classification. *Pattern Recognition Letters*, 68:190–196, 2015.

[109] James Townsend. Differentiating the singular value decomposition, 2016.

[110] Pavan Turaga, Ashok Veeraraghavan, Anuj Srivastava, and Rama Chellappa. Statistical computations on grassmann and stiefel manifolds for image and video-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2273–2286, 2011.

[111] Pavan Turaga, Ashok Veeraraghavan, Anuj Srivastava, and Rama Chellappa. Statistical computations on grassmann and stiefel manifolds for image and video-based recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 33(11):2273–2286, 2011.

[112] Bindu Verma and Ayesha Choudhary. Framework for dynamic hand gesture recognition using grassmann manifold for intelligent vehicles. *IET Intelligent Transport Systems*, 12(7):721–729, 2018.

[113] Paul Viola and Michael J Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[114] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Computer Vision, IEEE International conference on*, pages 3551–3558, 2013.

[115] Ruiping Wang, Huimin Guo, Larry S Davis, and Qionghai Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 2496–2503. IEEE, 2012.

[116] Ruiping Wang, Shiguang Shan, Xilin Chen, and Wen Gao. Manifold-manifold distance with application to face recognition based on image set. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[117] Satoshi Watanabe. Karhunen-loeve expansion and factor analysis: theoretical remarks and application. In *Trans. on 4th Prague Conf. Information Theory, Statistic Decision Functions, and Random Processes Prague*, pages 635–660, 1965.

[118] Satosi Watanabe. Evaluation and selection of variables in pattern recognition. *Computer and Information Science II*, pages 91–122, 1967.

[119] Xing Wei, Yue Zhang, Yihong Gong, Jiawei Zhang, and Nanning Zheng. Grassmann pooling as compact homogeneous bilinear pooling for fine-grained visual classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 355–370, 2018.

[120] Jie Xie, Juan G Colonna, and Jinglan Zhang. Bioacoustic signal denoising: a review. *Artificial Intelligence Review*.

[121] O. Yamaguchi, K. Fukui, and K. Maeda. Face recognition using temporal image sequence. *Proc. International Conference on Automatic Face and Gesture Recognition*, pages 318–323, 1998.

[122] Osamu Yamaguchi and Kazuhiro Fukui. Smartface–a robust face recognition system under varying facial pose and expression. 2003.

[123] Minghai Yao, Xinyu Qu, Qinlong Gu, Taotao Ruan, and Zhongwang Lou. Online pca with adaptive subspace method for real-time hand gesture learning and recognition. *Wseas transactions on computers*, 9(6):583–592, 2010.

[124] Zhong-Qiu Zhao, Shou-Tao Xu, Dian Liu, Wei-Dong Tian, and Zhi-Da Jiang. A review of image set classification. *Neurocomputing*, 335:251–260, 2019.

[125] Xi Zhou, Kai Yu, Tong Zhang, and Thomas S Huang. Image classification using super-vector coding of local image descriptors. In *European conference on computer vision*, pages 141–154. Springer, 2010.

[126] Rui Zhu, Kazuhiro Fukui, and Jing-Hao Xue. Building a discriminatively ordered subspace on the generating matrix to classify high-dimensional spectral data. *Information Sciences*, 382:1–14, 2017.

[127] Yulian Zhu and Jing Xue. Face recognition based on random subspace method and tensor subspace analysis. *Neural Computing and Applications*, 28(2):233–244, 2017.

List of publications

1. Lincon S. Souza, Bernardo B. Gatto, Jing-Hao Xue, Kazuhiro Fukui,“Enhanced Grassmann Discriminant Analysis with Randomized Time Warping for Motion Recognition”, Pattern Recognition, vol.97, issue 1, pp.107028, 2020.

2. Lincon S. Souza, Naoya Sogi, Bernardo B. Gatto, Takumi Kobayashi, Kazuhiro Fukui,“An Interface between Grassmann manifolds and vector spaces”, Proc. 37th IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2020) Workshop on Differential Geometry in Computer Vision and Machine Learning, pp. 846-847, 2020.

3. Lincon S. Souza, Bernardo B. Gatto, Kazuhiro Fukui,“Classification of Bioacoustic Signals with Tangent Singular Spectrum Analysis”, Proc.44th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2019), pp. 351-355, 2019.

4. Lincon S. Souza, Bernardo B. Gatto, Kazuhiro Fukui, “Grassmann Singular Spectrum Analysis for Bioacoustics Classification”, Proc. 43th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2018), pp. 256-260, 2018.