Spatio-Temporal Tensor Analysis on Product Grassmann Manifolds and its Application to Action Recognition

March 2023

Bojan Batalo

Spatio-Temporal Tensor Analysis on Product Grassmann Manifolds and its Application to Action Recognition

Graduate School of Science and Technology
Degree Programs in Systems and Information Engineering
University of Tsukuba

March 2023

Bojan Batalo

Abstract

Complex multi-dimensional data containing both spatial and temporal information can be represented by tensors. Widely used in fields where such data needs to be treated in a unified way (e.g. mechanics, electrodynamics, neural networks), the very general tensorial representation lends to the use of well-defined mathematical operations from multilinear algebra. This leads to not only computationally efficient representational and analytic methods, but to ones that might be generalized to numerous application scenarios. There are various approaches to dealing with tensor-valued data; often they include extensions of classical statistic and machine learning tools from vector- or matrix-valued data. Notable examples include extending Principal Component Analysis (PCA) to Multilinear Principal Component Analysis (MPCA), Canonical Correlation Analysis (CCA) to Tensor Canonical Correlation Analysis (TCCA), and Linear Discriminant Analysis (LDA) to Multilinear Discriminant Analysis. Most of these methods rely on tensor decomposition in order to bridge the gap between linear and multilinear algebra. One common approach is to first decompose and then unify each tensor from a dataset into a single point on non-Euclidean space called product Grassmann manifold (PGM). In this way, a simple and compact representation is achieved, and discriminative analysis can be performed using distances between the points on the PGM. The downside of this approach is that, most often, it does not take into account the underlying nature of tensor data; namely, it does not distinguish spatial from temporal information, and treats them in the same way. In applications where sequential context is crucial, this loss of temporal information leads to a degradation in performance. In this thesis, we develop a general approach for representing and analyzing spatio-temporal tensors, and show their application to the task of action recognition. First, we address the key issues of PGM representation by explicitly modeling temporal information from spatio-temporal tensors. This is achieved by first representing the temporal tensor mode with a Hankel-like matrix, followed by a sequence-preserving linear subspace encoding which can be seamlessly unified with PGM representation. We showcase this representation, named Product Grassmann Manifold with Hankel-like Embedding (PGM-HLE), through tasks of visualization and clustering of spatio-temporal datasets, concretely those containing human actions. Second, we greatly improve this representation by incorporating the randomized generalization of Hankel-like matrices, and develop a classification algorithm, Temporal-Stochastic Product Grassmann Manifold (TS-PGM) which we demonstrate on the task of gesture and action recognition. Finally, we develop discriminative extensions to TS-PGM by using (1) kernel mapping to handle non-linearity in the data and (2) projection on generalized difference subspace (GDS) to reduce overlap between class subspaces.

Acknowledgements

This thesis would not have been possible without the help of many fine people around me. Though some of them might not be fully aware of the impact they had, their part was instrumental in keeping me strong enough to push through the challenges that these studies have presented.

I would like to thank the Japanese Ministry of Education, Culture, Science and Technology (MEXT) for granting me the scholarship, and therefore the opportunity and the means to pursue doctoral studies in Japan. This gratitude also extends to the University of Tsukuba, which, in addition, granted me the freedom and support to focus on my research only.

My wholehearted thanks goes out to Prof. Kazuhiro Fukui, who mentored me, guided my research, and always kept his faith in me. His optimistic attitude, stalwart in the face of adversity and the toughest of circumstances, inspired me to never give up on a project and persevere until the end.

I would like to thank the rest of my thesis committee members: Prof. Takumi Kobayashi, Prof. Yoshihiro Kanamori, Prof. Itaru Kitahara and Prof. Keisuke Kameyama, for excellent questions during the first ever in-person presentation of my work, and so many encouraging comments. Their lively discussion invigorated me and granted motivation to further pursue a research career.

I am very grateful to Dr. Keisuke Yamazaki for granting the opportunity to work as a research assistant in his team at AIST. His keen insights improved my research skills in many ways, and offered a different perspective on what a research career can be. His professional, yet laid-back attitude made the entire experience immensely enjoyable.

Many thanks go to Prof. Neil Millar, who gave me the chance to work with him, and explore a very interesting research topic. Our weekly discussions were something I always looked forward to, not only for the chance to learn about writing, academic discourse and the English language in general, but also for the sake of great conversation itself. It was fun to solve problems, those we predicted and those we did not, and finally see our efforts yield great results.

The path of every Ph.D. student begins with a paved road which gradually, but certainly, turns to a mountain trail through unclaimed lands. I have had the exquisite fortune of working together with Dr. Lincon S. Souza and Dr. Bernardo B. Gatto, who paved the way for me, and they paved it well. They did not only teach me how to conduct research; they *showed* me how to be a researcher in spirit. With them, every conversation carried a well-meaning lesson, filled with anecdotes and relatable stories. They taught me the most valuable skill a researcher, and perhaps a person can have - the ability to turn failure into opportunity. The work that is the backbone of this thesis has their fingerprints all over it. I thank them very much for their efforts, time, patience and energy. Special thanks to Lincon, without whose help this thesis would not have been possible. He has supported me at every single step of the way, knowing when to help me, and perhaps even more importantly, when to let me fight on my own.

I would like to thank the members of Computer Vision Laboratory: Dr. Erica Kido Shimomoto and Dr. Naoya Sogi for being role model Ph.D. students, always ready to support; Huang Yuantian, Guoqing Hao and everyone else for making lab life much more interesting, with conversations about research and life. Many thanks to Ms. Hiroko Sawabe for her inexhaustible patience and kindness.

I have moved from my home country to pursue the Ph.D. degree, leaving behind most of my friends. Dušan Radisavljević, however, embarked on this journey with me and has supported me

through thick and thin. Though we each fight our own battles, knowing that help is right around the corner freed my mind to do its best, ready to take the next challenge without fear. I want to thank him for every late night call, each friendly visit and adventure, and the blessing of time simply spent together when conversation seemed superfluous. It made the the whole difference.

I would also like to thank Marko Čančar and Nikola Aleksić for keeping in touch from back home, sharing memories when they were most needed, and keeping me in line when I complained too much; thanks to all my other friends who have not forgotten me during these years. Special thanks to Djordjije Džudović for his encouragements and philosophical debates that kept my brain working, even when it was about to fall over from fatigue.

The thought of making so many good friends outside of my home country has never crossed my mind, yet it happened. I am truly grateful for their presence during my entire studies, as without them, I would have been left to fight this battle more alone than I thought possible. Thanks to Sergey Pavlov, Yeldar Toleubay, Kamilla Enikeeva and Kara Dinissa for their constant presence, patience when I was losing my mind, and the will to keep hanging out after I had skipped on many encounters due to research obligations. I will never forget their part in this.

The biggest supporter through this journey, by far, has been my girlfriend Maha Mahyub. She magnificently held on through every annoying, unexpected bump on the road. She offered kindness, patience, and love, sorely needed and a blessing in difficult moments. She was amazing throughout, a ray of sunshine every step of the way. Thanks for reminding me, every day, and especially when I thought I don't need such a reminder, that there are things far more important in life than work. You made this battle worth it.

Finally, I would like to thank my parents, Snežana and Stevo, my sister Tamara, and my grandmother Venka. There are no words good enough to describe how grateful I am for their support. Luckily, they know.

Contents

List of Figures

Chapter 1

Introduction

It has become common knowledge that technological advances are exponential, and that any advance opens up avenues for further development and expansion, not only in the field in which the advance occurred, but also in its adjacent fields. For example, the rise of computer science and computational devices has opened up previously unimaginable advances in classical scientific fields such as mathematics, physics, chemistry, medicine, etc. This allowed for breakthroughs to occur in those fields, which spill over and affect their adjacent fields, often in a circular fashion.

One such phenomena can be observed in the context of data and data processing. In recent years there has been a significant shift in the approach to handling data in virtually every technological and scientific field. Advances in sensor technology led to increasing availability of various types of increasingly complex data from real-world, and even simulated phenomena. This in turn created a need for a wide array of methods which are able to consume and process such data. If such methods are proven successful, new findings could lead to further breakthroughs and advancements. Thus, the task of processing increasingly complex data becomes paramount for the future development of any scientific field.

Sometimes it can be difficult to intuitively understand what construes *complex data*. Take camera sensors for example. Not very long ago only black-and-white photographs were available, comprised entirely of black and white pixel data. Then, color information was added to each pixel, as well as the ability to record a succession of such images in the form of a video, adding movement information. Various other forms of imaging have been developed as well, such as thermal and hyperspectral imaging, depth imaging, specialized medical imaging, and so on. Each new development adds more information and context, which can be critical for a given application. Accordingly, computational representations of such data grow increasingly complex, from a simple matrix of pixels, to multi-dimensional arrays containing color and depth information that change over time.

Multi-dimensional data can now be found in every scientific field, from image and video processing, hyperspectral and medical imaging, signal processing, internet data processing, recommendation systems, and many others [44, 75, 97, 130, 84, 11, 119, 92, 42, 76]. This, as mentioned before, calls for increasingly sophisticated data representation and processing methods, ones that are, most of the time, not trivial to implement. Another issue is that such methods are generally tailored to a specific application, and cannot be generalized to a broad set of tasks, even in the same domain. Thus, a lot of time and effort is spent implementing highly-specialized methods when a more generalized

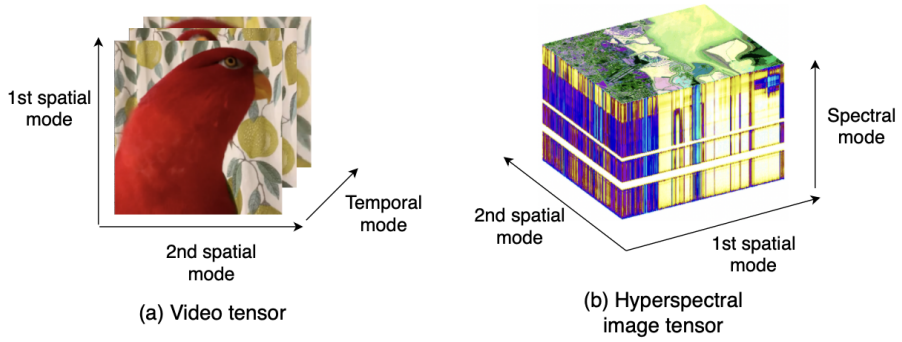

Figure 1.1: Examples of tensors and their modes. On image (a) a spatio-temporal tensor in the form of a video is shown, containing two spatial modes and one temporal mode. Image (b) shows a hyperspectral image, another example of a multidimensional data point; this tensor contains two spatial modes and one spectral mode.

approach could provide the same performance, with the added benefit of being employable on different tasks. Of course, one could argue that developing generalized methods is even more difficult and time-consuming, as it requires strong theoretical background and a very good insight into all the possible use cases. Even when successfully developed, a generalized method often needs to be fine-tuned to a specific task in order to be usable at all, which again requires time and effort.

In this thesis we argue that there exists a middle-ground between these two extremes; a more balanced approach than a fully general or highly specialized one. It is possible to pick such a starting point which distances the author from a specific application just enough to provide a decent amount of generalizability without greatly sacrificing performance, while also not pushing them too far away into the realm of pure theory. One way of achieving this is to find a key representation for a family of problems, and focus on developing an array of techniques that deal with such a representation.

An interesting family of problems is video data processing, in the context of human gesture and action recognition tasks. These tasks have increased in importance in recent years, with the abundance of video data generated from various digital cameras, including, but not limited to, smartphones, CCTVs, webcams, car dash cameras, etc. In almost all of these videos, of utmost interest are human actions. There are many examples of such applications, including monitoring pedestrians crossing a street, detecting sign language or emotion from a webcam, spotting anomalies in human behaviour in order to prevent injury or harm to other people, detecting sports movements and patterns for analytical and entertainment purposes, and many others. The key aspect of this data is that it contains both spatial and temporal information - movement is essentially a change of position across time.

Complex multi-dimensional data containing both spatial and temporal information can be naturally represented by *tensors*. Widely used in fields where such data needs to be treated in a unified way (e.g. mechanics, electrodynamics, neural networks), the very general tensorial representation lends to the use of well-defined mathematical operations from multilinear algebra. This leads not only to computationally efficient representational and analytic methods, but to ones that might be generalized to numerous application scenarios. Videos are especially well-suited for representation

with high-order tensors by the virtue of having at least two spatial and one temporal dimension, and the intuition that the interplay between these dimensions contains vital information.

A *tensor* is essentially a multi-dimensional array, where the number of indices used to access an element of the tensor correlates with the order of such a tensor. For example, a pixel in a grayscale video can be accessed by three indices (x-coordinate, y-coordinate, and the number of the frame), making it a third-order tensor. Tensors can be regarded as a generalization of vectors and matrices, where vectors are first-order tensors, and matrices are second-order tensors. Examples of spatio-temporal and hyperspectral tensors are shown on Figure 1.1

However, most methods in machine learning deal exclusively with vector or matrix-valued data. Often, a multi-dimensional data point is preprocessed in such a way to fit the form of a vector or matrix, to enable the application of a standard machine learning recognition method. This breaks apart the inherent multi-dimensional structure of the data point and incurs a loss of information, which might degrade recognition performance. In the case of videos, such treatment may break spatial and temporal connections of pixels. Another problem is the exponential increase in dimensionality of feature space when a tensor is vectorized; a 100x100 resolution video with 30 frames becomes a $300,000$-dimensional vector. This leads to an issue commonly known as the *curse of dimensionality*; dealing with such high-dimensional vectors is computationally expensive, and often does not work due to lack of training data for a very specific application.

There are studies and methods that attempt to deal with tensors in their original form and circumvent these issues. In this thesis, we would like to contribute to the field of tensor analysis by proposing general representational methods for spatio-temporal tensors. Our key idea is that with this approach, it is possible to strike the fine balance between general and highly-specific, and propose a set of tools based on tensor representation and multilinear algebra which can be used for a whole range of applications dealing with spatio-temporal tensors, such as human gesture and action recognition.

To this end, we first provide a survey of tensor analysis methods in the next section which are of special interest, and discuss some of their problems. We then position our proposed methods alongside established work.

1.1 Evolution and overview of tensor analysis methods

It was not immediately obvious to the scientific community that tensor representation is necessarily well-suited for complex, multilinear data. However, over the past three decades, continuous research on data representation and pattern analysis methods has naturally led to such a conclusion; as a result, a plethora of tensor-based methods have been developed, for an equally numerous array of applications.

It is prudent to first consider the versatility of tensor representation, and consequently, tensor-based methods, before delving into the survey. Initial efforts were made in standard pattern recognition tasks such as face recognition [125, 69, 68, 40, 126, 127, 64] and 3D object recognition [87]. It made sense due to natural representation of images as matrices of grayscale pixels, i.e. second-order tensors, and 3D objects as third-order tensors. Quickly, these efforts spilled over to other computer vision recognition tasks such action and gait recognition from videos [10, 28, 63, 66, 110], where videos are viewed as second-, third- or higher-order tensors; further, sequences of 3D skeletons for

3D skeleton action recognition [4, 53] are also easily modeled by higher-order tensors. Not surprisingly, tensors were soon found to be very useful for processing data other than standard images and videos, such as hyperspectral cubes [85], and fMRIs [34], as well as other various tasks including image compression and retrieval [128, 36], texture rendering [115], and space-time super-resolution [94].

Fields other than computer vision have seen a rise in tensor-based methods. Signal processing, for example, a field rich with multichannel and multi-source signal data has seen various tensor-based methods [97, 12]. Another significant branch of computer science greatly benefiting from tensorial representation is data mining [105, 106, 83], where significant work has been done to discover underlying correlations between individual dimensions of multi-dimensional data. Other than that, tensors were found handy to process biological data such as EEG signals [57, 67] and DNA sequences [129]. There has even been work on music genre classification [81]. Finally, tensors have found their use in solving general regression problems [29], parameter estimation for latent variable models [1] and many others not covered in this thesis.

In this section we will briefly cover the evolution and generalization of pattern recognition methods from firstly relying on pure vector representation, to utilizing matrices when they seem a natural choice for data representation, to finally considering higher-order tensors. Then, we delve into several tensor-based methods of particular note and interest to our research direction; this allows us to position our method and its contribution alongside established work.

Note that we mostly restrict our survey to tensor-based methods for computer vision tasks, particularly those involving video data in general, and action or hand gesture sequences in particular, to keep in line with our main research goal.

1.1.1 From linear to multilinear perspective on data

The history of pattern recognition methods based on linear algebra and second-order statistics. In this subsection we cover the most notable methods related to our proposed research. Figure 1.2 depicts some of those methods, along with their relations to each other.

Principal component analysis (PCA) [16] **and linear discriminant analysis (LDA)** [3] are among the first methods used for various pattern recognition tasks. Certainly, they have become extremely influential to the landscape of pattern recognition and machine learning. The shared philosophy behind PCA and LDA is relatively simple: given a set of data points, find orthogonal projections (*principal components*) whose linear combinations explain most of the variability of the data set. They differ, however, in their approaches: PCA is an *unsupervised* approach, meaning there is no prior knowledge or concept of the data point belonging to a certain class; LDA is a *supervised* approach, where the concept of class exists and data points are split among several classes, with projections being optimized so that they achieve the best separability among classes. There are various ways to obtain principal components, but usual approach is through **eigenvalue decomposition (EVD)** or **singular value decomposition (SVD)**.

Both methods assume that a data point is represented by a single vector. One of the first applications of PCA was through **Eigenfaces** [111] for the task of face recognition. A set of images, each image represented by a vector, is processed and decomposed through eigenvalue decomposition. Resulting eigenvectors, or principal components, are then used as projective vectors for images to create a new feature space, one that is significantly reduced in dimension. A linear classifier is

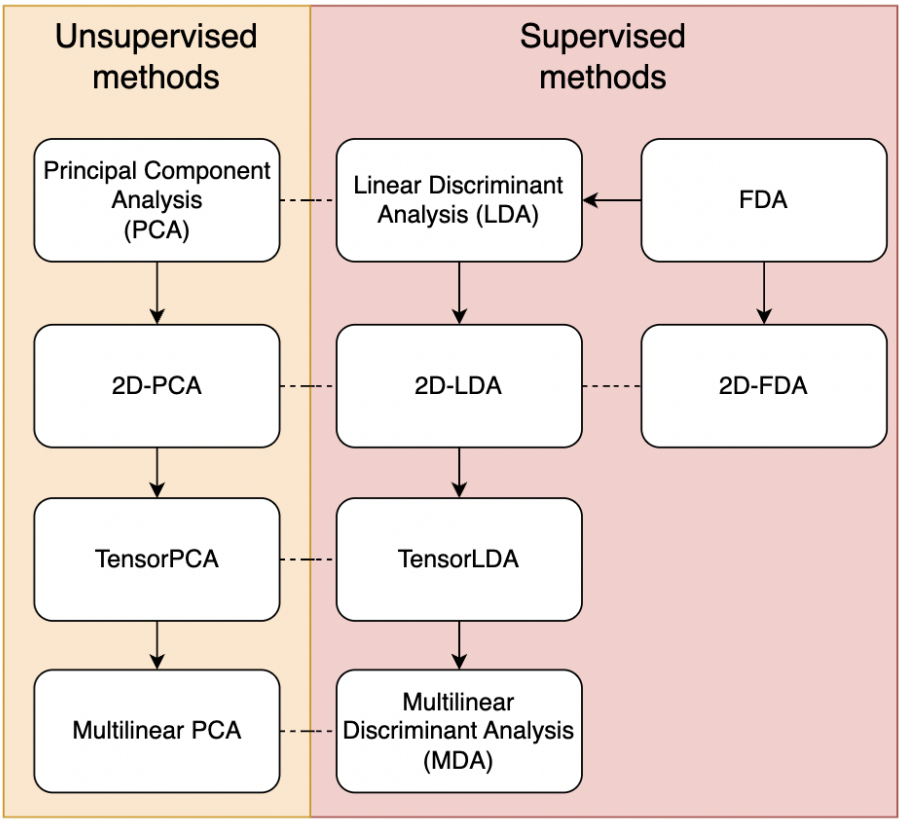

Figure 1.2: A simplified map of key pattern recognition methods ranging from linear to multilinear approaches. Arrows indicate predecessor-successor relationship, while dotted lines indicate joint underlying concepts. Extensions and application-dependent modifications of methods have been omitted from the map.

then trained on these projected images. However, the **Eigenfaces** method has been shown to exhibit vulnerability to changes in illumination, pose, and other variations other than the identity of the face. Thus, it was quickly surpassed by many other methods; however, it served as a philosophical cornerstone for a plethora of methods on the basis of PCA. A supervised version of this method, called **Fisherfaces** [6], has been implemented to combat the lack of discriminability, following the philosophy of LDA. Further, due to LDA being a linear method, it may sometimes fail to capture well the more complex distributions of data in the feature space; kernel techniques such as **Kernel Direct Discriminant Analysis (KDDA)** [69] are used to address this issue, which work by applying a kernel trick to transform the feature space from a linearly non-separable space to a linearly-separable space.

In general, approaches based on PCA and LDA suffer from two drawbacks regardless of the application scenario. First, reorganizing data to fit the vector form often yields vectors of very high dimension, giving way to a *curse of dimensionality*, where it becomes exponentially more difficult to obtain good prediction performance as the number of dimensions increases; sometimes this exacerbates the *small sample size* problem in applications where not a lot of data is available.

Second, the act of reshaping initial data structures to vector form potentially breaks apart data cohesion. This led to development of first methods utilizing matrices instead of vectors, such as **two-dimensional PCA (2D-PCA)** [126], **two-dimensional LDA (2D-LDA)** [60, 58], **two-dimensional Fisher Discriminant Analysis (2D-FDA)** [52] and their extensions and various modification for specific applications.

2D-PCA [126] is among the first attempts to generalize PCA to non-vector data. It forgoes vector-based description of images in favour of matrices. Though this idea sprang from computer vision, it is possible to generalize to other types of data; the main idea is that when a data point naturally represented a matrix is transformed into a vector, the correlations between elements break down and a data loss is incurred. 2D-PCA introduces a method for computing a covariance matrix from matrices instead of vectors, as is done by regular PCA. Then, the covariance matrix is decomposed, via eigenvalue decomposition or SVD, as in regular PCA and principal components are obtained as eigenvectors; eigenvectors corresponding to highest eigenvalues explain most of the variance of the data. This process is less computationally expensive as the decomposition is executed on a significantly smaller covariance matrix; besides, a preprocessing step of converting images to feature vectors is not required. Though 2D-PCA partially succeeded in moving data representation from vectors to matrices, it is worth noting that resulting principal components are still vectors, therefore some of the interpretability is lost as original data is in matrix form. This also means that variation along only one matrix axis is captured, so a certain information is incurred.

There are several extensions to 2D-PCA throughout the literature that try to address some of its problems. One of them is **alternate 2D-PCA (2PCA)** [131] which addresses the problem of 2D-PCA that occurs when obtaining principal components; 2D-PCA obtains them row-wise, while 2PCA allows for column-wise principal components. Experimental results have shown though, that there is no significant increase in performance if column-wise principal components are chosen over row-wise principal components. Another method is **Extended 2D-PCA (E2D-PCA)** [86] which utilizes both the row and column orientation simultaneously, but again with limited improvement. Finally, **Color PCA** [98], attempts to address a general flaw of matrix-based image representation, where only grayscale values of pixels are used and color information is lost. Namely, in Color PCA the three image matrices (each representing one color channel) are concatenated and 2D-PCA proceeds as usual; this resulted in only slight improvements.

Analogously, there are several attempts to generalize LDA into **2D-LDA**. One [60] proposes an iterative method to solve a version of the Fisher criterion and find Foley-Sammon optimal set of discriminant vectors. It does not always have the best result, as there is no closed-form solution to the optimization problem, unlike in the case of regular LDA. Another one [58], more successfully, tries to solve the Fisher criterion through between-class and within-class scatter matrix optimization, though it is also an iterative algorithm with no theoretical guarantees for convergence. Following the same idea of between-class and within-class scatter matrices, a **2D-FDA** [52] has been proposed, where features are extracted with FDA instead of PCA before solving the Fisher criterion.

The first true tensor-based methods have been developed on the backs of multilinear algebra and higher-order statistics; their aim was precisely to deal with problems that come from representing multidimensional data with vectors and matrices by using higher-order tensors. Previously mentioned methods are based in classical linear algebra and second-order statistics; tensor-based approaches are based in multilinear algebra and higher-order statistics. The research direction born out of this basis is often referred to as *multilinear subspace learning*. As before, the first of

these methods were developed for face and action recognition, before spreading to other application domains.

TensorFaces [114] is a tensor-based extension of PCA, mainly used for the task of face recognition. It aims to address the downsides of regular PCA such as inability to handle variations in the data; the key idea is that decomposing tensors will be able to account for such variation. Technically, the entire set of faces is represented by a higher-order tensor, where each dimension of the tensor corresponds to a certain variability: person, illumination, pose, etc. Then, an n-mode SVD [50] is used to decompose and obtain principal components for each of the tensor modes, i.e. each of the variables of faces. This is one of the first methods to introduce n-mode SVD, a method often used by the methods that follow. Unfortunately, the proposed n-mode SVD is not a "true" tensor SVD, considering it lacks some properties of regular matrix SVD, however, it is a step in the right direction. Another problem with Tensorfaces is that they are not constructed with discriminative methods in mind; therefore, they lack the classification capacity in some regard. Finally, the images are still treated as vectors, just like in original PCA.

Some of the issues of TensorFaces were addressed by the introduction of **TensorPCA** and its discriminant version, **TensorLDA** [9]. Firstly, these methods addressed the problem of representing images as vectors; instead, each face/image is modeled as a 2D tensor (2nd-order tensor) and is able to retain the spatial correlations of pixels. The underlying framework is similar between TensorPCA and TensorLDA; the main dfference is in their unsupervised and supervised nature respectively. The key idea is the following: suppose an image to be a 2D matrix (2nd-order tensor); there exists a tensor subspace such that when projecting the original image onto it, most of the variability of the image is retained. This concept is analogous to standard PCA; the only difference is that in PCA, principal components are vectors, whereas in TensorPCA and TensorLDA, the "principal components" are matrices. In the case of TensorPCA, tensor subspace is found by solving an optimization problem with the main goal of reducing reconstruction error. On the other hand, TensorLDA utilizes an approach to regular LDA, where between-class and within-class matrices are used to achieve best separability. In both methods, distance between two images (two matrices) is defined as a Frobenius norm. These approaches resulted in much lower computation time due to significantly reduced dimensionality, while achieving better classification results. However, both methods are tested only on face recognition task, and may fare poorly on more complicated tasks, especially those factoring data which is not easily linearly separable.

However, a **multilinear PCA (MPCA)** [64, 65] has been developed which generalizes PCA for higher-order tensors. The key idea of MPCA is to find a tensor projection that maximizes the total tensor scatter, i.e., the projection that captures most of the input tensor variance, analogously to the objective of regular PCA. The MPCA approach is iterative, and assumes, like PCA, that the dimension of reduced space is known ahead of time. However, in most problems determining this dimension is a non-trivial task; thus, MPCA includes a term in its objective function for finding an optimal subspace dimension based on the wanted amount of compression. MPCA has been proposed as a feature extraction mechanism for the task of object detection and recognition; extracted MPCA features can be used with LDA to perform classification.

MPCA is often used along with other techniques to tackle various problems. For example, a boosting algorithm is used to learn an ensemble of MPCA-generated tensor subspaces for the gait recognition problem [63]. A different boosting method utilizes MPCA features to learn several weak LDA learners which work in unison to produce a strong classifier [66]. One method combines

random subspace method [37] with MPCA to reduce the number of selected features and thus increase computational efficiency [118]. An unsupervised variant has been developed [120] as well, with the aim to increase robustness to variability in the dataset, and also to incompleteness of the dataset.

Multilinear Discriminant Analysis (MDA) [125], sometimes referred to in literature as Discriminant analysis with Tensor Representation (DATER), has been proposed as a multilinear generalization of PCA and LDA. The motivation came from the core idea of 2D-PCA in representing objects with higher-order tensors; though, while 2D-PCA represents images with matrices, MDA takes this one step further and considers higher-order tensors in general. It is a general approach which has been primarily utilized on computer vision problems such as face recognition. MDA formulates a *discriminant tensor criterion* (DTC) and a *k-mode* optimization algorithm; they are used in tandem to maximize the between-class scatter matrix and minimze the within-class scatter matrix. It follows the same philosophy as that of LDA, with the main difference here being that instead of a vector-based metric, a tensor-based one, DTC, is used for optimization. However, one downside of this algorithm is uncertainty of convergence; namely, there are no guarantees that the optimization will find a suitable result. In addition, finding an optimal number of dimensions during the dimensionality reduction step remains an open problem.

Tensor-based methods have not only been used for regular image or video data. An example is classification of hyperspectral data done in [85], where a generalization of PCA is used to decompose tensors representing hyperspectral data, containing two spatial modes and one spectral mode. On the other hand, they are also used to represent similarities/differences between complex data such as fMRI images in [34], which are then decomposed and used in an SVD classification method. In [127] a step has been made towards treating a set of images as a sequence of matrices, rather than a set of vector merged into a matrix. The set of matrices is then approximated with matrices of lower rank to improve compression and retain only important information; however, the solution is not of a closed-form, but iterative instead, and does not guarantee convergence.

One glaring downside of most of previously mentioned methods is that tensors are not explicitly used to represent each complex data point; instead, they are mostly used as a representation for the whole set of data, or some advanced features extracted from the dataset which contain information relevant to the application domain. In the following section we cover methods that treat data points as tensors, and are of more interest to our research.

1.1.2 Representing data points with tensors

First tensor-based methods build upon general ideas dimensionality reduction and linear separation of PCA and LDA. However, as mentioned many times before, the key point of tensor-based methods is to represent individual data points as higher-order tensors, whereas most machine learning methods work only with vectors. To address this gap, however, they took some insights from fields of pattern set recognition and pattern set matching, such as *subspace methods* [17], and modified them to work for tensors. For instance, a video can be viewed as a set of images; the whole set can be decomposed with PCA and represented by its principal components, and then compared to another video by measuring similarities of their respective principal components.

This very simple concept has led to development of several methods for classification of video data, mainly for tasks of hand gesture and action recognition. Two notable examples are based

on **Canonical Correlation Analysis (CCA)** [35], named **Tensor Canonical Correlation Analysis (TCCA)** [49] and **Discriminant Canonical Correlation (DCC)** [47]. The core idea of these methods lies in extending the concept of CCA from finding canonical correlations between two sets of vectors to finding correlations between two multi-dimensional arrays, namely higher-order tensors.

Tensor Canonical Correlation Analysis (TCCA) [49] is a classic tensor-matching method applied to action recognition, which achieves the generalization of CCA by factorizing the tensors into sets of matrices from tensor modes and finding canonical correlations between them. It essentially executes 6 CCA processes, split into two categories: joint-shared-modes (tensors share two modes and CCA is executed on the remaining mode) and single-shared-modes (tensors share one mode and CCA is executed on the remaining two modes). This generates 6 sets of features, which are then selected and optimized using weak learners and AdaBoost [15] boosting algorithm. Feature selection is an important step in TCCA as features obtained from different tensor modes contain different spatio-temporal information, and are not equal in terms of discriminative power. The final selected features are classified using a nearest neighbour classifier. TCCA assumes that tensors to be matched are of same dimension or shape, a common assumption shared by many tensor matching methods. This can pose some problems for action recognition, in cases of varying video lengths, different camera size, etc., which are usually dealt with extensive preprocessing steps.

Discriminant Canonical Correlation (DCCA) [47] is an evolved version of the TCCA which utilizes a discriminative mechanism, as well as advanced hand-crafted features. After factorizing the tensors, discriminative projection matrices are learned through an algorithm analogous to to Linear Discriminant Analysis (LDA) [48]. This operation minimizes between-class correlations and maximises within-class correlations, greatly increasing the classification performance in the newly created feature space. Additionally, SIFT [62] features are extracted from the spatio-temporal tensors to significantly increase representational power and robustness to changes in scale and rotation. They use edge-based features in contrast to intensity-based features of regular grayscale image pixels, and in most applications are shown to be a better choice. DCC is a good example of a simple extension to the base method of TCCA, yet one that yields high increase in performance. TCCA and DCC are related to PCA through the fact that they compute similarity between two tensors based on respective principal components (canonical vectors) of factorized tensor modes.

Previously discussed methods, though they have a tensor-first approach, mostly rely on statistical analyses to achieve good recognition performance. This approach is completely in line with traditional pattern recognition methods such as PCA and LDA. However, an argument could be made that tensor representation allows for a geometric interpretation of subspaces obtained from their decompositions. Such approach could yield more nuanced analysis of individual factors of each tensor mode.

This interpretation is explored by several methods, with the key idea of viewing subspaces as elements of special manifolds, Stiefel and Grassmann manifolds. These methods include **Tangent Bundle (TB)** [73, 70], **Product Grassmann Manifold (PGM)** [74, 72] and **n-mode Generalized Difference Subspace (n-mode GDS)** [25].

Tangent Bundles (TB) [73, 70] is a method which explores the geometrical interpretation of tensors. Its key idea is that a data tensor is related to a tangent bundle on a special manifold such as Stiefel or Grassmannian. A tensor is decomposed using High-Order Singular Value Decomposition (HOSVD) into a set of matrices; these matrices are then projected onto a space tangent to a special

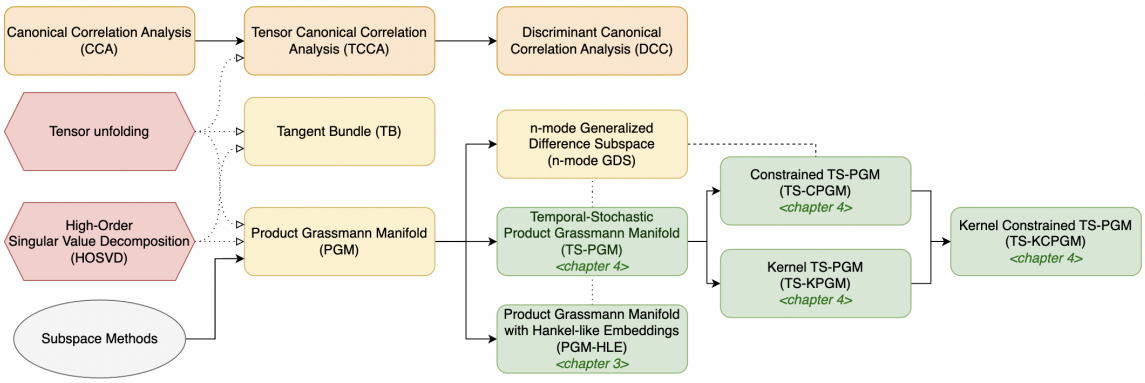

Figure 1.3: A map of related concepts and methods. We position our proposed methods (green) alongside tensor (orange) and manifold (yellow) based methods.

manifold, through the manifold charting operation. To perform classification, an intrinsic distance metric based on tangent bundles is used in a nearest neighbour fashion; there is no training required.

A similar approach using HOSVD is **Product Grassmann Manifold (PGM)** [74, 72]. In PGM, the key idea is to represent tensors as single points on non-Euclidean space called product Grassmann manifold. Tensors are decomposed into modes, each belonging to a factor Grassmann manifold. These factor Grassmannians are unified through a product manifold termed Product Grassmann Manifold. This results in a mapping of tensor structures to respective points on PGM. Thus, further analysis can be performed using geodesic distance, which is intrinsic to the product manifold. A least squares regression is defined on the product manifold, using its geodesic distance, and is used for classification purposes.

Following the same idea of PGM representation and expanding it with discriminative ability, **n-mode GDS** [25] has been developed. The key idea here is to apply a Generalized Difference Subspace (GDS) [18, 19] projection just before unifying factor manifolds on the product manifold. The projection is done for each manifold separately; it orthogonalizes subspaces created from tensor modes, removes overlap between them and thus increases discriminative capability of the representation. Similar to some other tensors methods that rely on tensor factorization, the authors of n-mode GDS added a weighing mechanism to tensor based on a modified Fisher criterion. They have developed a Fisher criterion which factors tensors instead of vectors, as well as an optimization algorithm for such criterion. Then, they are able to learn the a set of weights which achieves best classification performance on a certain dataset.

The downside of PGM and n-mode GDS is that it does not take into account the underlying nature of tensor data; namely, it does not distinguish spatial from temporal information, and treats them in the same way. In applications where sequential context is crucial, such as ordering of frames in a video, this loss of temporal information potentially leads to a degradation in performance.

The work proposed in this thesis is heavily inspired by previously discussed methods. However, we would like to position our methods at the intersection of tensor, manifold and subspace based methods, depicted on Figure 1.3. Like the above-mentioned tensor-based and manifold-based methods, our proposed method is rooted in the tensor decomposition. However, we do not utilize

CCA like TCCA and DCC do; instead, we perform our analysis on the product manifold. While PGM and n-mode GDS also use product manifolds to achieve good results on tasks such as gesture and action recognition, they treat all tensor modes equally. On the other hand, in our proposed methods we explicitly model the temporal information contained within the spatio-temporal tensor, and consider it as another factor manifold. Though many methods rely on advanced feature extraction, we do not; our primary goal is not to achieve the best possible performance in a specific task, but rather to develop a generalized representation technique for spatio-temporal tensors.

1.2 Objectives

In this thesis, we aim to develop a general approach for representing and analyzing spatio-temporal tensors, particularly videos, and show their application to the task of action recognition. The proposed methods should be general enough to be extendable and used for various other applications, so long as they factor spatio-temporal tensors; however, it should not veer too far into the theoretical side so as to be very difficult to adapt to new applications. Therefore, we propose methods based on sturdy theory developed in previous work related to tensor decomposition and product manifold geometry, while taking into consideration specific characteristics of spatio-temporal tensors.

1.3 Contributions

In this section of the thesis we outline our two main contributions to the fields of tensor analysis and subspace methods.

As our first contribution, we develop a representation technique for spatio-temporal tensors called Product Grassmann Manifold with Hankel-like Embedding (PGM-HLE). This is a general representational approach which can be used for a wide variety of tasks factoring spatio-temporal tensors. We address the key issues of product Grassmann manifold representation by explicitly modeling temporal information, by first representing the temporal tensor mode with a Hankel-like matrix, and then encoding it with a subspace which we term Hankel-like Embedding. This special subspace preserves temporal information, and can be seamlessly unified with standard PGM representation. We show the utility of our representation on data exploratory tasks such as visualization and clustering of spatio-temporal tensor datasets, with a focus on human gesture and action recognition.

As our second contribution, we develop a classification framework called Temporal-Stochastic Product Grassmann Manifold (TS-PGM), with several extensions. We greatly improve the Hankel-like Embedding from our previous work and address its memory and performance issues. This is achieved by creating its randomized generalization, Temporal-Stochastic Tensor (TST) features, which can also seamlessly be used with PGM to create a good representation for spatio-temporal tensors. Then, such representation is used by the TS-PGM framework for the purposes of classification in the tasks of human gesture and action recognition. Additionally, we develop discriminative extensions to TS-PGM by using (1) kernel mapping to handle non-linearity in the data and (2) projection on generalized difference subspace (GDS) to reduce overlap between class subspaces; either extension can be used individually or in combination with the other one for maximum increase in classification performance.

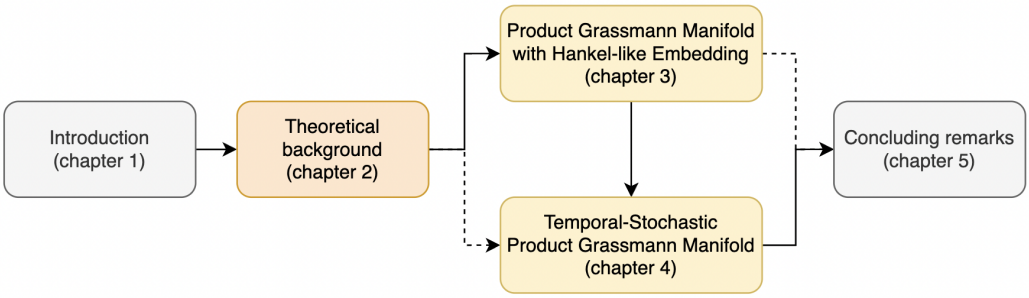

Figure 1.4: Organizational map of the thesis. For best reading experience, readers are advised to follow the path indicated by full lines: reading all chapters, starting with Chapter 1 and finishing with Chapter 5. However, Chapters 3 and 4 are written independently from each other, allowing readers to study only the one they are interested in, without significant loss of context.

In summary, we offer contributions to the field of tensor analysis by leveraging key ideas of subspace-based methods, with an explicit focus on preserving temporal information. Our proposed representational methods are evaluated in standard applications such as visualization, clustering and classification on datasets containing human gestures and actions.

1.4 Thesis organization

The rest of this thesis is organized as follows.

- In Chapter 2, we cover the theoretical concepts and tools that serve as the basis of our work: tensors and tensor unfolding; linear subspaces and their connection to Grassmann manifold; and conclude with product Grassmann manifold, unifying all previous concepts and laying the foundation for our main contributions.
- In Chapter 3 we introduce Product Grassmann Manifold with Hankel-like Embedding (PGM-HLE), a representational method for spatio-temporal tensors, along with its application to exploratory tasks such as dataset visualization and clustering.
- In Chapter 4 we introduce Temporal-Stochastic Product Grassmann Manifold (TS-PGM), a classification framework for spatio-temporal tensors, and its discriminative extensions. We then showcase these frameworks on tasks of hand gesture and action recognition.
- In Chapter 5 we conclude the thesis by summarizing the achieved results and offering suggestions for future work.

Chapter 2

Theoretical background

In this chapter we introduce the theoretical building blocks for the approaches proposed in this thesis. First, we introduce the general concept of tensors, focusing on its key attributes, and the cornerstone operation of tensor unfolding. Next, we cover basic terminology and assumptions of subspace methods, followed by their connection to manifold theory. Finally, we explore the product Grassmann manifold approach to tensor representation, and weigh its advantages and disadvantages in the context of this thesis' main objective.

The common notation used in this thesis is as follows: scalars are denoted by lowercase letters and sets are denoted by uppercase letters. Vectors and matrices are denoted by boldface lowercase and uppercase letters respectively. Tensors are represented by calligraphic letters, i.e., $\mathcal{X}$, and subspaces are denoted by script letters, i.e., $\mathscr{P}$. Given a matrix $\boldsymbol{A} \in \mathbb{R}^{w \times h}$, $\boldsymbol{A}^\top \in \mathbb{R}^{h \times w}$ represents its transposed matrix.

2.1 Tensors

Tensors are multidimensional arrays; formally, an n-th order tensor is an element of the tensor product between n vector spaces. The *order* of the tensor denotes the number of dimensions which can be indexed. It can be helpful to view higher-order tensors as generalizations of vectors and matrices, which are first-order and second-order tensors, respectively, as depicted on Figure 2.1. Often, in literature *order* is used synonymously with *mode* or *way*; however, it can cause confusion to use the terms interchangeably. Therefore, in this thesis we use *mode* to refer to a specific dimension of a higher-order tensor, i.e. first *mode* denotes the first dimension of a higher-order tensor.

Tensors are used to represent a variety of multi-dimensional data. One interesting type of tensors are spatio-temporal tensors, containing spatial and temporal information. In this thesis, we mainly focus on such spatio-temporal tensors of the third-order as the means of depicting grayscale video data. Consider a spatio-temporal tensor $\mathcal{X} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$. Dimensions (modes) of sizes d_1, d_2 and d_3 essentially represent the height, width and number of frames of said grayscale video.

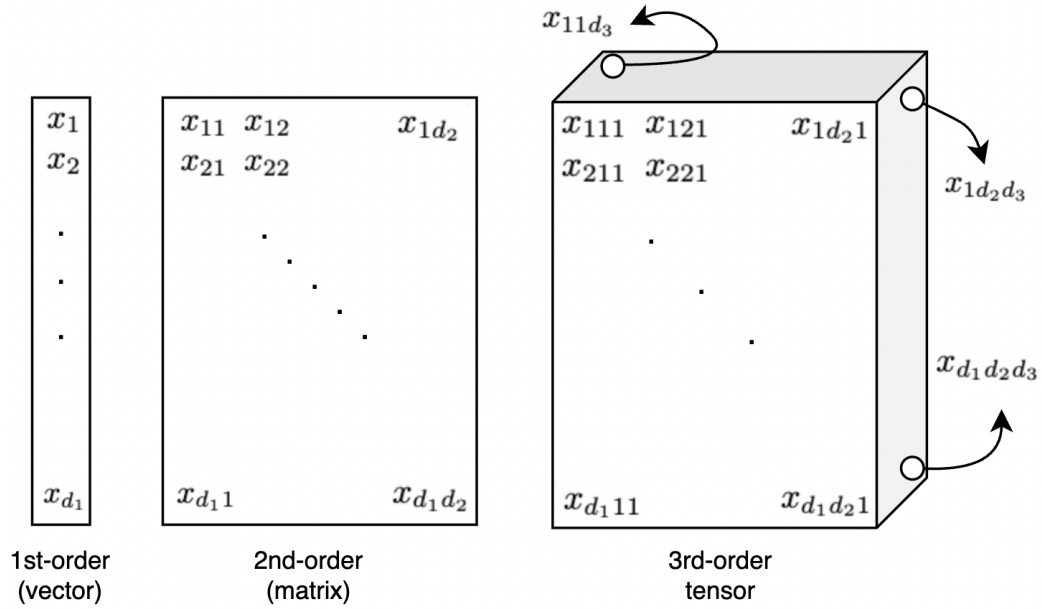

Figure 2.1: Illustration of the concept of tensor order. A first-order tensor is commonly known as a vector; likewise, a second-order tensor is known as a matrix. Third-order tensors are the simplest of higher-order tensors. Order refers to the number of indexable dimensions.

2.1.1 Tensor unfolding

An important tensor operation is tensor *unfolding*, also commonly referred to as *matricization* [51]. It is a procedure to rearrange the tensor into a matrix, reducing it along one specified mode. In essence, a tensor is sliced along one mode, and the slices are concatenated to create a matrix. In this work, we modify this procedure slightly, by adding an additional step; each of the slices is vectorized, with such vectors then forming columns of the final matrix. Usually it is important to maintain consistency in enumerating tensor modes, as well as to keep the order of slice vectorization.

Let $\mathcal{X} \in \mathbb{R}^{d_1 \times \ldots \times d_n}$ be an n-order tensor. The objective of tensor unfolding is to transform tensor $\mathcal{X}$ into a set of matrices $X = \{\boldsymbol{X}_j\}_{j=1}^{n}$. More generally, the definition is as follows: let mat_a be the matricization operator of a tensor along all its modes except mode a. This operator reshapes an n-order tensor into a matrix with d_a columns. With this, we can define matricization as follows:

$$\text{mat}_j \, \mathcal{X} = \boldsymbol{X}_j, \tag{2.1}$$

$$X = \{\boldsymbol{X}_j\}_{j=1}^{n}, \tag{2.2}$$

with set X containing mode matrices $\boldsymbol{X}_j \in \mathbb{R}^{f_j \times d_j}$, where $f_j = \prod_{k=1|k\neq j}^{n} d_k$.

To better understand the intuition behind tensor unfolding, consider the following example of a tensor $\mathcal{X} \in \mathbb{R}^{4\times 2\times 3}$, shown on 2.3.

$$\mathcal{X} \in \mathbb{R}^{4\times2\times3} = \left[\begin{bmatrix}1 & 5\\ 2 & 6\\ 3 & 7\\ 4 & 8\end{bmatrix}, \begin{bmatrix}9 & 13\\ 10 & 14\\ 11 & 15\\ 12 & 16\end{bmatrix}, \begin{bmatrix}17 & 21\\ 18 & 22\\ 19 & 23\\ 20 & 24\end{bmatrix}\right]. \tag{2.3}$$

Tensor $\mathcal{X}$ is a third-order tensor, therefore it has three modes. By applying mat operator to each individual mode, we get a set of mode matrices $X = \{\boldsymbol{X}_1 \in \mathbb{R}^{6\times4}, \boldsymbol{X}_2 \in \mathbb{R}^{12\times2}, \boldsymbol{X}_3 \in \mathbb{R}^{8\times3}\}$:

$$\boldsymbol{X}_1 \in \mathbb{R}^{6\times4} = \begin{bmatrix}1 & 2 & 3 & 4\\ 5 & 6 & 7 & 8\\ 9 & 10 & 11 & 12\\ 13 & 14 & 15 & 16\\ 17 & 18 & 19 & 20\\ 21 & 22 & 23 & 24\end{bmatrix}, \boldsymbol{X}_2 \in \mathbb{R}^{12\times2} = \begin{bmatrix}1 & 5\\ 9 & 13\\ 17 & 21\\ 2 & 6\\ 10 & 14\\ 18 & 22\\ 3 & 7\\ 11 & 15\\ 19 & 23\\ 4 & 8\\ 12 & 16\\ 20 & 24\end{bmatrix}, \boldsymbol{X}_3 \in \mathbb{R}^{8\times3} = \begin{bmatrix}1 & 9 & 17\\ 2 & 10 & 18\\ 3 & 11 & 19\\ 4 & 12 & 20\\ 5 & 13 & 21\\ 6 & 14 & 22\\ 7 & 15 & 23\\ 8 & 16 & 24\end{bmatrix} \tag{2.4}$$

For example, a video seen as a tensor of size $h \times w \times t$, where h, w and t represent height, width and number of frames, respectively, can be unfolded into $X = \{\boldsymbol{X}_1 \in \mathbb{R}^{(wt)\times h}, \boldsymbol{X}_2 \in \mathbb{R}^{(ht)\times w}, \boldsymbol{X}_3 \in \mathbb{R}^{(wh)\times t}\}$.

2.2 Subspaces

Linear vector *subspaces* of high-dimensional vector spaces are a linear algebra concept, commonly applied in pattern set recognition and dimensionality reduction problems. Its core idea lies in the premise that, when working with real world data, it is reasonable to assume that the data does not span the entirety of a high-dimensional feature space, but only a small portion. Thus, representing the pattern set with a subspace it spans not only simplifies data representation, but offers a wide range of tools for further analysis.

Subspace representation has been widely used in the field of pattern recognition, particularly for the problem of pattern *set* recognition. A pattern set of m patterns, namely d-dimensional feature vectors $\boldsymbol{X} \in \mathbb{R}^{d\times m}$ can be compactly represented by a linear subspace $\mathscr{P}$. spanned by its orthonormal basis vectors $\boldsymbol{P} \in \mathbb{R}^{m\times n}$. Because the size of a subspace n is much smaller than the feature space d, a compact representation is achieved, while preserving most of the variance that explains the underlying set of data. Such subspaces can then be compared through various metrics for the task of pattern analysis.

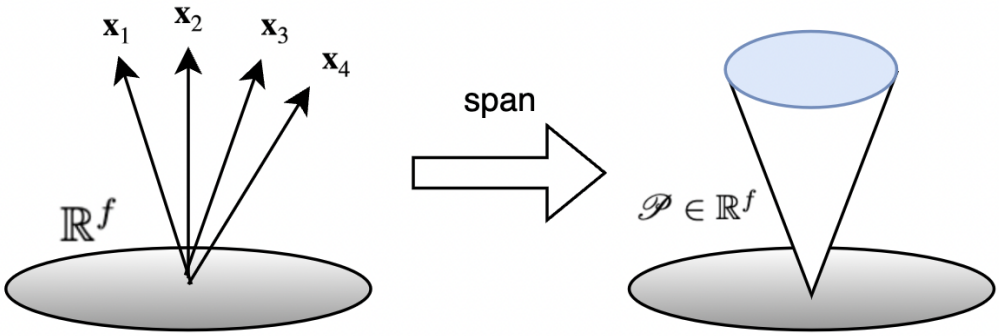

Figure 2.2: A set of vectors in a vector space $\mathbb{R}^f$ spans a subspace $\mathscr{P}$ of that vector space. This model is widely used for various pattern recognition tasks.

2.2.1 Subspace construction

There are several methods to construct a subspace from a pattern set in the form of a data matrix. A common approach is to perform *principal component analysis* (PCA) on the pattern set, and take the principal components as basis vectors of a subspace. Consider again an m-sized pattern set of d-dimensional feature vectors arranged into a matrix $\boldsymbol{X} \in \mathbb{R}^{d \times m}$, with feature vectors forming the columns of said matrix.

PCA is done by applying singular value decomposition (SVD) on the autocorrelation matrix of $\boldsymbol{X}$:

$$\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top = \boldsymbol{X}\boldsymbol{X}^\top. \tag{2.5}$$

Matrix $\boldsymbol{U}$ contains eigenvectors in its columns, while matrix $\boldsymbol{\Lambda}$ contains their matching eigenvalues in the diagonal, sorted in the descending order. The final step is to select n eigenvectors corresponding to n highest eigenvalues, and arrange them into the subspace basis matrix $\boldsymbol{P} \in \mathbb{R}^{d \times n}$.

In most subspace-based methods, the number of subspace dimensions n is considered a tuneable hyperparameter. This means that there is usually no way to determine the value which yields best recognition performance, as it can vary based on application, type of data, etc. Therefore, the best value is usually determined by hyperparameter tuning.

However, there exists a useful heurestic which can speed up this process. Eigenvectors corresponding to highest eigenvalues explain most of the variance of the considered pattern set. A contribution rate $\mu(n)$ can be calculated, which measures the ration of total variance contained in n eigenvectors:

$$\mu(n) \leq \frac{\sum_{i=1}^{n} \lambda_i}{\sum_{i=1}^{r} \lambda_i}, \tag{2.6}$$

where r is the rank of $\boldsymbol{X}\boldsymbol{X}^\top$. If multiplied by 100%, $\mu(n)$ signifies the percentage of variance contained in the subspace. Using Eq. 2.6 it is possible to control the trade-off between compactness and representational power of the subspace. The lower n is, the more compact subspace is; however, it might not contain enough information for discriminative analysis. A good starting point for

parameter search is to select n so that it explains about 90% of variance, and then adjust according to the performance on a given task and dataset.

2.2.2 Similarity between subspaces

For the purposes of classification and other discriminatory analyses, comparison of two subspaces is conduced through a similarity defined as minimal canonical angles between two subspaces [18]. Consider subspaces $\mathscr{P}$ and $\mathscr{Q}$, spanned by basis vectors $\boldsymbol{P}$ and $\boldsymbol{Q}$ respectively. Their canonical angles $\{0 \leq \theta_1, ..., \theta_m \leq \frac{\pi}{2}\}$ can be computed by SVD as:

$$\boldsymbol{P}^\top \boldsymbol{Q} = \boldsymbol{U}_p \boldsymbol{\Sigma} \boldsymbol{U}_q. \tag{2.7}$$

$\boldsymbol{U}_p$ and $\boldsymbol{U}_q$ are the canonical vectors, and $\boldsymbol{\Sigma} = \text{diag}(\lambda_1, \ldots, \lambda_k)$ is a diagonal matrix with k singular values $\{\lambda_l\}_{l=1}^k$.

The canonical angles $\{\theta_l\}_{l=1}^k$ can be obtained as $\{\arccos \lambda_l\}_{l=1}^m$. Thus, we can define the similarity between subspaces $\mathscr{P}$ and $\mathscr{Q}$ as:

$$s(\mathscr{P}, \mathscr{Q}) = \frac{1}{k} \sum_{l=1}^{k} \cos^2 \theta_l. \tag{2.8}$$

The values of similarity function 2.8 are bounded between 0 and 1. If the value of similarity is 0, then the two subspaces are completely orthogonal; on the other hand, if the value is 1, then the two subspaces completely overlap.

2.2.3 Grassmann Manifold and Subspace Representation

A Grassmann manifold, or Grassmannian, $\mathbb{G}(k, f)$ can be defined as a set of all k-dimensional subspaces of $\mathbb{R}^f$. It is a smooth and compact manifold, and allows for a geometric interpretation of subspaces that form it. For example, a k-dimensional subspace $\mathscr{P}$ can be viewed as a single point on the *Grassmann manifold* $\mathbb{G}(k, f)$. In such an interpretation, the similarity between subspaces based on canonical angles is related to the *geodesic distance* between their corresponding points on the manifold. Concretely, they are inversely correlated; a *dis*similarity based on canonical angles corresponds to geodesic distance on the Grassmannian. An example of this interpretation is depicted on Figure 2.3.

2.2.4 Product Grassmann Manifold

Tensor $\mathcal{X}$ can be decomposed to achieve a compact subspace representation using an *n-mode SVD*, also known as *High-Order SVD (HOSVD)*. First, tensor $\mathcal{X}$ is unfolded into a set of mode matrices $\{\boldsymbol{X}_j\}_{j=1}^n$. SVD is then executed on every $\boldsymbol{X}_j$ in the following manner:

$$\boldsymbol{U}_j \boldsymbol{\Lambda}_j \boldsymbol{U}_j^\top = \boldsymbol{X}_j \boldsymbol{X}_j^\top. \tag{2.9}$$

Depending on the dimensions of $\boldsymbol{X}_j$, SVD can also be performed efficiently as $\boldsymbol{V}_j \boldsymbol{\Lambda}_j \boldsymbol{V}_j^\top = \boldsymbol{X}_j^\top \boldsymbol{X}_j$, with $\boldsymbol{U}_j$ and $\boldsymbol{V}_j$ being related as $\boldsymbol{X}_j = \boldsymbol{U}_j \boldsymbol{\Lambda}^{\frac{1}{2}} \boldsymbol{V}_j^\top$. Finally, the equation of HOSVD is as follows:

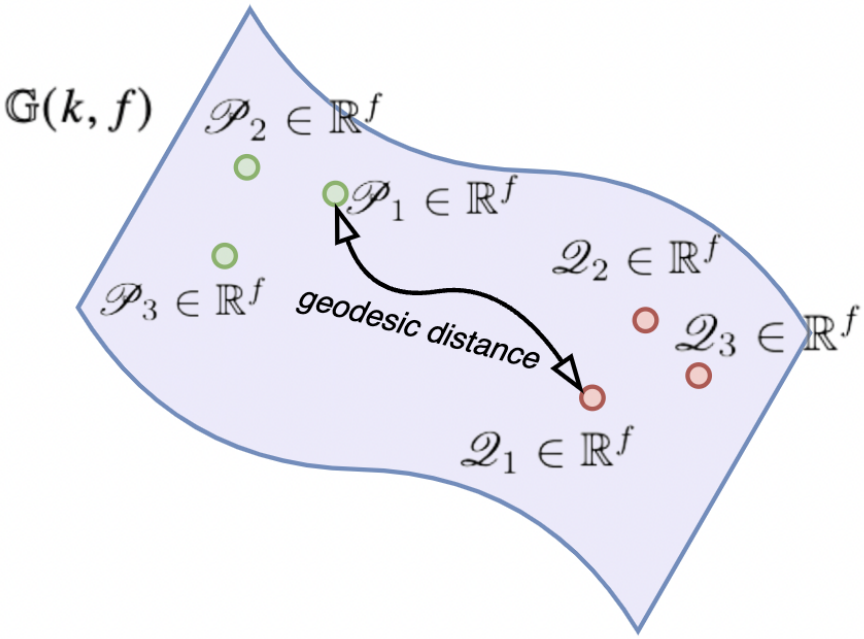

Figure 2.3: Subspaces can be interpreted as points on Grassmannian $\mathbb{G}(k, f)$. The distance between these points is *geodesic distance*; this distance is defined by the canonical angless between two subspaces.

$$\mathcal{X} = \mathcal{C} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 ... \times_n \boldsymbol{U}_n, \tag{2.10}$$

where $\times_i, i \in 1, \dots n$ denotes the Cartesian product operator. In this decomposition, core tensor $\mathcal{C}$ contains values that can be viewed as a multilinear correlation between the columns of the orthogonal factor matrices $\{\boldsymbol{U}_j\}_{j=1}^n$, which contain the singular vectors for each unfolded matrix $\boldsymbol{X}_j$.

Then, selecting subspaces from each vector space spanned by the eigenvectors $\boldsymbol{U}_j \in \{\boldsymbol{U}_j\}_{j=1}^n$ results in a set of subspaces $S_p = \{\mathscr{P}_j\}_j^n$. These subspaces are spanned by basis vectors $\{\boldsymbol{P}_j\}_j^n$, where $\boldsymbol{P}_j \in \mathbb{R}^{f_j \times k_j}$, containing k_j eigenvectors corresponding to the highest k_j eigenvalues, selected from $\boldsymbol{U}_j$ and $\boldsymbol{\Lambda}_j$ respectively. Thus, the tensor $\mathcal{X}$ is mapped to a set of mode subspaces S_p.

Every $\mathscr{P}_j$ can be considered as a point on a factor Grassmannian $M_j(k_j, d_j)$, where k_j and d_j are dimensions of subspace $\mathscr{P}_j$ and feature space of mode j, respectively. To unify these subspace representations, a product Grassmann manifold (PGM) can be constructed from a set of factor manifolds $M = \{M_j\}_{j=1}^n$ using the Cartesian product:

$$M_{pgm} = M_1 \times_1 ... \times_n M_n = (\mathscr{P}_1, ..., \mathscr{P}_n,). \tag{2.11}$$

Therefore, each tensor is represented as a point on M_{pgm}, enabling classification on the product manifold via a metric defined on this space. The geodesic distance between two points on the manifold is considered as a natural choice of dissimilarity due to its utilization of the manifold

surface itself [2]. Canonical angle-based similarity between tensors $\mathcal{X}$ and $\mathcal{Y}$, represented by sets of subspaces $\{\mathscr{P}_j\}_{j=1}^n$ and $\{\mathscr{Q}_j\}_{j=1}^n$, is defined as:

$$\rho(\mathcal{X}, \mathcal{Y}) = \frac{1}{n}\sqrt{\sum_{j=1}^{n} s(\mathscr{P}_j, \mathscr{Q}_j)^2}. \tag{2.12}$$

The similarity as defined in Eq. 2.12 is bounded between 0 and 1. Thus, a simple distance metric $d(\mathcal{X}, \mathcal{Y})$ on the PGM corresponds to $d(\mathcal{X}, \mathcal{Y}) = 1 - \rho(\mathcal{X}, \mathcal{Y})$; if the distance between two tensors on PGM is 0, they are identical; if it is 1, they are as far away as possible from each other.

Chapter 3

Product Grassmann Manifold with Hankel-like Embedding

In this chapter we describe our proposed method for representing tensors containing temporal information, based on Product Grassmann Manifold (PGM) and specially introduced temporal encoding. We evaluate our approach on hand gesture and action recognition datasets as exemplars of temporal tensor datasets, aiming to show the versatility of our representation.

The background of the proposed method is discussed in Section 3.1. The details of the proposed method are explained in Section 3.2, with experimental results shown in Section 3.3.

3.1 Background

Data exploration and representation has been at the forefront of problems tackled by the pattern recognition community for a long time. Increase of multi-dimensional data sources, such as various sensors and cameras, has been followed by a steady rise of representation techniques. Their common aim is to naturally mold and express rich information contained in the data in a unified way. Often, the first step in any research project is visualizing the given dataset and discovering natural patterns in the data. Such representation then enables analysis via visualisations, clustering and classification, allowing for a deeper insight into the structure of a given dataset. In this chapter we consider representation of multi-dimensional data with temporal information, and their visualization and clustering as first steps of data exploration.

As mentioned previously in this thesis, a representation method of particular interest is the *tensorial* form, where the nature of multi-dimensional data is preserved without changing its inherent structure. Concretely, a single data point is represented as a single tensor. In addition to the simplicity of singular representation, this approach brings the benefits of established multi-linear algebra, allowing for efficient computations on tensor data. Applications such as hand gesture and action recognition [54, 14, 41], medical data analysis and imaging [75, 130, 84, 119, 42, 76], multi-spectral imaging [44, 55, 45, 8, 43] and others [92, 11] may benefit from tensor representation.

One way to efficiently represent a tensor as a single data point is representation on the product Grassmann manifold (PGM) [72], rooted in tensor unfolding [51] and subspace representation [122, 80, 18]. An n-dimensional tensor can be unfolded along each of its dimensions, generating

n modes [51]. With each mode essentially presenting a unique look into the data contained within the tensor, by analyzing separate modes it is possible to extract information inaccessible by other means. For example, a video can be viewed as a 3-dimensional tensor containing two spatial and one temporal mode, and each mode can be compactly represented by a respective linear subspace it spans.

A Grassmann manifold (GM) is defined as a set of linear subspaces of same dimension, and can be geometrically interpreted as a surface where these subspaces are points on the manifold. Therefore, there is a GM corresponding to each of the tensor modes. A single manifold expresses geometrical relations between subspaces of the same mode via geodesic distance, enabling discriminative analysis within the manifold [113, 103, 102, 100, 33, 30].

However, utilizing information from each mode in a unified manner requires the construction of a PGM from distinct factor (mode) manifolds [71, 23]. Analysis can then be performed on the PGM in a similar vein, taking into account that chordal distance between points on PGM is equivalent to the Cartesian product of geodesic distances on respective mode Grassmannians [71, 32]. The downside is that representing all tensor modes in the same way can lead to some loss of mode-specific information. For videos, this means potential loss of discriminative temporal features, as linear subspaces do not fully preserve sequence information [25].

We address the lack of explicit handling of temporal information, by exploiting the fact that subspace representations of data on any number of factor manifolds can be used on the PGM. To this end, we model the temporal mode with a Hankel-like matrix, which can then be encoded with a sequence-preserving linear subspace and incorporated with regular tensor mode representations via PGM. We refer to this representation as *product Grassmann Manifold with Hankel-like embedding* (PGM-HLE), shown on Fig. 3.1. The idea of Hankel-like representations is strongly motivated by its recent applications for sequential data, as investigated in the literature [96, 101, 24, 56, 107].

Further analysis of tensors on PGM-HLE is done in the context of chordal distance between two points on the manifold [27, 104], a metric that unifies distances between subspaces within a single factor Grassmannian [7]. In this way, we provide valuable temporal context at a minimal increase in computational complexity. Besides, our approach demonstrates usability of specialized representations via PGM.

Our main contributions are summarized as follows:

1. We introduce product Grassmann Manifold with Hankel-like embedding (PGM-HLE), a temporal tensor representation method based on tensor decomposition and Hankel-like matrix.

2. Next, we show the benefit of using PGM-HLE through visualizations of basis vectors for specific modes.

3. In addition, we demonstrate a simple use-case of tensor dataset visualization with t-SNE and PGM-HLE.

4. Finally, we evaluate spectral clustering on PGM-HLE with hand gesture and action recognition datasets.

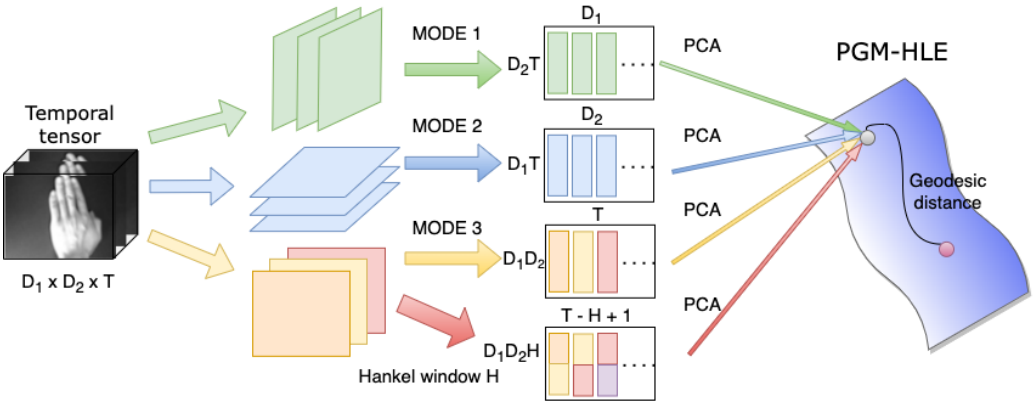

Figure 3.1: A temporal tensor can be unfolded along dimensions D_1, D_2 and T to generate mode features. Hankel-like matrix is created from temporal mode by applying a sliding window of size H to preserve sequential information. Unified tensor representation is created on product Grassmann manifold with Hankel-like embedding (PGM-HLE).

3.2 Proposed method

In this section, we describe the proposed tensor representation on PGM-HLE. An elementary overview of the method is shown on Figure 3.1, with a more detailed look provided in Figure 3.2. First, we formulate the general problem of tensor representation. Next, we explain the n-mode tensor representation of spatio-temporal features via linear subspaces and then introduce Hankel-like embedding of temporal modes. We describe our full tensor representation on the PGM with Hankel-like embedding. Finally, we show how PGM-HLE enables distance and similarity-based algorithms, with examples of visualization and clustering algorithms such as t-SNE and spectral clustering (SC).

Notation is as follows. Scalars are denoted by lowercase letters and sets are denoted by uppercase letters. Vectors and matrices are denoted by boldface lowercase and uppercase letters respectively. Calligraphic letters denote tensors and script letters denote subspaces. Given a matrix $\boldsymbol{A} \in \mathbb{R}^{w \times h}$, $\boldsymbol{A}^\top \in \mathbb{R}^{h \times w}$ denotes its transpose.

3.2.1 Basic idea

For simplicity, in this work, multi-dimensional data points are regarded as 3-mode tensors $\mathcal{X}$ of size $d_1 \times d_2 \times d_t$, where the mode of size d_t is a temporal mode. However, the proposed approach can be generalized to temporal tensors with n modes. In its raw form, the data can be rather unwieldy and uninformative. Therefore we compactly represent a tensor $\mathcal{X}$ with a set of *mode subspaces*. This representation has multiple advantages: it allows parallel processing, and helps finding correlations among various factors inherent in each mode.

We formulate the tensor representation problem as follows: Let $X = \{\mathcal{X}_i\}_{i=1}^n$ be a dataset of n tensors. In addition, let $T(\mathcal{X})$ be a transformation of a tensor in its raw form to its representation on PGM-HLE. Finally let $\rho(\mathcal{X}, \mathcal{Y})$ be the similarity function between two tensors $\mathcal{X}$ and $\mathcal{Y}$ defined by

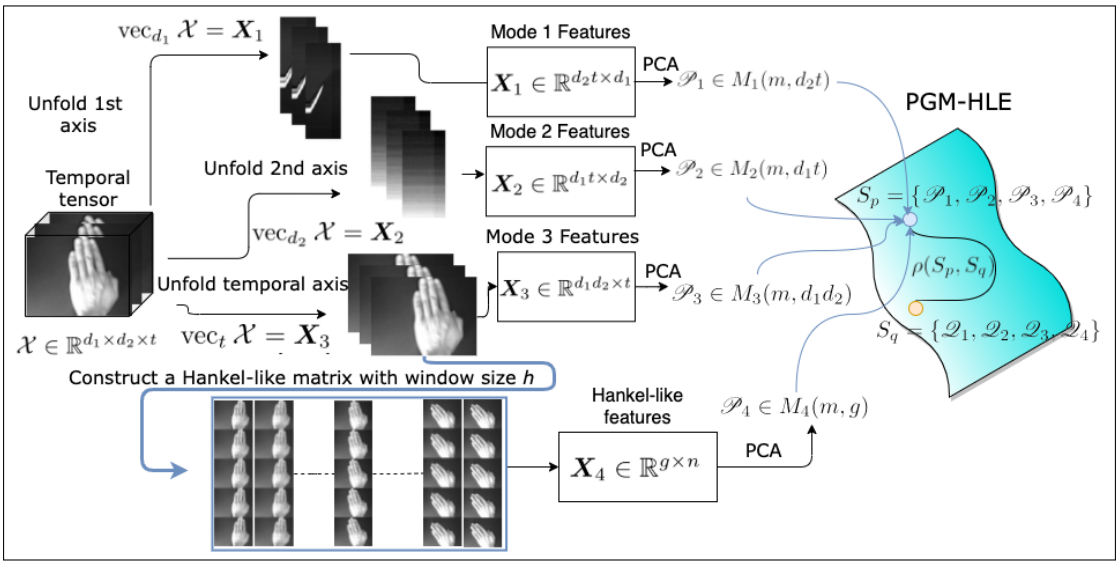

Figure 3.2: A greyscale tensor $\mathcal{X}$ is unfolded to generate mode features $\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3$, and Hankel-like matrix $\boldsymbol{X}_4$. non-centered PCA is performed to generate a set of subspace basis vectors. Tensors $\mathcal{X}$ and $\mathcal{Y}$, represented by sets of subspaces S_p and S_q are compared using geodesic distance $\rho(S_p, S_q)$.

geometric properties of PGM-HLE.

We consider optimization problems where we minimize a function F dependent on some sort of distance $d(\mathcal{X}, \mathcal{Y})$ between tensors $\mathcal{X}$ and $\mathcal{Y}$. This can be written as:

$$\min F(d(\mathcal{X}, \mathcal{Y})), \tag{3.1}$$

where $\mathcal{X}, \mathcal{Y} \in X = \{\mathcal{X}_i\}_{i=1}^{n}$. We aim to create a transformation $T(\mathcal{X})$ which provides similarity $\rho(\mathcal{X}, \mathcal{Y})$ as an interface for solving Eq. (3.1) with $d(\mathcal{X}, \mathcal{Y}) = 1 - \rho(\mathcal{X}, \mathcal{Y})$.

3.2.2 n-mode Tensor Representation with Linear Subspaces

Let $\mathcal{X} \in \mathbb{R}^{d_1 \times d_2 \times t}$ be a 3-dimensional tensor, where t represents the temporal dimension. Tensor $\mathcal{X}$ is unfolded into a set of matrices $X = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3\}$, a process known as *matricization*. Consider a video as a tensor of size $t \times h \times w$, where w, h and t are width, height and number of frames, respectively. This tensor is unfolded into $X = \{\boldsymbol{X}_1 \in \mathbb{R}^{(wt) \times h}, \boldsymbol{X}_2 \in \mathbb{R}^{(ht) \times w}, \boldsymbol{X}_3 \in \mathbb{R}^{(wh) \times t}\}$, with each mode representing concatenated slices along a specific tensor dimension. Therefore, $\mathcal{X}$ can be decomposed to achieve a compact subspace representation using *n-mode SVD* [51], defined as:

$$\mathcal{X} = \mathcal{C} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3. \tag{3.2}$$

Core tensor $\boldsymbol{C}$ contains values analogous to eigenvalues of *SVD*, while matrices $\{\boldsymbol{U}_j\}_{j=1}^{n=3}$ contain the singular vectors for each unfolded matrix $\boldsymbol{X}_j$, and expression $\boldsymbol{U}_j \boldsymbol{\Lambda}_j \boldsymbol{U}_j^\top = \boldsymbol{X}_j \boldsymbol{X}_j^\top$ holds.

Subsequently, we select a subspace spanned by eigenvectors $\boldsymbol{U}_j$, resulting in a set of subspaces $S_p = \{\mathscr{P}_j\}_j^{n=3}$, spanned by basis vectors $\{\boldsymbol{P}_j\}_j^{n=3}$, where $\boldsymbol{P}_j \in \mathbb{R}^{f_j \times m_j}$, containing m_j eigenvectors

corresponding to the highest m_j eigenvalues. Different m_j can be selected, as each mode exhibits different properties and levels of information density. In summary, tensor $\mathcal{X}$ is mapped to a set of mode subspaces S_p. This offers a compact representation and an opportunity to analyze each tensor mode independently. Both spatial and temporal information are captured by first and second mode unfoldings. However, as *SVD* does not inherently preserve sequence information, some temporal features may be lost from the temporal mode unfolding.

3.2.3 Hankel-like Embedding of Temporal Modes

It is possible to preserve sequential features from temporal tensor modes within a linear subspace. Consider the temporal mode of tensor $\mathcal{X} \in \mathbb{R}^{d_1 \times d_2 \times t}$:

$$\text{mat}_t\, \mathcal{X} = \boldsymbol{X}_t = \boldsymbol{X}_3 = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t]. \tag{3.3}$$

A *Hankel-like* matrix $\boldsymbol{H}$ can be created by applying a sliding window of size h over the columns of matrix $\boldsymbol{X}_3$. This creates a set of n lagged frame sequences comprised of h frames, arranged into matrix $\boldsymbol{H} \in \mathbb{R}^{g \times n}$ as follows:

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \ldots & \boldsymbol{x}_{t-h+1} \\ \boldsymbol{x}_2 & \boldsymbol{x}_3 & \ldots & \boldsymbol{x}_{t-h+2} \\ \vdots & & \ddots & \vdots \\ \boldsymbol{x}_h & \boldsymbol{x}_{h+1} & \ldots & \boldsymbol{x}_t \end{bmatrix}. \tag{3.4}$$

The number of columns of matrix $\boldsymbol{H}$ is determined by the total length of sequence $[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t]$, given by relationship $n = t - h + 1$, and the number of rows is $g = h \times f_t$. In our method, h is considered a hyperparameter.

In temporal mode $\boldsymbol{X}_3$, the ordering of columns carries sequence information, which is lost when applying *SVD* to create a subspace representation. However, H preserves temporal information by embedding it in its columns. We then construct a compact subspace representation in the following manner:

$$\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top = \boldsymbol{H}\boldsymbol{H}^\top. \tag{3.5}$$

$\boldsymbol{U} \in \mathbb{R}^{g \times n}$ contains eigenvectors as columns and $\boldsymbol{\Lambda} \in \mathbb{R}^{n \times n}$ is a diagonal matrix with n eigenvalues. Basis vectors of a subspace are obtained by selecting m eigenvectors $\boldsymbol{P}_4 = [\boldsymbol{v}_1, ..., \boldsymbol{v}_m]$ corresponding to the m highest eigenvalues. Resulting subspace $\mathcal{Q}_4$ spanned by $\boldsymbol{P}_4$ exhibits sequence-preserving qualities in each of the eigenvectors, which are generalized to the whole subspace. We call this representation Hankel-like embedding (HLE). Additionally, HLE is fully compatible with subspaces modeling spatio-temporal data from 3.2.2.

3.2.4 Product Grassmann Manifold with Hankel-like Embedding

By using the two approaches described in subsections 3.2.2 and 3.2.3, a temporal tensor $\mathcal{X} \in \mathbb{R}^{d_1 \times d_2 \times t}$ is represented by a set of subspaces $S_p = \{\mathcal{P}_j\}_{j=1}^4$, containing mode subspaces $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}_3$ and the HLE subspace $\mathcal{P}_4$. Every $\mathcal{P}_j$ is a point on a Grassmann manifold $M_j(m_j, d_j)$, where

m_j and d_j are dimensions of subspace $\mathscr{P}_j$ and feature space respectively. A unified representation is constructed on product Grassmann manifold from a set of factor manifolds $M = \{M_j\}_{j=1}^4$ as follows:

$$M = M_1 \times M_2 \times M_3 \times M_t = (\mathscr{P}_1, \mathscr{P}_2, \mathscr{P}_3, \mathscr{P}_4), \tag{3.6}$$

where $\times$ denotes Cartesian product. Therefore, each spatio-temporal tensor is represented as a single point on M. Further data analysis is possible using a metric defined on PGM, namely the geodesic distance between two points on the PGM. This is a natural choice of dissimilarity due to its utilization of the manifold surface [2].

To define geodesic distance on M, we first define geodesic distances between points (subspaces) on factor manifolds. These distances are parametrized in terms of canonical angles between subspaces, defined as minimal angles between two subspaces [18]. Given subspaces $\mathscr{P}$ and $\mathscr{Q}$ and their basis vectors $\boldsymbol{P}$ and $\boldsymbol{Q}$, canonical angles $\{0 \leq \theta_1, ..., \theta_m \leq \frac{\pi}{2}\}$ can be computed by SVD as:

$$\boldsymbol{P}^\top \boldsymbol{Q} = \boldsymbol{U}_p \boldsymbol{\Sigma} \boldsymbol{U}_q. \tag{3.7}$$

$\boldsymbol{U}_p$ and $\boldsymbol{U}_q$ contain canonical vectors, and $\boldsymbol{\Sigma} = \text{diag}(\kappa_1, \ldots, \kappa_r)$ is a diagonal matrix with m singular values $\{\kappa_l\}_{l=1}^m$. Canonical angles $\left\{\theta_l\right\}_{l=1}^m$ can be obtained as $\left\{\cos^- 1(\kappa_l)\right\}_{l=1}^m$. Similarity between subspaces $\mathscr{P}$ and $\mathscr{Q}$ is then defined as:

$$s(\mathscr{P}, \mathscr{Q}) = \frac{1}{m} \sum_{l=1}^{m} \cos^2 \theta_l. \tag{3.8}$$

Using this similarity in each factor manifold, we define the similarity on PGM. Tensors $\mathcal{X}$ and $\mathcal{Y}$ represented by sets of subspaces $\{\mathscr{P}_j\}_{j=1}^4$ and $\{\mathscr{Q}_j\}_{j=1}^4$ is defined as:

$$\rho(\mathcal{X}, \mathcal{Y}) = \frac{1}{n} \sqrt{\sum_{j=1}^{n=4} s(\mathscr{P}_j, \mathscr{Q}_j)^2}. \tag{3.9}$$

As individual similarities $s(\mathscr{P}_j, \mathscr{Q}_j)$ are bounded between 0 and 1, division by number of factor manifolds n is introduced to maintain same bounds for final similarity metric. This enables the conversion of similarity to distance as $d(\mathcal{X}, \mathcal{Y}) = 1 - \rho(\mathcal{X}, \mathcal{Y})$. Having defined the similarity between points on PGM, it is possible to conduct further analysis of tensor datasets by considering their layout in the manifold space. We explore an example of dataset visualization and clustering in 3.2.5 and 3.2.6 respectively.

3.2.5 Multilinear t-SNE

t-SNE [112] is a well known data visualization algorithm able effectively map multi-dimensional data to two or three dimensions, while preserving both global and local structure. This is achieved by representing distances between data points as probabilities under Gaussian distribution, and minimizing Kullback-Leibler divergence between joint probability distributions A and B in high-dimensional and low-dimensional spaces respectively. The following Eq. (3.10) represents this cost function and is optimized using gradient descent:

$$C = KL(A||B) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{3.10}$$

Consider two data points x_i and x_j in high-dimensional space, and their low-dimensional mappings y_i and y_j. Here, p_{ij} is the probability of choosing x_j as a closely-related neighbour of x_i under a Gaussian distribution centered on x_i. Analogously, q_{ij} is the same probability with respect to y_i and y_j. Probability p_{ij}, given by the following equation:

$$p_{ij} = \frac{\exp(-||x_i - x_j||^2 / 2\sigma^2)}{\sum_{k \neq i} \exp(-||x_k - x_i||^2 / 2\sigma^2)}, \tag{3.11}$$

is calculated explicitly as part of t-SNE algorithm. It assumes Euclidean distance between points x_i and x_j, which is defined for multi-dimensional vectors. However, if data points x_i and x_j are instead multi-linear tensors $\mathcal{X}_i$ and $\mathcal{X}_j$, the Euclidean distance is not defined, and Eq. (3.11) cannot be solved. Therefore, t-SNE cannot be utilized to visualise temporal tensor datasets.

We have provided an interface to solve this problem in 3.2.4, by introducing a similarity function between two tensors based on the PGM with Eq. (3.9). Thus, we modify Eq. (3.11) in the following manner:

$$p_{ij} = \frac{\exp(-d(\mathcal{X}_i, \mathcal{X}_j) / 2\sigma^2)}{\sum_{k \neq i} \exp(-d(\mathcal{X}_k, \mathcal{X}_i) / 2\sigma^2)}, \tag{3.12}$$

where $d(\mathcal{X}, \mathcal{Y}) = 1 - \rho(\mathcal{X}, \mathcal{Y})$. By using Eq. (3.12) it is possible to optimize the Kullback-Leibler divergence as defined in Eq. (3.10) for datasets with temporal tensor datasets and make appropriate visualizations with t-SNE.

3.2.6 PGM-HLE with spectral clustering

A variety of clustering algorithms relies on constructing symmetric matrices based on pairwise distances or similarities between all data points in the dataset. One notable example is spectral clustering [78]. Given a set of points $S = \{s_1, s_2, \ldots, s_n\} \in \mathbb{R}^l$, the first step in spectral clustering is to construct an affinity matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, where $\boldsymbol{A}_{ij} = \exp(-||s_i - s_j||^2 / 2\sigma^2)$, $i \neq j$ and $A_{ii} = 0$. However, if we consider a set of multilinear tensors $X = \{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n\}$ as a set of data points, the similarity defined in Eq. (4.13) can be used to rewrite the formation of affinity matrix to:

$$\boldsymbol{A}_{ij} = \rho(\mathcal{X}_i, \mathcal{X}_j). \tag{3.13}$$

Having constructed the affinity matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ using Eq. (3.13), spectral clustering algorithm proceeds with standard steps, without modifications, as defined in [78]. A major advantage of PGM-HLE representation is that via Eq. (4.13) it provides an interface for using various algorithms on tensor datasets, otherwise defined only for vector-valued data.

3.3 Experimental Results

In this section we cover experimental results involving PGM-HLE. First, we describe datasets used in the experiments. Next, we showcase the representational power of PGM-HLE in two ways: (1)

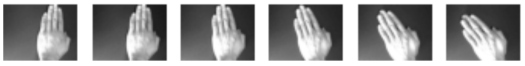

Figure 3.3: Uniformly sampled frames of temporal mode from CMB dataset.

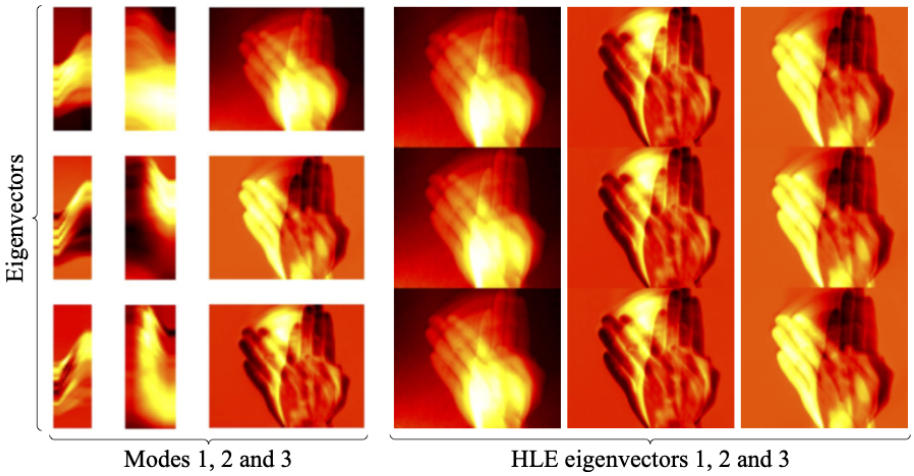

Figure 3.4: Subspace basis vectors of tensor modes 1, 2, 3 and Hankel-like embedding, exhibiting different form and information.

by visualizing the distribution of samples in a hand gesture dataset and (2) by evaluating a clustering algorithm on hand gesture and action recognition datasets.

3.3.1 Datasets

We use **Cambridge Hand Gesture** [49] (CMB) and **UT-Kinect** [124] (UT) datasets in visualization and clustering experiments. **CMB** is a benchmark dataset for hand gesture recognition, and comprises 9 classes with 100 videos per class. It is divided equally into five sets based on illumination settings. We perform clustering experiments on each set, and on the entire dataset to evaluate robustness against illumination change. **UT** contains skeleton data of 10 action classes, performed by 10 subjects in front of a Kinect device. There are 200 sequences in total. The data contains x, y and z coordinates of 20 skeleton joints.

Tensors in **CMB** dataset are of shape $h \times w \times t$, where h, w and t stand for height, width and sequence length respectively, while in **UT** they are of shape $c \times j \times t$, where c, j and t stand for coordinate, joint and sequence length respectively. As all tensors have variable temporal dimension t, we resize them to $12 \times 16 \times 30$ (CMB) and $3 \times 20 \times 20$ (UT). We use video data as it is easy to visualize and interpret subspace basis vectors, as well as evaluate formed clusters.

3.3.2 Visualizations of Hankel-like subspaces

As videos are simple to interpret, we can visualize eigenvectors spanning subspaces of tensor modes, including the Hankel-like embedding (HLE) of temporal mode. This provides insight into

information contained within these representations, as well as grounds for further interpretability. An example tensor $\mathcal{X}$ from **CMB** dataset, a closed hand moving to the left, is shown on 3.3.

We then decompose tensor $\mathcal{X}$ into three modes, obtain a Hankel-like matrix, perform non-centered PCA and construct respective subspaces. On 3.4 we depict first three eigenvectors of each mode. It can be clearly seen that each mode carries different information, Modes 1 and 2 are somewhat difficult to interpret, but they contain different spatio-temporal information depicting movement of a hand from left to right. Eigenvectors of mode 3 are very similar to eigenvectors of Hankel-like subspace, with the latter being almost a concatenation of the former. However, in 3.3.4 we show the effect of this information on clustering accuracy.

3.3.3 Tensor visualization on PGM-HLE

To investigate the effectiveness of PGM-HLE on **CMB** dataset, we use t-SNE on 1) baseline vectorized representation of tensors, 2) subspaces of individual modes, including the HLE and 3) on the PGM-HLE, and compare these visualizations. As t-SNE is not a deterministic algorithm, we run it 10 times and pick the one with lowest KL value [123]. Results are depicted on 3.5.

Baseline setting 1) results in the worst visualization, with a high KL divergence score of 0.371, while setting 3) achieves the best, with the lowest KL score of 0.208. Modes 1-4 in setting 2) showcase that each mode carries information of different characteristics and quality with respect to separability, with respective KL scores of 0.213, 0.271, 0.282 and 0.242. For example, mode 2 seems capable of separating all classes of 'contract' shape, mode 1 successfully extracts rightward movements and mode 3 'flat' hand shapes.

HLE appears very similar to temporal mode 3, which is expected due to underlying temporal information of both representations. However, separability between clusters in HLE is somewhat higher and easier to notice. For example, HLE is able to group samples of classes 'v-rightward', 'v-leftward' and 'spread-rightward', while improving on the separability of all 'flat' hand shapes. This indicates that there might be some merit in utilizing information from HLE. Finally, it can be clearly seen that PGM-HLE produces superior results in terms of separability and cluster interpretability in addition to lowest KL score.

3.3.4 Spectral clustering on TS-PGM

To investigate contributions of different tensor modes on clustering accuracy, we use spectral clustering (SC), a simple and fast algorithm. Both **CMB** and **UT** datasets contain labels, which we use to evaluate the accuracy as defined as in [26]. In short, cluster class is determined by the labels of majority members, and accuracy is defined as number of correctly clustered data points divided by number of total samples. Results are presented in 3.1.

Clustering performance differs across tensor modes. On the entire **CMB** dataset, modes 1 and 3 perform similarly at *75.55%* and *75.44%*, with mode 2 performing worse at *69.77%*. Performances vary in subsets of **CMB**, most likely due to different illumination settings affecting spatio-temporal features. All three modes significantly outperform the baseline at *18.80%*, indicating valuable information contained within them. Furthermore, HLE performs the best compared to individual modes on **CMB**, offering noticeable improvement. In **UT** dataset mode 1 outperforms other two modes and the baseline at *80.90%*. Unlike **CMB**, the nature of modes 1 and 2 is harder to interpret

Dataset	Baseline	M1	M2	M3	HLE	PGM	PGM-HLE
CMB S1	32.94%	88.88%	87.77%	84.44%	*95.00%*	97.22%	**98.33%**
CMB S2	16.88%	74.44%	80.00%	78.88%	*83.88%*	86.11%	**88.33%**
CMB S3	21.44%	74.44%	70.55%	72.77%	*78.88%*	81.11%	**82.22%**
CMB S4	27.55%	74.44%	71.11%	63.33%	*76.66%*	80.00%	**83.88%**
CMB S5	28.61%	78.88%	81.66%	80.55%	*84.44%*	86.66%	**88.33%**
Cambridge	18.80%	75.55%	69.77%	75.44%	*83.11%*	84.88%	**86.66%**
UT-Kinect	61.30%	*80.90%*	62.31%	65.32%	77.38%	92.46%	**93.96%**

Table 3.1: Spectral clustering results on CMB and UT. All tensor modes outperform the baseline. HLE works better than regular modes 1, 2 and 3, except on UT. PGM-HLE outperforms all baselines with 86.66% (CMB) and 93.96% (UT).

due to the structure of skeletal data. However, it is noticeable that HLE significantly improves the performance of temporal mode from *65.32%* to *77.38%*.

It worth noting that M1 provides superior accuracy compared to HLE when PGM and PGM-HLE are not available. This behavior may happen when some of the classes present very similar shapes where the ordering of the observations over time does not define their semantics.

Unifying information from different tensor modes consistently improves accuracy in all cases, shown on PGM and PGM-HLE performances. In PGM [72], a tensor is represented as a point on product Grassmann manifold, and serves as a baseline for the idea of unifying tensor modes. PGM-HLE offers additional context by utilizing specialized encoding of temporal information, and the improvement is consistent across all datasets.

3.4 Summary

In this chapter we introduced a method for representing temporal tensors based on established multilinear algebra. We use the PGM geometry to naturally unify representations of tensor modes and Hankel-like embedding of temporal information and apply the geodesic distance to investigate the relationship between temporal tensors. We further demonstrate the use of geodesic distance as an general interface for solving optimization and clustering problems.

Using this interface, we performed t-SNE visualizations and spectral clustering of temporal tensor datasets containing video and skeletal data, giving some weight to the strategy of unified representation on PGM, special treatment of temporal information via Hankel-like embeddings and finally the idea of geodesic distance as a general interface for solving various problems. Specifically in the context of video datasets, this approach may prove valuable as it allows simple and fast analysis of data in its raw form, without the need for significant data pre-processing or pre-training of heavy representational models.

As potential future research steps, we will consider several directions. First would be to evaluate the proposed method on various types of multilinear data, such a relational and signal data. Specifically, we believe that unified representation on PGM would be effective in utilizing side information in addition video data, such as sound information, movement information via gyroscope signals,

etc. Secondly, we plan to investigate different Riemannian manifolds in order to leverage their different characteristics and similarity metrics that they provide, as potential improvements to the representational aspect of our method. Lastly, a potential future direction includes extending the proposed method to consider applying kernel trick to handle potential non-linearity in tensor modes and other data.

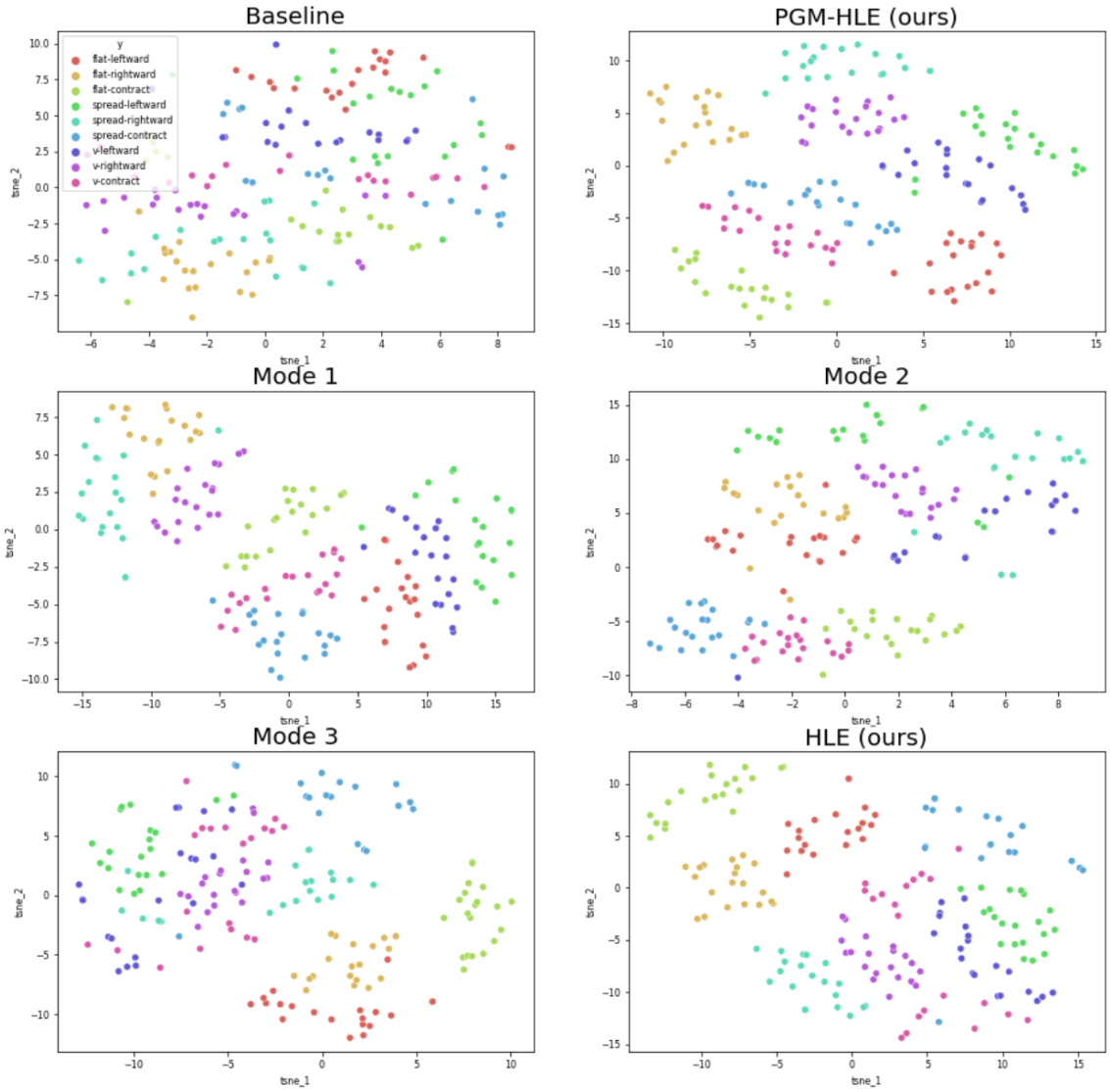

Figure 3.5: t-SNE visualizations of baseline vectorized representation, regular tensor modes, HLE and PGM-HLE for Cambridge Hand Gesture dataset. This dataset combines three hand shapes - 'flat', 'spread' and 'v' with three movements - 'leftward', 'rightward' and 'contract'. Baseline is very cluttered. Modes and HLE offer different perspectives and provide high discrimination among the clusters. PGM-HLE has the best defined visualization obtained by unifying those perspectives.

Chapter 4

Temporal-Stochastic Product Grassmann Manifold

In this chapter we propose a classification method for action recognition, based on PGM representation and greatly improved temporal encoding, which we evaluate on hand gesture, action and skeleton action recognition tasks.

The background of the proposed method is discussed in Section 4.1. The details of the proposed method are explained in Section 4.3, with experimental results shown in Section 4.4.

4.1 Background

Previous work has shown the merit of representing tensors on the PGM. Further, we have demonstrated in the previous chapter the value of specialized temporal encoding and unifying it with the PGM. However, we have used a naive, though well-established approach based on Hankel-like matrix to achieve this, that might not be best suited for classification problems requiring efficient discriminative mechanisms.

Multi-dimensional data are used in a variety of tasks, such as video processing, medical and hyperspectral image analysis, internet data processing, recommendation systems and many others [44, 75, 97, 130, 84, 11, 119, 92, 42, 76]. As the complexity of such data increases, better techniques are needed to accurately and efficiently represent the richness of the information contained therein. Commonly, representation methods are often tailored to be task specific, thus enabling optimization for a particular application. For example, video processing may require development of algorithms that treat the data as a series of frames in order to identify correlates. The disadvantages of such approaches are that optimized algorithms may not be generalizable to other tasks.

An alternative involves representing the data as tensors. Tensors, which can be regarded as generalizations of matrices, are widely used in fields where complex multi-dimensional data needs to be treated in a unified way (e.g. mechanics, electrodynamics, neural networks). Representing the data in these more abstract terms opens up the opportunity for the application of well-defined mathematical operations, in particular those from multilinear algebra, allowing for more efficient computation. In addition, these operations can then be applied to different kinds of tensor data, in direct contrast to more restricted bespoke algorithms.

In this chapter, we build on a method of tensor representation that is based on multilinear algebra [74, 72]. This approach, which we refer to here as Product Grassmann Manifold (PGM), starts with a unified representation of multi-dimensional data-points as tensors, applies a number of multilinear algebraic transformations to decompose the tensors, before finally reunifying the representation. The resulting representations are more compact, by over 90% [72, 73, 25], and the algorithm is simple to implement and computationally efficient.

In PGM, tensors undergo three mathematical operations: (1) tensor unfolding [51]; (2) subspace representation [122, 80]; and (3) unification on the Product Grassmann manifold (PGM) [72]. In **tensor unfolding** an n-dimensional tensor is reorganized along each of its dimensions to generate n modes, a process also known as *matricization*. For example, video data can be viewed as a 3-dimensional tensor containing two spatial and one temporal mode. The analysis of the modes can reveal information inaccessible when viewed in its original tensor form.

Subspace representation is then applied to each tensor mode, replacing the original mode data with the respective linear subspace which it spans. While retaining the essence of the original data, this representation is significantly more compact. These subspaces can also be viewed as points on Grassmann Manifolds (GM), which are defined as a sets of linear subspaces of the same dimension. In this case, each of the tensor modes corresponds to a single factor Grassmann Manifold. These factor manifolds express geometrical relations between points. The similarity/dissimilarity of the points can then be calculated through the geodesic distance on the manifold, allowing for discriminative analysis to be performed within the manifold [113]. At this stage, however, representations of each mode are still disconnected.

In the final step, individual factor manifolds are merged. The geodesic distances on respective factor manifolds can be unified as their Cartesian product [71, 23, 32]. This allows for discriminative analysis on what is known as the **Product Grassmann Manifold** (PGM) [74, 71]. As a method, PGM, thus, decomposes tensors to reveal information, represents it more compactly, and finally unifies those representations. However, one downside of the PGM approach is that all tensor modes are treated and represented equally. In the case of videos, for example, because linear subspaces do not fully preserve sequence information, this may result in the loss of discriminative temporal features [25].

In this chapter, we propose (1) a method that incorporates into PGM the explicit handling of temporal information, and (2) two extensions of this method. Our proposed method takes advantage of the fact that the Product Grassmann Manifold can comprise any number of manifolds [117]. In order to accurately represent tensor sequence information, we employ a procedure similar to Randomized Time Warping (RTW) [107]. It involves randomly sampling ordered data points on a temporal tensor mode, and constructing a low-dimensional subspace. This representation, which we refer to as Temporal-Stochastic Tensor (TST) features, is treated as an additional manifold and reduces the loss of discriminative temporal features. In this way, we provide valuable temporal context at a minimal increase in computational complexity. We refer to this approach as Temporal-Stochastic PGM (TS-PGM). In our application to video data, TS-PGM has four factor Grassmann Manifolds, three originating from linear subspaces of tensor modes, and one from specialized Temporal-Stochastic (TS) subspaces. On this unified representation, we perform classification using a modified Mutual Subspace Method (MSM) [39, 77]. Specifically, in order to account for the curvature of the manifold, we modify MSM by replacing the canonical angle based [7] similarity function with geodesic distance calculated on the TS-PGM [27, 104, 2]. An overview of the proposed

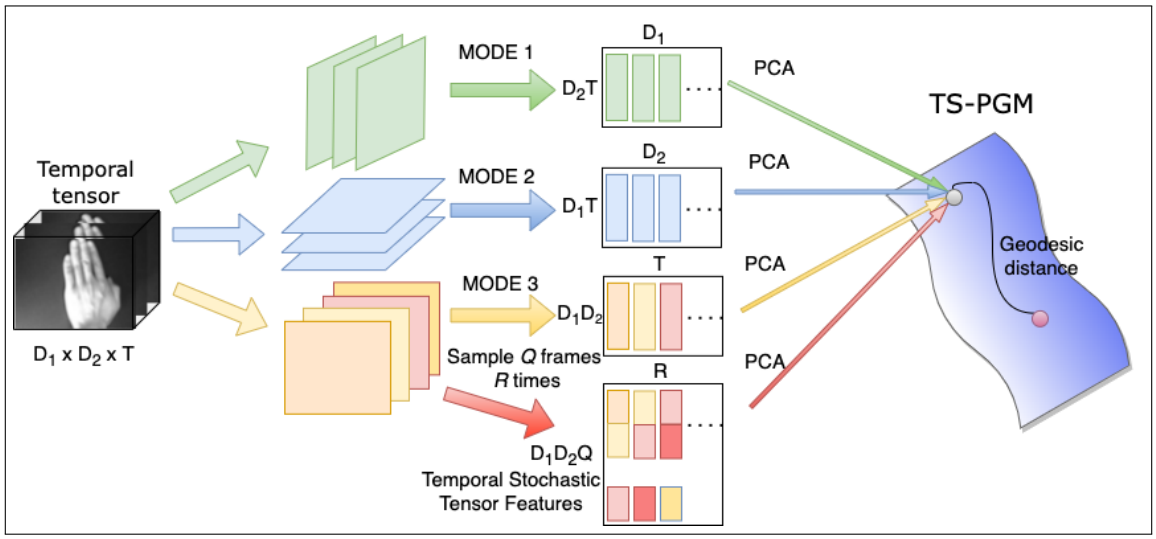

Figure 4.1: Overview of proposed method. A temporal tensor can be unfolded along dimensions D_1, D_2 and T to generate mode features. Temporal Stochastic Tensor (TST) features are created from the temporal mode by randomly sampling Q frames R times. This generates a feature matrix capable of preserving sequential information. Unified tensor representation is created on the temporal-stochastic product Grassmann manifold (TS-PGM).

method is shown in Figure 4.1.

We extend the TS-PGM method to address two drawbacks. Firstly, because subspaces are generated independently of each other, they are not necessarily optimized for classification. To alleviate this problem, before the use of TS-PGM, we project subspaces onto a Generalized Difference Subspace (GDS) [18, 19], which has a quasi-orthogonalization property. This projection method is known as n-mode GDS [25] when used on n-dimensional tensor data. Secondly, TS-PGM relies on linear subspaces which cannot accurately represent non-linear data distributions. To address this problem, we utilize a kernel mapping and construct subspaces in kernel space [21, 20, 79, 88], which can be used both with TS-PGM and its discriminative extension that relies on n-mode GDS projection.

As videos are a common exemplar of temporal tensors, we evaluate the proposed methods on motion and action recognition datasets. Our main contributions are summarized as follows:

1. We introduce specialized Temporal-Stochastic Tensor features to explicitly represent and preserve temporal tensor information.

2. We show the benefit of this representation and its integration with regular tensor modes via Temporal-Stochastic PGM.

3. We employ n-mode GDS projection to enhance discriminative capability of TS-PGM.

4. We apply kernel mappings to TS-PGM to handle non-linear data.

5. We evaluate TS-PGM and its enhancements on motion and action recognition datasets, outperforming traditional subspace methods.

The rest of the paper is organized as follows. The next section describes related work. Then, Section 4.3 describes the proposed method in detail, including the creation of TST features, construction of TS-PGM and the classification flow. In Section 4.4, we briefly introduce datasets used and describe experiments and achieved results. Finally, in Section 4.5, we conclude and offer suggestions for future work.

4.2 Related Work

Subspace-based methods have been extensively used to solve pattern-set problems, including video classification, by representing pattern-sets with low-dimensional linear subspaces. Most notable subspace-based method of this kind, originally developed for handwritten character recognition and face recognition, is Mutual Subspace Method (MSM) [77]. In a concrete classification problem, a linear subspace is constructed for each class from training data; providing reference models for these classes. An input pattern-set is represented as a subspace in the same manner. The MSM then classifies the input pattern-set by computing similarities between input and reference (class) subspaces based on multiple canonical angles. This concept has since been extended in various ways for different applications.

One such notable extension is Randomized Time Warping (RTW) [107], a method for hand gesture and action recognition. RTW generalizes a classic algorithm for time-series analysis, Dynamic Time Warping (DTW), by substituting the problem of computing similarities between sequences with similarities between subspaces. This is done through repeated and constrained random sampling of feature vectors from a given video sequence, which are then modeled by a subspace. Then, classification is performed either in an MSM-like manner using the canonical angles, or by treating subspaces as points on a Grassmann manifold and applying Grassmann Discriminant Analysis (GDA) [31].

An interesting alternative direction in video classification, closely related to our work, is explored by tensor-based methods such as Discriminant Canonical Correlation (DCC) [47] and Tensor Canonical Correlation Analysis (TCCA) [49]. Both methods take on a tensor-first approach to human gesture and action recognition, and rely on tensor decomposition and the concept of Canonical Correlation Analysis (CCA) [35]. TCCA extends the definition of CCA to multi-dimensional arrays such as tensors, by factorizing them into sets of matrices, before creating sets of projection matrices between input and reference tensors, such that they maximize their mutual canonical correlations.

DCC is an evolved version of the previous method which incorporates discriminative information learned through an algorithm analogous to Linear Discriminant Analysis (LDA) [48] between factorized matrices of a single tensor themselves. In addition, DCC utilizes SIFT features extracted from the spatio-temporal tensors to significantly increase performance.

A family of methods based on manifolds and product manifolds has been designed for tensors with spatio-temporal information, specifically videos. Tangent Bundle [73] is a method that relies on High-Order Singular Value Decomposition (HOSVD) to factorize the tensor; resulting matrices are projected onto a tangent space, by charting them on a special manifold, such as Stiefel or Grassmann manifolds. Classification is then performed as a nearest neighbour using intrinsic distance on those

manifolds. A similar approach using HOSVD is done in Product Grassmann Manifold (PGM) [74, 72]. In PGM, tensors are decomposed into modes, each belonging to a factor Grassmannian manifold. Then, each tensor is represented as a point on the PGM, where points can be compared using the geodesic distance. Such representation on the PGM is used to formulate a least squares regression based on the geodesic distance, and is used for classification purposes.

The PGM representation is further endowed with discriminative ability using Generalized Difference Subspace (GDS) [18, 19] in a method called n-mode GDS [25]. Here, right before unification on the PGM, resultant modes are projected onto a GDS, which orthogonalizes subspaces, thus decreasing overlap and increasing discriminative capability. A variant of this method, n-mode weighted GDS (n-mode wGDS), additionally includes learning the weights which dictate the contribution of each mode to the PGM representation. This is done with an optimization algorithm using an extended definition of Fisher's score [25].

Our proposed method is heavily inspired by subspace representation in MSM and RTW, as well as their classification flow; however, we handle multiple reference subspaces per class, and discrimination is done on the product manifold. While RTW assumes the use of an explicit discriminative algorithm through GDA, our proposed method relies only on the geodesic distances generated during the training stage. Although we have been inspired by the sampling concept introduced by RTW, our approach is less constrained and is adapted to work with tensor data.

Like the above-mentioned tensor-based and manifold-based methods, our proposed method is rooted in the tensor decomposition. However, we do not utilize CCA like TCCA and DCC do; instead, we perform our analysis on the product manifold. While PGM and n-mode GDS also use product manifolds to achieve good results on tasks such as gesture and action recognition, they treat all tensor modes equally. On the other hand, in our proposed method we explicitly model the temporal information contained within the tensor.

It is worth mentioning that our method is designed for general recognition tasks involving spatio-temporal tensors. Therefore, unlike most recognition methods, it does not rely on extraction of features such as optical flow, SIFT, SURF, intensity gradients, or deep learning features such as a specialized skeleton encoding; nor does it use a learning-based approach to classification. This is because, while those features can be very effective, our primary goal is not to achieve the best possible performance in a specific task, but rather to develop a generalized representation technique for temporal tensors.

4.3 Proposed method

Similarly to Section 2.2.4, we model a single multi-dimensional data point with a single n-mode tensor $\mathcal{X}$, represented as a set of *mode subspaces*. Then we address the tensor classification problem in two steps: 1) we represent individual tensors by their n-mode subspaces and 2) we classify tensors by the strategy of nearest neighbor (1-NN) by using the similarity between these n-mode representations.

In this section we describe the proposed TS-PGM. First, we formulate the problem of tensor classification. Next, we explain the n-mode Spatio-Temporal Tensor (STT) representation, followed by the introduction of our main idea - Temporal-Stochastic Tensor (TST) features. Finally, we describe our classification algorithm based on these representations on the Product Grassmann

Manifold (PGM). In addition, we also briefly discuss computational complexity of our proposed methods.

We use the following notation: scalars are denoted by lowercase letters and sets are denoted by uppercase letters. Vectors and matrices are denoted by boldface lowercase and uppercase letters respectively. Tensors are represented by calligraphic letters, i.e., $\mathcal{X}$, and subspaces are denoted by script letters, i.e., $\mathscr{P}$. Given a matrix $\boldsymbol{A} \in \mathbb{R}^{w \times h}$, $\boldsymbol{A}^\top \in \mathbb{R}^{h \times w}$ represents its transposed matrix.

4.3.1 Problem Formulation

In this chapter, we choose to model a single multi-dimensional data point with a single n-mode tensor $\mathcal{X}$. As tensors are complex data structures, it may be difficult to analyze the data in its raw form. Therefore, in the context of our framework, we opt to represent tensors in the compact form of a set of *mode subspaces*. This approach has several advantages, such as parallel processing of tensor modes and reduced storage footprint due to subspace representation. Additionally, it enables correlation analysis among various factors inherent in respective modes. We formulate the tensor classification problem in two steps. First, we represent individual tensors by their n-mode subspaces. Then, we classify tensors by the strategy of nearest neighbor (1-NN) by using the similarity between these n-mode representations.

More formally, the problem of classification can be defined as follows: let $\{(\mathcal{X}_i,\ y_i)\}_{i=1}^{m}$ be a set of m reference tensors. All tensors have n modes, and are of sizes $d_1 \times d_2 \times \ldots \times d_{n-1} \times d_n$, with $d_n = d_t$ indicating the dimension of the temporal mode. Each tensor $\mathcal{X}_i$ is paired with a respective label $y_i \in \mathbb{C} = \{1, \cdots, l\}$. Given a novel query tensor $\mathcal{Y}$, we want to correctly predict it as belonging to one of the categories $\mathbb{C}$.

4.3.2 Spatio-temporal Tensor Features

In this subsection, we explain the n-mode tensor representation. Let $\mathcal{X} \in \mathbb{R}^{d_1 \times \ldots \times d_{n-1} \times d_n}$ be an n-mode tensor with temporal dimension $d_n = d_t$. The tensor $\mathcal{X}$ is unfolded into a set of matrices $X = \{\boldsymbol{X}_j\}_{j=1}^{n}$, a process known as *matricization* or *unfolding* [51]. For example, consider a video seen as a tensor of size $h \times w \times t$, where h, w and t represent height, width and number of frames, respectively. This video is unfolded into $X = \{\boldsymbol{X}_1 \in \mathbb{R}^{(wt) \times h}, \boldsymbol{X}_2 \in \mathbb{R}^{(ht) \times w}, \boldsymbol{X}_3 \in \mathbb{R}^{(wh) \times t}\}$. In essence, each mode represents concatenated slices along a specific tensor dimension.

More generally, the definition is as follows: let mat_a be the vectorization operator of a tensor along all its modes excepted mode a. This operator reshapes an n-mode tensor into a matrix with a columns. For simpler indexation, let $d_n = d_t$. With this, we can define matricization as follows:

$$\text{mat}_{d_j}\, \mathcal{X} = \boldsymbol{X}_j, \tag{4.1}$$

$$X = \{\boldsymbol{X}_j\}_{j=1}^{n}, \tag{4.2}$$

with set X containing mode matrices $\boldsymbol{X}_j \in \mathbb{R}^{f_j \times d_j}$, where $f_j = \prod_{k=1|k\neq j}^{n} d_k$.

The set of modes X can be decomposed to achieve a compact subspace representation using an *n-mode SVD*, also known as *High-Order SVD (HOSVD)*. Thus, a tensor $\mathcal{X}$ is decomposed in the following manner:

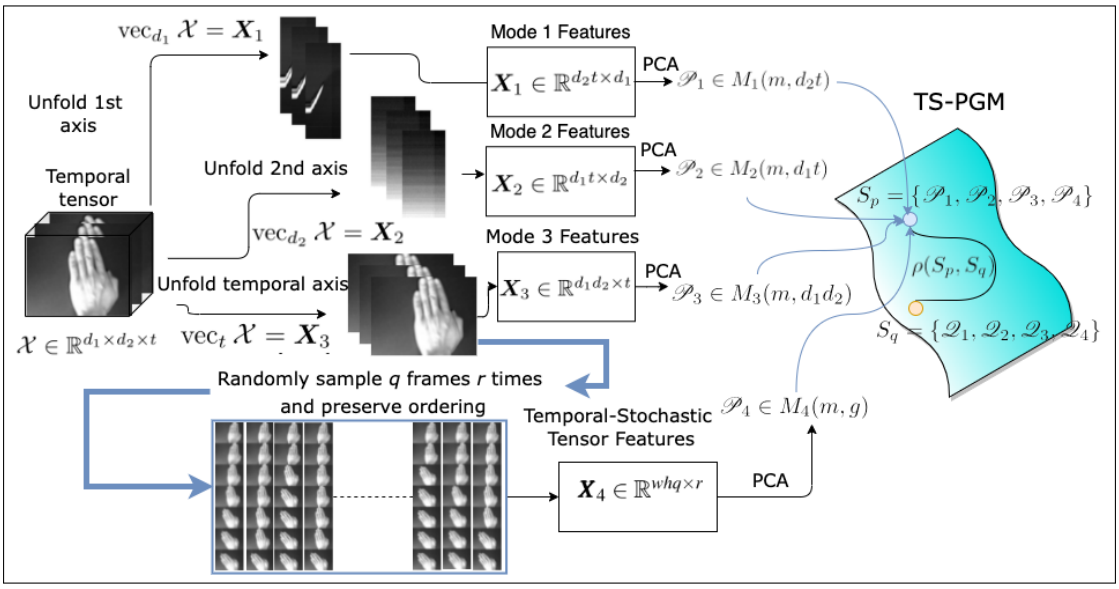

Figure 4.2: Detailed overview of proposed method. A grayscale tensor $\mathcal{X}$ is unfolded using matricization to generate mode features $\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3$. From $\boldsymbol{X}_3$, TST features $\boldsymbol{X}_4$ are generated by sampling q frames r times. On each feature set, PCA is performed to generate a set of subspaces P. Tensors $\mathcal{X}$ and $\mathcal{Y}$, represented by P and Q can then be compared using geodesic distance $\rho(P, Q)$.

$$\mathcal{X} = \mathcal{C} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \ldots \times_n \boldsymbol{U}_n, \tag{4.3}$$

where $\times_i, i \in 1, \ldots n$ denotes the Cartesian product operator. In this decomposition, core tensor $\mathcal{C}$ contains values that can be viewed as a multilinear correlation between the columns of the orthogonal factor matrices $\{\boldsymbol{U}_j\}_{j=1}^n$, which contain the singular vectors for each unfolded matrix $\boldsymbol{X}_j$, i.e.:

$$\boldsymbol{U}_j \boldsymbol{\Lambda}_j \boldsymbol{U}_j^\top = \text{corr}(\boldsymbol{X}_j), \tag{4.4}$$

$$\text{corr}(\{\boldsymbol{x}_{j,i}\}) = \sum_i \boldsymbol{x}_{j,i} \boldsymbol{x}_{j,i}^\top \tag{4.5}$$

Here, corr is a correlation matrix between all columns $\{\boldsymbol{x}_{j,i}\}_i$ of the matrix $\boldsymbol{X}_j$ with respect to an inner product. For ease of calculation, $\text{corr}(\boldsymbol{X}_\text{j}) = \boldsymbol{X}_\text{j} \boldsymbol{X}_\text{j}^\top$ in the simplest case..

We then select subspaces from each vector space spanned by the eigenvectors $\boldsymbol{U}_j \in \{\boldsymbol{U}_j\}_{j=1}^n$ resulting in a set of subspaces $S_p = \{\mathscr{P}_j\}_j^n$. These subspaces are spanned by basis vectors $\{\boldsymbol{P}_j\}_j^n$, where $\boldsymbol{P}_j \in \mathbb{R}^{f_j \times v_j}$, containing k_j eigenvectors corresponding to the highest k_j eigenvalues, selected from $\boldsymbol{U}_j$ and $\boldsymbol{\Lambda}_j$ respectively. Note that different k_j can be selected for each mode subspace as each mode exhibits different properties and levels of information density. In summary, the tensor $\mathcal{X}$ is mapped to a set of mode subspaces S_p.

In our tensor classification framework we compute a set of reference subspaces for each class in the training (in contrast to a set of reference subspaces per sample tensor). Therefore, in the training

phase we employ a variation of the procedure above to model a set of reference tensors (belonging to the same class) with a single set of mode class subspaces. For class $c \in \mathbb{C}$ with m_c samples, we compute:

$$\boldsymbol{U}_{j,c}\boldsymbol{\Lambda}_{j,c}\boldsymbol{U}_{j,c}^{\top} = \frac{1}{m_c}\sum_{i|y_i=c}^{m} \text{corr}(\boldsymbol{X}_{j,i}). \tag{4.6}$$

The cth class subspace $\mathscr{W}_j$ is spanned by basis vectors $\boldsymbol{W}_{j,c}$ obtained from the leading k eigenvectors, i.e. the columns of $\boldsymbol{U}_{j,c}$. Class c is represented by a set of *mode class subspaces* $\{\boldsymbol{W}_{j,c}\}_{j=1}^{n}$.

The resulting set of mode subspaces offers a compact representation of tensor modes and an opportunity to analyze each mode independently. However, some temporal features may be lost in this representation, as a consequence of subspace extraction. We discuss this issue in more detail in the next section, as well as offering a solution by introducing Temporal-Stochastic Tensor features.

4.3.3 Temporal-Stochastic Tensor Features

For tensors containing temporal information, such as videos, applying the approach described in section 4.3.2 may not be ideal. This is due to loss of this temporal information when executing the Singular Value Decomposition (SVD) in Eq.(4.4) on the auto-correlation matrix defined in Eq.(4.5). Consider the unfolded temporal mode of a tensor $\mathcal{X}$:

$$\begin{aligned} \text{mat}_n\, \mathcal{X} = \boldsymbol{X}_n = \\ = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{d_t}], \end{aligned} \tag{4.7}$$

where mode matrix $\boldsymbol{X}_n$ is of size $f_n \times d_t$, with $f_n = \prod_{j=1}^{n-1} d_j$. Vector $\boldsymbol{x}_t \in \mathbb{R}^{f_n}$ $(t = 1, \ldots, d_t)$ is a column of $\boldsymbol{X}_n$, and can be thought of as a frame of the sequence determined by the temporal mode. Temporal information is embedded here in the ordering of columns in $\boldsymbol{X}_n$.

However, this ordering is lost when creating subspace representation of the temporal mode. The first step is to compute an auto-correlation matrix, a $f_n \times f_n$ matrix containing correlation values of each frame with itself. Then, SVD is performed on this auto-correlation matrix in Eq.(4.4). Regardless of the ordering of columns in the auto-correlation matrix, SVD produces the same eigenvectors and eigenvalues, meaning that the temporal information directly embedded into the ordering of columns is lost.

To address this issue, we introduce the Temporal-Stochastic Tensor (TST) features to extract the temporal features that the mode subspaces cannot preserve. Consider again the unfolded temporal mode $\boldsymbol{X}_n$ in Eq.(4.7) and the ordered set $S = \{\boldsymbol{x}_t\}_{t=1}^{d_t}$ containing ordered columns of $\boldsymbol{X}_n$.

A TST feature vector $\boldsymbol{s} = [z_1^{\top}, z_2^{\top}, ...z_q^{\top}]^{\top}, \boldsymbol{s} \in \mathbb{R}^g$ $(g = q f_{d_t})$ is constructed by randomly sampling q frames from the ordered set S, while preserving their original ordering. Specifically, if $t(z_i)$ (with $i = 1, \ldots, q$) is the original order of a frame, then the samples $z_1, ..., z_q \in S$ needs to satisfy the following condition: $t(z_1) < ... < t(z_q)$.

This sampling procedure is repeated r times, resulting in a set of TST feature vectors $\{\boldsymbol{s}_i\}_1^r$. The numbers of sampled frames q and repetitions r are considered hyperparameters, and, depending on the dataset, should be tuned accordingly. We compute a TST subspace to represent the distribution of TST feature vectors in the traditional subspace manner:

$$\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\top} = \text{corr}([\boldsymbol{s}_1, \ldots, \boldsymbol{s}_r]). \tag{4.8}$$

$\boldsymbol{U} \in \mathbb{R}^{g \times r}$ is a matrix with eigenvectors as columns and $\boldsymbol{\Lambda} \in \mathbb{R}^{r \times r}$ is a diagonal matrix with r eigenvalues. The basis vectors of the TST subspace are obtained by selecting k eigenvectors $\boldsymbol{P}_T = [\boldsymbol{v}_1, ..., \boldsymbol{v}_k]$ corresponding to the k highest eigenvalues. The resulting TST subspace $\mathscr{P}_n$ exhibits sequence-preserving qualities in each of the eigenvectors, which are generalized to the whole subspace. This procedure is used to map a tensor $\mathcal{X}$ to $\mathscr{P}_n$.

Analogous to Section 4.3.2, we compute a reference TST subspace for each class in the training. Using a variation of the procedure described in this section, we model a set of reference sequences, belonging to the same class, with a single TST subspace. For class $c \in \mathbb{C}$ with m_c samples, we compute:

$$\boldsymbol{U}_c\boldsymbol{\Lambda}_c\boldsymbol{U}_c^{\top} = \frac{1}{m_c} \sum_{i|y_i=c}^{m_c} \text{corr}([\boldsymbol{s}_{1,i}, \ldots, \boldsymbol{s}_{r,i}]). \tag{4.9}$$

The basis vectors $\boldsymbol{W}_{t,c}$ of cth class subspace $\mathscr{P}_{t,c}$ are obtained from the leading k eigenvectors, i.e. the columns of $\boldsymbol{V}_c$.

4.3.4 Temporal-Stochastic Product Grassmann Manifold

By using the two approaches described in previous subsections, we can map an n-dimensional temporal tensor $\mathcal{X} \in \mathbb{R}^{d_1 \times \cdots \times d_n}$ to a set of subspaces $S_p = \{\mathscr{P}_j\}_{j=1}^{n+1}$, spanned by basis vectors $\{\boldsymbol{P}_j\}_{j=1}^{n+1}$. This set is comprised of subspace representations of n Spatio-Temporal Tensor modes $\{\mathscr{P}_j\}_{j=1}^{n}$ and the TST subspace $\mathscr{P}_t$. Every $\mathscr{P}_j$ can be considered as a point on a factor Grassmannian $M_j(k_j, d_j)$, where k_j and d_j are dimensions of subspace $\mathscr{P}_j$ and feature space of mode j, respectively. To unify these subspace representations, we construct Temporal-Stochastic Product Grassmann Manifold from a set of factor manifolds $M = \{M_j\}_{j=1}^{n+1}$ by Eq. 4.10 using the Cartesian product:

$$M = M_1 \times_1 ... \times_n M_n \times_{n+1} M_{n+1} = (\mathscr{P}_1, ..., \mathscr{P}_n, \mathscr{P}_{n+1}). \tag{4.10}$$

Therefore, each temporal tensor is represented as a point on M, enabling classification on the product manifold via a metric defined on this space. The geodesic distance between two points on the manifold is considered a natural choice of dissimilarity due to its utilization of the manifold surface itself [2].

Geodesic distance on M is defined through geodesic distances between points on all factor manifolds $\{M_j\}_{j=1}^{n+1}$. In general, the geodesic distance on a single Grassmann Manifold M_i is related to a similarity parametrized in terms of the canonical angles [18] between subspaces in the following manner: consider subspaces $\mathscr{P}$ and $\mathscr{Q}$, spanned by basis vectors $\boldsymbol{P}$ and $\boldsymbol{Q}$ respectively. Their canonical angles $\{0 \leq \theta_1, ..., \theta_m \leq \frac{\pi}{2}\}$ can be computed by SVD as:

$$\boldsymbol{P}^{\top}\boldsymbol{Q} = \boldsymbol{U}_p\boldsymbol{\Sigma}\boldsymbol{U}_q. \tag{4.11}$$

$\boldsymbol{U}_p$ and $\boldsymbol{U}_q$ are the canonical vectors, and $\boldsymbol{\Sigma} = \text{diag}(\lambda_1, \ldots, \lambda_k)$ is a diagonal matrix with k singular values $\{\lambda_l\}_{l=1}^{k}$. The canonical angles $\left\{\theta_l\right\}_{l=1}^{k}$ can be obtained as $\left\{\arccos \lambda_l\right\}_{l=1}^{m}$. Thus, we can define the similarity between subspaces $\mathscr{P}$ and $\mathscr{Q}$ as:

$$s(\mathscr{P}, \mathscr{Q}) = \frac{1}{k} \sum_{l=1}^{k} \cos^2 \theta_l. \tag{4.12}$$

The similarity function on Product Grassmann Manifold M is defined by combining similarities $\{s(\mathscr{P}_j, \mathscr{Q}_j)\}_{j=1}^{n+1}$ on each of $n+1$ manifolds. Formally, the similarity between two tensors $\mathcal{X}$ and $\mathcal{Y}$, represented with sets of subspaces S_p and S_q, respectively, is defined as:

$$\rho(S_p, S_q) = \left(\sum_{j=1}^{n+1} \left(s(\mathscr{P}_j, \mathscr{Q}_j) \right)^2 \right)^{1/2}. \tag{4.13}$$

A distance is inversely related to the similarity function ρ. Finally, relying on the similarity function ρ, we can define the Temporal-Stochastic Product Grassmann Manifold (TS-PGM) classification framework for temporal tensors. An overview of the proposed method is shown on Figure 4.2. For the sake of clarity and simplicity, we illustrate TS-PGM for the case of tensors with $n = 3$ modes and size $h \times w \times t$. However, this approach can be generalized to n-mode tensors. As defined in Section 4.3.1, having a set of labeled reference tensors $\{(\mathcal{X}_i, y_i)\}_{i=1}^{m}$, with $y_i \in \mathbb{C} = \{1, \cdots, c\}$ and an input tensor $\mathcal{Y}$, the training phase consists of:

1. **The unfolding stage:** each reference tensor $\mathcal{X}_i$ is unfolded into 3 spatio-temporal modes $\boldsymbol{X}_{1,i}, \boldsymbol{X}_{2,i}$ and $\boldsymbol{X}_{3,i}$. In parallel, a set of TST features $\{\boldsymbol{s}_s\}_{s=1}^{r}$ is generated, with q frames sampled r times. For convenience, these features can be arranged into matrix $\boldsymbol{X}_{4,i}$. Therefore, a set of feature matrices $X_i = \{\boldsymbol{X}_{j,i}\}_{j=1}^{4} = \{\boldsymbol{X}_{1,i} \in \mathbb{R}^{wt \times h}, \boldsymbol{X}_{2,i} \in \mathbb{R}^{ht \times w}, \boldsymbol{X}_{3,i} \in \mathbb{R}^{wh \times t}, \boldsymbol{X}_{4,i} \in \mathbb{R}^{qhw \times r}\}$ is created for every tensor $\mathcal{X}_i$.

2. **The subspace creation stage:** for each class $c \in \mathbb{C}$, the class correlations are aggregated and a set of basis vectors $\{\boldsymbol{W}_{j,c}\}_{j=1}^{4}$ is computed according to Eq. 4.6 for spatio temporal features $\boldsymbol{X}_1$, $\boldsymbol{X}_2$ and $\boldsymbol{X}_3$ and Eq. 4.9 for TST features $\boldsymbol{X}_4$. These basis vectors span a set of reference class subspaces $S_c = \{\mathscr{W}_{j,c}\}_{j,c=1}^{4,m_c}$.

The testing phase for a query tensor $\mathcal{Y}$ consists of:

1. **The unfolding stage:** tensor $\mathcal{Y}$ is unfolded into 3 spatio-temporal modes $\boldsymbol{Y}_1$, $\boldsymbol{Y}_2$ and $\boldsymbol{Y}_3$, and the set of TST features $\{\boldsymbol{s}_s\}_{s=1}^{r}$, arranged into matrix $\boldsymbol{Y}_4$, is generated with q frames sampled r times. This results in a set of feature matrices $Y = \{\boldsymbol{X}_j\}_{j=1}^{4} = \{\boldsymbol{Y}_1 \in \mathbb{R}^{wt \times h}, \boldsymbol{Y}_2 \in \mathbb{R}^{ht \times w}, \boldsymbol{Y}_3 \in \mathbb{R}^{wh \times t}, \boldsymbol{Y}_4 \in \mathbb{R}^{qhw \times r}\}$.

2. **The input subspace creation stage:** a set of basis vectors $\{\boldsymbol{Q}_j\}_{j=1}^{4}$ is generated using Eq. 4.4 for spatio-temporal features $\boldsymbol{Y}_1$, $\boldsymbol{Y}_2$ and $\boldsymbol{Y}_3$ and Eq. 4.8 for TST features $\boldsymbol{Y}_4$, spanning a set of input subspaces $S_q = \{\mathscr{Q}_j\}_{j=1}^{4}$.

3. **The similarity calculation stage**: similarities $\{\rho(S_c, S_q)\}_{c=1}^{l}$ are calculated between the set of input subspaces S_q and each set of class subspaces W_c for each reference class $c \in \mathbb{C}$, using Eq. 4.13.

4. **The classification stage:** finally, tensor $\mathcal{Y}$ is assigned the label of class c corresponding to the label of the reference set of subspaces with highest similarity of $\{\rho\}_{i=1}^{4}$, i.e., prediction $\text{pred}(\mathcal{Y}) = \text{argmax}_c\ \rho(S_c, S_q)$.

4.3.5 Constrained TS-PGM

A major drawback of TS-PGM is that subspaces are generated independently of each other [80], meaning there is no reason to a priori assume they are optimal for classification purposes. To directly address this, Generalized Difference Subspace (GDS) has been developed [18]. It works by projecting subspaces to a special subspace (GDS) containing only the essential components for discrimination between classes. It has also been utilized for tensor analysis [25].

Given a set of c (≥ 2) n-dimensional reference class subspaces $\{\mathscr{P}_i\}_{i=1}^{c}$ with their basis matrices $\{\boldsymbol{P}_i\}_{i=1}^{c}$, a basis matrix of a sum subspace $\mathscr{G}$ can be created in the following manner:

$$\boldsymbol{G} = \frac{1}{c}\sum_{i=1}^{c}\boldsymbol{P}_i. \tag{4.14}$$

Sum subspace $\mathscr{G}$ contains information of all the classes comprising it. To obtain discriminative information, its basis matrix $\boldsymbol{G}$ is decomposed using SVD:

$$\boldsymbol{G} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^{\top}, \tag{4.15}$$

where $\boldsymbol{U}$ contains eigenvectors, and $\boldsymbol{\Lambda}$ eigenvalues in descending order. GDS $\mathscr{D}$ is obtained by removing the leading principal components from sum subspace $\mathscr{G}$. This is done by selecting k eigenvectors from $\boldsymbol{U}$ corresponding to *lowest* eigenvalues in $\boldsymbol{\Lambda}$. Resulting subspace $\mathscr{D}$ contains only differences between all subspaces $\{\mathscr{P}_i\}_{i=1}^{c}$. Discriminative information is then used for the purposes of classification, by projecting original reference class subspaces $\{\mathscr{P}_i\}_{i=1}^{c}$ onto the GDS $\mathscr{D}$, resulting in a set of reference class subspaces $\hat{P} = \{\hat{\boldsymbol{P}}_i\}_{i=1}^{c}$ with enhanced discriminative capabilities:

$$\hat{\boldsymbol{P}}_i = orth(\boldsymbol{D}^{\top}\boldsymbol{P}_i). \tag{4.16}$$

with $orth$ representing the Gram-Schmidt orthogonalization. TS-PGM utilizing GDS projection is referred to as **Temporal-Stochastic Constrained PGM** (TS-CPGM). In the training stage, after completing steps 1 and 2 of standard TS-PGM, the resulting set of reference class subspaces $S_c = \{\mathscr{W}_{j,c}\}_{j,c=1}^{4,m_c}$, spanned by $\{\boldsymbol{W}_{j,c}\}_{j,c=1}^{4,m_c}$ is used to create a set of Generalized Difference Subspaces $D = \{\mathscr{D}_j\}_{j=1}^{4}$, spanned by $\{\boldsymbol{D}_j\}_{j=1}^{4}$. In our case, j belongs in $\{1, 2, 3, 4\}$, representing four subspaces; three Spatio-Temporal Tensor modes and one Temporal-Stochastic representation. This is accomplished using equations (4.14) and (4.15). Original subspaces from set S_c are then projected onto D using (4.16) like so:

$$\hat{\boldsymbol{W}}_{j,c} = orth(\boldsymbol{D}_c^{\top}\boldsymbol{W}_{j,c}), \tag{4.17}$$

for every $j \in \{1, 2, 3, 4\}$ and $c \in \{1, \ldots, m_c\}$. Set $\{\hat{\boldsymbol{W}}_{j,c}\}_{j,c=1}^{4,m_c}$ spans the set of projected reference class subspaces $\{\hat{\mathscr{W}}_{j,c}\}_{j,c=1}^{4,m_c}$. To perform classification, an input tensor undergoes steps 1 and 2 of standard TS-PGM and a set of input subspaces $S_q = \{\mathscr{Q}_j\}_{j=1}^{4}$ is projected onto the set

of Generalized Difference Subspaces D using Eq. (4.16). The resulting set $\{\hat{\mathcal{Q}}_j\}_{j=1}^4$, spanned by $\{\boldsymbol{Q}_j\}_{j=1}^4$ is compared to reference class subspaces $\{\hat{\mathscr{W}}_{j,c}\}_{j,c=1}^{4,m_c}$ using similarity in Eq. (4.13) as normal.

4.3.6 Kernel Extension of TS-PGM

The second drawback of TS-PGM lies in the linear nature of subspaces. While theoretically subspaces offer decent and compact representation for individual classes, it is often the case that real-world application data comprises non-linear data distributions. In such cases, linear subspaces usually exhibit a high degree of overlap, thus significantly reducing the classification performance of subspace-based methods [21, 88].

An established way of dealing with this issue is to generate subspace representations through kernel PCA [90, 91], by mapping input patterns into a high dimensional feature space via a nonlinear map ϕ. Linear subspaces created in the mapped feature space correspond to non-linear subspaces in original input space.

Consider a set of patterns $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d)$ in an n-dimensional vector space I and mapping function $\phi(\boldsymbol{x})$. It maps input patterns from I to an f-dimensional feature space $F : \phi : \mathbb{R}^n \rightarrow \mathbb{R}^f, \boldsymbol{x} \rightarrow \phi(\boldsymbol{x})$. PCA can then be performed in the non-linear space F by calculating the inner product $(\phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{y}))$.

Due to high (possibly infinite) dimensionality of space F, it can be hard to calculate the inner product. However, if the nonlinear map ϕ is defined through a kernel function $k(\boldsymbol{x}, \boldsymbol{y})$, the inner products $(\phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{y}))$ can be calculated from the inner products $(\boldsymbol{x} \cdot \boldsymbol{y})$. This technique is known as the "kernel trick". Commonly, an exponential function $k(x, y) = exp(-\frac{||x-y||^2}{\sigma^2})$ is used.

In the context of our framework, kernel PCA is used to construct mode subspaces for Spatio-Temporal and Temporal-Stochastic Tensor features. In standard TS-PGM, Eq. (4.5) is used for computing the auto-correlation matrix, the first step in PCA. However, for kernel PCA, we use ϕ to map the vectors to non-linear space, and thus the equation becomes:

$$\text{corr}(\{\boldsymbol{x}_{j,i}\}) = \sum_i \phi(\boldsymbol{x}_{j,i})^\top \phi(\boldsymbol{x}_{j,i}) \tag{4.18}$$

Having computed the auto-correlation in non-linear space, the flow of kernel TS-PGM continues the same way as standard TS-PGM. We compute a set of reference subspaces for each class in the training equations Eq. (4.4) and Eq. (4.8) for spatio-temporal and temporal-stochastic tensor features, respectively. Then, for an input tensor, we compute a set of subspaces using Eq. (4.6) and Eq. (4.9). Similarity between a set of input subspaces and a set of reference subspaces is performed in the same way as in standard TS-PGM. We refer to this enhanced method as **Temporal-Stochastic Kernel PGM** (TS-KPGM).

It is worth noting that GDS projection from Section 4.3.5 can be used in conjunction with the kernel mapping, leveraging the benefits of both extensions simultaneously, at the cost of increased computational requirements and memory usage. This final extension is called **Temporal-Stochastic Kernel Constrained Product Grassmann Manifold** (TS-KCPGM).

Mode	Reference	Input	Similarity
M1	$O(c(wt)(hm)^2)$	$O(wth^2)$	$O(ck^2i)$
M2	$O(c(ht)(wm)^2$	$O(htw^2)$	$O(ck^2i)$
M3	$O(c(wh)(tm)^2)$	$O(wht^2)$	$O(ck^2i)$
TST	$O(c(qhw)(rm)^2)$	$O(qhwr^2)$	$O(ck^2i)$

Table 4.1: Varying time complexities of SVD operations in TS-PGM. The costliest operation is computing the reference subspaces; with TST features being slightly more expensive than regular tensor modes. Computing input subspaces and similarities on the product manifold is relatively inexpensive.

4.3.7 Computational Complexity of Proposed methods

Finally, we discuss the computational complexity of our methods in terms of Big O notation. The main bottleneck of our method is the SVD, required for computing reference subspaces, input subspaces, and similarities between them. In general, the upper bound time complexity of the SVD is $O(mn^2)$, for an $m \times n$ matrix. This makes our method require exponential execution time. However, the actual computation time differs based on a number of parameters such as dataset and tensor size, and chosen reference and input subspace dimensions. In practice, subspace dimensions are usually very small.

Further discussion is carried out for a classification problem with c classes, each of which contains m tensors. For simplicity, all tensors are of the same size $h \times w \times t$. To obtain TST features, q frames are sampled r times from each tensor. Reference subspaces are of dimension k, while input subspaces are of dimension i, where $k >= i$. A general overview of SVD complexities is shown in Table 4.1.

In the case of TS-PGM, computation of reference subspaces mostly depends on class and sample sizes c and m, tensor dimensions h, w, t and sampling parameters q and r, and is the most costly operation in our method. Input subspaces, on the other hand, are obtained very fast and depend only on tensor dimensions, while the computation of similarities depends on class size c and subspace dimensions k and i.

Kernel extension to TS-PGM additionally requires computation of kernel space prior to constructing reference subspaces; therefore, its cost is higher compared to base TS-PGM and greatly depends on the size of the dataset. On the other hand, introducing GDS projection to TS-PGM marginally increases the complexity in the training step by requiring an additional SVD operation per mode to compute the GDS.

4.4 Experimental Results

In this section, we evaluate proposed methods on several hand gesture, action and skeleton-based action recognition datasets. We first describe the datasets we use in our experiments, as well as their respective experimental settings. Secondly, we evaluate the validity of tensor representation with a sequence-preserving TST subspace and analyze the contribution of this representation on the TS-PGM compared to standard tensor modes through ablation experiments. Thirdly, we show that

extensions of TS-PGM with n-mode GDS projection and kernel mapping achieve better performance compared to basic TS-PGM. Lastly, we compare our methods to relevant subspace-based and neural network-based approaches and discuss their advantages and drawbacks.

4.4.1 Datasets

Cambridge Gesture (CMB) is a simple hand motion dataset that contains 9 classes (3 hand shapes combined with 3 hand motions), with 100 video sequences per class, resulting in 900 videos in total. It is divided equally into five sets based on illumination settings. We use one set with normal illumination for training, and the rest for testing.

Northwestern Hand Gesture (NW) is a more complex dataset containing 10 gestures performed by 15 subjects with 7 different hand shapes. This totals 1,050 video sequences. A convention is to consider hand shapes and subjects to be variations for every gesture performed, therefore the dataset has 10 classes. We use 550, 250 and 250 sequences for training, validation and testing, respectively.

KTH dataset consists of 2,391 action sequences in video format. It contains six types of actions recorded in four different scenarios, with total of 25 subjects recorded performing the actions. The dataset is split according to subjects, with 8, 8 and 9 subjects for train, validation and test sets, respectively.

UT-Kinect (UT) dataset contains videos, depth sequences, and skeleton data of 10 action classes, performed by 10 subjects in front of a Kinect device. There are 200 sequences in total. In our experiments we utilize skeleton data, which contain x, y and z coordinates of 20 skeleton joints, under a Leave-One-Out Cross Validation (LOOCV) experimental protocol.

All datasets contain sequences of variable length which we resize to equal number of frames using linear interpolation. This is a key pre-processing step, as tensors are required to be of the same dimensions for most tensor-based methods. However, TST features can be extracted before this interpolation, thus potentially preserving additional information. The dimensions of resulting tensors are $12 \times 16 \times 30$ (CMB), $12 \times 16 \times 20$ (NW), $16 \times 16 \times 30$ (KTH) and $3 \times 20 \times 20$ (UT).

4.4.2 Spatio-temporal and temporal-stochastic tensor features

First, we put our main question to the test - how well do Temporal-Stochastic Tensor (TST) features capture the temporal information compared to regular Spatio-Temporal Tensor (STT) modes? We do this through a classification ablation study on all datasets, by classifying subspace representations of both STT modes and TST features using Mutual Subspace Method (MSM) [77], an established method for classifying subspace-represented data, specifically pattern-sets. It is a suitable method for evaluating the accuracy obtained from singular modes, as it also relies on canonical angles to measure similarity between subspaces, and TS-PGM could be considered its natural multilinear extension.

Class reference subspaces and input subspaces are obtained as described in section 4.3.4. However, canonical angles between input and class reference subspaces are calculated using Eq. 4.11, and input is classified as the class of the most similar reference subspace. Note that there is no unification using Product Grassmann Manifold, and only information contained in either one of the tensor modes or TST features is used for classification. Dimensions of reference and input subspaces are chosen experimentally and fixed for all classes, meaning that every class is represented with the

Dataset	M1	M2	M3	TST
Cambridge S1	91.11%	76.66%	82.77%	**96.66%**
Cambridge S2	78.88%	77.22%	82.77%	**93.33%**
Cambridge S3	85.55%	81.11%	84.44%	**92.77%**
Cambridge S4	83.88%	76.66%	73.33%	**94.44%**
Cambridge ALL	81.25%	79.02%	73.33%	**93.33%**
Northwestern	88.53%	82.31%	27.11%	**91.37%**
KTH	61.14%	68.48%	74.62%	**75.55%**
UT-Kinect	93.50%	74.44%	79.00%	**94.00%**

Table 4.2: Ablation study of MSM classification performances on singular tensor modes and Temporal-Stochastic Tensor (TST) features. Standard modes are labeled as M1, M2 and M3, depending on the tensor axis along which the matricization was executed. Classification utilizing TST features outperforms regular tensor modes in all cases, usually with a significant margin.

same number of dimensions. Results are shown in Table 4.2.

It can be seen that regular STT modes do not perform consistently across datasets. Mode 1 performs the best on *Cambridge*, *Northwestern* and *UT-Kinect*, while mode 3 (temporal) outperforms it on *KTH*. This implies the imbalance of information contained within individual tensor modes, with respect to a given dataset. For *CMB*, *NW* and *UT*, spatial information is more valuable than temporal, as shown by superior performance of modes 1 and 2, while *KTH* benefits more from temporal mode. However, TST representation consistently achieves the best performance, outperforming regular modes by far. This is possibly due to the ability of TST features to represent both global and local temporal information, offering significant improvement over a regular temporal mode. These results give us confidence in using TST features for representing temporal tensors.

Next, we test the benefit of unifying different representations through the Product Grassmann Manifold geometry and compare performance of singular modes. In essence, we aim to show that incorporating explicit temporal information is highly beneficial, compared both to combined representations on PGM and especially individual representation. We use PGM to classify combinations of three STT modes and we use TS-PGM to classify unified TST feature and STT mode representations. For both methods the dimensions of reference and input subspaces are fixed across modes and classes and are determined experimentally, as in the previous experiment. Results are shown in Table 4.3.

Using tensor modes together, via PGM, offers robustness and better performance compared to individual representations. PGM outperforms even MSM using TST features, though the gap in performance is not as large. Joint representation on TS-PGM beats all approaches by a fair margin. As mentioned previously, the contribution of different modes varies depending on the dataset. However, TS-PGM offers robustness to this variance, in addition to leveraging additional temporal information using TST features. It is worth noting that in our approach, the contributions of factors on PGM and TS-PGM are uniform. This means that when computing the similarity between points on the product manifold, there is no weighting strategy applied to the contribution of similarities of each of the factor manifolds and all factors are considered equal. Further modifying the significance of each factor is possible by learning weights, like in [25].

Dataset	Single Mode	TST	PGM	TS-PGM
Cambridge	81.25%	93.33%	93.33%	**97.08%**
Northwestern	88.53%	91.37%	93.05%	**94.98%**
KTH	74.62%	75.55%	77.28%	**80.15%**
UT-Kinect	93.50%	94.00%	95.00%	**99.00%**

Table 4.3: Comparison of best performances from single tensor modes and TST features to unified representations of tensor modes on PGM and TS-PGM. Results shows the benefit of unifying individual representations on the PGM, while TS-PGM demonstrates that incorporating explicit temporal information via TST features beats other representation methods.

Dataset	TS-PGM	TS-CPGM	TS-KPGM	TS-KCPGM
Cambridge S1	98.88%	**100%**	99.44%	**100%**
Cambridge S2	96.66%	98.33%	97.22%	**98.88%**
Cambridge S3	97.77%	**98.33%**	97.77%	97.77%
Cambridge S4	97.77%	**99.44%**	98.33%	**99.44%**
Cambridge ALL	95.08%	97.50%	97.36%	**97.77%**

Table 4.4: Evaluation of discriminative and kernel extesions to TS-PGM representation. The effect of GDS projection to increase discriminative capabilities can be seen in results of TS-CPGM, achieving increase of at least 1% in all cases. Applying kernel mapping improves both base TS-PGM and TS-CPGM, demonstrated by results of TS-KPGM and TS-KCPGM respectively.

4.4.3 Discriminative extensions and evaluation with current methods

In this experiment, we evaluate proposed discriminative extensions to TS-PGM on all *Cambridge* dataset illumination subsets. As mentioned in Section 4.3.5, we employ n-mode GDS projection [25], resulting in the constrained TS-PGM (TS-CPGM) method. We also utilize a kernel mapping to implement kernel TS-PGM (TS-KPGM) and kernel TS-CPGM (TS-KCPGM), explained in Section 4.3.6.

Regular Temporal-Stochastic Product Grassmann Manifold requires setting up three hyperparameters: dimensions of reference and input subspaces, and the number of angles to consider when calculating geodesic distance. Constrained extension of TS-PGM further requires setting up the dimension of Generalized Difference Subspace (GDS). In kernel extensions, as we are using Gaussian exponential function for the kernel trick, where the value of σ is one additional hyperparameter to be considered.

In our experiments, the dimensions of n-mode GDS for TS-CPGM, as well as the kernel size for TS-KPGM and TS-KCPGM are fixed for all modes and classes and are experimentally determined, and the best performance is recorded for each method. Results on *Cambridge* sets are shown in Table 4.4.

The base TS-PGM has the lowest performance which is because it is merely a representation method without a discriminative mechanism. As expected, both extensions offer better performance. Utilizing n-mode GDS projection, which is considered a very good discriminative method [18, 25],

yields stronger results, and offers constant improvement on all sets. TS-CPGM achieves greater performance by at least *1%* in all settings, and even achieves 100% accuracy on set 1. Projection onto n-mode GDS removes overlap between sets of class reference subspaces, making it easier to classify an input tensor. However, in some cases the dimension of the GDS can be hard to determine. Too few dimensions results in the loss of discriminative information, while too many introduces noise, thus negatively impacting the classification performance.

Tackling non-linearity in the data through kernel mapping offers slight (around *0.5%*), but constant and robust, improvement over the baseline, as shown by performance of TS-KPGM. While not as convincing as the constrained version, it still offers a *2%* performance boost in the case of all data being used. It is worth mentioning that while the kernel trick does not offer as big of a boost in performance as n-mode GDS projection, it is comparatively easier to determine proper kernel size. In our experiments, σ can be any value in the range of 10 to 300 without affecting the results significantly. In addition, kernel mapping can be used to strengthen the constrained version of TS-PGM, resulting in TS-KCPGM. By utilizing both the n-mode GDS projection and kernel trick, TS-KCPGM achieves the best results, outperforming the baseline in all cases.

4.4.4 Comparison with related methods

In this section, we compare our proposed method to related work on hand gesture and action recognition, paying special attention to other traditional tensor-based methods. In addition, we consider state of the art methods on the employed datasets - tensor-based, neural-network based or otherwise. We follow standard experimental protocols defined for each dataset. It is important to note that for our method, we utilize exclusively *grayscale* image features from the *Cambridge* and *Northwestern* datasets, unlike some of the other methods.

We compare TS-PGM and and its extensions to established tensor-based methods on *Cambridge* hand gesture dataset. These methods include Discriminative Canonical Correlation (DCC) [47], Tensor Canonical Correlation Analysis (TCCA) [49], Product Grassmann Manifold (PGM) [72] and Tangent Bundle (TB) [73]. Further, we observe performances of closely related methods of Randomized Time Warping (RTW) [107], n-mode Generalized Difference Subspace (n-mode GDS) [25] and n-mode weighted GDS (n-mode wGDS). Finally, Spatio-Temporal Covariance Descriptors (Cov3D) [89] and Key Frames Extraction Method and Feature Fusion Strategy (Key Frames) [108] are considered. Results are shown in Table 4.5.

Results demonstrate the effectiveness of TS-PGM compared to traditional tensor-based methods such as DCC, TCCA, PGM and TB, with smallest performance margin being 6% in case of TB, and largest being 21% in case of DCC. Interestingly, improvement over manifold-based methods of PGM and its discriminative extension n-mode GDS can be noted with 9% and 3% increase. This lends credibility to our idea of incorporating explicit temporal data with TST features.

Spatio-Temporal Covariance Descriptors (Cov3D) [89] are covariance matrix-based features extracted from a number of windows within the video which are found to be the most discriminative. As the Cov3D are symmetric positive definite matrices existing on a Riemannian manifold, a Riemannian learning method is utilized for final classification. However, it can be seen that base TS-PGM outperforms Cov3D by 4%, even though it relies on, essentially, a 1-nearest neighbour classifier based on geodesic distance. This further shows that the quality of representation can outweigh the effectiveness of a discriminative method.

Method	Cambridge
DCC [47]	76%
TCCA [49]	82%
PGM [72]	88%
TB [73]	91%
Cov3D [89]	93%
RTW [107]	93.6%
n-mode GDS [25]	93%
n-mode wGDS [25]	94%
Ours (TS-PGM)	*97.08%*
Ours (TS-KPGM)	*97.36%*
Ours (TS-CPGM)	*97.50%*
Ours (TS-KCPGM)	*97.77%*
Key Frames [108]	**98.23%**

Table 4.5: Comparison of proposed methods with established tensor and subspace based methods on *Cambridge* dataset. It can be clearly seen that TS-PGM offers competitive results, outperforming all methods except Key Frames [108]. TS-KPGM, TS-CPGM and TS-KCPGM all deliver improved performance over the base method, with TS-KCPGM trailing behind the best performing method by only 0.5%.

As noted before, boosting representational and discriminative power of TS-PGM by kernel mapping and GDS projection yields the best results, shown by TS-KCPGM.

The only method outperforming TS-KCPGM, though only by 0.5%, is Key Frames [108], a hand gesture recognition-specific method. It detects the most discriminant frames in a video by calculating entropy of motion histograms of each frame, and finding peaks in the entropy curve through density clustering. Advanced motion and appearance features are then extracted from key frames, before fusing them and classifying using SVM. The difference in performance could be attributed to TS-KCPGM not utilizing hand crafted spatial or temporal features such as HOG [13], SURF [5] or SIFT 3D [93], but instead relying only on grayscale image representation. However, these features can be easily incorporated into TS-PGM by replacing specific tensor modes, and proceeding with individual subspace representation and then unification on TS-PGM as usual.

On *Northwestern* hand gesture dataset, we compare our proposed TS-PGM to several methods designed specifically for hand gesture recognition. [95] calculate Divergence Fields (DF) from optical flows between frames in order to find salient regions for feature extraction; classification is performed by matching against the pre-trained database using the term-frequency inverse document frequency (TF-IDF) algorithm. On the other hand, [61] utilize genetic programming to evolve a set of simple 3D sequence-processing operators to obtain a set of strong high-level, spatio-temporal descriptors.

Results for *Northwestern* hand gesture dataset are shown in table 4.6. While more challenging than the *Cambridge* dataset, TS-PGM demonstrates competitive results, with average accuracy of 94.98% over 20 trials, though not outperforming other methods. Lower performance of TS-PGM can be explained by our choice of features, which are simple grayscale features, not tailored for

Method	Northwestern
Ours (TS-PGM)	*94.98% ± 3.02%*
DF + TF-IDF [95]	95.8%
Genetic Programming [61]	96.1%
Key Frames [108]	**96.89% ± 1.08%**

Table 4.6: Comparison of proposed method with related methods on *Northwestern* dataset. Although not outperforming other methods, TS-PGM yields competitive results, while relying on much simpler image descriptors such as grayscale features.

any specific recognition task. High variance of 3% indicates possible overfitting, which could be explained by the fact that we use the same number of dimensions for all reference subspaces. This can negatively impact the performance, as not all classes and tensor modes require the same number of dimensions to be effectively represented, making it difficult to find optimal dimension for all reference subspaces. A solution would be to allow a degree of freedom when selecting subspace dimensions for a particular class or tensor mode.

Finally, by testing our proposed representation method on skeleton data contained within the *UT-Kinect* dataset, we show its capability to represent different types of temporal tensors. Results are shown in Table 4.7. As TS-PGM is a general framework for encoding temporal tensors, it is possible to utilize skeleton data directly as tensor input, without any preprocessing. TST sampling is performed on the temporal axis, while the tensor is decomposed in the same manner as a grayscale video tensor.

On this dataset, base TS-PGM achieves excellent, near state of the art performance, doing better than most other approaches, which utilize Graph Based Skeleton Modeling [46], Time Warped ARMA Models [99] and Lie Groups [116]. It even outperforms deep learning approaches such as [59] which is based on LSTM networks with attention mechanism, [22] that relies on graph convolutional neural networks, and [109] that utilizes deep reinforcement learning. The only method outperforming TS-PGM is the approach based on subspace clustering and temporal pruning, proposed by [82], highly specialized for human action recognition using 3D skeletons.

These results show promise for the ability to represent general temporal tensors on TS-PGM. In addition, they show that it is feasible to use skeleton extraction methods as a powerful preprocessing step for human action recognition under the TS-PGM framework.

The proposed method does not rely on pretraining, or extracting features such as optical flow [38], and uses only raw tensor data while achieving competitive results. It does not require a large amount of training data, as demonstrated on small sample datasets. Another advantage of our method is robustness to different illumination settings, which is a well-known property of subspace representation. The proposed methods handle prototypes of considerably smaller size than original tensors, benefiting hardware with limited memory and processing capacities. Further, scalability is achievable due to the independence of the subspaces, allowing for utilization of parallel processing techniques to speed up computation. Besides, new classes can be quickly adjusted to incorporate new knowledge in our methods. It is worth noting that TS-PGM and its extensions work by focusing only the representational aspect of data, which leaves room for further improvement by using explicit discriminative methods such as GDA [31], or other learning methods.

Method	UT-Kinect
Kao et al. [46]	96.0%
Sogi et al. [99]	97.0%
Vemulapalli et al. [116]	97.1%
Liu et al. [59]	98.5%
Tang et al. [109]	98.5%
Gao et al. [22]	98.5%
Ours (TS-PGM)	*99.0%*
Paoletti et al. [82]	**99.5%**

Table 4.7: Comparison of proposed method with related methods on *UT-Kinect* dataset. In general, tensors containing temporal modes can be represented on the TS-PGM, not necessarily only videos. In UT-Kinect, human actions are defined by tensors containing skeleton coordinates over a period of time. Classification on TS-PGM yields very competitive results, outperforming most related methods, while requiring no special preprocessing.

4.5 Summary

In this chapter we introduced the Temporal-Stochastic Product Grassmann Manifold representation and classification framework for temporal tensors, based on Product Grassmann Manifold and Temporal-Stochastic Tensor features. Action and gesture recognition tasks were used to evaluate the proposed method since their data structures are typically presented as tensors.

We address the loss of temporal information occurring in traditional PGM method by creating Temporal-Stochastic Tensor features through a random sampling method. We then use the PGM geometry to naturally unify representations of tensor modes and Temporal-Stochastic Tensor features, allowing for the application of the geodesic distance to investigate the relationship between temporal tensors. Additionally, we employ n-mode GDS projection to extract discriminative features and improve our base method. We further implement a kernel mapping to handle nonlinear data distributions. Experimental results revealed that the proposed method is superior to the conventional PGM and subspace-related methods, confirming the usefulness of explicit temporal representation via Temporal-Stochastic Features in unison with tensor modes. In addition, we show the possibility of using different types of temporal tensors, such as skeleton data, highlighting the generalizability of our method.

A potential future direction would include extending the proposed method to consider hybrid varieties of kernels, approaching each mode with an optimal kernel. This strategy may provide more informative features since it would deal with the nonlinear nature of the data distribution in each mode. Due to recent advances in differentiability of the Singular Value Decomposition computation [121], it would be interesting to design an end-to-end framework able to learn better representations with algorithms such as stochastic gradient descent on the Riemannian Manifold [102]. We also plan to investigate different manifolds for future work and evaluate the proposed method on more complex and varied data.

Chapter 5

Concluding remarks

In this thesis we introduced basic representation methods for tensors containing temporal information, followed by algorithms for visualization, clustering and classification in the domain of computer vision. First, we introduced a method for representing temporal tensors by using the PGM geometry to naturally unify representations of tensor modes and Hankel-like embedding of temporal information. Geodesic distance on the PGM-HLE can be used to analyze the relationship between temporal tensors, and used as a general interface for solving optimization and clustering problems. To demonstrate this, we performed t-SNE visualizations and spectral clustering of temporal tensor datasets containing video and skeletal data, giving some weight to the strategy of unified representation on PGM, special treatment of temporal information via Hankel-like embeddings and finally the idea of geodesic distance as a general interface for solving various problems. Specifically in the context of video datasets, this approach may prove valuable as it allows simple and fast analysis of data in its raw form, without the need for significant data pre-processing or pre-training of heavy representational models.

Then, we extended and improved this idea by introducing Temporal-Stochastic Product Grassmann Manifold (TS-PGM), a classification framework for action recognition. We greatly enhanced the temporal encoding of tensors with Temporal-Stochastic Tensor features, with can likewise be seamlessly integrated with the PGM geometry. The discriminative abilities of TS-PGM are improved by introducing generalized difference subspace (GDS) projection, resulting in a constrained TS-PGM (TS-CPGM). We also introduced kernel mapping to TS-PGM to alleviate non-linearity issues that often occur in application data, culminating in the kernel TS-PGM (TS-KPGM). Both of these extensions can be utilized at the same time. We demonstrate proposed classification frameworks for tasks of hand gesture, action and skeleton action recognition. All experimental results show the effectiveness of proposed methods. Proposed methods are general, assume only tensors containing temporal modes as input, and can be utilized in a variety of applications which we have not explored at present.

Concerning our future work, there are several areas in which our algorithms could be improved. Most crucial problem to address would be better selection of frames for best temporal encoding, as TST features are not necessarily optimal for classification purpose due to randomness. Further, it would be very valuable to investigate the use of various features other than grayscale images, as this will open up the pathway to many computer vision applications. One such interesting application

is anomaly detection, where it would theoretically be possible to detect changes in a video by observing changes in multiple tensor modes simultaneously, which could yield improvement over tracking the changes only on raw video. A more daring expansion into applications can be done by considering data such as 4D videos, 4D computed tomography (CT) scans, 4D magnetic resonance imaging (MRI) scans, and other forms of various higher-order tensor data. Another, very interesting perspective, is combining representations of multiple signals or videos through the PGM geometry, as well as considering data with mutliple modalities. We hope that this basis will foster development of algorithms targeting other applications.

Bibliography

[1] Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *Journal of machine learning research*, 15:2773–2832, 2014.

[2] Sherif Azary and Andreas Savakis. Grassmannian sparse representations and motion depth surfaces for 3d action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 492–499, 2013.

[3] Suresh Balakrishnama and Aravind Ganapathiraju. Linear discriminant analysis-a brief tutorial. *Institute for Signal and information Processing*, 18(1998):1–8, 1998.

[4] Panagiotis Barmpoutis, Tania Stathaki, and Stephanos Camarinopoulos. Skeleton-based human action recognition through third-order tensor representation and spatio-temporal analysis. *Inventions*, 4(1):9, 2019.

[5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[6] Peter N. Belhumeur, Joao P Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):711–720, 1997.

[7] Åke Björck and Gene H Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of computation*, 27(123):579–594, 1973.

[8] Salah Bourennane, Caroline Fossati, and Alexis Cailly. Improvement of classification for hyperspectral images based on tensor modeling. *IEEE Geoscience and Remote Sensing Letters*, 7(4):801–805, 2010.

[9] Deng Cai, Xiaofei He, and Jiawei Han. Subspace learning based on tensor analysis. Technical report, 2005.

[10] Rama Chellappa, Amit K Roy-Chowdhury, and S Kevin Zhou. Recognition of humans and their activities using video. *Synthesis Lectures on Image, Video & Multimedia Processing*, 1(1):1–173, 2005.

[11] Huiyuan Chen and Jing Li. Modeling relational drug-target-disease interactions via tensor factorization with multiple web sources. In *The World Wide Web Conference*, pages 218–227, 2019.

[12] Andrzej Cichocki, Danilo Mandic, Lieven De Lathauwer, Guoxu Zhou, Qibin Zhao, Cesar Caiafa, and Huy Anh Phan. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE signal processing magazine*, 32(2):145–163, 2015.

[13] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[14] Wenwen Ding, Kai Liu, Evgeny Belyaev, and Fei Cheng. Tensor-based linear dynamical systems for action recognition from 3d skeletons. *Pattern Recognition*, 77:75–86, 2018.

[15] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[16] Karl Pearson F.R.S. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[17] Kazuhiro Fukui. Subspace methods. *Computer Vision: A Reference Guide*, pages 1–5, 2020.

[18] Kazuhiro Fukui and Atsuto Maki. Difference subspace and its generalization for subspace-based methods. *IEEE transactions on pattern analysis and machine intelligence*, 37(11):2164–2177, 2015.

[19] Kazuhiro Fukui, Naoya Sogi, Takumi Kobayashi, Jing-Hao Xue, and Atsuto Maki. Discriminant feature extraction by generalized difference subspace. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[20] Kazuhiro Fukui, Björn Stenger, and Osamu Yamaguchi. A framework for 3d object recognition using the kernel constrained mutual subspace method. In *Asian Conference on Computer Vision*, pages 315–324. Springer, 2006.

[21] 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.

[22] Xiang Gao, Wei Hu, Jiaxiang Tang, Jiaying Liu, and Zongming Guo. Optimized skeleton-based action recognition via sparsified graph regression. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 601–610, 2019.

[23] Xizhan Gao, Quansen Sun, Haitao Xu, and Jianqiang Gao. Sparse and collaborative representation based kernel pairwise linear regression for image set classification. *Expert Systems with Applications*, 140:112886, 2020.

[24] 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.

[25] Bernardo B Gatto, Eulanda M dos Santos, Alessandro L Koerich, Kazuhiro Fukui, and Waldir SS Junior. Tensor analysis with n-mode generalized difference subspace. *Expert Systems with Applications*, 171:114559, 2021.

[26] Bernardo B Gatto, Eulanda M dos Santos, Marco AF Molinetti, and Kazuhiro Fukui. Multilinear clustering via tensor fukunaga–koontz transform with fisher eigenspectrum regularization. *Applied Soft Computing*, page 107899, 2021.

[27] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU press, 2013.

[28] Richard D Green and Ling Guan. Quantifying and recognizing human movement patterns from monocular video images-part ii: applications to biometrics. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(2):191–198, 2004.

[29] Weiwei Guo, Irene Kotsia, and Ioannis Patras. Tensor learning for regression. *IEEE Transactions on Image Processing*, 21(2):816–827, 2011.

[30] Jihun Hamm. *Subspace-based learning with Grassmann kernels*. PhD thesis, University of Pennsylvania, 2008.

[31] 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, 2008.

[32] 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):113–136, 2015.

[33] Mehrtash Harandi, Conrad Sanderson, Chunhua Shen, and Brian C Lovell. Dictionary learning and sparse coding on grassmann manifolds: An extrinsic solution. In *Proceedings of the IEEE international conference on computer vision*, pages 3120–3127, 2013.

[34] David R Hardoon and John Shawe-Taylor. Decomposing the tensor kernel support vector machine for neuroscience data with structured labels. *Machine learning*, 79(1):29–46, 2010.

[35] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

[36] Xiaofei He. Incremental semi-supervised subspace learning for image retrieval. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 2–8, 2004.

[37] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844, 1998.

[38] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[39] Katsushi Ikeuchi. *Computer vision: A reference guide*. Springer Publishing Company, Incorporated, 2014.

[40] Anil K Jain, Arun Ross, and Salil Prabhakar. An introduction to biometric recognition. *IEEE Transactions on circuits and systems for video technology*, 14(1):4–20, 2004.

[41] Chengcheng Jia and Yun Fu. Low-rank tensor subspace learning for rgb-d action recognition. *IEEE Transactions on Image Processing*, 25(10):4641–4652, 2016.

[42] Heidi Johansen-Berg and Timothy EJ Behrens. *Diffusion MRI: from quantitative measurement to in vivo neuroanatomy*. Academic Press, 2013.

[43] Mohamad Jouni, Mauro Dalla Mura, and Pierre Comon. Hyperspectral image classification using tensor cp decomposition. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 1164–1167. IEEE, 2019.

[44] Mohamad Jouni, Mauro Dalla Mura, and Pierre Comon. Hyperspectral image classification based on mathematical morphology and tensor decomposition. *Mathematical Morphology-Theory and Applications*, 4(1):1–30, 2020.

[45] Charilaos I Kanatsoulis, Xiao Fu, Nicholas D Sidiropoulos, and Wing-Kin Ma. Hyperspectral super-resolution: A coupled tensor factorization approach. *IEEE Transactions on Signal Processing*, 66(24):6503–6517, 2018.

[46] Jiun-Yu Kao, Antonio Ortega, Dong Tian, Hassan Mansour, and Anthony Vetro. Graph based skeleton modeling for human activity analysis. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2025–2029. IEEE, 2019.

[47] Tae-Kyun Kim and Roberto Cipolla. Gesture recognition under small sample size. In *Asian conference on computer vision*, pages 335–344. Springer, 2007.

[48] 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.

[49] Tae-Kyun Kim, Shu-Fai Wong, and Roberto Cipolla. Tensor canonical correlation analysis for action classification. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[50] Tamara G Kolda. Orthogonal tensor decompositions. *SIAM Journal on Matrix Analysis and Applications*, 23(1):243–255, 2001.

[51] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[52] Hui Kong, Eam Khwang Teoh, Jian Gang Wang, and Ronda Venkateswarlu. Two-dimensional fisher discriminant analysis: forget about small sample size problem [face recognition applications]. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 2, pages ii–761. IEEE, 2005.

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

[54] Piotr Koniusz, Lei Wang, and Anoop Cherian. Tensor representations for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[55] Damien Letexier, Salah Bourennane, and Jacques Blanc-Talon. Nonorthogonal tensor matricization for hyperspectral image filtering. *IEEE Geoscience and Remote Sensing Letters*, 5(1):3–7, 2008.

[56] Binlong Li, Mustafa Ayazoglu, Teresa Mao, Octavia I Camps, and Mario Sznaier. Activity recognition using dynamic subspace angles. In *CVPR 2011*, pages 3193–3200. IEEE, 2011.

[57] Jie Li, Liqing Zhang, Dacheng Tao, Han Sun, and Qibin Zhao. A prior neurophysiologic knowledge free tensor-based scheme for single trial eeg classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 17(2):107–115, 2008.

[58] Ming Li and Baozong Yuan. 2d-lda: A statistical linear discriminant analysis for image matrix. *Pattern Recognition Letters*, 26(5):527–532, 2005.

[59] Jun Liu, Gang Wang, Ping Hu, Ling-Yu Duan, and Alex C Kot. Global context-aware attention lstm networks for 3d action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1647–1656, 2017.

[60] Ke Liu, Yong-Qing Cheng, and Jing-Yu Yang. Algebraic feature extraction for image recognition based on an optimal discriminant criterion. *Pattern recognition*, 26(6):903–911, 1993.

[61] Li Liu and Ling Shao. Synthesis of spatio-temporal descriptors for dynamic hand gesture recognition using genetic programming. In *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–7. IEEE, 2013.

[62] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[63] Haiping Lu, KN Plataniotis, and AN Venetsanopoulos. Boosting lda with regularization on mpca features for gait recognition. In *2007 Biometrics Symposium*, pages 1–6. IEEE, 2007.

[64] Haiping Lu, Konstantinos N Plataniotis, and Anastasios N Venetsanopoulos. Multilinear principal component analysis of tensor objects for recognition. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 2, pages 776–779. IEEE, 2006.

[65] Haiping Lu, Konstantinos N Plataniotis, and Anastasios N Venetsanopoulos. Mpca: Multilinear principal component analysis of tensor objects. *IEEE transactions on Neural Networks*, 19(1):18–39, 2008.

[66] Haiping Lu, Konstantinos N Plataniotis, and Anastasios N Venetsanopoulos. Boosting discriminant learners for gait recognition using mpca features. *EURASIP Journal on Image and Video Processing*, 2009:1–11, 2009.

[67] Haiping Lu, Konstantinos N Plataniotis, and Anastasios N Venetsanopoulos. Regularized common spatial patterns with generic learning for eeg signal classification. In *2009 Annual International Conference of the IEEE Engineering in medicine and biology society*, pages 6599–6602. IEEE, 2009.

[68] Haiping Lu, Jie Wang, and Konstantinos N Plataniotis. A review on face and gait recognition: System, data, and algorithms. *Advanced Signal Processing*, pages 303–330, 2017.

[69] Juwei Lu, Konstantinos N Plataniotis, and Anastasios N Venetsanopoulos. Face recognition using kernel direct discriminant analysis algorithms. *IEEE transactions on Neural Networks*, 14(1):117–126, 2003.

[70] Yui Man Lui. Tangent bundles on special manifolds for action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(6):930–942, 2011.

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

[72] Yui Man Lui. Human gesture recognition on product manifolds. *The Journal of Machine Learning Research*, 13(1):3297–3321, 2012.

[73] Yui Man Lui and J Ross Beveridge. Tangent bundle for human action recognition. In *2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, pages 97–102. IEEE, 2011.

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

[75] Yuan Luo, Fei Wang, and Peter Szolovits. Tensor factorization toward precision medicine. *Briefings in bioinformatics*, 18(3):511–514, 2017.

[76] Anant Madabhushi and George Lee. Image analysis and machine learning in digital pathology: Challenges and opportunities. *Medical image analysis*, 33:170–175, 2016.

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

[78] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

[79] Yasuhiro Ohkawa and Kazuhiro Fukui. Hand shape recognition based on kernel orthogonal mutual subspace method. In *MVA*, pages 122–125, 2009.

[80] Erkki Oja. *Subspace methods of pattern recognition*, volume 6. John Wiley & Sons, 1983.

[81] Yannis Panagakis, Constantine Kotropoulos, and Gonzalo R Arce. Non-negative multilinear principal component analysis of auditory temporal modulations for music genre classification. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):576–588, 2009.

[82] Giancarlo Paoletti, Jacopo Cavazza, Cigdem Beyan, and Alessio Del Bue. Subspace clustering for action recognition with covariance representations and temporal pruning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6035–6042. IEEE, 2021.

[83] Evangelos E Papalexakis, Christos Faloutsos, and Nicholas D Sidiropoulos. Tensors for data mining and data fusion: Models, applications, and scalable algorithms. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(2):1–44, 2016.

[84] Thomas Papastergiou, Evangelia I Zacharaki, and Vasileios Megalooikonomou. Tensor decomposition for multiple-instance classification of high-order medical data. *Complexity*, 2018, 2018.

[85] Nadine Renard and Salah Bourennane. Dimensionality reduction based on tensor modeling for classification methods. *IEEE Transactions on Geoscience and Remote Sensing*, 47(4):1123–1131, 2009.

[86] M. Safayani, M.T. Manzuri Shalmani, and M. Khademi. Extended two-dimensional pca for efficient face representation and recognition. In *2008 4th International Conference on Intelligent Computer Communication and Processing*, pages 295–298, 2008.

[87] Harkirat S Sahambi and Khashayar Khorasani. A neural-network appearance-based 3-d object recognition using independent component analysis. *IEEE transactions on neural networks*, 14(1):138–149, 2003.

[88] Hitoshi Sakano, Naoki Mukawa, and Taichi Nakamura. Kernel mutual subspace method and its application for object recognition. *Electronics and Communications in Japan (Part II: Electronics)*, 88(6):45–53, 2005.

[89] Andres Sanin, Conrad Sanderson, Mehrtash T Harandi, and Brian C Lovell. Spatio-temporal covariance descriptors for action and gesture recognition. In *2013 IEEE Workshop on applications of Computer Vision (WACV)*, pages 103–110. IEEE, 2013.

[90] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International conference on artificial neural networks*, pages 583–588. Springer, 1997.

[91] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[92] Thomas Schultz and Hans-Peter Seidel. Estimating crossing fibers: A tensor decomposition approach. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1635–1642, 2008.

[93] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 357–360, 2007.

[94] Eli Shechtman, Yaron Caspi, and Michal Irani. Space-time super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):531–545, 2005.

[95] Xiaohui Shen, Gang Hua, Lance Williams, and Ying Wu. Dynamic hand gesture recognition: An exemplar-based approach from motion divergence fields. *Image and Vision Computing*, 30(3):227–235, 2012.

[96] Qiquan Shi, Jiaming Yin, Jiajun Cai, Andrzej Cichocki, Tatsuya Yokota, Lei Chen, Mingxuan Yuan, and Jia Zeng. Block hankel tensor arima for multiple short time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5758–5766, 2020.

[97] Nicholas D Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E Papalexakis, and Christos Faloutsos. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 65(13):3551–3582, 2017.

[98] Pawan Sinha, Benjamin Balas, Yuri Ostrovsky, and Richard Russell. Face recognition by humans: Nineteen results all computer vision researchers should know about. *Proceedings of the IEEE*, 94(11):1948–1962, 2006.

[99] 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.

[100] Lincon S Souza, Bernardo B Gatto, and Kazuhiro Fukui. Grassmann singular spectrum analysis for bioacoustics classification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 256–260. IEEE, 2018.

[101] Lincon S Souza, Bernardo B Gatto, Jing-Hao Xue, and Kazuhiro Fukui. Enhanced grassmann discriminant analysis with randomized time warping for motion recognition. *Pattern Recognition*, 97:107028, 2020.

[102] Lincon S Souza, Naoya Sogi, Bernardo B Gatto, Takumi Kobayashi, and Kazuhiro Fukui. An interface between grassmann manifolds and vector spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 846–847, 2020.

[103] Lincon S Souza, Naoya Sogi, Bernardo B Gatto, Takumi Kobayashi, and Kazuhiro Fukui. Grassmannian learning mutual subspace method for image set recognition. *arXiv preprint arXiv:2111.04352*, 2021.

[104] GW Stewart and JG Sun. Computer science and scientific computing. matrix perturbation theory, 1990.

[105] Jimeng Sun, Dacheng Tao, and Christos Faloutsos. Beyond streams and graphs: dynamic tensor analysis. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 374–383, 2006.

[106] Jimeng Sun, Dacheng Tao, Spiros Papadimitriou, Philip S Yu, and Christos Faloutsos. Incremental tensor analysis: Theory and applications. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2(3):1–37, 2008.

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

[108] Hao Tang, Hong Liu, Wei Xiao, and Nicu Sebe. Fast and robust dynamic hand gesture recognition via key frames extraction and feature fusion. *Neurocomputing*, 331:424–433, 2019.

[109] Yansong Tang, Yi Tian, Jiwen Lu, Peiyang Li, and Jie Zhou. Deep progressive reinforcement learning for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5323–5332, 2018.

[110] Dacheng Tao, Xuelong Li, Xindong Wu, and Stephen J Maybank. General tensor discriminant analysis and gabor features for gait recognition. *IEEE transactions on pattern analysis and machine intelligence*, 29(10):1700–1715, 2007.

[111] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

[112] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[113] Claudio Varini, Andreas Degenhard, and Tim W Nattkemper. Isolle: Lle with geodesic distance. *Neurocomputing*, 69(13-15):1768–1771, 2006.

[114] M Alex O Vasilescu and Demetri Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *European conference on computer vision*, pages 447–460. Springer, 2002.

[115] M Alex O Vasilescu and Demetri Terzopoulos. Tensortextures: Multilinear image-based rendering. In *ACM SIGGRAPH 2004 Papers*, pages 336–342. 2004.

[116] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 588–595, 2014.

[117] Boyue Wang and Junbin Gao. Unsupervised learning on grassmann manifolds for big data. In *Multimodal Analytics for Next-Generation Big Data Technologies and Applications*, pages 151–180. Springer, 2019.

[118] Jin Wang, Armando Barreto, Lu Wang, Yu Chen, Naphtali Rishe, Jean Andrian, and Malek Adjouadi. Multilinear principal component analysis for face recognition with fewer features. *Neurocomputing*, 73(10-12):1550–1555, 2010.

[119] Long Wang, JL Wang, ZL Cheng, L Ran, and Z Yin. Personalized medicine recommendation based on tensor decomposition. *Comput Sci*, 42:225–229, 2015.

[120] Mengjiao Wang, Yannis Panagakis, Patrick Snape, and Stefanos P Zafeiriou. Disentangling the modes of variation in unlabelled data. *IEEE transactions on pattern analysis and machine intelligence*, 40(11):2682–2695, 2017.

[121] Wei Wang, Zheng Dang, Yinlin Hu, Pascal Fua, and Mathieu Salzmann. Robust differentiable svd. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[122] Satosi Watanabe and Nikhil Pakvasa. Subspace method of pattern recognition. In *Proc. 1st. IJCPR*, pages 25–32, 1973.

[123] Martin Wattenberg, Fernanda Viégas, and Ian Johnson. How to use t-sne effectively. *Distill*, 1(10):e2, 2016.

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

[125] Shuicheng Yan, Dong Xu, Qiang Yang, Lei Zhang, Xiaoou Tang, and Hong-Jiang Zhang. Multilinear discriminant analysis for face recognition. *IEEE Transactions on image processing*, 16(1):212–220, 2006.

[126] Jian Yang, David Zhang, Alejandro F Frangi, and Jing-yu Yang. Two-dimensional pca: a new approach to appearance-based face representation and recognition. *IEEE transactions on pattern analysis and machine intelligence*, 26(1):131–137, 2004.

[127] Jieping Ye. Generalized low rank approximations of matrices. In *Proceedings of the twenty-first international conference on Machine learning*, page 112, 2004.

[128] Jieping Ye, Ravi Janardan, and Qi Li. Gpca: An efficient dimension reduction scheme for image compression and retrieval. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 354–363, 2004.

[129] Jieping Ye, Tao Li, Tao Xiong, and Ravi Janardan. Using uncorrelated discriminant analysis for tissue classification with gene expression data. *IEEE/ACM Transactions on computational biology and bioinformatics*, 1(4):181–190, 2004.

[130] Yeyang Yu, Jin Jin, Feng Liu, and Stuart Crozier. Multidimensional compressed sensing mri using tensor decomposition-based sparsifying transform. *PloS one*, 9(6):e98441, 2014.

[131] Daoqiang Zhang and Zhi-Hua Zhou. (2d) 2pca: Two-directional two-dimensional pca for efficient face representation and recognition. *Neurocomputing*, 69(1-3):224–231, 2005.

List of publications

1. Bojan Batalo, Lincon S. Souza, Bernardo B. Gatto, Naoya Sogi, Kazuhiro Fukui, “Temporal-Stochastic Tensor Features for Action Recognition”, Machine Learning with Applications, Volume 10, 100407, 2022.

2. Bojan Batalo, Lincon S. Souza, Bernardo B. Gatto, Naoya Sogi, Kazuhiro Fukui, “Analysis of Temporal Tensor Datasets on Product Grassmann Manifold”, 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 4868-4876, 2022.