Journal of Machine Learning Research 13 (2012) 3297-3321 Submitted 5/12; Revised 10/12; Published 11/12

Human Gesture Recognition on Product Manifolds

Yui Man Lui LUI@CS.COLOSTATE.EDU
Department of Computer Science
Colorado State University
Fort Collins, CO 80523-1873, USA

Editors: Isabelle Guyon and Vassilis Athitsos

Abstract

Action videos are multidimensional data and can be naturally represented as data tensors. While tensor computing is widely used in computer vision, the geometry of tensor space is often ignored. The aim of this paper is to demonstrate the importance of the intrinsic geometry of tensor space which yields a very discriminating structure for action recognition. We characterize data tensors as points on a product manifold and model it statistically using least squares regression. To this aim, we factorize a data tensor relating to each order of the tensor using Higher Order Singular Value Decomposition (HOSVD) and then impose each factorized element on a Grassmann manifold. Furthermore, we account for underlying geometry on manifolds and formulate least squares regression as a composite function. This gives a natural extension from Euclidean space to manifolds. Consequently, classification is performed using geodesic distance on a product manifold where each factor manifold is Grassmannian. Our method exploits appearance and motion without explicitly modeling the shapes and dynamics. We assess the proposed method using three gesture databases, namely the Cambridge hand-gesture, the UMD Keck body-gesture, and the CHALEARN gesture challenge data sets. Experimental results reveal that not only does the proposed method perform well on the standard benchmark data sets, but also it generalizes well on the one-shot-learning gesture challenge. Furthermore, it is based on a simple statistical model and the intrinsic geometry of tensor space.

Keywords: gesture recognition, action recognition, Grassmann manifolds, product manifolds, one-shot-learning, kinect data

1. Introduction

Human gestures/actions are the natural way for expressing intentions and can be instantly recognized by people. We use gestures to depict sign language to deaf people, convey messages in noisy environments, and interface with computer games. Having automated gesture-based communication would broaden the horizon of human-computer interaction and enrich our daily lives. In recent years, many gesture recognition algorithms have been proposed (Mitra and Acharya, 2007; Wang et al., 2009; Bilinski and Bremond, 2011). However, reliable gesture recognition remains a challenging area due in part to the complexity of human movements. To champion the recognition performance, models are often complicated, causing difficulty for generalization. Consequently, heavy-duty models may not have substantial gains in overall gesture recognition problems.

In this paper, we propose a new representation to gesture recognition based upon tensors and the geometry of product manifolds. Since human actions are expressed as a sequence of video frames, an action video may be characterized as a third order data tensor. The mathematical framework

for working with high order tensors is multilinear algebra which is a useful tool for characterizing multiple factor interactions. Tensor computing has been successfully applied to many computer vision applications such as face recognition (Vasilescu and Terzopoulos, 2002), visual tracking (Li et al., 2007), and action classification (Vasilescu, 2002; Kim and Cipolla, 2009). However, the geometrical aspect of data tensors remains unexamined. The goal of this paper is to demonstrate the importance of the intrinsic geometry of tensor space where it provides a very discriminating structure for action recognition.

Notably, several recent efforts (Lui, 2012a) have been inspired by the characteristics of space and the associated construction of classifiers based upon the intrinsic geometry inherent in particular manifolds. Veeraraghavan et al. (2005) modeled human shapes from a shape manifold and expressed the dynamics of human silhouettes using an autoregressive (AR) model on the tangent space. Turaga and Chellappa (2009) extended this framework and represented the trajectories on a Grassmann manifold for activity classification. The use of tangent bundles on special manifolds was investigated by Lui (2012b) where a set of tangent spaces was exploited for action recognition. Age estimation was also studied using Grassmann manifolds (Turaga et al., 2010). The geodesic velocity from an average face to the given face was employed for age estimation where the space of landmarks was interpreted as a Grassmann manifold. Lui and Beveridge (2008) characterized tangent spaces of a registration manifold as elements on a Grassmann manifold for face recognition. The importance of the ordering on Stiefel manifolds was demonstrated by Lui et al. (2009) and an illumination model was applied to synthesize such elements for face recognition. These successes motivate the exploration of the underlying geometry of tensor space.

The method proposed in this paper characterizes action videos as data tensors and demonstrates their association with a product manifold. We focus attention on the intrinsic geometry of tensor space, and draw upon the fact that the geodesic on a product manifold is equivalent to the Cartesian product of geodesics from multiple factor manifolds. In other words, elements of a product manifold are the set of all elements inherited from factor manifolds. Thus, in our approach, action videos are factorized to three factor elements using Higher Order Singular Value Decomposition (HOSVD) in which the factor elements give rise to three factor manifolds. We further extend the product manifold representation to least squares regression. In doing so, we consider the underlying geometry and formulate least squares regression as a composite function. As such, we ensure that both the domain values and the range values reside on a manifold through the regression process. This yields a natural extension from Euclidean space to manifolds. The least squares fitted elements from a training set can then be exploited for gesture recognition where the similarity is expressed in terms of the geodesic distance on a product manifold associated with fitted elements from factor manifolds.

We demonstrate the merits of our method on three gesture recognition problems including hand gestures, body gestures, and gestures collected from the Microsoft Kinect™ camera for the one-shot-learning CHALEARN gesture challenge. Our experimental results reveal that our method is competitive to the state-of-the-art methods and generalizes well to the one-shot-learning scheme, yet is based on a simple statistical model. The key contributions of the proposed work are summarized as follows:

- A new way of relating tensors on a product manifold to action recognition.
- A novel formulation for least squares regression on manifolds.
- The use of appearance and motion without explicitly modeling shapes or dynamics.

- A simple pixel-based representation (no silhouette or skeleton extraction).
- No extensive training and parameter tuning.
- No explicit assumption on action data.
- Competitive performance on gesture recognition.
- Applicable to other visual applications.

The rest of this paper is organized as follows: Related work is summarized in Section 2. Tensor algebra, orthogonal groups, and Grassmann manifolds are reviewed in Section 3. The formulation of the proposed product manifold is presented in Section 4 and is further elaborated with examples in Section 5. The statistical modeling on manifolds is introduced in Section 6. Section 7 reports our experimental results. Section 8 discusses the effect of using raw pixels for action recognition. Finally, we conclude this paper in Section 9.

2. Related Work

Many researchers have proposed a variety of techniques for action recognition in recent years. We highlight some of this work here, including bag-of-features models, autoregressive models, 3D Fourier transforms, tensor frameworks, and product spaces.

In the context of action recognition, bag-of-features models (Dollar et al., 2005; Wang et al., 2009; Bilinski and Bremond, 2011) may be among the most popular methods wherein visual vocabularies are learned from feature descriptors and spatiotemporal features are typically represented by a normalized histogram. While encouraging results have been achieved, bag-of-features methods have heavy training loads prior to classification. In particular, feature detection and codebook generation can consume tremendous amounts of time if the number of training samples is large. Recently, Wang et al. (2009) have evaluated a number of feature descriptors and bag-of-features models for action recognition. This study concluded that different sampling strategies and feature descriptors were needed to achieve the best results on alternative action data sets. Similar conclusions were also found by Bilinski and Bremond (2011) where various sizes of codebooks are needed for different data sets in order to obtain peak performances.

Another school of thought for action classification is using an autoregressive (AR) model. Some of the earliest works involved dynamic texture recognition (Saisan et al., 2001) and human gait recognition (Bissacco et al., 2001). These works represented actions using AR models. The authors found that the most effective way to compare dynamics was by computing the Martin distance between AR models. Veeraraghavan et al. (2005) modeled human silhouettes based on Kendall's theory of shape (Kendall, 1984) where shapes were expressed on a shape manifold. This method modeled the dynamics of human silhouettes using an AR model on the tangent space of the shape manifold. The sequences of human shapes were compared by computing the distance between the AR models. Turaga and Chellappa (2009) investigated statistical modeling with AR models for human activity analysis. In their work, trajectories were considered a sequence of subspaces represented by AR models on a Grassmann manifold. As such, the dynamics were learned and kernel density functions with Procrustes representation were applied to density estimation.

Three-dimensional Fourier transform has been demonstrated as a valuable tool in action classification. Weinland et al. (2006) employed Fourier magnitudes and cylindrical coordinates to represent motion templates. Consequently, the action matching was invariant to translations and rotations

around the z-axis. Although this method was view invariant, the training videos needed to be acquired from multiple cameras. Rodriguez et al. (2008) synthesized a filter respond using the Clifford Fourier transform for action recognition. The feature representation was computed using spatiotemporal regularity flow from the xy-parallel component. The advantage of using Clifford algebra is the direct use of vector fields to Fourier transform.

Data tensors are the multidimensional generalizations to matrices. Vasilescu (2002) modeled the joint angle trajectories on human motion as a set of factorized matrices from a data tensor. Signatures corresponding to motion and identity were then extracted using PCA for person identification. Kim and Cipolla (2009) extended canonical correlation analysis to the tensor framework by developing a Tensor Canonical Correlation Algorithm (TCCA). This method factorized data tensors to a set of matrices and learned a set of projection matrices maximizing the canonical correlations. The inner product was employed to compute the similarity between two data tensors. The use of SIFT features with CCA was also considered for gesture recognition by Kim and Cipolla (2007). Recently, nonnegative tensor factorization has been exploited for action recognition by Krausz and Bauckhage (2010) where action videos were factorized using a gradient descent method and represented as the sum of rank-1 tensors associated with a weighting factor. As a result, the appearance was captured by the basis images and the dynamics was encoded with the weighting factor.

Product spaces have received attention in applications related to spatiotemporal interactions. Datta et al. (2009) modeled the motion manifold as a collection of local linear models. This method learned a selection of mappings to encode the motion manifold from a product space. Lin et al. (2009) proposed a probabilistic framework for action recognition using prototype trees. Shape and motion were explicitly learned and characterized via hierarchical K-means clustering. The joint likelihood framework was employed to model the joint shape-motion space. Li and Chellappa (2010) investigated the product space of spatial and temporal submanifolds for action alignment. Sequential importance sampling was then used to find the optimal alignment. Despite these efforts, the geometry of the product space has not been directly considered and the geodesic nature on the product manifold remains unexamined.

3. Mathematical Background

In this section, we briefly review the background mathematics used in this paper. Particularly, we focus on the elements of tensor algebra, orthogonal groups, Stiefel manifolds, and Grassmann manifolds.

3.1 Tensor Representation

Tensors provide a natural representation for high dimensional data. We consider a video as a third order data tensor $\in \mathbb{R}^{X \times Y \times T}$ where X, Y, and T are the image width, image height, and video length, respectively. High order data tensors can be regarded as a multilinear mapping over a set of vector spaces. Generally, useful information can be extracted using tensor decompositions. In particular, a Higher Order Singular Value Decomposition (HOSVD) (De Lathauwer et al., 2000) is considered in this paper because the data tensor can be factorized in a closed-form. A recent review paper on tensor decompositions can be found in Kolda and Bader (2009). Before we describe HOSVD, we illustrate a building block operation called matrix unfolding.

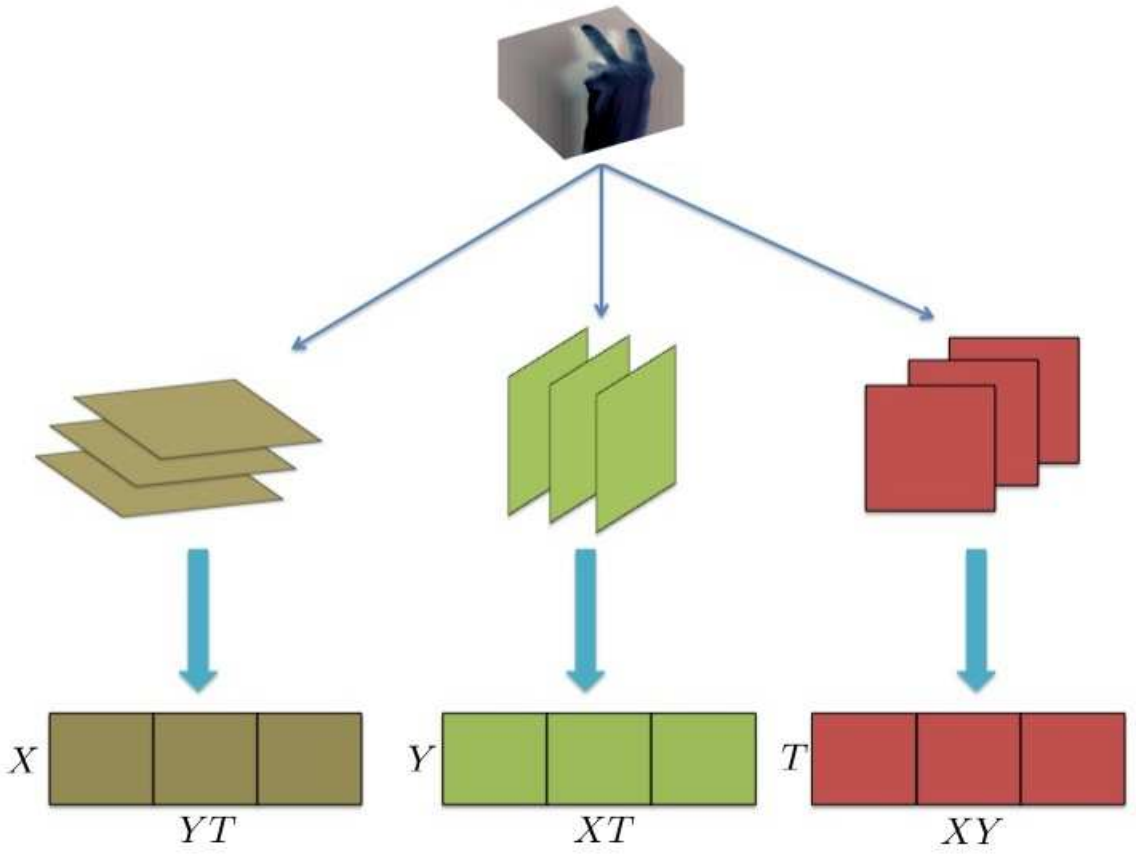


Figure 1: An example of matrix unfolding for a third order tensor. The illustration is for a video action sequence with two spatial dimensions X and Y and a temporal dimension T.

3.1.1 MATRIX UNFOLDING

Let $\mathcal{A}$ be an order N data tensor $\in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. The data tensor $\mathcal{A}$ can be converted to a set of matrices via a matrix unfolding operation. Matrix unfolding maps a tensor $\mathcal{A}$ to a set of matrices $A_{(1)}, A_{(2)}, \ldots, A_{(N)}$, where $A_{(k)} \in \mathbb{R}^{I_k \times (I_1 \times \cdots \times I_{k-1} \times I_{k+1} \cdots \times I_N)}$ is a mode-k matrix of $\mathcal{A}$. An example of matrix unfolding of a third order, that is, $N = 3$, tensor is given in Figure 1. As Figure 1 shows, we can slice a third order tensor in three different ways along each axis and concatenate these slices into three different matrices $A_{(1)}$, $A_{(2)}$, and $A_{(3)}$ where the rows of an unfolded matrix are represented by a single variation of the tensor and the columns are composed by two variations of the tensor.

3.1.2 HIGHER ORDER SINGULAR VALUE DECOMPOSITION

Just as a data matrix can be factorized using a Singular Value Decomposition (SVD), a data tensor can also be factorized using Higher Order Singular Value Decomposition (HOSVD), also known as multilinear SVD. HOSVD operates on the unfolded matrices $A_{(k)}$, and each unfolded matrix may

be factored using SVD as follows:

$$A_{(k)} = U^{(k)}\Sigma^{(k)}V^{(k)^T} \tag{1}$$

where $\Sigma^{(k)}$ is a diagonal matrix, $U^{(k)}$ is an orthogonal matrix spanning the column space of $A_{(k)}$ associated with nonzero singular values, and $V^{(k)}$ is an orthogonal matrix spanning the row space of $A_{(k)}$ associated with nonzero singular values. Then, an N order tensor can be decomposed using HOSVD as follows:

$$\mathcal{A} = \mathcal{S} \times_1 U^{(1)} \times_2 U^{(2)} \ldots \times_n U^{(N)}$$

where $\mathcal{S} \in \mathbb{R}^{(I_1 \times I_2 \times \cdots \times I_N)}$ is a core tensor, $U^{(1)}, U^{(2)}, \ldots, U^{(N)}$ are orthogonal matrices spanning the column space described in (1), and $\times_k$ denotes mode-k multiplication. The core tensor signifies the interaction of mode matrices and is generally not diagonal when the tensor order is greater than two.

3.2 Orthogonal Groups

Matrix Lie groups arise in various kinds of non-Euclidean geometry (Belinfante and Kolman, 1972). The General Linear Group[1] $\mathcal{GL}(n)$ is a set of nonsingular $n \times n$ matrices defined as:

$$\mathcal{GL}(n) = \{Y \in \mathbb{R}^{n \times n} : \det(Y) \neq 0\}.$$

The $\mathcal{GL}(n)$ is closed under a group operation, that is, matrix multiplication. This is because the product of two nonsingular matrices is a nonsingular matrix. Of practical importance here is the fact that elements of $\mathcal{GL}(n)$ are full rank and thus their row and column spaces span $\mathbb{R}^n$. A further subgroup of $\mathcal{GL}(n)$ is the orthogonal group denoted as:

$$\mathcal{O}(n) = \{Y \in \mathbb{R}^{n \times n} : Y^T Y = I\}.$$

It is known that the determinants of orthogonal matrices can be either $+1$ or -1 where the matrices with the determinant of 1 are rotation matrices and the matrices with the determinant of -1 are reflection matrices.

3.3 Stiefel Manifolds

The Stiefel manifold $\mathcal{V}_{n,p}$ is a set of $n \times p$ orthonormal matrices defined as:

$$\mathcal{V}_{n,p} = \{Y \in \mathbb{R}^{n \times p} : Y^T Y = I\}.$$

The Stiefel manifold $\mathcal{V}_{n,p}$ can be considered a quotient space of $\mathcal{O}(n)$ so we can identify an isotropy subgroup H of $\mathcal{O}(n)$ expressed as $\left\{ \begin{bmatrix} I_p & 0 \\ 0 & Q_{n-p} \end{bmatrix} : Q_{n-p} \in \mathcal{O}(n-p) \right\}$ where the isotropy subgroup leaves the element unchanged. Thus, the Stiefel manifold can be expressed as $\mathcal{V}_{n,p} = \mathcal{O}(n) \,/\, \mathcal{O}(n-p)$. From a group theory point of view, $\mathcal{O}(n)$ is a Lie group and $\mathcal{O}(n-p)$ is its subgroup so that $\mathcal{O}(n) \,/\, \mathcal{O}(n-p)$ represents the orbit space. In other words, $\mathcal{V}_{n,p}$ is the quotient group of $\mathcal{O}(n)$ by $\mathcal{O}(n-p)$.

1. In this paper, we are only interested in the field of real number $\mathbb{R}$. Unitary groups may be considered in other contexts.

3.4 Grassmann Manifolds

When we impose a group action of $O(n)$ onto the Stiefel manifold, this gives rise to the equivalence relation between orthogonal matrices so that the elements of Stiefel manifolds are rotation and reflection invariant. In other words, elements are considered being equivalent if there exists a $p \times p$ orthogonal matrix Q_p which maps one point into the other. This equivalence relation can be written as:

$$\lfloor Y \rfloor = \{YQ_p : Q_p \in O(n)\} \tag{2}$$

where $\lfloor Y \rfloor$ is an element on the Grassmann manifold. Therefore, the Grassmann manifold $\mathcal{G}_{n,p}$ is a set of p-dimensional linear subspaces of $\mathbb{R}^n$ and its isotropy subgroup composes all elements of $\left\{ \begin{bmatrix} Q_p & 0 \\ 0 & Q_{n-p} \end{bmatrix} : Q_p \in O(p), Q_{n-p} \in O(n-p) \right\}$. The quotient representation of Grassmann manifolds is expressed as $\mathcal{G}_{n,p} = O(n) \,/\, (O(p) \times O(n-p)) = \mathcal{V}_{n,p} \,/\, O(p)$. As such, the element of the Grassmann manifold represents the orbit of a Stiefel manifold under the group action of orthogonal groups. More details on the treatment of Grassmann manifolds can be found in Edelman et al. (1998) and Absil et al. (2008).

4. Elements of Product Manifolds

This section discusses the elements of product manifolds in the context of gesture recognition. We illustrate the essence of product manifolds and the factorization of action videos. Further, we describe the realization of geodesic distance on the product manifold and its use for action classification.

4.1 Product Manifolds

A product manifold can be recognized as a complex compound object in a high dimensional space composed by a set of lower dimensional objects. For example, the product of a line with elements y in $\mathbb{R}^1$ and a solid circle with elements x in $\mathbb{R}^2$ becomes a cylinder with elements (x, y) in $\mathbb{R}^3$ as shown in Figure 2. Formally, this product topology can be expressed as:

$$\begin{aligned} I &= \{y \in \mathbb{R} : |y| < 1\}, \\ D^2 &= \{x \in \mathbb{R}^2 : |x| < 1\}, \\ D^2 \times I &= \{(x,y) \in \mathbb{R}^2 \times \mathbb{R} : |x| < 1 \text{ and } |y| < 1\} \end{aligned}$$

where D^2 and I are viewed as topological spaces.

The cylinder may be equally well interpreted as either a circle of intervals or an interval of circles. In general, a product manifold may be viewed as the cross section of lower dimensional objects. Formally, let $\mathcal{M}_1$, $\mathcal{M}_2$, ... , $\mathcal{M}_q$ be a set of manifolds. The set $\mathcal{M}_1 \times \mathcal{M}_2 \times \ldots \times \mathcal{M}_q$ is called the product of the manifolds where the manifold topology is equivalent to the product topology. Hence, a product manifold is defined as:

$$\begin{aligned} \mathcal{M} &= \mathcal{M}_1 \times \mathcal{M}_2 \times \cdots \times \mathcal{M}_q \\ &= \{(x_1, x_2, \ldots, x_q) : x_1 \in \mathcal{M}_1, x_2 \in \mathcal{M}_2, \ldots, x_q \in \mathcal{M}_q\} \end{aligned}$$

where $\times$ denotes the Cartesian product, $\mathcal{M}_k$ represents a factor manifold (a topological space), and x_k is an element in $\mathcal{M}_k$. Note that the dimension of a product manifold is the sum of all factor manifolds (Lee, 2003).

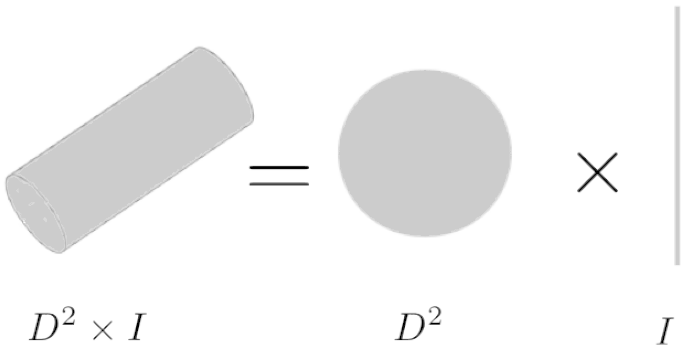


Figure 2: An example of a product manifold: A cylinder is a cross product of a circle and an interval.

The product manifold naturally expresses a compound topological space associated with a number of factor manifolds. For action video classification, third order data tensors are manifested as elements on three factor manifolds. As such, video data can be abstracted as points and classified on a product manifold.

4.2 Factorization in Product Spaces

As discussed in Section 3, HOSVD operates on the unfolded matrices (modes) via matrix unfolding in which the variation of each mode is captured by HOSVD. However, the traditional definition of HOSVD does not lead to a well-defined product manifold in the context of action recognition.

We observe that the column of every unfolded matrix $A_{(k)}$ is composed by multiple orders from the original data tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. This fact can also be observed in Figure 1. Let m be the dimension of the columns, $I_1 \times I_2 \times \cdots \times I_{k-1} \times I_{k+1} \cdots \times I_N$, and p be the dimension of the rows, I_k, for an unfolded matrix $A_{(k)}$. We can then assume that the dimension of the columns is greater than the dimension of the rows due to the nature of matrix unfolding for action videos, that is, $m > p$. This implies that the unfolded matrix $A_{(k)}$ only spans p dimensions.

Alternatively, one can factorize the data tensor using the right orthogonal matrices (Lui et al., 2010). From the context of action videos, the HOSVD can be expressed as:

$$\mathcal{A} = \hat{\mathcal{S}} \times_1 V^{(1)}_{\text{horizontal-motion}} \times_2 V^{(2)}_{\text{vertical-motion}} \times_3 V^{(3)}_{\text{appearance}}$$

where $\hat{\mathcal{S}}$ is a core tensor, $V^{(k)}$ are the orthogonal matrices spanning the row space with the first p rows associated with non-zero singular values from the unfolded matrices, respectively. Because we are performing action recognition on videos, the orthogonal matrices, $V^{(1)}_{\text{horizontal-motion}}$, $V^{(2)}_{\text{vertical-motion}}$, and $V^{(3)}_{\text{appearance}}$, correspond to horizontal motion, vertical motion, and appearance. Figure 3 shows some examples from the action decomposition.

From the factorization of HOSVD, each $V^{(k)}$ is a tall orthogonal matrix, thus it is an element on a Stiefel manifold. When we impose a group action of the orthogonal group, elements on the Stiefel manifold become rotation and reflection invariant. In other words, they are elements on the Grassmann manifold described in (2). As such, the action data are represented as the orbit of

elements on the Stiefel manifold under the rotation and reflection actions with respect to appearance and dynamics. Section 5 will discuss how we benefit from imposing such a group action on the Stiefel manifold.

4.3 Geodesic Distance on Product Manifolds

The geodesic in a product manifold $\mathcal{M}$ is the product of geodesics in $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_q$ (Ma et al., 1998; Begelfor and Werman, 2006). Hence, for any differentiable curve γ parametrized by t, we have $\gamma(t) = (\gamma_i(t), \gamma_j(t))$ where γ is the geodesic on the product manifold $\mathcal{M}$, and γ_i and γ_j are the geodesics on the factor manifold $\mathcal{M}_i$ and $\mathcal{M}_j$ respectively. From this observation, the geodesic distance on a product manifold may be expressed as a Cartesian product of canonical angles computed by factor manifolds.

Just as there are alternatives to induce a metric on a Grassmann manifold (Edelman et al., 1998) using canonical angles, the geodesic distance on a product manifold could also be defined in different ways. One possible choice is the chordal distance that approximates the geodesic via a projection embedding (Conway et al., 1996). Consequently, we define the geodesic distance on a product manifold as:

$$d_{\mathcal{M}}(\mathcal{A}, \mathcal{B}) = \| \sin\Theta \|_2 \tag{3}$$

where $\mathcal{A}$ and $\mathcal{B}$ are the N order data tensors, $\Theta = (\theta_1, \theta_2, \ldots, \theta_N)$, and $\theta_k \in \mathcal{G}_k$ is a set of canonical angles (Björck and Golub, 1973) computed independently from each factor (Grassmann) manifold.

This development of geodesic distance on the product manifold can be related back to our cylinder example where a circle in $\mathbb{R}^2$ and a line in $\mathbb{R}^1$ form a cylinder in $\mathbb{R}^3$ where $\mathbb{R}^3$ is the product space. Recall that a Grassmann manifold is a set of p-dimensional linear subspaces. In analogous fashion, the product of a set of $p_1, p_2, \ldots, p_N$ linear subspaces forms a set of product subspaces whose dimension is $(p_1 + p_2 + \ldots + p_N)$. The product subspaces are the elements on a product manifold. This observation is consistent with the Θ in (3) where the number of canonical angles agrees with the dimension of product subspaces on the product manifold.

Note that canonical angles θ_k are measured between $V_{\mathcal{A}}^{(k)}$ and $V_{\mathcal{B}}^{(k)}$ where each is an orthogonal matrix spanning the row space associated with nonzero singular values from a mode-k unfolded matrix. As such, an N order tensor in $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ would span N row spaces in $I_1, I_2, \ldots, I_N$, respectively, and the dimension of a product manifold is the sum of each order of a data tensor, that is, $(\sum_{i=1}^{N} = I_1 + I_2 + \ldots + I_N)$.

5. The Product Manifold Representation

The tensor representation on a product manifold models the variations in both space and time for action videos. Specifically, the product manifold captures the individual characteristics of spatial and temporal evolution through three factor manifolds. As such, one factor manifold is acquiring the change in time, resulting in the appearance (XY) component, while the other two capture the variations in horizontal and vertical directions, demonstrating the horizontal motion (YT) and vertical motion (XT). Putting all these representations together, geodesic distance on the product manifold measures the changes in both appearance and dynamics.

The aim of this section is to illustrate how the product manifold characterizes appearance and dynamics from action videos. To visualize the product manifold representation, let us consider

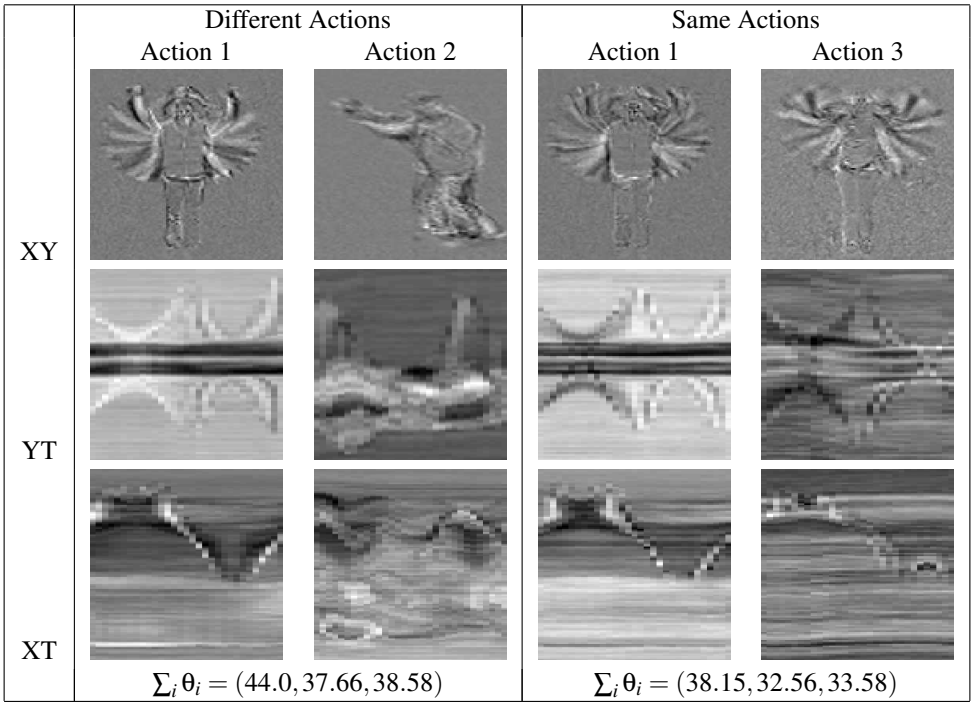


Figure 3: Examples of appearance and motion changes where the first row is the overlay appearances, the second and third rows are the overlay horizontal motion and vertical motion, and the bottom row gives the sum of canonical angles computed from each factorization of the pairs of canonical variates.

the example given in Figure 3 where the first row expresses the pairs of overlay appearance (XY) canonical variates, the second and third rows reveal the pairs of overlay horizontal motion (YT) and vertical motion (XT) canonical variates, and the bottom row gives the sum of canonical angles computed from the pairs of canonical variates. Note that the canonical variates are elements on Stiefel manifolds. In the first column, two distinct actions are factorized to canonical variates. We can see that all canonical variates exhibit very different characteristics in both appearance and motions. On the contrary, the second column shows the same action performed by different actors and the canonical variates are much more similar than the first column, resulting in smaller canonical angles overall.

One of the advantages of the product manifold representation is that actions do not need to be aligned in temporal space. To demonstrate this merit, we permute the frame order from action 3 denoted as action 4 and match it to action 1. Figure 4 shows the pairs of canonical variates between actions (1, 3) and actions (1, 4). We should first note that the appearance (XY) of action 3 and action 4 span the same space despite the visual differences resulting in the identical sum of canonical angles 38.15. This is because elements on the Grassmann manifold are rotation and reflection invariant from elements of the Stiefel manifold. This important concept is illustrated in Figure 5 where the exchange matrix $O(p)$ maps the appearance of action 4 to the appearance of action 3.

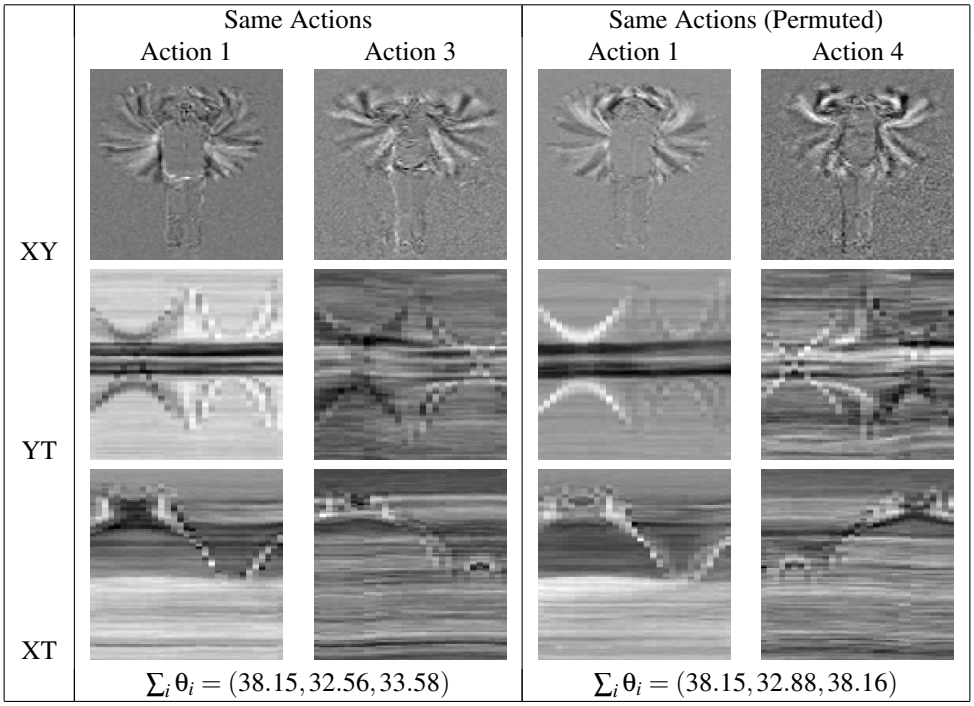


Figure 4: Examples of appearance and motion changes where Action 4 is a permuted version of Action 3. The canonical angles for the appearance indicates that the action is not affected by the frame order.

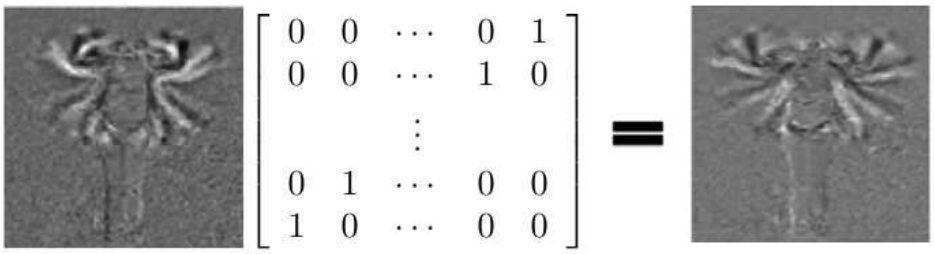


Figure 5: The characterization of the Grassmann manifold where a point is mapped to another point on the Stiefel manifold via an exchanged matrix. The group action is $(X, Q) \longmapsto XQ$ where $X \in \mathcal{V}_{n,p}$ and $Q \in O(p)$ so that elements on the Grassmann manifold are closed under the orthogonal matrix multiplication.

In the example given in Figure 4, the most prominent change is related to the motion in vertical directions (XT) between action 3 and action 4. This arises from the fact that the change of motion mostly occurs in the vertical direction when we permute the order of the video frames from action 3. Consequently, the sum of canonical angles in XT varies from 33.58 to 38.16 which is less similar to action 1. When we identify a waving hand moving from top to bottom and from bottom to

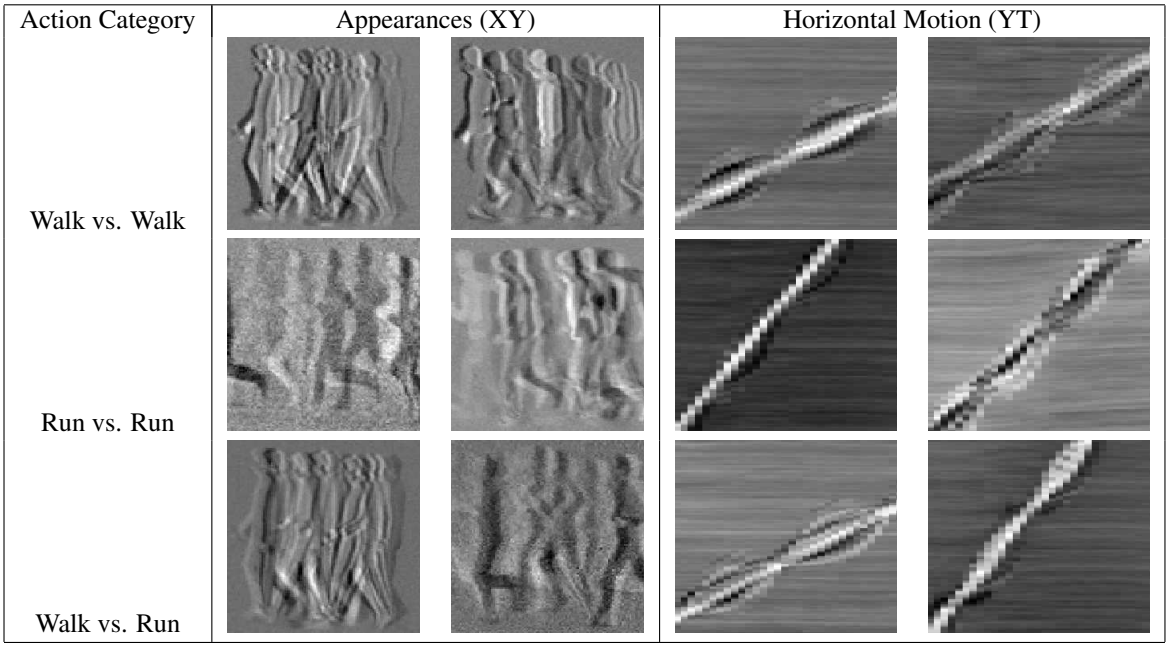


Figure 6: Illustration of capturing the rate of actions. The first column shows the change of appearance while the second column reveals the change of horizontal motion where the slopes exhibit the rate of motion.

top, the vertical motion is the key feature. Otherwise, a simple cyclical search can compensate such variation. As a result, the product manifold representation is resilient to misregistration in the temporal space for appearance while keeping the dynamics intact.

Another intriguing attribute of the product manifold representation is its ability to capture the rate of motion, which is useful in identifying some particular actions. Figure 6 shows the pairs of canonical variates of two similar actions - walking and running. First, we note that there is little information from the vertical motion since the movements of walking and running occur horizontally. The appearance differences between walking and running are not substantial, which is shown in the first column of Figure 6. The key information between walking and running is embedded in the horizontal motion (YT). While the structure of horizontal motion between walking and running is similar exhibiting a line-like pattern, they have very distinct slopes shown in the horizontal motion column of Figure 6. These slopes characterize the rate of motion and are the key factors in recognizing these types of actions. In particular, when walking and running are compared depicted in the third row of Figure 6, the idiosyncratic aspect is captured by the rate of horizontal motion. In general, it is possible to see the rate of motion through both motion representations depending on the type of actions.

6. Statistical Modeling

Least squares regression is one of the fundamental techniques in statistical analysis. It is simple and often outperforms complicated models when the number of training samples is small (Hastie

et al., 2001). Since video data do not reside in Euclidean space, we pay attention to the manifold structure. Here, we introduce a nonlinear regression framework in non-Euclidean space for gesture recognition. We formulate least squares regression as a composite function; as such, both domain and range values are constrained on a manifold through the regression process. The least squares fitted elements from a training set can then be exploited for gesture recognition.

6.1 Linear Least Squares Regression

Before we discuss the geometric extension, we will first review the standard form of least squares fitting. We consider a regression problem $y = A\beta$ where $y \in \mathbb{R}^n$ is the regression value, $A([a_1|a_2|\cdots|a_k]) \in \mathbb{R}^{n\times k}$ is the training set, and $\beta \in \mathbb{R}^k$ is the fitting parameter. The residual sum-of-squares can be written as:

$$R(\beta) = \| y - A\beta \|^2 \tag{4}$$

and the fitting parameter β can be obtained by minimizing the residual sum-of-squares error from (4). Then, we have

$$\hat{\beta} = (A^T A)^{-1} A^T y.$$

The regressed pattern from the training set has the following form

$$\hat{y} = A\hat{\beta} = A(A^T A)^{-1} A^T y. \tag{5}$$

The key advantage of least squares fitting is its simplicity and it intuitively measures the best fit of the data.

6.2 Least Squares Regression on Manifolds

Non-Euclidean geometry often arises in computer vision applications. We consider the nonlinear nature of space and introduce a geometric framework for least squares regression. First, we extend the linear least squares regression from (5) to a nonlinear form by incorporating a kernel function shown in the following

$$A(A \star A)^{-1}(A \star y)$$

where $\star$ is a nonlinear similarity operator. Obviously, $\star$ is equal to $x^T y$ in the linear case. In this paper, we employ the RBF kernel given as:

$$x \star y \quad = \quad \exp(-\frac{\sum_k \theta_k}{\sigma}) \tag{6}$$

where x and y are the elements on a factor manifold, θ_k is the canonical angle computed from the factor manifold, and σ is set to 2 in all our experiments. While other kernel functions can be considered, our goal is to demonstrate our geometric framework and choose a commonly used RBF kernel operator.

Considering the similarity measure given in (6), the regression model becomes three sub-regression estimators given by

$$\psi^{(k)}(y) = A^{(k)}(A^{(k)} \star A^{(k)})^{-1}(A^{(k)} \star y^{(k)}) \tag{7}$$

Algorithm 1 Weighted Karcher Mean Computation

1: Initialize a base point μ on a manifold
2: **while** not converged **do**
3: Apply the logarithmic map to the training samples Y_i to the base point μ
4: Compute the weighted average on the tangent space at the base point μ
5: Update the base point μ by applying the exponential map on the weighted average
6: **end while**

where k denotes the mode of unfolding, $A^{(k)}$ is a set of orthogonal matrices factorized from HOSVD, and $y^{(k)}$ is an orthogonal matrix from the unfolded matrix.

To gain a better insight on the regression model, we explore the geometrical interpretation from (7). Given p training instances, the first element, $A^{(k)}$, is a set of factorized training samples residing on a manifold. Furthermore, $(A^{(k)} \star A^{(k)})^{-1}$ produces a $p \times p$ matrix from the training set and $(A^{(k)} \star y^{(k)})$ would create a $p \times 1$ vector. Therefore, the rest of the regression provides a weighting vector characterizing the training data on a factor manifold as:

$$\mathrm{w} = (A^{(k)} \star A^{(k)})^{-1}(A^{(k)} \star y^{(k)})$$

where the weighting vector is in a vector space, that is, $\mathrm{w} \in \mathcal{V}$.

Now, we have a set of factorized training samples, $A^{(k)}$, on a manifold and a weighting vector, w, in a vector space. To incorporate these two elements with the least squares fitting given in (7), we make a simple modification and reformulate the regression as follows

$$\Psi^{(k)}(y) = A^{(k)} \bullet (A^{(k)} \star A^{(k)})^{-1}(A^{(k)} \star y^{(k)}) \tag{8}$$

where $\bullet$ is an operator mapping points from a vector space back to a factor manifold. By introducing an additional operator, we ensure that both the domain values $y^{(k)}$ and the range values $\Psi^k(y)$ reside on a manifold. From a function composition point of view, the proposed regression technique can be viewed as a composition map $\mathcal{G} \circ \mathcal{H}$ where $\mathcal{H} : \mathcal{M} \longrightarrow \mathcal{V}$ and $\mathcal{G} : \mathcal{V} \longrightarrow \mathcal{M}$ where $\mathcal{M}$ is a manifold and $\mathcal{V}$ is a vector space.

One possible way to realize the composition map, $\mathcal{G} \circ \mathcal{H}$, is to employ the tangent space and modify the Karcher mean (Karcher, 1977). The computation of Karcher mean considers the intrinsic geometry and iteratively minimizes the distance between the updated mean and all data samples via the tangent space. Since w is the weighting vector, it naturally produces the weight between training samples. All we need is to apply the weighting vector to weight the training samples on a factor manifold. This is equivalent to computing the weighted Karcher mean, which is an element of a manifold.

So far, our geometric formulation on least squares regression is very general. To make it specific for gesture recognition, we impose rotation and reflection invariance to the factorized element $V^{(k)}$ such that they are elements on a Grassmann manifold and the computation of the weighted Karcher mean can be realized. Here, we sketch the pseudo-code in Algorithm 1. As Algorithm 1 illustrates, the first step is to initialize a base point on a manifold. To do so, we compute the weighted average from the training samples in Euclidean space and project it back to the Grassmann manifold using QR factorization. Then, we iteratively update the base point on the Grassmann manifold. The update procedure involves the standard logarithmic map and the exponential map on Grassmann

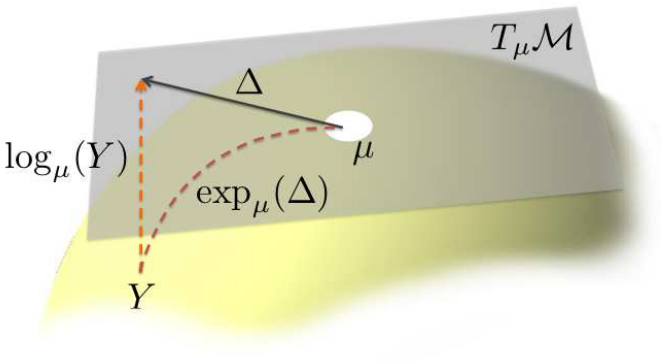


Figure 7: An illustration of logarithmic and exponential maps where Y and μ are points on a manifold, Δ is the tangent vector, and $T_\mu\mathcal{M}$ is the tangent space at μ.

manifolds (Edelman et al., 1998) described as follows

$$\log_\mu(Y_i) \quad = \quad U_1\Theta_1V_1^T$$

where μ is the base point for the tangent space, Y_i is a training instance factorized from the Grassmann manifold, $\mu_\perp\mu_\perp^TY_i(\mu^TY_i)^{-1} = U_1\Sigma_1V_1^T$, $\Theta_1 = \arctan(\Sigma_1)$, and $\mu_\perp$ is the orthogonal complement to μ.

$$\exp_\mu(\Delta) \quad = \quad \mu V_2\cos(\Sigma_2) + U_2\sin(\Sigma_2)$$

where Δ is the weighted tangent vector at μ and $\Delta = U_2\Sigma_2V_2^T$. From a geometric point of view, the logarithmic operator maps a point on a manifold to a tangent space whereas the exponential map projects a point in the tangent space back to the manifold. A pictorial illustration is given in Figure 7. In addition, the Karcher mean calculation exhibits fast convergence (Absil et al., 2004). Typically, convergence can be reached within 10 iterations in our experiments. A sample run is depicted in Figure 8 where expeditious reduction of residuals occurs in the first few iterations.

To perform gesture recognition, a set of training videos is collected. All videos are normalized to a standard size. During the test phase, the category of a query video is determined by

$$j^* = \underset{j}{\operatorname{argmin}}\, \mathcal{D}(Y, \Psi_j(Y))$$

where Y is a query video, Ψ_j is the regression instance for the class j given in (8), and $\mathcal{D}$ is a geodesic distance measure. Because the query gesture Y and the regression instance are realized as elements on a product manifold, we employ the chordal distance given in (3) for gesture classification.

In summary, the least squares regression model applies HOSVD on a query gesture Y and factorizes it to three sub-regression models ($\Psi_j^{(1)}$, $\Psi_j^{(2)}$, $\Psi_j^{(3)}$) on three Grassmann manifolds where regressions are performed. The distance between the regression output and query is then characterized on a product manifold; gesture recognition is achieved using the chordal distance. We note that

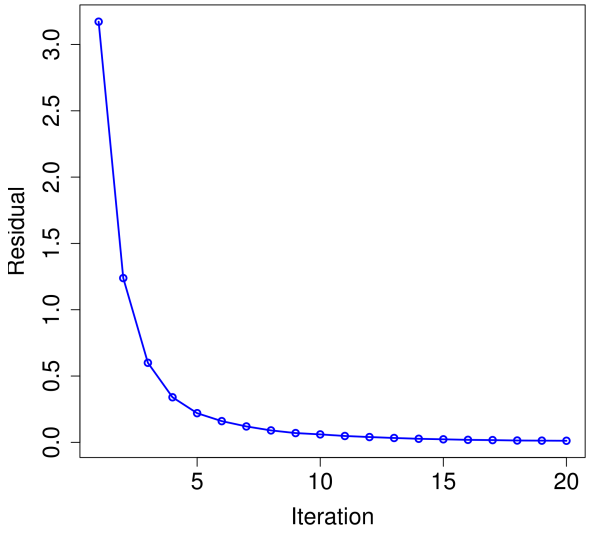


Figure 8: The residual values of tangent vectors.

our least squares framework is applicable to many matrix manifolds as long as the logarithmic and exponential maps are well-defined. Furthermore, when the kernel operator is $\star = x^T y$, $\log_x(y) = y$, and $\exp_x(\Delta) = x + \Delta$, the regression model in (8) becomes the canonical least squares regression in Euclidean space.

When statistical models exhibit high variance, shrinkage techniques are often applied (Hastie et al., 2001). We see that a simple regularization parameter turns least squares regression into ridge regression. This observation can also be applied to our non-Euclidean least squares regression framework.

7. Experimental Results

This section summarizes our empirical results and demonstrates the proficiency of our framework on gesture recognition. To facilitate comparison, we first evaluate our method using two publicly available gesture data sets namely Cambridge hand-gesture (Kim and Cipolla, 2009) and UMD Keck body-gesture (Lin et al., 2009). We further extend our method to the one-shot-learning gesture challenge (CHALEARN, 2011). Our experiments reveal that not only does our method perform well on the standard benchmark data sets, but also it generalizes well on the one-shot-learning gesture challenge.

7.1 Cambridge Hand-Gesture Data Set

Our first experiment is conducted using the Cambridge hand-gesture data set which has 900 video sequences with nine different hand gestures (100 video sequences per gesture class). The gesture data are collected from five different illumination sets labeled as Set1, Set2, Set3, Set4, and Set5. Example gestures are shown in Figure 9.

Figure 9: Hand gesture samples. Flat-Leftward, Flat-Rightward, Flat-Contract, Spread-Leftward, Spread-Rightward, Spread-Contract, V-Shape-Leftward, V-Shape-Rightward, and V-Shape-Contract.

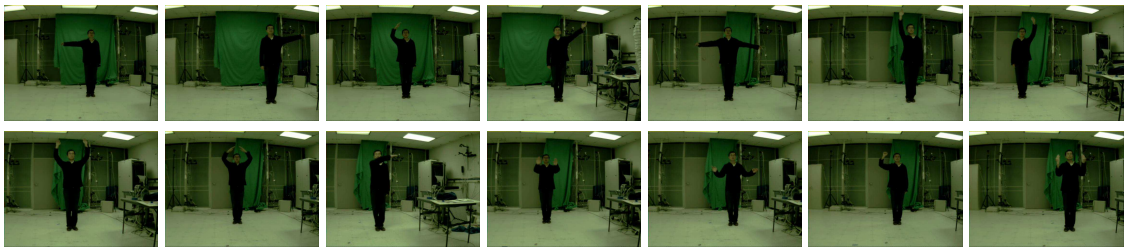


Figure 10: Body gesture samples. First row: Turn Left, Turn Right, Attention Left, Attention Right, Attention Both, Stop Left, and Stop Right. Second row: Stop Both, Flap, Start, Go Back, Close Distance, Speed Up, and Come Near.

We follow the experimental protocol employed by Kim and Cipolla (2009) where Set5 is the target set, and Set1, Set2, Set3, and Set4 are the test sets. The target Set5 is further partitioned into a training set and validation set (90 video sequences in the training set and 90 video sequences in the validation set). We employ five random trials in selecting the training and validation videos in Set5. The recognition results are summarized in Table 1 where the classification rates are the average accuracy obtained from five trial runs followed by the standard deviation. As Table 1 shows, our method performs very well across all illumination sets obtaining 91.7% average classification rate.

7.2 UMD Keck Body-Gesture Data Set

The UMD Keck body-gesture data set consists of 14 naval body gestures acquired from both static and dynamic backgrounds. In the static background, the subjects and the camera remain stationary whereas the subjects and the camera are moving in the dynamic environment during the performance of the gesture. There are 126 videos collected from the static scene and 168 videos taken from the dynamic environment. Example gestures are given in Figure 10.

We follow the experimental protocol proposed by Lin et al. (2009) for both static and dynamic settings. The region of interest is tracked by a simple correlation filter. In the static background, the protocol is leave-one-subject-out (LOSO) cross-validation. As for the dynamic environment, the gestures acquired from the static scene are used for training while the gestures collected from the dynamic environment are the test videos. The recognition results for both static and dynamic backgrounds are reported in Table 2. We can see that our method is competitive to the current state-of-the-art methods in both protocols. One of the key advantages of our method is its direct use of raw pixels while the prototype-tree (Lin et al., 2009), MMI-2+SIFT (Qiu et al., 2011), and CC K-

Method	Set1	Set2	Set3	Set4	Total
Graph Embedding (Yuan et al., 2010)	-	-	-	-	82%
TCCA (Kim and Cipolla, 2009)	81%	81%	78%	86%	82±3.5%
DCCA+SIFT (Kim and Cipolla, 2007)	-	-	-	-	85±2.8%
RLPP (Harandi et al., 2012)	86%	86%	85%	88%	86.3±1.3%
TB$\{\mathcal{V}_{n,p}\}$ (Lui, 2012b)	88%	84%	85%	87%	86±3.0%
PM 1-NN (Lui et al., 2010)	89%	86%	89%	87%	88±2.1%
Our Method	93%	89%	91%	94%	91.7±2.3%

Table 1: Recognition results on the Cambridge Hand-Gesture data set (Five trial runs).

Method	Static Setting	Dynamic Setting
HOG3D (Bilinski and Bremond, 2011)	-	53.6%
Shape Manifold (Abdelkadera et al., 2011)	82%	-
MMI-2+SIFT (Qiu et al., 2011)	95%	-
CC K-Means (Jiang et al., 2012))	-	92.9%
Prototype-Tree (Lin et al., 2009)	95.2%	91.1%
TB$\{\mathcal{V}_{n,p}\}$ (Lui, 2012b)	92.1%	91.1%
PM 1-NN (Lui et al., 2010)	92.9%	92.3%
Our Method	94.4%	92.3%

Table 2: Recognition results on the UMD Keck Body-Gesture data set.

means (Jiang et al., 2012) methods operate on silhouette images which require image segmentation prior to classification. This makes our representation more generic.

7.3 One-Shot-Learning Gesture Challenge

Microsoft Kinect™ has recently revolutionized gesture recognition by providing both RGB and depth images. To facilitate the adaptation to new gestures, CHALEARN (Guyon et al., 2012) has organized a one-shot-learning challenge for gesture recognition.

The key aspect of one-shot-learning is to perform machine learning on a single training example. As such, intra-class variability needs to be modeled from a single example or learned from different domains. While traditional machine learning techniques require a large amount of training data to model the statistical distribution, least squares regression appears to be more robust when the size of training samples is limited (Hastie et al., 2001). We employ our least squares regression framework and model the intra-class variability by synthesizing training examples from the original training instance. Consequently, we apply the same regression framework on the product manifold to the one-shot-learning gesture challenge.

One of the gesture variations is performing gesture positions. Our initial studies for frame alignment did not yield positive results due in part to the incidental features of the upper body. Since gesture positions are the key source of variations, we synthesize training examples for translational instances on both RGB and depth images. The synthesized examples are generated by shifting the entire action video horizontally and vertically. Specifically, we synthesize two vertically (up/down) and four horizontally (left/right) translated instances along with the original training example. As such, we have seven training instances for RGB and depth images, respectively. We stress that we

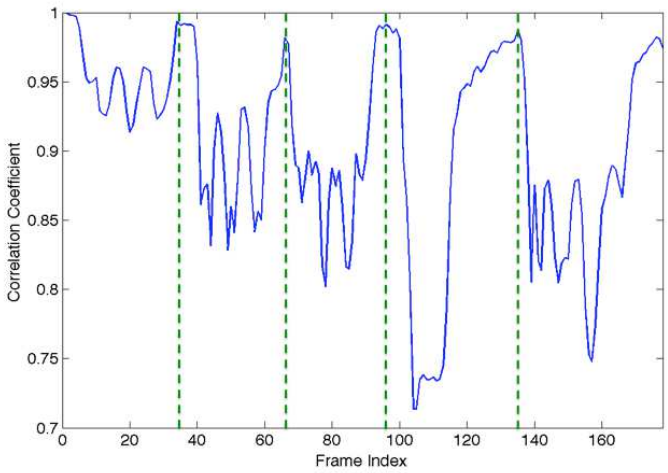


Figure 11: An illustration of temporal segmentation where the dash lines indicate the peak locations and the resting frames from the action sequence.

do not apply any spatial segmentation or intensity normalization to video data; alignment is the only variation that we synthesize for one-shot-learning. Our experiments on the training batches indicate that there is about 2% gain by introducing the translational variations.

We assess the effectiveness of the proposed framework on the development data set for the one-shot-learning gesture challenge. The development data set consists of 20 batches of gestures. Each batch is made of 47 gesture videos and split into a training set and a test set. The training set includes a small set of vocabulary spanning from 8 to 15 gestures. Every test video contains 1 to 5 gestures. Detailed descriptions of the gesture data can be found in Guyon et al. (2012).

Since the number of gestures varies for test videos, we perform temporal segmentation to localize each gesture segment. It is supposed that the actor will return to the resting position before performing a new gesture. Thus, we employ the first frame as a template and compute the correlation coefficient with subsequent frames. We can then localize the gesture segments by identifying the peak locations from the correlations; the number of gestures is the number of peaks + 1. An illustration of temporal segmentation is given in Figure 11 where the peak locations provide a good indication for the resting frames. Furthermore, we fix the spatial dimension to 32×32 and dynamically determine the number of frames by selecting 90% of the PCA energy from each training batch. Linear interpolation is then applied to normalize the video length.

The recognition performance is evaluated using the Levenshtein distance (Levenshtein, 1966), also known as edit distance. Table 3 shows the average errors over 20 batches. As Table 3 reveals, our method significantly outperforms the baseline algorithm (CHALEARN, 2011) and achieves 28.73% average Levenshtein distance per gesture on the development data set. Our method also ranks among the top algorithms in the gesture challenge (Guyon et al., 2012). This illustrates that our method can be effectively adopted for one-shot-learning from the traditional supervised learning paradigm.

	Baseline		Our Method	
Batch	TeLev%	TeLen%	TeLev%	TeLen%
devel01	53.33	12.22	13.33	4.44
devel02	68.89	16.67	35.56	14.44
devel03	77.17	5.43	71.74	20.65
devel04	52.22	30.00	10.00	2.22
devel05	43.48	10.87	9.78	7.61
devel06	66.67	17.78	37.78	14.44
devel07	81.32	19.78	18.68	3.30
devel08	58.43	12.36	8.99	5.62
devel09	38.46	9.89	13.19	1.10
devel10	75.82	21.98	50.55	1.10
devel11	67.39	18.48	35.87	2.17
devel12	52.81	5.62	22.47	4.49
devel13	50.00	17.05	9.09	2.27
devel14	73.91	22.83	28.26	3.26
devel15	50.00	8.70	21.74	0.00
devel16	57.47	17.24	31.03	6.90
devel17	66.30	32.61	30.43	4.35
devel18	70.00	28.89	40.00	11.11
devel19	71.43	15.38	49.45	3.30
devel20	70.33	36.26	35.16	12.09
Average	62.32	18.01	28.73	6.24

Table 3: Recognition results on the development data for the one-shot-learning challenge where TeLev is the sum of the Levenshtein distance divided by the true number of gestures and TeLen is the average error made on the number of gestures.

While our method performs well on the one-shot-learning gesture challenge, it is not a complete system yet. There are three particular batches that cause difficulties for our algorithm. These batches are devel03, devel10, and devel19 where the example frames are shown in Figure 12. These three batches share a common characteristic that the gesture is only distinguishable by identifying the hand positions. Since we do not have a hand detector, the gross motion dominates the whole action causing it to be confused with other similar gestures.

Another source of errors is made by the temporal segmentation. While the actor is supposed to return to the resting position before performing a new gesture, this rule has not always been observed. As a result, such variation introduces a mismatch between the template and subsequent frames resulting errors in partitioning the video sequence. The large error in devel03 is caused by the need for hand positions and temporal segmentation. Future work will focus on combining both appearance and motion for temporal segmentation.

Nevertheless, the experimental results from the Cambridge hand-gesture and the UMD Keck body-gesture data sets are encouraging. These findings illustrate that our method is effective in both hand gestures and body gestures. Once we have a reliable hand detector, we expect to further improve gesture recognition from a single training example. Currently, the processing time on 20 batches (2,000 gestures) including both training and testing is about 2 hours with a non-optimized MATLAB implementation on a 2.5GHz Intel Core i5 iMac.

Figure 12: Gesture samples on the one-shot-learning gesture challenge (devel03, devel10, and devel19).

8. Discussion

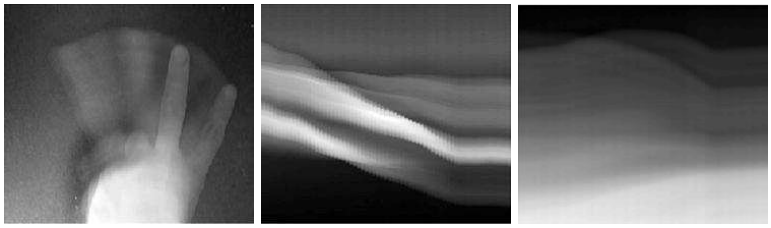
The proposed method is geometrically motivated. It decomposes a video tensor to three Stiefel manifolds via HOSVD where the orthogonal elements are imposed to Grassmannian spaces. As mentioned before, one of the key advantages of our method is its direct use of raw pixels. This gives rise to a practical and important question. *How robust can the raw pixel representation be against background clutter?*

To address this concern, we synthesize an illustrative example given in Figure 13. The first, second, and third columns depict the appearance, horizontal motion, and vertical motion of the gesture, respectively. A V-shape rightward gesture and a flat leftward gesture are shown in the first row and second row. We superpose a cluttered background on every frame of the flat leftward gesture exhibited in the third row. While the appearances between the uniform flat gesture and the cluttered flat gesture emerge differently, the deterioration on the dynamics is quite minimal. As a result, the gesture performed with the background clutter can still be discriminated against other gestures. Numerically, the sum of the canonical angles between the uniform (second row) and the cluttered background (third row) gestures is (56.09, 7.99, 9.17) resulting in a geodesic distance of 5.91 on the product manifold. In contrast, the sum of the canonical angles between the V-shape (first row) and the flat (second row) gestures is (76.35, 23.66, 18.42) yielding a geodesic distance of 8.29. In addition, when the V-shape gesture (first row) matches against the cluttered flat gesture (third row), the sum of the canonical angles is (76.09, 23.75, 18.84) and the geodesic distance is 8.31. This finding reveals that the geodesic distance between the uniform and cluttered background gestures are quite similar against inter-class gestures, while the geodesic distance is significantly smaller for the intra-class gestures. Hence, raw pixels can be directly exploited in our representation.

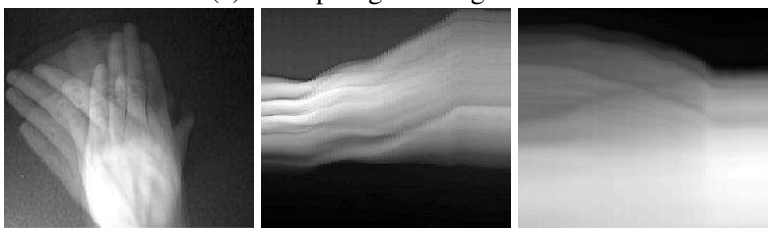
As technology advances, we can now separate the foreground and background more easily using a Kinect™ camera. We hypothesize that better recognition results may be obtained when the foreground gestures are extracted. On the other hand, our method can still perform gracefully when a cluttered background is present.

9. Conclusions

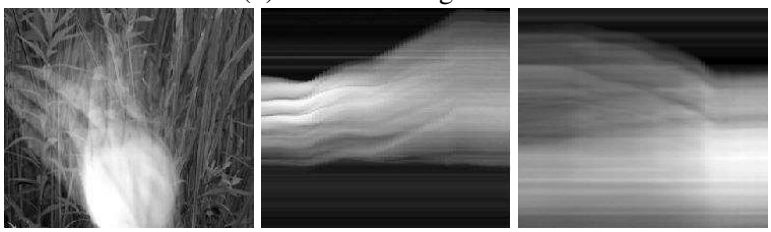
This paper promotes the importance of the underlying geometry of data tensors. We have presented a geometric framework for least squares regression and applied it to gesture recognition. We view action videos as third order tensors and impose them on a product manifold where each factor is



(a) V-shape rightward gesture.



(b) Flat leftward gesture.



(c) Superposed cluttered background on the flat leftward gesture.

Figure 13: The effect of background clutter. Appearance, horizontal motion, and vertical motion are depicted in the first, second, and third columns, respectively.

Grassmannian. The realization of points on these Grassmannians is achieved by applying HOSVD to a tensor representation of the action video. A natural metric is inherited from the factor manifolds since the geodesic on the product manifold is given by the product of the geodesic on the Grassmann manifolds.

The proposed approach provides a useful metric and a regression model based on latent geometry for action recognition. To account for the underlying geometry, we formulate least squares regression as a composite function. This formulation provides a natural extension from Euclidean space to manifolds. Experimental results demonstrate that our method is effective and generalizes well to the one-shot-learning scheme.

For longer video sequences, micro-action detection is needed which may be modeled effectively using HMM. Future work will focus on developing more sophisticated models for gesture recognition and other regression techniques on matrix manifolds for visual applications.

References

M. F. Abdelkadera, W. Abd-Almageeda, A. Srivastavab, and R. Chellappa. Gesture and action recognition via modeling trajectories on riemannian manifolds. *Computer Vision and Image Understanding*, 115(3):439–455, 2011.

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

P.-A Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

E. Begelfor and M. Werman. Affine invariance revisited. In *IEEE Conference on Computer Vision and Pattern Recognition, New York*, 2006.

J.G.F. Belinfante and B. Kolman. *A Survey of Lie Groups and Lie Algebras with Applications and Computational Methods*. SIAM, 1972.

P. Bilinski and F. Bremond. Evaluation of local descriptors for action recognition in videos. In *ICVS*, 2011.

A. Bissacco, A. Chiuso, Y. Ma, and S. Soatto. Recognition of human gaits. In *IEEE Conference on Computer Vision and Pattern Recognition, Hawaii*, pages 270–277, 2001.

Å. Björck and G.H. Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of Computation*, pages 579–594, 1973.

CHALEARN. Chalearn gesture dataset (cgd 2011), chalearn, california, 2011.

J.H. Conway, R.H. Hardin, and N.J.A. Sloane. Packing lines, planes, etc.: Packings in grassmannian spaces. *Experimental Mathematics*, 5(2):139–159, 1996.

A. Datta, Y. Sheikh, and T. Kanade. Modeling the product manifold of posture and motion. In *Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (in conjunction with ICCV)*, 2009.

L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21(4):1253–1278, 2000.

P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (in conjunction with ICCV)*, 2005.

A. Edelman, R. Arias, and S. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20(2):303–353, 1998.

I. Guyon, V. Athitsos, P. Jangyodsuk, B. Hammer, and H. J. E. Balderas. Chalearn gesture challenge: Design and first results. In *CVPR Workshop on Gesture Recognition*, 2012.

M. T. Harandi, C. Sanderson, A. Wiliem, and B. C. Lovell. Kernel analysis over riemannian manifolds for visual recognition of actions, pedestrians and textures. In *WACV*, 2012.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.

Z. Jiang, Z. Lin, and L. Davis. Class consistent k-means: Application to face and action recognition. *Computer Vision and Image Understanding*, 116(6):730–741, 2012.

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

D. Kendall. Shape manifolds, procrustean metrics and complex projective spaces. *Bull. London Math. Soc.*, 16:81–121, 1984.

T-K. Kim and R. Cipolla. Gesture recognition under small sample size. In *Asian Conference on Computer Vision*, 2007.

T-K. Kim and R. Cipolla. Canonical correlation analysis of video volume tensors for action categorization and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8): 1415–1428, 2009.

T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3), September 2009.

B. Krausz and C. Bauckhage. Action recognition in videos using nonnegative tensor factorization. In *International Conference on Pattern Recognition*, 2010.

J. Lee. *Introduction to Smooth Manifolds*. Springer, 2003.

V. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707–710, 1966.

R. Li and R. Chellappa. Group motion segmentation using a spatio-temporal driving force model. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

X. Li, W. Hu, Z. Zhang, X. Zhang, and G. Luo. Robust visual tracking based on incremental tensor subspace learning. In *IEEE International Conference on Computer Vision*, 2007.

Z. Lin, Z. Jiang, and L. Davis. Recognizing actions by shape-motion prototype trees. In *IEEE International Conference on Computer Vision*, 2009.

Y. M. Lui. Advances in matrix manifolds for computer vision. *Image and Vision Computing*, 30 (6-7):380–388, 2012a.

Y. M. Lui. Tangent bundles on special manifolds for action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(6):930–942, 2012b.

Y. M. Lui and J. R. Beveridge. Grassmann registration manifolds for face recognition. In *European Conference on Computer Vision, Marseille, France*, 2008.

Y. M. Lui, J. R. Beveridge, and M. Kirby. Canonical stiefel quotient and its application to generic face recognition in illumination spaces. In *IEEE International Conference on Biometrics : Theory, Applications and Systems, Washington, D.C.*, 2009.

Y. M. Lui, J. R. Beveridge, and M. Kirby. Action classification on product manifolds. In *IEEE Conference on Computer Vision and Pattern Recognition, San Francisco*, 2010.

Y. Ma, J. Košecká, and S. Sastry. Optimal motion from image sequences: A riemannian viewpoint, 1998. Technical Report No. UCB/ERL M98/37, EECS Department, University of California, Berkeley.

S. Mitra and T. Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, Cybernetics - Part C: Applications and Reviews*, 37:311–324, 2007.

Q. Qiu, Z. Jiang, and R. Chellappa. Sparse dictionary-based representation and recognition of action attributes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

M. Rodriguez, J. Ahmed, and M. Shah. Action mach: A spatio-temporal maximum average correlation height filter for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

P. Saisan, G. Doretto, Y-N. Wu, and S. Soatto. Dynamic texture recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

P. Turaga and R. Chellappa. Locally time-invariant models of human activities using trajectories on the grassmannian. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

P. Turaga, S. Biswas, and R. Chellappa. The role of geometry for age estimation. In *IEEE International conference Acoustics, Speech and Signal Processing*, 2010.

M. A. O. Vasilescu. Human motion signatures: Analysis, synthesis, recognition. In *International Conference on Pattern Recognition, Quebec City, Canada*, pages 456–460, 2002.

M. A. O. Vasilescu and D. Terzopoulos. Multilinear image analysis for facial recognition. In *International Conference on Pattern Recognition, Quebec City, Canada*, pages 511–514, 2002.

A. Veeraraghavan, A. K. Roy-Chowdhury, and R. Chellappa. Matching shape sequences in video with applications in human movement analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (12):1896–1909, 2005.

H. Wang, M. Ullah, A Klaser, I. Laptev, and C. Schmid. Evaulation of local spatio-temporal features for action recognition. In *British Machine Vision Conference*, 2009.

D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104:249–257, 2006.

Y. Yuan, H. Zheng, Z. Li, and D. Zhang. Video action recognition with spatio-temporal graph embedding and spline modeling. In *ICASSP*, 2010.