Accurate 3D Action Recognition using Learning on the Grassmann Manifold

Rim Slama, Hazem Wannous, Mohamed Daoudi, Anuj Srivastava

Accurate 3D Action Recognition using Learning on the Grassmann Manifold

Rim Slama[a,b], Hazem Wannous[a,b], Mohamed Daoudi[b,c], Anuj Srivastava[d]

[a] *University Lille 1, Villeneuve d'Ascq, France*
[b] *LIFL Laboratory / UMR CNRS 8022, Villeneuve d'Ascq, France*
[c] *Institut Mines-Telecom / Telecom Lille, Villeneuve d'Ascq, France*
[d] *Florida State University, Departement of Statistics, Tallahassee, USA*

Abstract

In this paper we address the problem of modelling and analyzing human motion by focusing on 3D body skeletons. Particularly, our intent is to represent skeletal motion in a geometric and efficient way, leading to an accurate action-recognition system. Here an action is represented by a dynamical system whose observability matrix is characterized as an element of a Grassmann manifold. To formulate our learning algorithm, we propose two distinct ideas: (1) In the first one we perform classification using a Truncated Wrapped Gaussian model, one for each class in its own tangent space. (2) In the second one we propose a novel learning algorithm that uses a vector representation formed by concatenating local coordinates in tangent spaces associated with different classes and training a linear SVM. We evaluate our approaches on three public 3D action datasets: MSR-action 3D, UT-kinect and UCF-kinect datasets; these datasets represent different

Email addresses: `rim.slama@telecom-lille.fr` (Rim Slama), `hazem.wannous@telecom-lille.fr` (Hazem Wannous), `mohamed.daoudi@telecom-lille.fr` (Mohamed Daoudi), `anuj@stat.fsu.edu` (Anuj Srivastava)

kinds of challenges and together help provide an exhaustive evaluation. The results show that our approaches either match or exceed state-of-the-art performance reaching 91.21% on MSR-action 3D, 97.91% on UCF-kinect, and 88.5% on UT-kinect. Finally, we evaluate the latency, i.e. the ability to recognize an action before its termination, of our approach and demonstrate improvements relative to other published approaches.

Keywords: Human action recognition, Grassmann manifold, observational latency, depth images, skeleton, classification.

1. Introduction

Human action and activity recognition is one of the most active research topics in the computer vision community due to its many challenging issues. The motivation behind the great interest granted to action recognition is the large number of possible applications in consumer interactive entertainment and gaming [1], surveillance systems [2], life-care and home systems [3]. An extensive literature around this domain can be found in a number of fields including pattern recognition, machine learning, and human-machine interaction [4, 5].

The main challenges in action recognition systems are the accuracy of data acquisition and the dynamic modelling of the movements. The major problems, which can alter the way actions are perceived and consequently be recognized, are: occlusions, shadows and background extraction, lighting condition variations and viewpoint changes. The recent release of consumer depth cameras, like Microsoft Kinect, has significantly lighten these difficulties that reduce the action recognition performance in 2D video. These cameras provide in addition to the RGB image a depth stream allowing to discern changes in depth in certain viewpoints.

More recently, Shotton et al. [6] have proposed a real-time approach for estimating 3D positions of body joints using extensive training on synthetic and real depth-streams. The accurate estimation obtained by such a low-cost acquisition depth sensor has provided new opportunities for human-computer-interaction applications, where popular gaming consoles involve the player directly in interaction with the computer. While these acquisition sensors and their accurate data are within everyone's reach, the next research challenge is activity-driven.

In this paper we address the problem of modelling and analyzing human motion in the 3D human joint space. Particularly, our intent is to represent skeletal joint motion in a compact and efficient way that leads to an accurate action recognition. Our ultimate goal is to develop an approach that avoids an overly complex design of feature extraction and is able to recognize actions performed by different actors in different contexts.

Additionally, we study the ability of our approach for reducing latency: in other words, to quickly recognize human actions from the smallest number of frames possible to permit a reliable recognition of the action occurring. Furthermore, we analyze the impact of reducing the number of actions per class in the training set on the classifier's accuracy.

In our approach, the spatio-temporal aspect of the action is considered and each movement is characterized by a structure incorporating the intrinsic nature of the data. We believe that 3D human joint motion data captures useful knowledge to understand the intrinsic motion structure, and a manifold representation of such simple features can provide discriminating structure for action recognition. This leads to manifold-based analysis, which has been successfully used in many computer vision applications such as visual tracking [7] and action recognition in 2D video [8, 9, 10, 11].

Our overall approach is sketched in Figure 1, which has the following modules:

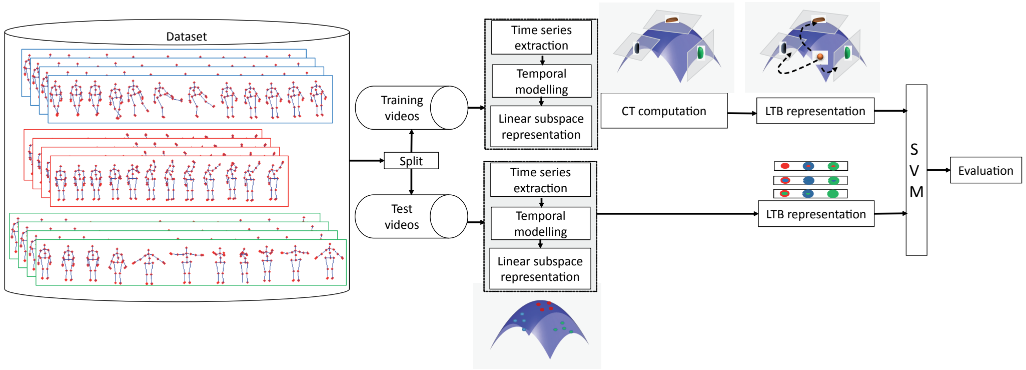


Figure 1: Overview of the approach. The illustrated pipeline is composed of two main modules: (1) temporal modelling of time series data and manifold representation (2) learning approach on the Control Tangent spaces on Grassman manifold, using Local Bundle Tangent representation of data.

First, given training videos recorded from depth camera, motion trajectories from the 3D human joint in Euclidean space are extracted as time series. Then, each motion represented by its time series is expressed as an autoregressive and moving average model (ARMA) in order to model its dynamic process. The subspace spanned by columns of the observability matrix of this model represents a point on a Grassmann manifold.

Second, using the Riemannian geometry of this manifold, we present a solution for solving the classification problem. We studied statistical modelling of inter- and intra-class variations in conjunction with appropriate tangent vectors on this manifold. While class samples are presented by a Grassmann point cloud, we propose to learn Control Tangent (CT) spaces which represent the mean of each class.

Third, each observation of the learning process is projected on all CTs to form a Local Tangent Bandle (LTB) representation. This step allows obtaining a discriminative parameterization incorporating class separation properties and providing the input to a linear SVM classifier.

While given an unknown test video, to recognize its belonging to one of N action classes, we apply the first step on the sequence to represent it as a point on the Grassmann manifold. Then, this point is presented by its LTB as done in learning step. In order to recognize the input action, SVM classifier is performed.

The rest of the paper is organized as follows: In section 1, the state-of-the-art is summarized and main contributions of this paper are highlighted. In section 2, parametric subspace-based modelling of 3D joint-trajectory is discussed. In Section 3, statistical tools developed on a Grassmann manifold are presented and a new supervised learning algorithm is introduced. In Section 4, the strength of the framework in term of accuracy and latency on several datasets are demonstrated. Finally, concluding remarks are presented in Section 5.

2. Related works

In this section two categories of related works are reviewed from two points of view: manifold-based approache and depth data representation.

We first review some related manifold based approaches for action analysis and recognition in 2D video. Then we focus on the most recent methods of action recognition from depth cameras.

2.1. Manifold approaches in 2D videos

Human action modelling from 2D video is a well studied problem in the literature. Recent surveys can be found in the work of Aggarwal et al. [12], Weinland et al. [13], and Poppe [4]. Beside classical methods performed in Euclidean space, a variety of techniques based on manifold analysis are proposed in recent years.

In the first category of manifold based approaches, each frame of action sequence (pose) is represented as an element of a manifold and the whole action is represented as a trajectory on this manifold. These approaches give solutions in the temporal domain to be invariant to speed and time using techniques like Dynamic Time Warping (DTW) to align action trajectories on the manifold. Also probabilistic grammatical models like Hidden Markov Model (HMM) are used to classify these actions presented as trajectories. Indeed, Veeraraghavan et al. [14] propose the use of human silhouettes extracted from video images as a representation of the pose. Silhouettes are then characterized as points on the shape space manifold and modelled by ARMA models in order to compare sequences using a DTW algorithm. In another manifold shape space, Abdelkader et al. [15] represent each pose silhouette as a point on the shape space of closed curves and each gesture is represented as a trajectory on this space. To classify actions, two approaches are used: a template-based approach (DTW) and a graphical model approach (HMM). Other approaches use skeleton as a representation of each frame, as works presented by Gong et al. [16]. They propose a spatio-Temporal Manifold (STM) model to analyze non-linear multivariate time series with latent spatial structure and apply it to recognize actions in the joint-trajectories space. Based on STM, they propose a Dynamic Manifold Warping (DMW) and a motion similarity metric to compare human action sequences both in 2D space using a 2D tracker to extract joints from images and in 3D space using Motion capture data. Recently, Gong et al. [17] propose a Kernelized Temporal Cut (KTC) as an extension of their previous work [16]. They incorporate Hilbert space embedding of distributions to handle the non-parametric and high dimensionality issues.

Some manifold approaches represent the entire action sequence as a point on an other special manifold. Indeed, Turaga et al. [18] involve a study of the geometric properties of the Grassmann and Stiefel manifolds, and give appropriate definitions of Riemannian metrics and geodesics for the purpose of video indexing and action recognition. Then, in order to perform the classification as a probability density function, a mean and a standard-deviation are learnt for each class on class-specific tangent spaces. Turaga et al. [19] use the same approach to represent complex actions by a collection of subsequence. These sub-sequences correspond to a trajectory on a Grassmann manifold. Both DTW and HMM are used for action modelling and comparison. Guo et al. [20] use covariance matrices of bags of low-dimensional feature vectors to model the video sequence. These feature vectors are extracted from segments of silhouette tunnels of moving objects and coarsely capture their shapes.

Without any extraction of human descriptor as silhouette and neither an explicit learning, Lui et al. [21] introduce the notion of tangent bundle to represent each action sequence on the Grassmann manifold. Videos are expressed as a third-order data tensor of raw pixel from action images, which are then factorized on the Grassmann manifold. As each point on the manifold has an associated tangent space, tangent vectors are computed between elements on the manifold and obtained distances are used for action classification in a nearest neighbour fashion. In the same way, Lui et al. [22] factorize raw pixel from images by high-order singular value decomposition in order to represent the actions on Stiefel and Grassmann manifolds. However, in this work where raw pixels are directly factorized as manifold points, there is no dynamic modelling of the sequence. In addition, only distances obtained between all tangent vectors are used for action classification and there is no training process on data.

Kernels [23, 24] are also used in order to transform subspaces of a manifold onto a space where Euclidean metric can be applied. Shirazi et al. [23] embed Grassmann manifolds upon a Hilbert space to minimize clustering distortions and then apply a locally discriminant analysis using a graph. Video action classification is then obtained by a Nearest-Neighbour classifier applied on Euclidean distances computed on the graph-embedded kernel. Similarly, Harandi et al. [24] propose to represent the spatio-temporal aspect of the action by subspaces elements of a Grassmann manifold. Then, they embed this manifold into reproducing kernel of Hilbert spaces in order to tackle the problem of action classification on such manifolds. Gall et al. [25] use multi-view system coupling action recognition on 2D images with 3D pose estimation, were the action-specific manifolds are acting as a link between them.

All these approaches cited above are based on features extracted from 2D video sequences as silhouettes or raw pixels from images. However, the recent emergence of low-cost depth sensors opens the possibility of revisiting the problem of activity modelling and learning using depth data-driven.

2.2. Depth data-driven approaches

Maps obtained by depth sensors are able to provide additional body shape information to differentiate actions that have similar 2D projections from a single view. It has therefore motivated recent research works, to investigate action recognition using the 3D information. Recent surveys [26, 27] are reporting works on depth videos. First methods used for activity recognition from depth sequences have tendency to extrapolate techniques already developed for 2D video sequences. These approaches use points in depth map sequences as a gray pixels in images to extract meaningful spatiotemporal descriptors. In Wanqing et al. [28], depth maps are projected onto the three orthogonal Cartesian planes ($X-Y$, $Z-X$, and $Z-Y$ planes) and the contours of the projections are sampled for each frame. The sampled points are used as *bag-of-points* to characterize a set of salient postures that correspond to the nodes of an *action graph* used to model explicitly the dynamics of the actions. Local feature extraction approaches like spatiotemporal interest points (STIP) are also employed for action recognition on depth videos. Bingbing et al.[29] use depth maps to extract STIP and encode Motion History Image (MHI) in a framework combining color and depth information. Xia et al [30] propose a method to extract STIP a on depth videos (DSTIP). Then around these points of interest they build a depth cuboid similarity feature as descriptor for each action. In the work proposed by Vieira et al. [31], each depth map sequence is represented as a 4D grid by dividing the space and time axes into multiple segments in order to extract SpatioTemporal Occupancy Pattern features (STOP). Also in Wang et al. [32], the action sequence is considered as a 4D shape but Random Occupancy Pattern (ROP) is used for features extraction. Yang et al.[33] employ Histograms of Oriented Gradients features (HOG) computed from Depth Motion Maps (DMM), as the representation of an action sequence. These histograms are then used as input to SVM classifier. Similarly, Oreifej et al. [34] compute a 4D histogram over depth, time, and spatial coordinates capturing the distribution of the surface normal orientation. This histogram is created using 4D projectors allowing quantification in 4D space.

The availability of 3D sensors has recently made possible to estimate 3D positions of body joints. Especially thanks to the work of Shotton et al. [6], where a real-time method is proposed to accurately predict 3D positions of body joints. Thanks to this work, skeleton based methods have become popular and many approaches in the literature propose to model the dynamic of the action using these features.

Xia et al. [35] compute histograms of the locations of 12 3D joints as a compact representation of postures and use them to construct posture visual words of actions. The temporal evolutions of those visual words are modeled by a discrete HMM. Yang et al. [36] extract three features, as pair-wise differences of joint positions, for each skeleton joint. Then, principal component analysis (PCA) is used to reduce redundancy and noise from feature, and it is also used to obtain a compact *Eigen Joints* representation for each frame.

Finally, a naïve-Bayes nearest-neighbour classifier is used for multi-class action classification. The popular Dynamic Time Warping (DTW) technique [37], well-known in speech recognition area, is also used for gesture and action recognition using depth data. The classical DTW algorithm was defined to match temporal distortions between two data trajectories, by finding an optimal warping path between the two time series. The feature vector of time series is directly constructed from human body joint orientation extracted from depth camera or 3D Motion Capture sensors. Reyes et al. [38] perform DTW on a feature vector defined by 15 joints on a 3D human skeleton obtained using PrimeSense NiTE. Similarly, Sempena et al. [39], by the 3D human skeleton model, use quaternions to form a 60-element feature vector. The obtained warping path, by classical DTW algorithm, between two time series is mainly subjected to some constraints: (1) boundary constraint which enforces the first elements of the sequences as well as the last one to be aligned to each other (2) monotonicity constraint which requires that the points in the warping path are monotonically spaced in time in the two sequences. This technique is relatively sensitive to noise as it requires all elements of the sequences to be matched to a corresponding elements of the other sequence. It also has a drawback related to its computational complexity incurring in quadratic cost. However, many works have been proposed to bypass its drawbacks by means of probabilistic models [40] or incorporating manifold learning approach [17, 16].

Recent research has carried on more complex challenge of in-line recognition systems for different applications, in which a trade-off between accuracy and latency can be highlighted. Ellis et al. [41] study this trade-off and employed a Latency Aware Learning (LAL) method, reducing latency when recognizing actions. They train a logistic regression-based classifier, on 3D joint position sequences captured by kinect camera, to search a single canonical posture for recognition. Another work is presented by Barnachon et al. [42], where a histogram-based formulation is introduced for recognizing streams of poses. In this representation, classical histogram is extended to integral one to overcome the lack of temporal information in histograms. They also prove the possibility of recognizing actions even before they are completed using the integral histogram approach. Tests are made on both 3D MoCap from TUM kitchen dataset [43] and RGB-D data from MSR-Action dataset [28].

Some hybrid approaches combining both skeleton data features and depth information were recently introduced, trying to combine positive aspects of both approaches. Azary et al. [44] propose spatiotemporal descriptors as time-invariant action surfaces, combining image features extracted using radial distance measures and 3D joint tracking. Wang et al. [45] compute local features on patches around joints for human body representation. The temporal structure of each joint in the sequence is represented through a temporal pattern representation called *Fourier Temporal Pyramid*. In Oreifej et al. [34], a spatiotemporal histogram (HON4D) computed over depth, time, and spatial coordinates is used to encode the distribution of the surface normal orientation. Similarly to Wang et al. [45], HON4D histograms [34] are computed around joints to provide the input of an SVM classifier. Althloothi et al. [46] represent 3D shape features based on spherical harmonics representation and 3D motion features using kinematic structure from skeleton.

Both feature are then merged using multi kernel learning method.

It is important to note that, to date, few works have very recently proposed to use manifold analysis for 3D action recognition. Devanne et al. [47], propose a spatiotemporal motion representation to characterize the action as a trajectory which corresponds to a point on Riemannian manifold of open curves shape space. These motion trajectories are extracted from 3D joints, and the action recognition is performed by K-Nearest-Neighbor method applied on geodesic distances obtained on open curve shape space. Azary et al. [48] use a Grassmannian representation as an interpretation of depth motion image (DMI) computed from depth pixel values. All DMI in the sequence are combined to create a motion depth surface representing the action as a spatiotemporal descriptor.

2.3. Contributions and proposed approach

On the one hand, approaches modelling actions as elements of manifolds [49, 50, 9] prove that it is an appropriate way to represent and compare videos. On the other hand, very few works deal with this task using depth images and it is still possible to improve learning step using these models. Besides, linear dynamic systems [51] show more and more promising results on the motion modelling since they exhibit the stationary properties in time, so they fit for action representation.

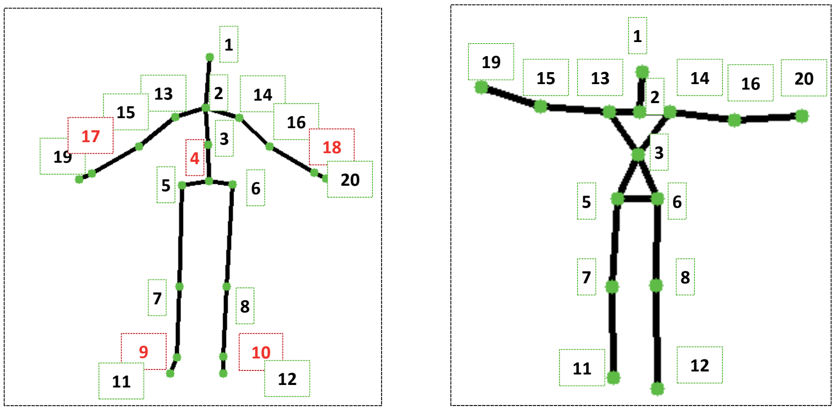
In this paper, we propose the use of geometric structure inherent in the Grassmann manifold for action analysis. We perform action recognition by introducing a manifold learning algorithm in conjunction with dynamic modelling process. In particular, after modelling motions as a linear dynamic systems using ARMA models, we are interested in a representation of each point on the manifold incorporating class separation properties. Our representation takes benefit of statistics in the Grassmann manifold and action classes representations on tangent spaces. From spatiotemporal point of view, each action sequence is represented in our approach as linear dynamical system acquiring the time series of 3D joint-trajectory. From geometrical point of view, each action sequence is viewed as a point on the Grassmann manifold. In terms of machine learning, a discriminative representation is provided for each action thanks to a set of appropriate tangent vectors taking benefit of manifold proprieties. Finally, the efficiency of the proposed approach is demonstrated on three challenging action recognition datasets captured by depth cameras.

3. Spatiotemporal modelling of action

The human body can be represented as an articulated system composed of hierarchical joints that are connected with bones, forming a skeleton. The two best-known skeletons provided by the Microsoft Kinect sensor, are those obtained by official Microsoft SDK, which contains 20 joints, and PrimeSense NiTE which contains only 15 joints (see Figure 2). The various joint configurations throughout the motion sequence produce a time series of skeletal poses giving the skeleton movement. In our approach, an action is simply described as a collection of time series of 3D positions of the joints in the hierarchical configuration.

3.1. Linear dynamic model

Let p_t^j denote the 3D position of a joint j at a given frame t i.e., $p^j = [x^j, y^j, z^j]_{j=1:J}$, with J is the number of joints. The joint position time-series



(a) (b)

Figure 2: Skeleton joint locations captured by Microsof Kinect sensor (a) using Microsoft SDK (b) using PrimeSense NiTE. Joint signification are: (1) head (2) shoulder center (3) spine (4) hip center (5/6) left/right hip (7/8) left/ ight knee (9/10) left/right ankle (11/12) left/right foot (13/14) left/right shoulder (15/16) left/right elbow (17/19) left/right wrist (19/20) left/right hand.

of joint j is $p_t^j = \{x_t^j, y_t^j, z_t^j\}_{j=1:J}^{t=1:T}$, with T the number of frames. A motion sequence can then be seen as a matrix collecting all time-series from J joints, i.e., $M = [p^1 p^2 \cdots p^T]$, $p \in \mathbb{R}^{3*J}$.

At this level, we could consider using DTW algorithm [37] to find optimal non-linear warping function to match these given time-series as proposed by [38, 39, 16]. However, we opted for a system combining a linear dynamic modelling with statistical analysis on a manifold, avoiding the boundary and the monotonicity constraints presented by classical DTW algorithm. Such a system is also less sensitive to noise due to the poor estimation of the joint locations, in addition to its reduced computational complexity.

The dynamic and the continuity of movement imply that the action can not be resumed as a simply set of skeletal poses because of the temporal information contained in the sequence. Instead of directly using original joint position time-series data, we believe that a linear dynamic system, like that often used for dynamic texture modelling, is essential before manifold analysis. Therefore, to capture both the spatial and the temporal dynamics of a motion, linear dynamical system characterized by ARMA models are applied to the 3D joint position time-series matrix M.

The dynamic captured by the ARMA [52, 53] model during an action sequence M can be represented as:

$$\begin{aligned} p(t) &= Cz(t) + w(t), \quad w(t) \sim N(0, R), \\ z(t+1) &= Az(t) + v(t), \quad v(t) \sim N(0, Q) \end{aligned} \tag{1}$$

where $z \in \mathbb{R}^d$ is a hidden state vector, $A \in \mathbb{R}^{d \times d}$ is the transition matrix and $C \in \mathbb{R}^{3*J \times d}$ is the measurement matrix. w and v are noise components modeled as normal with mean equal to zero and covariance matrix $R \in \mathbb{R}^{3*J \times 3*J}$ and $Q \in \mathbb{R}^{d \times d}$ respectively. The goal is to learn parameters of the model (A, C) given by these equations. Let $U \sum V^T$ be the singular value decomposition of the matrix M. Then, the estimated model parameters A and C are given by: $\hat{C} = U$ and $\hat{A} = \sum V^T D_1 V (V^T D_2 V)^{-1} \sum^{-1}$, where $D_1 = [0 \;\; 0, I_{\tau-1} \;\; 0]$, $D_2 = [I_{\tau-1} \;\; 0, 0 \;\; 0]$ and $I_{\tau-1}$ is the identity matrix of size $\tau - 1$.

Comparing two ARMA models can be done by simply comparing their observability matrices. The expected observation sequence generated by an ARMA model (A,C) lies in the column space of the extended observability matrix given by $\theta_\infty^T = [C^T, (CA)^T, (CA^2)^T, ...]^T$. This can be approximated by the finite observability matrix $\theta_m^T = [C^T, (CA)^T, (CA^2)^T, ..., (CA^2)^m]^T$

[18]. The subspace spanned by columns of this finite observability matrix corresponds to a point on a Grassmann manifold.

3.2. Grassmann manifold interpretation

Grassmannian analysis provides a natural way to deal with the problem of sequence matching. Especially, this manifold allows to represent a sequence by a point on its space and offers tools to compare and to do statistics on this manifold. The classification problem of sets of motions represented by a collection of features can be transformed to point classification problem on the Grassmann manifold.

In this work we are interested in Grassmann manifolds which definition is as below.

Definition: The Grassmann manifold $G_{n \times d}$ is a quotient space of orthogonal group $O(n)$ and is defined as the set of d-dimensional linear subspaces of $\mathbb{R}^n$. Points on the Grassmann manifold are equivalent classes of $n \times d$ orthogonal matrices, with $d < n$, where two matrices are equivalent if their columns span the same d-dimensional subspace.

Let μ denotes an element on $G_{n \times d}$, the tangent space to this element T_μ on $G_{n,d}$ is the tangent plane to the surface of the manifold at μ. It is possible to map a point U, of the Grassmann manifold, to a vector in the tangent space T_μ using the logarithm map as defined by Turaga et al. [18]. An other important tool in statistics is the exponential map $Exp_\mu : T_\mu(G_{n,d}) \rightarrow G_{n,d}$, which allows to move on the manifold.

Two points U_1 and U_2 on $G_{n,d}$ are equivalent if one can be mapped into the other one by $d \times d$ orthogonal matrix [54]. In other words, U_1 and U_2 are equivalent if the d columns of U_1 are rotations of U_2. The minimum length curve connecting these two points is the geodesic between them computed as:

$$d_{geod}(U_1, U_2) = \| [\theta_1, \theta_2, \cdots, \theta_i, \cdots, \theta_d] \|_2 \qquad (2)$$

where θ_i is the principal angle vector which can be computed through the SVD of $U_1^T U_2$.

4. Learning process on the manifold

Let $\{U_1, \cdots U_N\}$ be N actions represented by points on the Grassmann manifold. A common learning approach on manifolds is based on the use of only one-tangent space, which usually can be obtained as the tangent space to the mean (μ) of the entire data points $\{U_i\}_{i=1:N}$ without regard to class labels. All data points on the manifold are then projected on this tangent space to provide the input of a classifier. This assumption provide an accommodated solution to use a classical supervised learning on the manifold. However, this flattening of the manifold through tangent space is not efficient since the tangent space on the global mean can be far from other points.

A more appropriate way is to consider separate tangent spaces for each class at the class-mean. The classification is then performed in these individual tangent spaces as in [18].

Some other approaches explore the idea of tangent bundle as in Lui et al. [21, 22], in which all tangent planes of all data points on the manifold are considered. Tangent vectors are then computed between all points on a Grassmann manifold and action classification is performed thanks to obtained distances.

We believe that using several tangent spaces, obtained for each class of the training data points, is more intuitive. However, the question here is how to learn a classifier in this case?

In the rest of the section, we present a statistical computation of the mean in the Grassmann manifold [55]. Then, we propose two learning methods on this manifold taking benefit from tangent space class specific and tangent bundle [21]: Truncated Wrapped Gaussian (TWG) [56] and Local Tangent Bundle SVM (LBTSVM).

4.1. Mean computation on the Grassmann manifold

The Karcher mean [55] enables computation of a mean representative for a cluster of points on the manifold. This mean should belong to the same space as the given points. In our case, we need Karcher mean to compute averages on the Grassman manifold and more precisely means of each action class which represents the action at best. The algorithm exploits *log* and *exp* maps in a predictor/corrector loop until convergence to an expected point.

The computation of a mean can be used to perform an action classification solution. This can be done by a s simple comparison of an unknown action, represented as a point on the manifold, to all class-means and assigning it to the nearest one using the distance presented in Equation 2.

4.2. Truncated Wrapped Gaussian

In addition to the mean μ computed by Karcher mean on $\{U_i\}_{i=1:N}$, we look for the standard deviation value σ between all actions in each class of training data. The σ must be computed on $\{V_i\}_{i=1:N}$ where $V = exp_{\mu}^{-1}(U_i)$ are the projections of actions from the Grassmann manifold into the tangent space defined on the mean μ. The key idea here is to use the fact that the tangent space $T_\mu(G_{n,d})$ is a vector space.

Thus, we can estimate the parameters of a probability density function such as a Gaussian and then use the exponential map to wrap these parameters back onto the manifold using exponential map operator [18]. However, the exponential map is not a bijection for the Grassmann manifold. In fact, a line on tangent space, with infinite length, can be warpped around the manifold many times. Thus, some points of this line are going to have more than one image on $G_{n,d}$. It becomes a bijection only if the domain is restricted. Therefore, we can restrict the tangent space by a truncation beyond a radius of π in $T_\mu(G_{n,d})$. By truncation, the normalization constant changes for multivariate density in $T_\mu(G_{n,d})$. In fact, it gets scaled down depending on how much of the probability mass is left out of the truncation region.

Let $f(x)$ denotes the probability density function (pdf) defined on $T_\mu(G_{n,d})$ by :

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-x^2}{2\sigma^2}} \tag{3}$$

After truncation, an approximation of f gives:

$$\hat{f}(x) = \frac{f(x) \times \mathbf{1}_{|x|<\pi}}{z} \tag{4}$$

where z is the normalization factor :

$$z = \int_{-\pi}^{\pi} f(x) \times \mathbf{1}_{|x|<\pi} dx \tag{5}$$

Using Monte Carlo estimation, it can proved that the estimation of z is given by:

$$\hat{z} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_{|x_i|<\pi} \tag{6}$$

In practice, we employ wrapped Gaussians in each class-specific tangent space. Separate tangent space is considered for each class at its mean computed by Karcher mean algorithm. Predicted class of an observation point is estimated in these individual tangent spaces. In the training step, the mean, standard deviation and normalization factor in each class of actions are computed. The predicted label of unknown class action is estimated as a function of probability density in class-specific tangent spaces.

4.3. Local Tangent Bundle

We intent here to generalize a learning algorithm to work with data points which are geometrically lying to a Grassmann manifold. Using multiple class-specific tangent spaces is decidedly more relevant than single one. However, restrict the learning to only the mean and the standard-deviation in each tangent space, as in TGW method, is probably insufficient to classify complex actions with small inter-class variation. Our idea is to build a supervised classifier on the manifold but without limiting the learning process to distances computed on the tangent spaces as in [22].

We consider such data points to be embedded in higher dimensional representation providing a natural and implicit separation of directions. We use the notion of tangent bundle on the manifold to formulate our learning algorithm.

The tangent bundle of a manifold is defined in the literature as the manifold along with the set of tangent planes taken at all points on it. Each such a tangent plane can be equipped with a local Euclidean coordinate system. In our approach, we consider several "local" bundles, each one represents the tangent planes taken at all points belonging to a class from training dataset and expressed as class-specific local bundle.

We generate Control Tangents (CT) on the manifold, which represent all class-specific local bundles of data points. Each CT can be seen as the tangent space of the Karcher mean of all points belonging to the same class of points from only training data. Karcher mean algorithm can be employed here for mean computation.

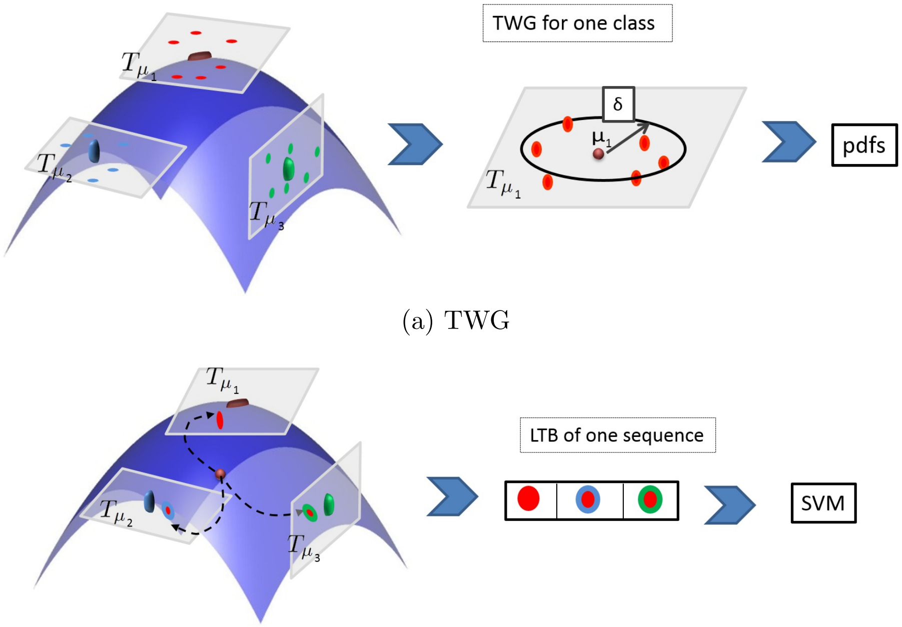
We introduce an upswing of the manifold learning so-called Local Tangent Bundle (LTB), in which proximities are required between each point on the manifold and all CTs. The LTB can be viewed as a parameterization of a point on the manifold which incorporates implicitly release properties in relation to all class clusters, by mapping this point to all CTs using logarithm map.

The LTBs can provide the input of a classifier, like the linear SVM classifier as in our case. In doing so, the learning model of the classifier is constructed using LTBs instead of classifying as function of the local distances (mean and standard-deviation) of the point from LTBs as in TWG method.

We finally notice that training a linear SVM classifier on our representation of points provided by LTB is more appropriate than the use of SVM with classical Kernel, like rbf, on original points on the manifold.

In experiments, we compare our learning approach LTBSVM to the classical one denoted as One-tangent SVM (TSVM), in which the mean is computed on the entire training dataset regardless to class labels. Then, all points on the manifold are projected on this later to provide the inputs of a linear SVM.

A graphical illustration of the manifold learning by TWG and LTB can be shown in Figure 3.



(a) TWG

(b) LTB

Figure 3: Conceptual TWG and LTB learning methods on the Grassmann manifold. (a) Actions belonging to the same class, illustrated with same color, are projected to the tangent space presented with their mean and then Gaussian function is computed on each tangent space, (b) An action is projected on all CTs, and thus construct a new observation is represented by its LTB.

5. Experimental results

This section summarizes our empirical results and provides an analysis of the performances of our proposed approach on several datasets compared to the state-of-the-art approaches.

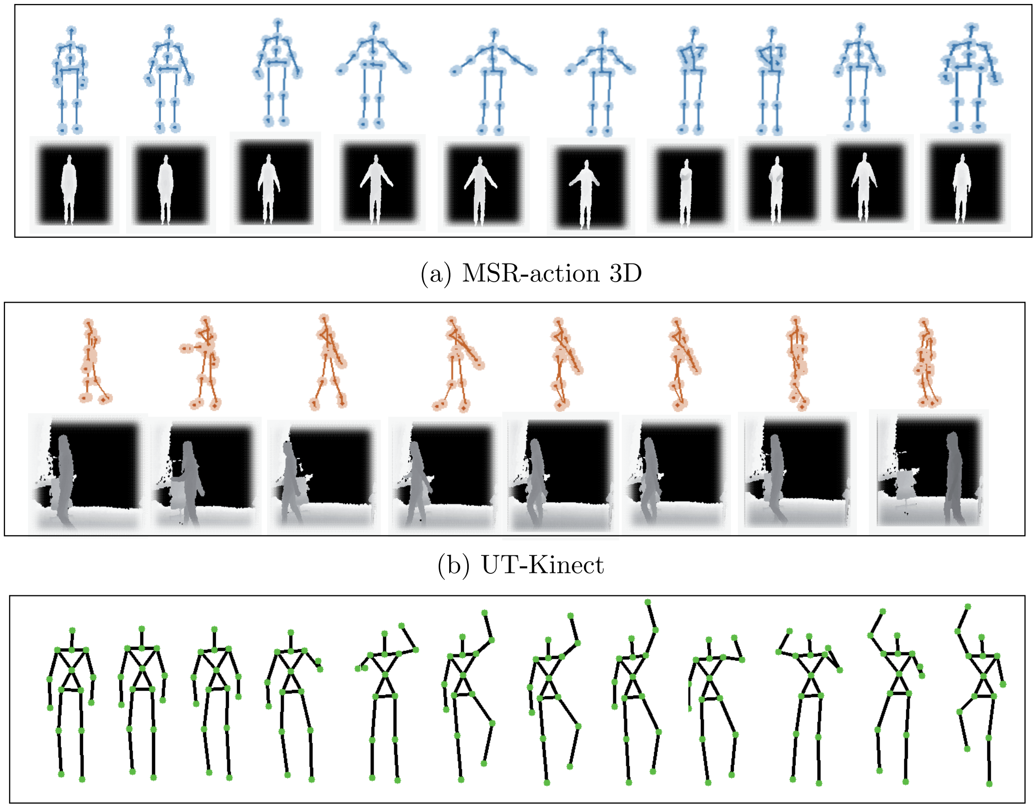
5.1. Data and features

We extensively experimented our proposed approach on three public 3D action datasets containing various challenges, including MSR-action 3D [28], UT-kinect [35] and UCF-kinect [41]. All details about these datasets: different types and number of motions, number of subjects executing these motions and the experimental protocol used for evaluation are summarized in Table 1. Examples of actions from these datasets are shown in Figure 4.

Dataset	Motions	Total number of actions	Experimental protocol
MSR-action 3D [28]	RGB + depth (320*240) + 20 joints: high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw X, draw tick, draw circle, hand clap, two hand wave, side-boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pick up and throw	10 subjects | 20 actions | 3 try ⇒ Total of 520 actions	50% Learning / 50% Testing
UT-kinect [35]	RGB + depth (320*240) + 20 joints: walk, sit down, stand up, pick up, carry, throw, push, pull, wave and clap hands	10 subject | 10 actions | 2 try ⇒ Total of 200 actions	leave-one-out cross-validation
UCF-kinect [41]	15 joints: balance, climb up, climb ladder, duck, hop, vault, leap, run, kick, punch, twist left, twist right , step forward, step back, step left, step right	16 subjects | 16 actions | 5 try ⇒ Total of 1280 actions	70% Learning / 30% Testing

Table 1: Overview of the datasets used in the experiments.

In all these datasets, a normalization step is performed in order to make the skeletons scale-invariant. For each frame, the hip center joint is first placed at the origin of the coordinate system. Then, a skeleton template is taken as reference and all the other skeletons are normalized such that their



(a) MSR-action 3D

(b) UT-Kinect

(c) UCF-kinect

Figure 4: Examples of human actions from datsets used in our experiments: (a) ’hand clap’ from MSR-action 3D , (b) ’walk’ from UT kinect and (c) ’climb ladder’ from UCF-kinect.

body part lengths are equal to the corresponding lengths of the reference skeleton. Each 3D joint sequence is represented as time series matrix of size $F \times T$ with T the number of frames in the sequence and F the number of features per frame. The number of features depends on the number of estimated joints (60 values for Microsoft SDK skeleton and 45 for PrimeSense NiTE skeleton). The dynamic of the activity is then captured using an ARMA model. In this process, a dimensionality reduction is needed and best subspace dimension "d" have been chosen using a 5-fold cross-validation on the training dataset. The parameter giving the best accuracy on the training set is kept for all experiments.

Each action is an element of the Grassmann manifold $G_{n \times d}$ with $n = m \times J$ where J represents the number of joints and d is the subspace dimension learnt on the training data. We set $m = d$, while m represents the truncation parameter of observation.

In our LTBSVM approach, we train a linear SVM on our LTB representations of points on the Grassmann manifold. We use a multi-class SVM classifier from LibSVM library [57], where the penalty parameter C is tuned using a 5-fold cross-validation on the training dataset.

We evaluate the performance of our approach for action recognition and explore the latency on recognition by evaluating the trade-off between accuracy and latency over varying number of actions. To allow a better evaluation of our approach, we conducted experiments respecting those made in the state-of-the-art approaches. We note here that other interesting datasets are available, like TUM kitchen dataset [43] which presents challenging short and complex actions. In our experiments we concentrated on three other datasets from depth sensors (such as kinect), chosen according to the challenges they contain, as occlusion, change of view and possibility to compare the latency. Details of the experiments are presented in the following sections.

5.2. MSR-Action 3D dataset

MSR-Action 3D [28] is a public dataset of 3D action captured by a depth camera. It consists of a set of temporally segmented actions where subjects are facing the camera and they are advised to use their right arm or leg if an action is performed by a single limb. The background is pre-processed clearing discontinuities and there is no interaction with objects in performed actions. Despite of all of these facilities, it is also a challenging dataset since many activities appear very similar due to small inter-class variation. Several works have already been conducted on this dataset. Table 2 shows the accuracy of our approach compared to the state-of-the-art methods. We followed the same experimental setup as in Oreifej et al. [34] and Jiang et al. [45], where first five actors are used for training and the rest for testing.

Our results obtained in this table correspond to four learning methods: simple Karcher Mean (KM), one Tangent SVM (TSVM), Truncated Wrapped Gaussian (TWG) and Local Tangent Bundle SVM (LTBSVM). Our approach using LTBSVM achieves an accuracy of 91.21%, exceeding the best method from the state-of-the-art proposed by Oreifej et al. [34]. We note that our approach is based on only skeletal joint coordinates as motion features, compared to other approaches, such as Oreifej et al. [34] and Wang et al. [32] which use the depth map or depth information around joint locations.

To evaluate the effect of the changing of the subspace dimensions, we conduct several tests on MSR-Action 3D dataset with different dimensions

Method	accuracy %
Histograms of 3D Joints [58]	78.97
Eigen Joints [36]	82.33
DMM-HOG [33]	85.52
HON4D [34]	85.80
Random Occupancy patterns [32]	86.50
Actionlet Ensemble [45]	88.20
HOH4D + D_{disc} [34]	88.89
TSVM on one tangent space	**74.32**
KM	**77.02**
TWG	**84.45**
LTBSVM	**91.21**

Table 2: Recognition accuracy (in %) for the MSR-Action 3D dataset using our approach compared to the previous approaches.

of subspaces. Figure 5 shows the variation of recognition performances with the change of the subspace dimension. We remark that until dimension 12, the recognition rate generally increase with the increase of the size of the subspaces dimensions. This is expected, since a small dimension causes a lack of information but also a big dimension of the subspace keeps noise and brings confusion between inter-classes. We also compare in this figure, our new introduced learning algorithm LBTSVM to TWG and KM.

To better understand the behavior of our approach according to the action type, the confusion matrix is illustrated in Figure 6. For most actions, about 11 classes of actions, video sequences are 100% correctly classified.

The classification error occurs if two actions are very similar, such as 'horizontal arm wave' and 'high arm wave'. Besides, one of most problematic action to classify is 'hammer' action which is frequently confused with 'draw X'. The particularity of these two actions is that they start in the same way but one finishes before the other. If we show only the first part of 'draw X' action and the whole sequence of 'hammer' action we can see that

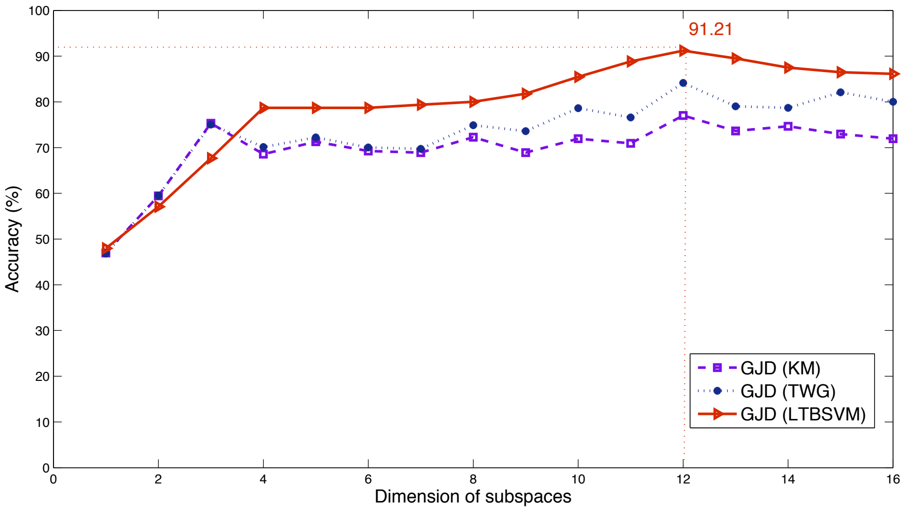


Figure 5: Recognition rate variation with learning approach and subspace dimension.

they are very similar. The same for 'hand catch' action which is confused with 'draw circle'. It is important to note that 'hammer' action is completely misclassified with the approach presented by Oreifej et al. [34] which presents the second better recognition rate after our approach.

While the focus of this paper is mainly on action recognition and latency reduction, some applications need to perform training step with a reduced amount of data. To study the effect of the amount of training dataset, we measured how the accuracy changed as we iteratively reduced the number of actions per class in the training dataset. Table 3 shows obtained accuracy results with different size of training dataset.

These results show that, in contrast to approaches that use HMM which require a large number of training data, our approach reveals robustness and efficiency. This robustness is due to the fact that the Control Tangents, which

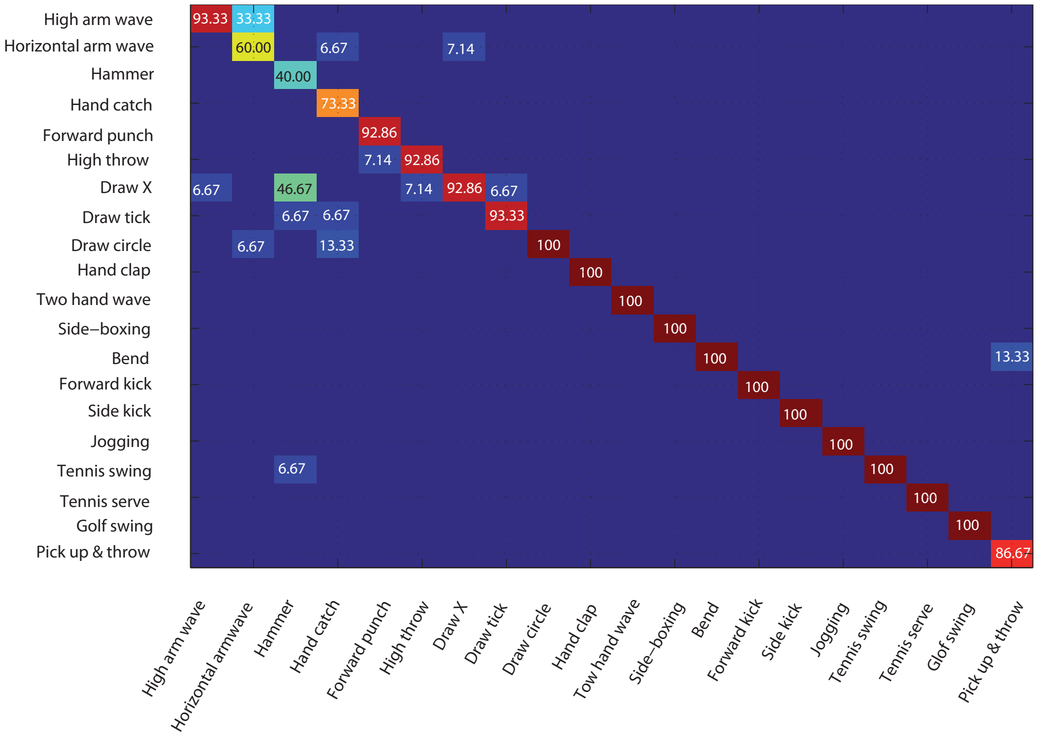


Figure 6: The confusion matrix for the proposed approach on MSR-Action 3D dataset. It is recommended to view the Figure on the screen.

play an important role in learning process, can be computed efficiently using small number of action points per class on the manifold.

5.3. UT-Kinect dataset

Sequences of this dataset are taken using one depth camera (kinect) in indoor settings and their length vary from 5 to 120 frames. We use this dataset because it contains several challenges:

- View change, where actions are taken from different views: right view, frontal view or back view.

Actions per class	Training dataset %	Accuracy %
5	37.17	73.36
6	44.23	77.64
7	51.13	83.10
8	58.36	84.79
9	65.54	88.51
10	72.49	89.18
11	79.95	87.83
12	86.24	88.85
13	91.07	90.20
14	95.91	90.54
15	100	91.21

Table 3: Recognition accuracy, obtained by our approach using LTBSVM on MSR-Action 3D dataset, with different size of training dataset.

- Significant variation in the realization of the same action: same action is done with one hand or two hands can be used to describe the 'pick up' action.

- Variation in duration of actions: the mean and standard-deviation are respectively for the whole actions 31.1 and 11.61 frames at 30 fps.

To compare our results with state-of-the-art approaches, we follow experiment protocol proposed by Xia et al. [35]. The protocol is leave-one-out cross-validation. In Table 4, we show comparison between the recognition accuracy produced by our approach and the approach presented by Xia et al. [35].

This table shows the accuracy of the five least-recognized actions in UT-kinect dataset and the five best-recognized actions. Our system performs the worst when the action represents an interaction with an object: 'throw', 'push', 'sit down' and 'pick up'. However, for the best five recognized actions, our approach improves the recognition rate reaching 100%. These actions

Action	Acc % Xia et al. [35]	Acc % LTBSVM
Walk	96.5	**100**
Stand up	91.5	**100**
Pick up	97.5	**100**
Carry	97.5	**100**
Wave	100	**100**
Throw	59	60
Push	81.5	65
Sit down	91.5	80
Pull	92.5	85
Clap hands	100	95
Overall	90.92	88.5

Table 4: Recognition accuracy (per action) for the UT-kinect dataset obtained by our approach using LTBSVM compared to Xia et al. [35].

contain variations in view point and realization of the same action. This means that our approach is view-invariant and it is robust to change in action types thanks to the used learning approach. The overall accuracy of Xia et al. [35] is better than our recognition rate. However on MSR Action3D database, the recognition rate obtained by this approach gives only 78.97%. This can be explained by the fact that this approach requires a large training dataset. Especially for complex actions which affect adversely the HMM classification in case of small samples of training.

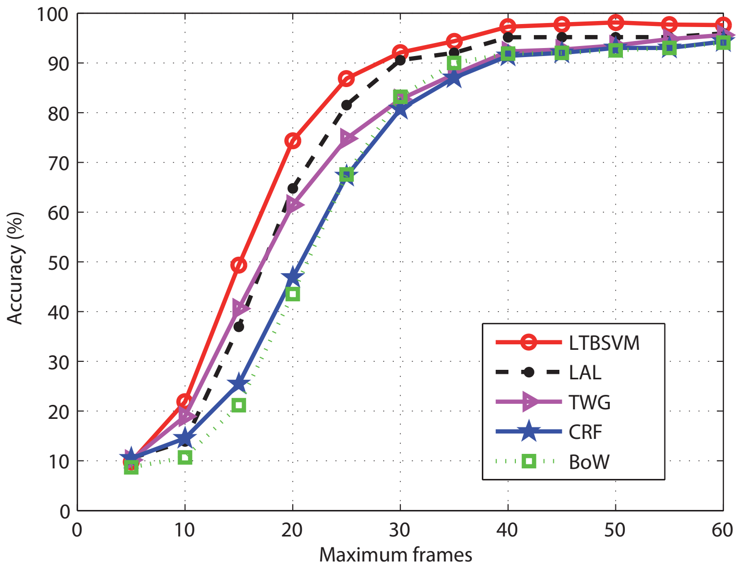
5.4. UCF-kinect dataset

In this experiment, our approach is evaluated in terms of latency, i.e. the ability for a rapid (low-latency) action recognition. The goal here is to automatically determine when a sufficient number of frames are observed to permit a reliable recognition of the occurring action. For many applications, a real challenge is to define a good compromise between ”making forced decision” on partial available frames (but potentially unreliable) and ”waiting” for the entire video sequence.

To evaluate the performance of our approach in reducing latency, we conducted our experiments on UCF-kinect dataset [41]. The skeletal joint locations (15 joints) over sequences of this dataset are estimated using Microsoft Kinect sensor and the PrimeSense NiTE. The same experimental setup as in Ellis et al. [41] is followed. For a total of 1280 action samples contained in this dataset, a 70% and 30% split is used for respectively training and testing datasets. From the original dataset, new subsequences were created by varying a parameter corresponding to the K first frames. Each new subsequence was created by selecting only the first K frames from the video. For videos shorter than K frames, the entire video is used. We compare the result obtained by our approach to those obtained by Latency Aware Learning (LAL) method proposed by Ellis et al. [41] and other baseline algorithms: Bag-of-Words (BoW) and Linear Chain Conditional Random Field (CRF), also reported by Ellis et al. [41].

As shown in Figure 7, our approach using LTBSVM clearly achieves improved latency performance compared to all other baseline approaches. Analysis of these curves shows that, accuracy rates for all other approaches are close when using small number of frames (less than 10) or a large number of frames (more than 40). However, the difference increases significantly in the middle range. The table joint to Figure 7 shows numerical results at several points along the curves in the figure. Thus, given only 20 frames of input, our system achieves 74.37%, while BOW, CRF recognition rate below 50% and LAL achieves 61.45%.

It is also interesting to notice the improvement of accuracy of 92.08%



Approach/frames	10	15	20	25	30	40	60
LTBSVM	**21.87**	**49.37**	**74.37**	**86.87**	**92.08**	**97.29**	**97.91**
TWG	18.95	40.62	61.45	74.79	82.7	92.29	95.62
LAL [41]	13.91	36.95	64.77	81.56	90.55	95.16	95.94
CRF [41]	14.53	25.46	46.88	67.27	80.70	91.41	94.06
BOW [41]	10.7	21.17	43.52	67.58	83.20	91.88	94.06

Figure 7: Accuracy vs. state-of-the-art approaches over videos truncated at varying maximum lengths. Each point of this curve shows the accuracy achieved by the classifier given only the number of frames shown in the x-axis.

obtained by LTBSVM compared to 82.7% obtained by TWG, with maximum frame number equal to 30. For a large number of frames, all of the methods perform globally a good accuracy with an improvement of the ours (97.91% comparing to 95.94% obtained by LAL proposed in Ellis et al. [41]). These results show that our approach can recognize actions at the desired accuracy with reducing latency.

Finally, the detail of recognition rates, when using the totality of frames in the sequence, are shown through the confusion matrix in Figure 8. Unlike what gives LAL, we can observe that the ’twist left’, ’twist right’ actions are not confused with each others. All classes of actions are classified with a rate more than 93.33% which gives a lot of confidence to our proposed learning approach.

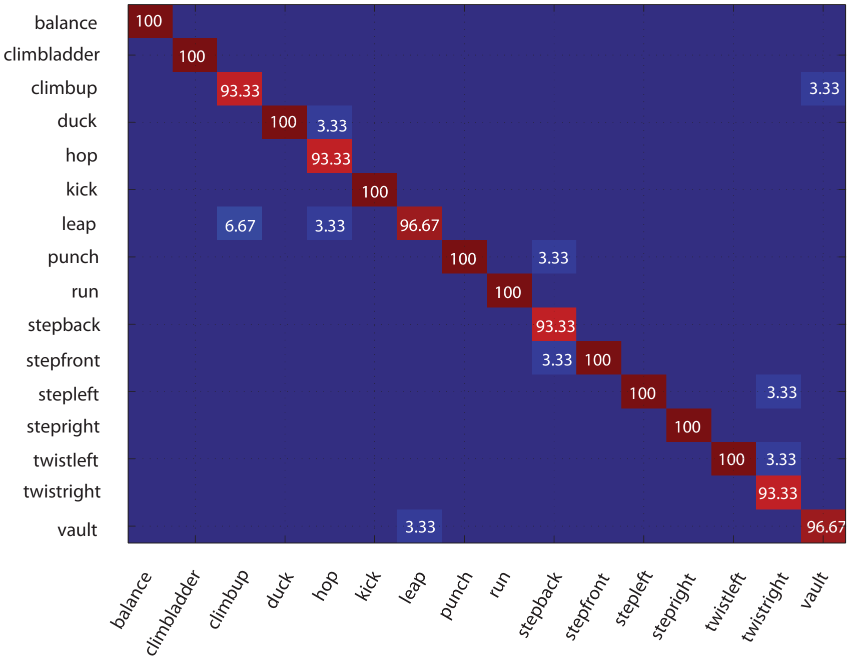


Figure 8: The confusion matrix for the proposed method on UCF-kinect dataset. Overall accuracy achieved 97.91%. It is recommended to view the figure on the screen.

5.5. *Discussion*

Manifold representation and learning. Data representation is one of the most important factors in the recognition approach, on which we must take a lot of consideration. Our data representation, like many state-of-the-art manifold techniques [19, 14, 21], consider the geometric space and incorporates the intrinsic nature of the data. In our framework, which is 3D joint-based, both geometric appearance and dynamic of human body are captured simultaneously. Furthermore, unlike the manifold approaches using silhouettes [14, 15, 18], or directly raw pixels [22, 19], our approach use informative geometric features, which capture useful knowledge to understand the intrinsic motion structure. Thanks to recent release of depth sensor, these features are extracted and tracked along the action sequence, while classical pixel-based manifold approaches relying on a good action localization, or on tedious feature extraction from 2D videos like silhouettes.

In terms of learning method, we generalized a learning algorithm to work with data points which are geometrically lying to a Grassmann manifold.

Other approaches are tested in the learning process on the manifold: one tangent space (TSVM) and class-specific tangent spaces (TWG). In the first one, recognition rate is low. In fact, the computation of the mean of all actions from all classes can be inaccurate. Besides, projections on this plane can lead to big deformations. A better solution is to operate on each class by computing its proper tangent space, as in TWG [56] which improve TSVM results (see Table 2). In our approach (LTBSVM), both Control Tangent and statistics on the manifold are used. The purpose was to formulate our learning algorithm using a discriminative parametrization which incorporate class separation properties. The particularity of our learning model is the incorporation of proximities relative to all Control Tangent spaces representing class clusters, instead of classifying using a function of local distances. The results in Table 2 demonstrate that the proposed algorithm is more efficient in action recognition scenario when inter-variation classes is present as a challenge.

Furthermore, the analysis of the impact of reducing the number of actions in the training set on the accuracy of the classifier show robustness. Even with a small number of actions in the training data recognition rates remain good as demonstrated in Table 3. However it is a limitation especially for approaches using an HMM learning because they require a large number of training dataset. Such as Xia et al. approach [35], which gives only 78.97% of recognition rate while performing cross subject test on MSR dataset.

Latency and Time computation. The evaluations in terms of latency have clearly revealed the efficiency of our approach for a rapid recognition. It is possible to recognize actions up to 95% using only 40 frames which is a good performance comparing to state-of-the-art approaches presented in [41]. Thus, our approach can be used for interactive systems. Particularly, in entertainment applications to resolve the problem of lag and improve some motion-based games.

Since the proposed approach is based on only skeletal joint coordinates, it is simple to calculate and it needs only a small computation time. In fact, with our current implementation written in C++, the whole recognition time takes 0.26 sec to recognize a sequence of 60 frames. The joint extraction and normalisation take 0.0001 sec, the Grassmann and the LTB representation take 0.0108 sec and the prediction on SVM takes 0.251 sec. These computation time are reported on UCF dataset, with Grassmann manifold dimension $n = 540$ and $d = 12$. We also reported the computation time needed to recognize actions while incorporating latency on UCF dataset. Figure 9 illustrates inline time recognition with time progression, after only 40 frames the recognition is given at the 0.94 sec within 97.29% of correctness rate. After 60 frames, in 1.3 sec the algorithm recognize correctly the action with

97.91%. All the computation time experiments are lunched on a PC having Intel Core i5-3350P (3.1 GHz) CPU, 4GB RAM and a PrimeSense camera for skeleton extraction giving about 60 skeleton/sec.

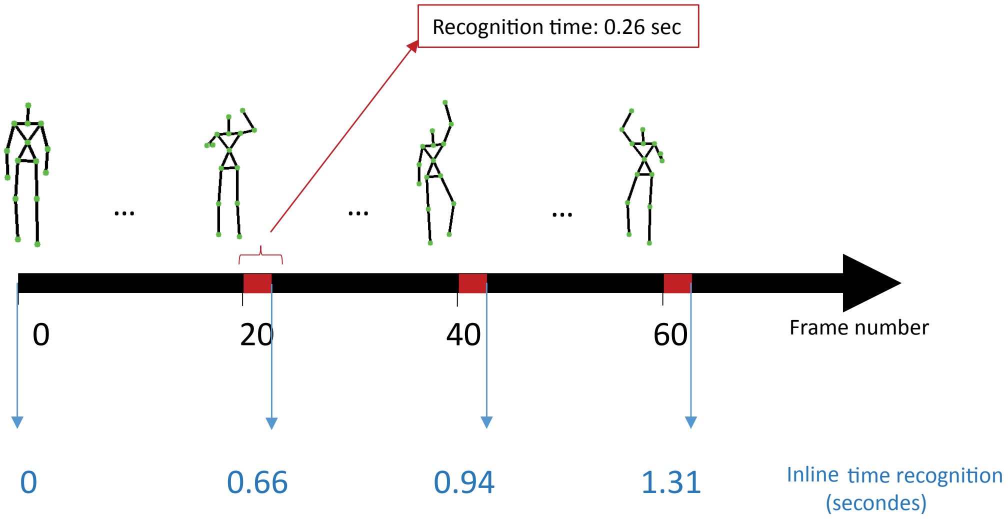

Recognition time: 0.26 sec
...
...
...
0
20
40
60
Frame number
0
0.66
0.94
1.31
Inline time recognition (secondes)

Figure 9: The computation time to perform 20 frames actions sequences is 0.26 sec by using our approach. The computation time is given for each actions frames sequences (e.g. 0.94 sec for 40 frames).

Limitations. Our proposed approach is a 3D joint-based framework derives a human action recognition from skeletal joint sequences. In the case of presence of object interaction in human actions, our approach do not provides any relevant information about objects and thus, action with and without objects are confused. This limitation can be leveraged in future by the use of additional features, which can be extracted from depth or color images associated to 3D joint locations.

6. Conclusion

In this paper, an effective framework for modelling and recognizing human motion in the 3D skeletal joint space is proposed. In this framework, sequence features are modeled temporally as subspaces lying to a Grassmannian manifold. A new learning algorithm on this manifold is then introduced. It embeds each action, presented as a point on the manifold, in higher dimensional representation providing natural separation directions. We formulated our learning algorithm using the notion of local tangent bundles on class clusters on the Grassmann manifold. The empirical results and the analysis of the performance of our proposed approach show promising results with high accuracies superior to 88% on three different datasets. The evaluation of our approach in terms of accuracy/latency reveals an important ability for a low-latency action recognition system. Obtained results show that with minimum number of frames, it provides the highest recognition rate.

We would encourage future works to extend our approach to investigate more challenging problems like human behaviour recognition. Finally, we plan to use additional features from depth or color images associated to 3D joint locations to solve the problem of human-object interaction.

References

[1] S. Fothergill, H. Mentis, P. Kohli, S. Nowozin, Instructing people for training gestural interactive systems, in: CHI Conference on Human Factors in Computing Systems, New York, NY, USA, 2012, pp. 1737–1746.

[2] W. Lao, J. Han, P. de With, Automatic video-based human motion analyzer for consumer surveillance system, in: IEEE Transactions on Consumer Electronics, Vol. 55, 2009, pp. 591–598.

[3] A. Jalal, M. Uddin, T. S. Kim, Depth video-based human activity recognition system using translation and scaling invariant features for life logging at smart home, in: IEEE Transactions on Consumer Electronics, Vol. 58, 2012, pp. 863–871.

[4] R. Poppe, A survey on vision-based human action recognition, in: Image and Vision Computing, Vol. 28, 2010, pp. 976–990.

[5] P. Turaga, R. Chellappa, V. S. Subrahmanian, O. Udrea, Machine recognition of human activities: A survey, in: IEEE Transactions on Circuits and Systems for Video Technology, Vol. 18, Piscataway, NJ, USA, 2008, pp. 1473–1488.

[6] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, Real-time human pose recognition in parts from single depth images, in: Machine Learning for Computer Vision, Vol. 411, 2013, pp. 119–135.

[7] C.-S. Lee, A. M. Elgammal, Modeling view and posture manifolds for tracking, in: IEEE International Conference on Computer Vision, 2007, pp. 1–8.

[8] Y. M. Lui, Advances in matrix manifolds for computer vision, in: Image and Vision Computing, Vol. 30, 2012, pp. 380 – 388.

[9] M. T. Harandi, C. Sanderson, S. Shirazi, B. C. Lovell, Kernel analysis on grassmann manifolds for action recognition, in: Pattern Recognition Letters, Vol. 34, 2013, pp. 1906 – 1915.

[10] M. Bregonzio, T. Xiang, S. Gong, Fusing appearance and distribution information of interest points for action recognition, in: Pattern Recognition, Vol. 45, 2012, pp. 1220 – 1234.

[11] S. O'Hara, Y. M. Lui, B. A. Draper, Using a product manifold distance for unsupervised action recognition, in: Image and Vision Computing, Vol. 30, 2012, pp. 206 – 216.

[12] J. Aggarwal, M. Ryoo, Human activity analysis: A review, in: ACM Computing Surveys, Vol. 43, 2011, pp. 1–43.

[13] D. Weinland, R. Ronfard, E. Boyer, A survey of vision-based methods for action representation, segmentation and recognition, in: Computer Vision and Image Understanding, Vol. 115, 2011, pp. 224–241.

[14] A. Veeraraghavan, A. Roy-Chowdhury, R. Chellappa, Matching shape sequences in video with applications in human movement analysis, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, 2005, pp. 1896–1909.

[15] M. F. Abdelkader, W. Abd-Almageed, A. Srivastava, R. Chellappa, Silhouette-based gesture and action recognition via modeling trajectories on riemannian shape manifolds, in: Computer Vision and Image Understanding, Vol. 115, 2011, pp. 439 – 455.

[16] D. Gong, G. Medioni, Dynamic manifold warping for view invariant action recognition, in: IEEE International Conference on Computer Vision, Barcelona, Spain, 2011, pp. 571–578.

[17] D. Gong, G. Medioni, X. Zhao, Structured time series analysis for human action segmentation and recognition, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PP, 2014, pp. 1–1.

[18] P. Turaga, A. Veeraraghavan, A. Srivastava, R. Chellappa, Statistical computations on grassmann and stiefel manifolds for image and video-based recognition, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, 2011, pp. 2273–2286.

[19] P. Turaga, R. Chellappa, Locally time-invariant models of human activities using trajectories on the grassmannian, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 2435–2441.

[20] K. Guo, P. Ishwar, J. Konrad, Action recognition in video by sparse representation on covariance manifolds of silhouette tunnels, in: Recognizing Patterns in Signals, Speech, Images and Videos, Vol. 6388, 2010, pp. 294–305.

[21] Y. M. Lui, J. R. Beveridge, Tangent bundle for human action recognition, in: IEEE International Conference on Automatic Face and Gesture Recognition, 2011, pp. 97–102.

[22] Y. M. Lui, Tangent bundles on special manifolds for action recognition, in: IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22, 2012, pp. 930–942.

[23] S. Shirazi, M. T. Har, C. S, A. Alavi, B. C. Lovell, Clustering on grassmann manifolds via kernel embedding with application to action analysis, in: International Conference on Image Processing, 2012, pp. 781–784.

[24] M. T. Harandi, C. Sanderson, S. Shirazi, B. C. Lovell, Kernel analysis on grassmann manifolds for action recognition, in: Pattern Recognition Letters, Vol. 34, 2013, pp. 1906 – 1915.

[25] J. Gall, A. Yao, L. Van Gool, 2D action recognition serves 3d human pose estimation, in: European Conference on Computer Vision, Vol. 6313, 2010, pp. 425–438.

[26] L. Chen, H. Wei, J. Ferryman, A survey of human motion analysis using depth imagery, in: Pattern Recognition Letters, Vol. 34, 2013, pp. 1995 – 2006.

[27] M. Ye, Q. Zhang, L. Wang, J. Zhu, R. Yang, J. Gall, A survey on human motion analysis from depth data, in: Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications, Vol. 8200, 2013, pp. 149–187.

[28] W. Li, Z. Zhang, Z. Liu, Action recognition based on a bag of 3D points, in: IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2010, pp. 9–14.

[29] B. Ni, G. Wang, P. Moulin, Rgbd-hudaact: A color-depth video database for human daily activity recognition, in: International Conference on Computer Vision Workshops, 2011, pp. 1147–1153.

[30] L. Xia, J. Aggarwal, Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2834–2841.

[31] A. W. Vieira, E. R. Nascimento, G. L. Oliveira, Z. Liu, M. F. Campos, STOP: Space-time occupancy patterns for 3D action recognition from depth map sequences, in: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Vol. 7441, 2012, pp. 252–259.

[32] J. Wang, Z. Liu, J. Chorowski, Z. Chen, Y. Wu, Robust 3D action recognition with random occupancy patterns, in: European Conference on Computer Vision, 2012, pp. 872–885.

[33] X. Yang, C. Zhang, Y. Tian, Recognizing actions using depth motion maps-based histograms of oriented gradients, in: international conference on ACM Multimedia, New York, NY, USA, 2012, pp. 1057–1060.

[34] O. Oreifej, Z. Liu, Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences, in: IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 2013, pp. 716–723.

[35] L. Xia, C.-C. Chen, J. K. Aggarwal, View invariant human action recognition using histograms of 3D joints, in: Computer Vision and Pattern Recognition Workshops, 2012, pp. 20–27.

[36] X. Yang, Y. Tian, Eigenjoints based action recognition using naive bayes nearest neighbor, in: Computer Vision and Pattern Recognition Workshops, 2012, pp. 14–19.

[37] T. Giorgino, Computing and visualizing dynamic time warping alignments in R: The dtw package, in: Journal of Statistical Softwar, Vol. 31, 2009, p. 1–24.

[38] M. Reyes, G. Dominguez, S. Escalera, Featureweighting in dynamic timewarping for gesture recognition in depth data, in: IEEE International Conference on Computer Vision Workshops, 2011, pp. 1182–1188.

[39] S. Sempena, N. Maulidevi, P. Aryan, Human action recognition using Dynamic Time Warping, in: International Conference on Electrical Engineering and Informatics, 2011, pp. 1–5.

[40] M. Bautista, A. Hernndez-Vela, V. Ponce, X. Perez-Sala, X. Bar, O. Pujol, C. Angulo, S. Escalera, Probability-based dynamic time warping for gesture recognition on RGB-D data, in: Advances in Depth Image Analysis and Applications, Vol. 7854, 2013, pp. 126–135.

[41] C. Ellis, S. Z. Masood, M. F. Tappen, J. J. Laviola, Jr., R. Sukthankar, Exploring the trade-off between accuracy and observational latency in action recognition, in: International Journal of Computer Vision, Vol. 101, 2013, pp. 420–436.

[42] M. Barnachon, S. Bouakaz, B. Boufama, E. Guillou, Ongoing human action recognition with motion capture, in: Pattern Recognition, Vol. 47, 2014, pp. 238 – 247.

[43] M. Tenorth, J. Bandouch, M. Beetz, The TUM kitchen data set of everyday manipulation activities for motion tracking and action recognition,

in: International Conference on Computer Vision Workshops, 2009, pp. 1089–1096.

[44] S. Azary, A. Savakis, A spatiotemporal descriptor based on radial distances and 3D joint tracking for action classification, in: IEEE International Conference on Image Processing, 2012, pp. 769–772.

[45] J. Wang, Z. Liu, Y. Wu, J. Yuan, Mining actionlet ensemble for action recognition with depth cameras, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1290–1297.

[46] S. Althloothi, M. H. Mahoor, X. Zhang, R. M. Voyles, Human activity recognition using multi-features and multiple kernel learning, in: Pattern Recognition, Vol. 47, 2014, pp. 1800 – 1812.

[47] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, A. Del Bimbo, Space-time pose representation for 3D human action recognition, in: Workshop on Social Behaviour Analysis ICIAP, Vol. 8158, 2013, pp. 456–464.

[48] S. Azary, A. Savakis, Grassmannian sparse representations and motion depth surfaces for 3D action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2013, pp. 492–499.

[49] X. Zhang, Y. Yang, L. Jiao, F. Dong, Manifold-constrained coding and sparse representation for human action recognition, in: Pattern Recognition, Vol. 46, 2013, pp. 1819 – 1831.

[50] R. Li, P. Turaga, A. Srivastava, R. Chellappa, Differential geometric

representations and algorithms for some pattern recognition and computer vision problems, in: Pattern Recognition Letters, Vol. 43, 2014, pp. 3 – 16.

[51] H. Wang, C. Yuan, G. Luo, W. Hu, C. Sun, Action recognition using linear dynamic systems, in: Pattern Recognition, Vol. 46, 2013, pp. 1710 – 1718.

[52] G. Doretto, A. Chiuso, Y. N. Wu, S. Soatto, Dynamic textures, in: International Journal of Computer Vision, Vol. 51, 2003, pp. 91–109.

[53] A. Bissacco, A. Chiuso, Y. Ma, S. Soatto, Recognition of human gaits, in: IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, 2001, pp. 52–57.

[54] A. Edelman, T. A. Arias, S. T. Smith, The geometry of algorithms with orthogonality constraints, in: SIAM Journal on Matrix Analysis and Applications, Vol. 20, 1998, pp. 303–353.

[55] A. Srivastava, E. Klassen, S. Joshi, I. Jermyn, Shape analysis of elastic curves in euclidean spaces, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, 2011, pp. 1415–1428.

[56] S. Kurtek, A. Srivastava, E. Klassen, Z. Ding, Statistical modeling of curves using shapes and related features, in: Journal of the American Statistical Association, Vol. 107, 2012, pp. 1152–1165.

[57] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, in: ACM Transactions on Intelligent Systems and Technology, Vol. 2, 2011, pp. 1–27.

[58] L. Xia, C.-C. Chen, J. K. Aggarwal, View invariant human action recognition using histograms of 3D joints., in: IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2012, pp. 20–27.