Classification of Multidimensional Time-Evolving Data using Histograms of Grassmannian Points

Kosmas Dimitropoulos, *Member, IEEE,* Panagiotis Barmpoutis, Alexandros Kitsikidis and Nikos Grammalidis, *Member, IEEE*

***Abstract*—In this paper we address the problem of classifying multidimensional time-evolving data in dynamic scenes. To take advantage of the correlation between the different channels of data, we introduce a generalized form of a stabilized higher-order linear dynamical system (sh-LDS) and we represent the multidimensional signal as a third order tensor. In addition, we show that the parameters of the proposed model lie on a Grassmann manifold and we attempt to address the classification problem through the study of the geometric properties of the sh-LDS's space. Moreover, to tackle the problem of non-linearity of the observation data, we represent each multidimensional signal as a cloud of points on the Grassmann manifold and we create a codebook by identifying the most representative points. Finally, each multidimensional signal is classified by applying a bag-of-systems approach having first modeled the variation of the class of each codeword on its tangent space instead of the sh-LDS's space. The proposed methodology is evaluated in three different application domains, namely video-based surveillance systems, dynamic texture categorization and human action recognition, showing its great potential.**

***Index Terms*—Linear dynamical systems, grassmann geometry, higher order decomposition, multidimensional signal processing.**

I. Introduction

MANY computer vision problems dealing with the analysis of dynamic scenes involve the modeling and classification of multidimensional time evolving data. Applications in this area usually concern dynamic textures (i.e., non-rigid objects such as water, fire, smoke, etc.) categorization, human activity modeling and recognition, gait analysis, face recognition and event detection from video sequences. To model the statistical properties of such data, it is often sensible to assume each observation to be correlated to the value of an underlying latent variable, or state, that is evolving over the course of the sequence [1].

A well known dynamical system, which is used by a number of state-of-the-art techniques [2] to model a wide variety of spatio-temporal data, is the first order autoregressive moving average (ARMA) model with white zero-mean independent and identically distributed Gaussian input, also known as Linear Dynamical System (LDS). LDS-based approaches have been successfully used for modeling time series in engineering, economics and social sciences.

In all of these application domains, the temporal variation of the data sequence is modeled as a LDS after a system identification process aiming to define the parameters of the dynamical model. Several approaches have been proposed for this purpose based on EM algorithm [3], non-iterative subspace methods [4] or principal components analysis [5]. The comparison between two LDS systems is then performed using a similarity metric, such as a distance [6], [7] or kernel [8], with Martin distance [9] to be one of the most common approaches. After the definition of the similarity metric, standard classifiers such as Nearest Neighbors or Support Vector Machines are usually used for the final categorization of a query data sequence [10].

To address the problem of the multi-dimensionality of data, most of the researchers often make a simplifying assumption of the data structure, which leads to the concatenation of data into a simple vector representing a point that follows a trajectory as time evolves (e.g., in the case of dynamic texture analysis, each frame is unfolded into a vector containing only grayscale information or the concatenated pixels' intensities of all channels, i.e., R, G and B components). This representation of data is driven by the fact that standard LDS systems extract dynamic information from a single element, i.e., a vector representation of data, and thus they do not fully exploit any hidden correlation between the different channels of data. Apart from taking advantage of the multi-dimensionality of data, significant improvement in the classification results can be achieved through the study of the geometric properties of the space in which the parameters of the model lie. Hence, instead of using subspace angles between the models, this study will allow us not only to define a more efficient similarity metric in the non-Euclidean space of the dynamical model, known as Grassmann manifold, but also to model the variation of the classes on this space. However, an inherent

Manuscript submitted 15 June, revised 11 October, accepted 8 November 2016.

K. Dimitropoulos, P. Barmpoutis, A. Kitsikidis and N. Grammalidis are with Information Technologies Institute, ITI - CERTH, 6th km Charilaou-Thermi Rd, Thessaloniki, 57001 GREECE (e-mail: dimitrop@iti.gr, panbar@iti.gr, ajinchv@iti.gr, ngramm@iti.gr).

limitation of all LDS systems is the fact that they try to model non-linear observation data using a system of linear equations. To address the problem of non linear structure of data other methods from the area of video semantic recognition have proposed the development of semi-supervised feature selection algorithms, e.g., Han et al. [11] proposed a framework of semi-supervised feature selection via spline regression. Here, we attempt to address the problem by considering the multidimensional evolving data, as a cloud of points (instead of a single point) on the Grassmann manifold and we create a codebook in order to represent each multidimensional signal as a histogram of Grassmannian points. More specifically, the main contributions of this paper are summarized as follows:

- We propose a novel methodology for the modeling and classification of multidimensional time evolving data. The proposed methodology aims to increase the robustness of linear dynamical systems by taking advantage of the multidimensionality of data and the geometric properties of the descriptor's space.
- We introduce a generalized form of a new stabilized higher-order linear dynamical system (sh-LDS), which can be used in various classification problems dealing with multidimensional time evolving data represented as a third order tensor. By applying a generalization of the singular value decomposition, we can extract a new descriptor, which enables us to i) exploit the correlation between the different channels of data, ii) model multidimensional signals of different temporal size and compare them without needing to employ first any sub-sampling technique for the alignment of signals, iii) reduce the computational cost by applying dimensionality reduction to the space of the hidden variable and iv) improve robustness by introducing a stabilization process based on a convex optimization technique.
- By taking advantage of the geometric properties of sh-LDS's space, we propose a new descriptor, namely Histograms of Grassmannian Points (HoGP), in order to improve the classification of multidimensional signals. Moreover, to address the problem of non-linearity of data, we represent each multidimensional signal as a cloud of points on the Grassmann manifold and we create a codebook by identifying the most representative of them. For the better classification of a sh-LDS model to one of the representative codewords, in our method we attempt to model the variation of the class of each codeword on its tangent space instead of the sh-LDS's space.
- To demonstrate the reproducibility of the proposed method, we carried out extensive tests in three different application domains: i) video-based surveillance systems, ii) dynamic texture categorization and iii) human action recognition. More specifically, ten different datasets were used, while the performance of the proposed method was compared against a number of different variations of LDS systems and other state of the art algorithms.

The remainder of this paper is organized as follows. Related work is presented in Section II. Section III presents the generalized form of sh-LDS model and the proposed HoGP descriptor. Finally, experimental results are discussed in Section IV, while conclusions are drawn in Section V.

II. Related Work

Time evolving data is usually modeled by temporal state-space methods, such as Hidden Markov Models (HMMs), Conditional Random Fields (CRFs) or dynamical systems. Brand et. al [12] presented algorithms for coupling and training HMMs to model interactions between processes that may have different state structures and degrees of influence on each other, while Zhuang et. al [13] proposed HMM supervectors, to improve patch-based GMM supervector approaches [14], and then calculated their Kullback-Leibler divergence. On the other hand, in [15] CRFs were used for contextual motion recognition, while in [16] a two-layer model was proposed along with CRFs for encoding actions and viewpoint-specific poses.

LDSs are considered as a more general form of HMMs [17] and they have shown promising results in a variety of applications. To this end, several techniques have been proposed aiming to address the problem of system identification in linear dynamical systems [4]. However, the method proposed by Doreto et. al [5] for dynamic texture analysis is considered as the most versatile approach. The method aims to define the parameters of the system using a fast closed-form suboptimal method based on principal component analysis, i.e., the decomposition of the image in a simple linear form. Before the decomposition, each frame is transformed into a vector representation containing grayscale information or the concatenated color components.

To provide a more flexible and natural way of decomposition, a model based on tensor representation and Tucker decomposition was presented in [18] for dynamic texture synthesis, i.e., for the creation of artificial textures. Tensor representation and factorization are widely used in multimedia applications, since they can effectively preserve the structure information. For instance, Guo et al [19] proposed a tensor learning regression framework based on CP decomposition, while Zhang et al. [20] presented a novel tensor bag of words model for multimedia analysis. More recently, for classification purposes in dynamic texture analysis, a higher order model based on tensor representation and factorization was proposed in our previous work [21] for the identification of smoke in video surveillance applications. The model was based on the consideration that the orthogonal matrix corresponding to the decomposition of the input signal unfolded along the time axis can be used as the mapping matrix of the system. However, since this model was developed as part of an ad-hoc solution for smoke detection in video sequences, there were some limitations for its general application in other classification problems. The main limitation stems from the fact that the size of the descriptor parameters is associated with the size of the sequence, hence, the model was only applicable to spatio-temporal cuboids, while the computational cost could increase significantly for long sequences. To this end, in this paper, we propose a

generalized sh-LDS model, which not only addresses the aforementioned problems by considering the observation data as a collection of time evolving signals, represented as a third order tensor, but it also enables the dimensionality reduction of the descriptor parameters and the stabilization of the model.

For the comparison of two dynamical models most of the researchers focus on the definition of distances or kernels. More specifically, a measure of the distance between two ARMA processes based on their cepstrum coefficients was presented by Martin in [9], while in [6] a notion of subspace angles between two ARMA models was defined as the principal angles between the column spaces generated by the observability matrices of the two models. Chan and Vasconcelos [7] proposed a probabilistic kernel based on Kullback-Leibier (KL) divergence between Gauss-Markov processes. The KL-kernels were derived for dynamic textures in both image and hidden state space. Recent advances on distances between probabilistic models from the area of 3D object or image retrieval could be also investigated for their application to dynamic texture classification For instance, Wang et al. [22] presented a discriminative probabilistic object modeling approach defining the distance between two objects as the upper bound of the KL divergence of the corresponding probabilistic models, while in [23] a compact semantic method along with a semantic distance matrix learning approach were proposed.

In [24], Chan and Vasconcelos presented a new dynamical model based on kernel PCA and the computation of Martin distance between the kernel dynamic textures. A family of kernels based on Binet-Cauchy theorem was presented in [8] and later Binet-Cauchy distance for LDSs based on kernel PCA was employed in [25]. More recently, Ravichandran et al. [26] attempted to model video sequences with a collection of LDSs, which are then used as features in a bag of systems approach, while in our previous work in [27] we used traditional LDSs and Martin distance to create histograms, as part of an ad-hoc algorithm, for the detection of flame in video surveillance systems. On the other hand, Turaga et al. [28] studied the geometry of the LDS space and proposed algorithms for supervised and unsupervised clustering on the Grassmann and Stiefel manifold for face and action recognition, while in [29] a kernel analysis on Grassmann manifold for action recognition was presented. In this paper, we propose a new algorithm, namely histogram of Grassmannian points (HoGP), aiming to improve the classification of multidimensional signals in the space of sh-LDS descriptor. Experimental results with different variations of LDS models and various distances have shown the great potential of the proposed algorithm.

III. Stabilized Higher Order LDS

A linear dynamical system is associated with a first order ARMA process with white zero mean IID Gaussian input. More specifically, the stochastic modeling of both signal dynamics and appearance is encoded by two stochastic processes, in which dynamics are represented as a time-evolving hidden state process $x(t) \in R^n$ and the observed data $y(t) \in R^d$ (e.g., for a video frame, d indicates the number of pixels in a frame $y(t)$) as a linear function of the state vector:

$$x(t+1) = Ax(t) + Bv(t) \quad (1)$$

$$y(t) = \bar{y} + Cx(t) + w(t) \quad (2)$$

where $A \in R^{nxn}$ is the transition matrix of the hidden state, while $C \in R^{dxn}$ is the mapping matrix of the hidden state to the output of the system (Table I contains the basic notations and their definitions). The quantities $w(t)$ and $Bv(t)$ are the measurement and process noise respectively, with $w(t) \sim N(0,R)$ and $Bv(t) \sim N(0,Q)$, while $\bar{y} \in R^d$ is the mean value of the observation data. The LDS descriptor, $M_{LDS} = (A, C)$, contains both the appearance information of the observation data $Y=[y(1), y(2),...., y(N)]$ modeled by C, and its dynamics that are represented by A. For the estimation of the system parameters, several approaches have been proposed based either on EM algorithm [3] or non-iterative subspace methods [4]. Since these approaches require high computational cost, a suboptimal method was proposed in [5], according to which the columns of the mapping matrix C can be considered as an orthonormal basis, e.g., a set of principal components. To this end, we decompose matrix Y (before the decomposition, we first need to subtract from Y the mean value $\bar{y}$) by applying singular value decomposition:

$$Y = U\Sigma V^T = CX \quad (3)$$

where the columns of the mapping matrix C are the principal components, i.e., $C=U$, and $X = [x(1), x(2), \dots, x(N)] = \Sigma V^T$ are the estimated states of the system. If we define $X_1 = [x(1), x(2), \dots, x(N-1)]$ and $X_2 = [x(2), x(3), \dots, x(N)]$, the transition matrix A, containing the dynamics of the signal, can be easily computed by using least squares as:

$$A = X_2 X_1^T (X_1 X_1^T)^{-1} \quad (4)$$

TABLE I
Basic Notations and Definitions

Symbol	Definition
Y	observed time series
X	hidden state time series
$y(t)$	observed data for time instant t
$x(t)$	hidden state for time instant t
d	dimension of the observed data
n	dimension of the hidden state
M	number of elements
N	number of samples
A	transition matrix
C	mapping matrix
U	orthogonal matrix
S	core tensor
a_{sh}	stabilized transition matrix in vector form
M_{sh}	stabilized higher order LDS descriptor
O_m^T	finite observability matrix of size m
V	tangent vector
G	Grassmannian point

A. The generalized form of higher-order LDS

As can be easily noticed from equation (3), LDS model

exploits information from only one element (or channel), thus, in the case of multidimensional time evolving data the concatenation of different components into one single element, i.e., *Y*, is required. To address this limitation, in this paper we propose a new model, which is called stabilized higher order LDS (sh-LDS), and we present a generalized form of the model for its application to various classification problems dealing with multidimensional time evolving data.

More specifically, let's consider a multidimensional time series, which is represented by a tensor $Y \in R^{MxNxd}$ where *M* is the dimension of data, i.e., the number of data elements, *N* is the number of samples and *d* indicates the number of observed data in each element per sample (e.g., for a color video sequence the number of elements, *M*, is equal to three, *N* is the number of frames and *d* indicates the number of pixels in a frame), as shown in Fig.1. To extract the dynamics of the multidimensional signal we need to define a stochastic process as in (1) and (2) and define parameters *A* and *C*. Since in our case the sequence of the observation data is represented by a third order tensor, we need to apply a generalization of the singular value decomposition for higher order tensors, such as higher-order SVD analysis [30].

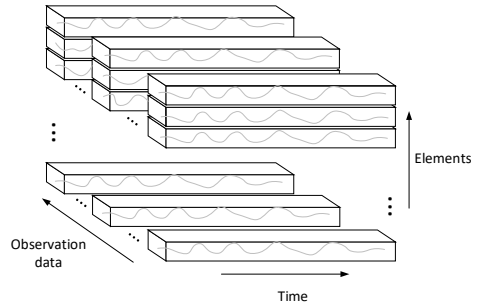

Fig. 1. The multidimensional time series is represented by tensor *Y* of size *M*x*N*x*d*

More specifically, we first subtract from *Y* the temporal average $\bar{Y}$ in order to construct a zero mean tensor in the time axis and then we decompose *Y* by applying higher order SVD analysis:

$$Y = S \times_1 U_{(1)} \times_2 U_{(2)} \times_3 U_{(3)} \quad (5)$$

where $S \in R^{MxNxd}$ is the core tensor, $U_{(1)} \in R^{MxM}$, $U_{(2)} \in R^{NxN}$ *and* $U_{(3)} \in R^{dxd}$ are orthogonal matrices containing the orthonormal vectors spanning the column space of the matrix unfolding $Y_{(i)}$ and $\times_i$ denotes the *i*-mode product between a tensor and a matrix [30], with *i*=1,2 and 3. Since the columns of the mapping matrix *C* of the stochastic process need to be orthonormal, we can easily choose one of the three orthogonal matrices of equation (5) to be equal to *C*. In addition, given the fact that the choice of matrices *A*, *C* and *Q* in equations (1) and (3) is not unique, we can consider $C=C_h=U_{(3)} \in R^{dxd}$ and

$$X = S \times_1 U_{(1)} \times_2 U_{(2)} \quad (6)$$

Hence, equation (5) can be reformulated as follows:

$$Y = X \times_3 C_h \Leftrightarrow Y_{(3)} = C_h X_{(3)} \quad (7)$$

where $Y_{(3)}$ and $X_{(3)}$ indicate the unfolding along the third dimension of tensors *Y* and *X* respectively. Similarly, by unfolding tensor *X* of equation (6) along the third dimension, we can easily define matrices X_1 and X_2. The transition matrix $A_h \in R^{dxd}$, containing the dynamics of the multidimensional time series, can then be easily computed by using least squares as in equation (4).

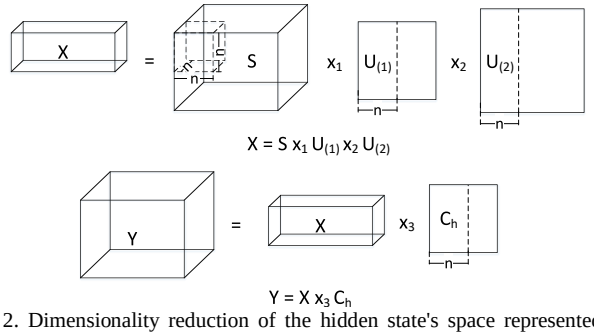

Fig. 2. Dimensionality reduction of the hidden state's space represented by tensor *X*.

Hence, the new descriptor M_h consists of two parameters, matrices A_h and C_h, which both belong to R^{dxd}. This is something that was expected, as in this generalized form of our descriptor, we have considered that i) the signal in each time instant *t* and for each element *e* (with *e*=1,2...*M*) is represented as $y_e(t) \in R^d$ and ii) the mapping matrix of the hidden state is equal to the orthogonal matrix that contains the orthonormal vectors spanning the space of the third dimension of tensor *Y* (i.e., the number of the observed data in each channel per sample). In this way, the size of matrices A_h and C_h is completely independent of time, i.e., the number of samples, and the dimension of the signal. Especially, the first enables us to model multidimensional signals of different temporal sizes (i.e., different number of frames or samples) and compare them without needing to apply first any sub-sampling technique in order to align the signals.

However, since the size of the descriptor's parameters is related to the total number of the observed data, the size of the two matrices can be significantly increased affecting thereby the computational cost of the method, as shown in the experimental results. To address the problem, we need to reduce the dimension of the hidden space, which is represented by tensor *X*. To produce the best approximation of *X*, we select the first *n* orthonormal columns of the orthogonal matrices $U_{(1)}$ and $U_{(2)}$, generating the truncated matrices $U_{(1)}^{(n)} \in R^{Mxn}$ and $U_{(2)}^{(n)} \in R^{Nxn}$ respectively, as well as the best approximation of the core tensor *S*, i.e., $S^{(n)} \in R^{nxnxn}$, with *n*<*d*. After the truncation of tensor *X*, i.e., $X^{(n)} \in R^{MxNxn}$, the dimension of the orthogonal matrix C_h needs to be reduced as well in order for equation (7) to be valid, as shown in Fig.2.

For the estimation of the transition matrix A_h from equation (4), we need to define the new matrices $X_1^{(n)}$ and $X_2^{(n)} \in R^{nx(MN-1)}$ after the unfolding of tensor $X^{(n)}$. Hence, the size of the transition matrix $A_h^{(n)} \in R^{nxn}$ depends solely on the dimension of the hidden state, while as expected the truncated

mapping matrix $C_h^{(n)} = U_{(3)}^{(n)}$ is of size *dxn*, i.e., $C_h^{(n)} \in R^{dxn}$.

B. The stabilized model

In order to ensure the stability of the system, the spectral radius of the transition matrix $A_h \in R^{nxn}$ needs to be smaller than 1, i.e., $|\lambda_1(A_h)| \leq 1$, where λ_1 denotes the first eigenvalue of matrix A_h, considering the eigenvalues in descending order of magnitude. However, the estimated matrix A_h from equation (4), i.e., using least squares, may be unstable although the system is stable [31]. As shown in the experimental results, stabilization can increase significantly the performance of the descriptor, however, this process is usually ignored and as a result locally optimal values of the system's parameters are used.

In this sub-section, we aim to improve the stability of the higher-order linear dynamical system by applying an approximation solution based on a convex optimization technique [32]. More specifically, to ensure a stable solution we reformulate equation (4) as follows:

$$A_{sh} = arg\min_{a_h}(a_h P a_h - 2q^T a_h + r) \tag{8}$$

where A_{sh} is the stable transition matrix, $a_h = vec(A_h^{(n)})$, $q = vec(X_1^{(n)} X_2^{(n)^T})$ with $a_h, q \in R^{n^2}$, $r = tr(X_2^{(n)^T} X_2^{(n)})$ with $r \in R$ and $P = I_n \otimes (X_1^{(n)^T} X_1^{(n)})$ with $P \in R^{n^2 xn^2}$. Here, I_n is the *nxn* identity matrix, $tr(\cdot)$ indicates the trace of a matrix, $vec(\cdot)$ operator converts a matrix to vector and $\otimes$ denotes the Kronecker product.

Our main constraint here in order to enforce stability criterion in (8) is that the spectral radius of A_{sh} should not be greater than 1. By decomposing the transition matrix $A_{sh} = U\Sigma V^T$ or $\Sigma = U^T A_{sh} V$, the stability criterion can be expressed as follows:

$$\lambda_1 = u_1^T A_{sh} v_1 \leq 1 \Rightarrow tr(v_1 u_1^T A_{sh}) \leq 1 \tag{9}$$

where u_1^T and v_1 are the singular vectors corresponding to eigenvalue λ_1. By setting $g = vec(u_1 v_1^T)$ and $a_{sh} = vec(A_{sh})$ the quadratic problem can be defined as:

$$\begin{aligned} &\text{minimize} \quad a_{sh} P a_{sh} - 2q^T a_{sh} + r \\ &\text{subject to} \quad g^T a_{sh} \leq 1 \end{aligned} \tag{10}$$

Starting with the initial value $a_{sh} = a_h$, the new transition matrix, $A_{sh} = mat(a_{sh})$ (where $mat(\cdot)$ operator converts a vector to matrix), is iteratively estimated until the stability criterion is satisfied.

Hence, the stabilized higher-order LDS descriptor (sh-LDS) can be defined as $M_{sh} = (A_{sh}, C_{sh})$, where $A_{sh} \in R^{nxn}$ is the stabilized matrix resulting from the solution of the quadratic problem and C_{sh} is equal to the truncated orthogonal matrix $C_h^{(n)}$. Similarly the stabilized higher-order LDS model is defined in matricized form by the following stochastic process:

$$\begin{aligned} x(t+1) &= mat(a_{sh})\, x(t) + Bv(t) \\ y(t) &= \bar{y} + U_{(3)}^{(n)} x(t) + w(t) \end{aligned} \tag{11}$$

where $y(t) \in R^{dxM}$ is the multidimensional observation data in time instant t and $x(t) \in R^{nxM}$ is similarly the multidimensional hidden state for the same time instant.

IV. Codebook Creation Based on Grassmann Geometry

Having defined the feature descriptor, i.e., $M_{sh} = (A_{sh}, C_{sh})$, we subsequently need to define a similarity metric between two sh-LDSs. However, the parameters of a sh-LDS descriptor do not lie in a Euclidean space. A common approach to address the problem is to estimate the Martin distance between two sh-LDSs, M_{sh}^1 and M_{sh}^2. The estimation of the Martin distance is based on the calculation of the subspace angles between the columns spaces of the extended observability matrices of the two models. More specifically, for a sh-LDS model, the extended observability matrix $O_\infty^T \in R^{\infty xn}$ is defined as:

$$O_\infty^T(M_{sh}) = [C_{sh}^T, (C_{sh}A_{sh})^T, (C_{sh}A_{sh}^2)^T, \dots] \tag{12}$$

and the Martin distance between the two sh-LDSs can be estimated as follows:

$$D_{M_{sh}}(M_{sh}^1, M_{sh}^2)^2 = -\ln \prod_i \cos^2\theta_i \tag{13}$$

where θ_i are the subspace angles [6] between the models.

However, an alternative, more efficient way to define a notion of similarity between sh-LDS descriptors can be achieved through the study of the geometric properties of the space in which the parameters of the descriptor lie. This study will also allow us to model the variations of the sh-LDS classes, which is a key issue in a classification problem. To this end, we approximate the extended observability matrix O_∞^T with the finite observability matrix $O_m^T \in R^{mdxn}$:

$$O_m^T(M_{sh}) = [C_{sh}^T, (C_{sh}A_{sh})^T, (C_{sh}A_{sh}^2)^T, \dots, (C_{sh}A_{sh}^{m-1})^T] \tag{14}$$

where *n* is the dimension of the state space, *d* the dimension of the observed features and *m* is usually chosen to be equal to *n*. Each column of the finite observability matrix can be considered as a *n*-dimensional subspace of R^{md} space. By applying a Gram-Scmidt orthonormalization procedure [33], the subspace spanned by the columns of O_m^T can be represented by an orthonormal basis:

$$O_m^T = GR \tag{15}$$

where $G \in R^{mdxn}$ is an orthogonal matrix, containing an orthonormal basis of the subspace, and matrix $R \in R^{nxn}$. The orthonormal martrix G corresponds to a point on a Grassmann manifold and for this reason we can also claim that a sh-LDS model can be considered as a Grassmannian point.

A. Grassmann Geometry

A Grassmann manifold can be considered as a quotient of the special orthogonal group *SO(n)*, i.e., the subset of all orthogonal matrices with determinant equal to +1. This simply means that we can extend the notion of tangent spaces, geodesics etc. from the base manifold *SO(n)* to the quotient space of Grassmann manifold. Since it can be easily shown that *SO(n)* is a Riemannian manifold [28], we can claim that Grassmann manifolds are endowed with a Riemannian structure, i.e., they form a special class of Riemannian manifolds [29], [34]. Hence, the distance between two sh-LDS models can be considered as the Riemannian distance between two subspaces, that is, the length of the shortest geodesic connecting the two Grassmannian points.

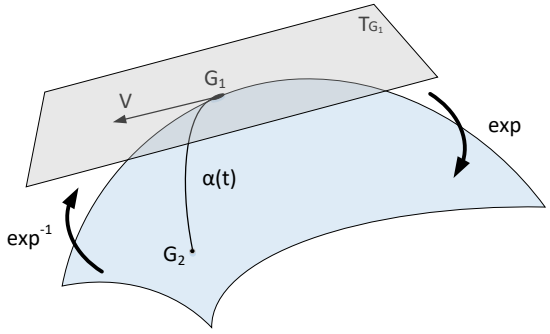

Fig. 3. The tangent space at point G_1 and the geodesic connecting G_1 and G_2 on the Grassman manifold. The exponential map enables us to map points of the tangent space onto the manifold.

Let us consider two sh-LDS models, represented by two orthogonal matrices G_1 and G_2, which are two points on a Grassmann manifold. As shown in Fig.3, T_{G_1} is the tangent space at point G_1 and *α(t)*, with $t \in [0,1]$, is a geodesic between G_1 and G_2 on manifold. The geodesic is a parametric curve on the manifold, starting at G_1 for t=0, i.e., *α(0)*, with velocity *V*. The velocity of the curve is represented by a vector on the tangent space of G_1. To map a point from the tangent space T_{G_1} onto manifold, we need a transformation function, so that the distance from G_1 on the tangent space (a tangent space is a Euclidean space) is the same as the geodesic distance between G_1 and the projected point on the manifold. This transformation function is called exponential map and leads to the estimation of the geodesic on the manifold:

$$a(t) = exp_{G_1}(tV) = G_1 exp(tV) \quad (16)$$

On the other hand, the inverse transformation, i.e., the inverse exponential map, enables us to map a Grassmannian point on a tangent space of another point, while preserving the distance between the points. In other words, using the inverse exponential map we can move from a Grassmann manifold to a Euclidean space, such as the tangent space of a manifold's point. Hence, the similarity metric between two sh-LDS descriptors, G_1 and G_2, can be defined as follows:

$$d(G_1, G_2) = \left\| exp_{G_2}^{-1} G_1 \right\|_F \quad (17)$$

where $\|.\|_F$ is the matrix Frobenius norm. Here we have to note that one can use various norms, e.g. $\|.\|_2$, however in our experiments we decided to use Frobenius norm since it gave us the best results. To estimate the inverse exponential map in equation (17), we first need to compute the orthogonal completion O_r of G_1:

$$O_r = I_n - \begin{bmatrix} G_{11} - I_n \\ G_{12} \end{bmatrix} [I_n - G_{11}^T]^{-1} [G_{11}^T - I_{dn} \quad G_{12}^T] \quad (18)$$

where $I_n \in R^{nxn}$ is an identity matrix, while $G_{11} \in R^{nxn}$ and $G_{12} \in R^{(md-n)xn}$ are the upper and lower parts of G_1 respectively. Subsequently, in order to estimate the direction matrix *B* from point G_1 to G_2, we compute the thin CS decomposition of matrix $O_r^T G_2$:

$$O_r^T G_2 = \begin{bmatrix} v_1 & 0 \\ 0 & \tilde{v}_2 \end{bmatrix} \begin{bmatrix} \Gamma(t) \\ -\Sigma(t) \end{bmatrix} S_1^T \quad (19)$$

with matrices $\Gamma(t)$ and $\Sigma(t) \in R^{nxn}$ to be diagonals with elements $\gamma_i = cos(t\theta_i)$ and $\sigma_i = sin(t\theta_i)$ respectively. By setting t=1, we can easily compute the angles θ_i and construct the diagonal matrix Θ, with diagonal elements the computed angles. The direction matrix $B \in R^{(md-n)xn}$, specifying the direction and speed of geodesic flow, is related to matrix Θ, since the singular value decomposition of *B* is equal to $\tilde{v}_2 \Theta v_1$. Hence, the tangent vector *V*, which is a skew symmetric matrix containing *B* as element, can be defined as:

$$V = \begin{bmatrix} 0 & B^T \\ -B & 0 \end{bmatrix} \quad (20)$$

The tangent vector, which is the result of the inverse exponential map, lies in the tangent space and through equation (17) enables us to measure the distance between two sh-LDS descriptors.

B. Histograms of Grassmannian Points

The inverse exponential map allows us to estimate a similarity metric between two Grassmannian points, however this metric by itself is not enough for the classification of multidimensional time series. The first reason is that it gives us no information about the variation of different classes in a classification problem and the second one is the inherent non-linearity of the observation data. The latter simply means that a point on a Grassmann manifold is a representation of a linear system, such as a sh-LDS, which, however, is used for the modeling of non-linear observation data. To address this problem, we propose the segmentation of the original multidimensional signal into equally sized elementary signals that can be more efficiently represented by a linear model i.e., a sh-LDS model. In this way, we can claim that each multidimensional time series can be represented by a set, or a "cloud", of points on a Grassmann manifold instead of a single point. Fig.4 illustrates an example of human motion, which is represented as a cloud of points on the Grassmann manifold by using a sliding window that divides the signal into equally sized elementary segments, i.e., a point is created in each time instant. As it shown in the experimental results section, this representation of the non-linear data improves significantly the

classification results compared with the use of a single descriptor for the representation of the whole time series.

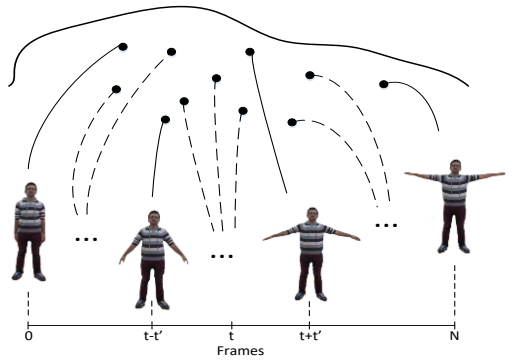

Fig. 4. The multidimensional time series, i.e., a human motion, is represented as a set of points on the Grassmann manifold. Each point corresponds to an elementary segment (time interval) of the multidimensional time series.

Therefore, if we manage to identify the most representative points on the manifold, i.e., codewords, we can easily define a codebook and then apply a bag of systems approach, where sh-LDSs play the role of feature descriptors. In this case each multidimensional time series can be represented as a histogram of the most representative Grassmannian points (HoGP). However, there are two significant problems that should be addressed here. First, we need to define a suitable notion of the "mean" on the Grassmann manifold and second we have to ensure that the estimation of the mean in a finite set of points on the manifold leads to a deterministic solution of the problem.

To fulfill the aforementioned requirements, we propose, as a first step, the identification of the most representative K Grassmannian points among the existing points of the set and then the estimation of the K means. To this end, we apply a K-medoids classification approach [35] considering as medoid of each class C_k, the local minimizer of function F_k:

$$F_k(m_k) = \frac{1}{n_k}\sum_{j=1}^{n_k} d(m_k, G_{kj}) \quad (21)$$

where k=1,...,K, n_k indicates the number of points G_{kj} in class k, m_k is the medoid of the class and $d(\cdot)$ denotes the distance between two Grassmannian points (see equation (17)).

The medoid m_k of a class C_k is the best approximation of its mean among the existing points of the class. However, the identification of the medoids is not enough for the creation of the codebook. To define the words of the codebook, we need to find the mean of each class, known as the Karcher mean [36]. Instead of picking points at random, we consider the K medoids as the initial means of the classes, thus ensuring the deterministic convergence of the algorithm. Subsequently, we map the Grassmannian points of the class on the tangent plane of the current mean and we estimate the average tangent vector according to the following equation:

$$\overline{V_k} = \frac{1}{n_k}\sum_{j=1}^{n_k} exp_{m_k}^{-1}(G_{kj}) \quad (22)$$

The new mean $\hat{m}_k$ of the class is considered then as the projected point on the manifold, which is estimated by moving m_k towards the direction of the average tangent vector $\overline{V_k}$ (typically setting factor t equal to 0.5):

$$\hat{m}_k = exp_{m_k}(t\overline{V_k}) \quad (23)$$

The new average tangent vector and mean of C_k class (equations (22) and (23)) are iteratively computed until the algorithm converges or the maximum iterations are exceeded.

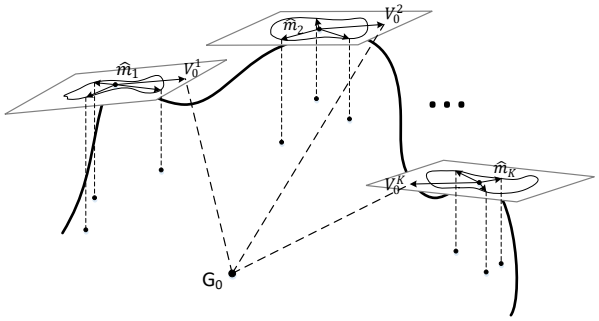

Fig. 5. Classification of a Grassmannian point to one of the K representative codewords for the creation of HoGP descriptor.

The estimation of the mean $\hat{m}_k$ of each class C_k leads to the creation of a codebook consisting of the most representative Grassmannian points. To generate the HoGP representation of a multidimensional time evolving data, we need to classify each elementary segment of data, represented as a point on the manifold, to one of the K codewords (experimental results with various codebook sizes are presented in Section V). To do so, we first need to reidentify the members of the K classes, based on the estimated means, $\hat{m}_k$, and then to model their variation. Since by mapping a point on the tangent space of a mean point, the geodesic distance between the two points remains the same as the Euclidean distance between the mean and the tangent vector, we will attempt to model the variation of K classes on the tangent spaces of the estimated means.

Let's consider that a class C_k, with mean $\hat{m}_k$, consists of n_k Grassmannian points G_j, with j=1....n_k. Using the inverse exponential map, we map each point on the tangent space of the mean point and we estimate the tangent vectors V_j^k. The variation of each class is then modeled by the covariance matrix of the tangent vectors V_j^k as follows (see also Algorithm 1):

$$S_k = cov\left(\left[exp_{\hat{m}_k}^{-1}G_1 \middle| exp_{\hat{m}_k}^{-1}G_2 \middle| \dots \middle| exp_{\hat{m}_k}^{-1}G_{n_k}\right]\right) = cov([V_1^k | V_2^k | \dots | V_{n_k}^k]) \quad (24)$$

Therefore, the distance of an arbitrary point G_0 (see Fig.5) from a codeword with mean $\hat{m}_k$ and covariance matrix S_k can be defined as follows:

$$d_k = tr(V_0^k S_k^{-1} (V_0^k)^T) \quad (25)$$

where $tr(\cdot)$ indicates the trace of a matrix and V_0^k is the tangent vector of point G_0 on the tangent space of $\hat{m}_k$.

Algorithm 1 HoGP descriptor: Codebook creation

Step 1: Given a set of multidimensional signals $\{Y_i\}$

Step 2: Divide the signals into equally sized elementary signals and produce a cloud of points $\{G_i\}$ on the Grassmann manifold

Step 3: Apply a *K*-medoids classification approach to define the m_k medoids of C_k classes.

Step 4: For each class estimate the Karcher mean $\widehat{m}_k$:

Step 4.1: $\widehat{m}_k = m_k$

Step 4.2: for each i=1,....,n_k map the Grassmannian points G_{kj} of the class on the tangent plane of the mean and estimate the average tangent vector:

$\overline{V_k} = \frac{1}{n_k}\sum_{j=1}^{n_k} exp_{m_k}^{-1}(G_{kj})$

Step 4.3: Compute the new mean: $\widehat{m}_k = exp_{m_k}(t\overline{V_k})$, with t=0.5

Step 4.4: Repeat steps 4.2 and 4.3 until the algorithm converges or the maximum iterations are exceeded.

Step 5: Re-identify the members of the *K* classes, based on the estimated $\widehat{m}_k$ means.

Step 6: Compute the covariance matrix of each class:

$S_k = cov([V_1^k | V_2^k | ... | V_{n_k}^k])$

Algorithm 2 Classification of HoGP descriptor

Step 1: Given a multidimensional signal $Y \in R^{MxNxd}$

Step 2: Represent the signal as a cloud of Grassmannian points:

For each elementary segment:

Step 2.1: HOSVD: $Y = S \times_1 U_{(1)} \times_2 U_{(2)} \times_3 U_{(3)}$

Step 2.2: Dimensionality reduction for $n<d$

Step 2.3: Estimate matrices C_{sh} and A_h: $C_{sh}^{(n)} = U_{(3)}^{(n)}$ and $A_h = X_2 X_1^T (X_1 X_1^T)^{-1}$

Step 2.4: Estimate the stabilized matrix $A_{sh} = mat(a_{sh})$ by solving the quadratic problem of equation (10)

Step 2.4: Define the Grassmannian point $G \in R^{mdxn}$ through equation (15)

End For

Step 3: Classify each point to one of the *K* codewords:

Step 3.1: Estimate the tangent vector V_0^k of each point on the tangent space of each codeword $\widehat{m}_k$

Step 3.2: Calculate the distance of each point from the *K* codewords: $d_k = tr(V_0^k S_k^{-1} (V_0^k)^T)$

Step 4: Define the HoGP descriptor $h = [h_1, h_2, ..., h_k]^T$

Step 5: Infer the label of the descriptor using a SVM classifier.

Finally, each multidimensional time-series is represented as a Term Frequency (TF) histogram $h = [h_1, h_2, ..., h_k]^T$ of the predefined codewords and a multi-class SVM classifier is trained with the distributions of these codewords. For the classification of a new time series, HoGP representation is estimated and the extracted histogram is provided to the SVM classifier to infer the label of the class (Algorithm 2).

V. Experimental Results

For the evaluation of the proposed methodology we conducted extensive tests in three different application domains: i) video-based surveillance systems, ii) dynamic texture categorization and iii) human action recognition. In the application domain of video based surveillance, we selected two application scenarios for early warning systems, namely flame and smoke detection. Our initial goal here was to apply sparse sampling and evaluate the performance of our methodology by exploiting information from all channels. To further examine the performance of our algorithm with more than three channels, in the case of smoke detection, we decided to add another one by creating a synthesized channel information. Since in both of these cases (flame and smoke) we deal with a binary problem (e.g., flame or flame colored object) in the next application domain of dynamic texture categorization our main goal was to evaluate the performance of our method by applying a dense sampling approach in a multiclass problem using a dataset containing videos with water, flags, flowers, fountains etc.. Finally, to demonstrate the generality of our approach, we also applied our method to more noisy multidimensional signals, such as those produced by depth motion sensors or video cameras, for the classification of action sequence dynamics. More specifically, ten different datasets were used in total, while the performance of the proposed method was compared against a number of state of the art approaches.

A. Video-based Surveillance Applications

1) Fire Detection

In this section, we aim to present a multivariate comparison of the proposed methodology against a number of different variations of linear dynamical systems. Our goal here is twofold: first we want to define the optimum set of parameters for our algorithm in dynamic texture analysis and second we want to demonstrate its superiority against other state of the art LDS approaches by taking into account all these factors that can directly affect the experimental results. Hence, in this way we can ensure a fair comparison between the different approaches. For the experimental evaluation of our method, we used two well known datasets containing fire and non-fire video sequences, such as the Firesense dataset [37] and the dataset used by Ko et al. in [38]. In order to avoid a dense sampling approach (this is a significant requirement in automatic video-based flame detection systems), we adopted for all algorithms the pre-processing step proposed in [27] for the identification of candidate image patches, i.e., those image patches for which there is a non-trivial indication of flame existence. In order to be able to classify a frame as flame or non-flame frame, i.e., to determine the time of the incident, we segmented each video in equally sized subsequences using a sliding time window *T* and we represented each subsequence as a histogram using a bag of features approach. In the experimental results of this section, we estimated the number of correctly detected frames, either flame or non-flame, out of the total number of frames in the datasets, which is a typical approach for the evaluation of flame/smoke detection algorithms.

More specifically, for the generalized sh-LDS descriptor,

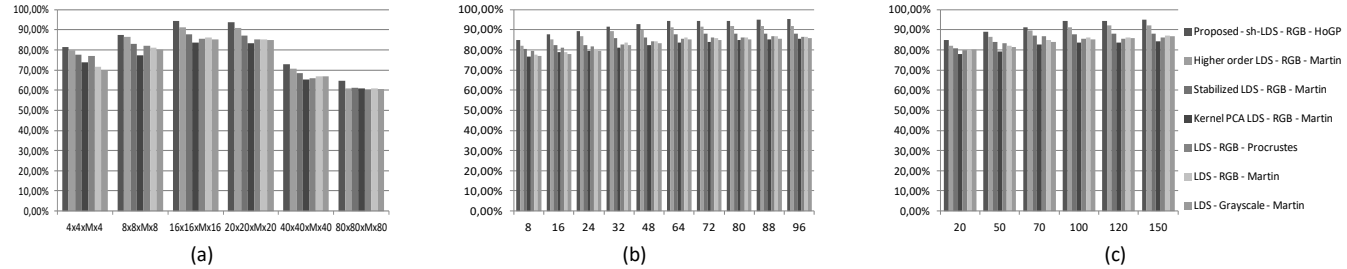

Fig. 6. Total detection rates using the datasets in [32] and [33] with different parameters sets: a) spatio-temporal patches of various sizes, b) different codebooks and c) various time window lengths.

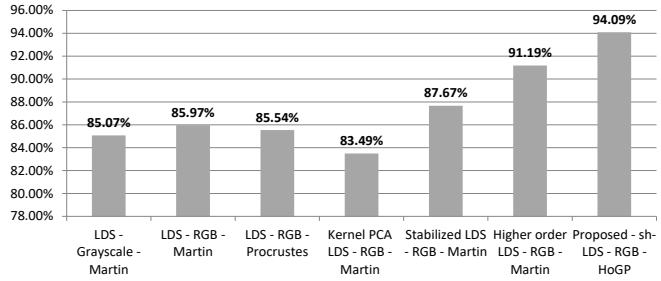

Fig. 7. Comparison of the proposed approach with six different LDS-based approaches using the best parameter set.

Fig. 8. A flame detection example from Firesense dataset.

we considered each image patch as a multidimensional time series represented by a tensor $Y \in R^{MxNxd}$, where M is the number of channels (i.e., three for an RGB image), N is the number of frames, i.e., the temporal length of the spatiotemporal image patch and d indicates the total number of pixels in a patch i.e., d depends on the size of the image patch. Similarly, for the LDS descriptor, we initially used grayscale information and then we concatenated the pixels' intensities of all three channels. Finally, for the h-LDS descriptor we created spatio-temporal cuboids as proposed in [21] for smoke detection. The proposed HoGP algorithm using as feature the generalized sh-LDS descriptor was compared against six different LDS-based approaches: LDSs using Martin distance based in [26] (using both grayscale and RGB data), Procrustes distance proposed in [39] and also used in [40], a non-linear LDS approach based on Kernel PCA as proposed in [26], a stabilized LDS model introduced in [32] and a higher-order LDS [21] using Martin distance as a similarity metric. In order to have a fair comparison, we used the same training and testing set for all methods and we applied a common bag-of-systems approach using as features the above descriptors.

Fig. 6 illustrates classification results with various patch sizes, number of codewords and time window lengths. More specifically, we compared the proposed method using six different patch sizes, ten codebook sizes and six time window lengths, i.e., twenty two comparison tests. For running the experiments, we adopted the following procedure: initially the size of the codebook was set to 64, while the temporal length of the sliding window to 100, since these values were successfully used in the past for LDS-based classification [27], and we run experiments with various patch sizes. Subsequently, we selected the best patch size and we compared the algorithms with various codebooks and finally, in order to produce the results of Fig. 6c, we kept the best patch and codebook size from the two previous series of experiments and we compared the methods using different time window lengths. As one can easily see in Fig. 6, the proposed method outperforms all other approaches independently of the parameters set used in the experiments. The best experimental results, which are illustrated in Fig. 7, are produced for patch size of 16x16x16, codebook size of 64 clusters (as we can see in Fig. 6b by using more than 64 codewords does not lead to a significant improvement in the results) and for time window length equal to 100. Fig. 8 illustrates indicatively a flame detection example using a video from Firesense dataset.

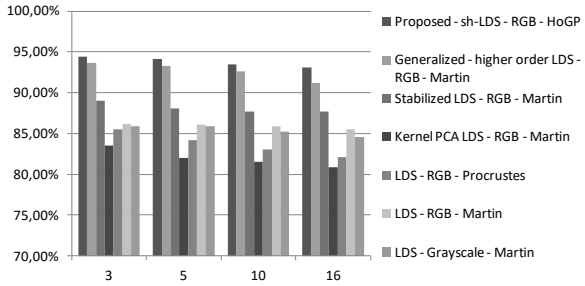

Fig. 9. Comparison of the proposed method against six different approaches and different hidden state's dimensions.

After the definition of the above parameters, we also attempted to examine the effect of the LDS size, i.e., the dimension n of the hidden state after the dimensionality reduction process described in Section II, in the performance of the proposed method. Fig. 9 displays the detection performance of the method against the choice of the hidden state's dimension and the other state of the art approaches (except for the higher order LDS [21], where dimensionality reduction is not applicable, therefore in Fig. 9 we present experimental results using the generalized higher order LDS and Martin distance). We can see that independently of the choice of the hidden state's dimension (as we explained in Section II, this choice affects the size of the descriptor's parameters, i.e., $A_{sh} \in R^{nxn}$ and $C_{sh} \in R^{dxn}$) the proposed method outperforms all other approaches. By reducing the size

of the descriptor's parameters, the detection performance seems to be improved, however, we cannot claim that in this application scenario the choice of the LDS size affects significantly the performance of the descriptor.

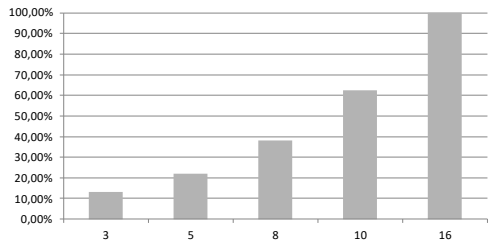

Fig. 10. Estimation of the computational cost with various dimensions as a percentage of the highest one, i.e., 16.

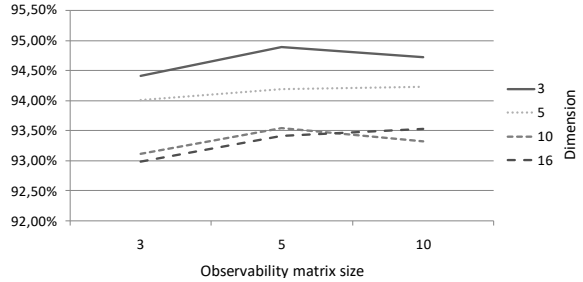

Fig. 11. Detection rates with various observability matrix sizes (horizontal axis) and hidden state dimensions.

On the other hand, the main advantage of selecting a low dimension in the space of the hidden variable lies in the reduction of the computation cost, as shown in Fig. 10. More specifically, in Fig. 10 we present experimental results of the computational costs of the method with various dimensions as a percentage of the highest one, i.e., n=16. As expected, by reducing the size of the descriptor, the computational cost becomes smaller, with that of n=3, i.e., $A_{sh} \in R^{3x3}$ and $C_{sh} \in R^{dx3}$, to provide the best results with a computational cost reduction of around 88% with respect to that of n=16. More specifically, 77.28% of the computational cost corresponds to the estimation of the sh-LDS descriptors (with the higher order SVD analysis and stabilization process to require 35.42% and 56.57% of the sh-LDS computational cost respectively), while the rest of the processes, i.e., the estimation of HoGP descriptor and the classification process require 20.82% and 1. 9% of the total computation cost. The average frame rate of the proposed method was 4.81fps (using a PC with Core i5, 2.4GHz processor), which is considered adequate for this application. Compared to other approaches, there is a slight reduction in the computational efficiency (higher order LDS: 5.4fps, stabilized LDS: 5.1fps, LDS-Martin: 5.7fps, kernel PCA-LDS: 5.89fps and LDS-procrustes:5.97fps) mainly due to the higher order SVD analysis and the stabilization process. However, as shown in the results the tensor representation of data does not increase prohibitively the computational burden, since the higher order decomposition takes place in the first step of the method.

Finally, in order to study the effect of the size of the observability matrix, m, in the performance of our method, we conducted tests with various m values and hidden state dimensions, i.e., different n values. As we can see in Fig. 11, the detection performance increases when we reduce the size of the observability matrix, with the best result to be produced for m=5 and n=3. Here, we have to note that the experimental results presented in this paragraph can be improved if we combine the proposed method with other spatio-temporal characteristics of flame as presented in [27], however, the development of an ad-hoc solution for the detection of fire is beyond the scope of this paper.

2) Smoke Detection

The case of smoke detection from video-based systems has many similarities to the problem of flame detection that we discussed in the previous paragraph. In this case our main goal is to identify smoke and discriminate it from other objects in nature that appear similar characteristics with smoke, e.g., clouds, shadows etc.. However, we have to keep the computation cost low, as in the case of flame, hence, a sparse sampling approach needs to be applied as well in this case. Since the visualization of the feature space of HOG descriptor can provide us extra information for the detection of smoke, in this experimental study we aim to examine the performance of our algorithm with more than three channels, i.e., R, G and B color components. To this end, we created a synthesized channel information, i.e., a fourth channel H, by visualizing the feature space of HOG descriptor, as proposed in [41].

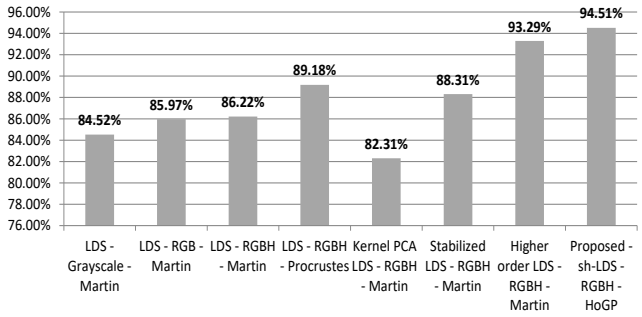

Fig. 12. Detection rates of all methods using the two smoke datasets.

More specifically, for the evaluation of the proposed method we used two popular datasets, the Bilkent [42] and the Visor [43] datasets, containing videos with smoke and non smoke frames (more than 24,000 frames in total). Moreover, we applied the same sparse sampling approach to all algorithms presented in Fig. 12, as proposed in [21]. For the standard LDS approach we initially used grayscale information and then we concatenated the RGB data and the RGBH data. For the kernel-PCA approach we also concatenated the RGBH data, while for the higher order LDS descriptor we created a tensor of size 16x16x4x16. Finally, for the proposed generalized sh-LDS descriptor we created a tensor of 4x16x256. As we can see in Fig. 12, the proposed method outperforms all other approaches with a detection rate of 94.51% (for the experimental results presented in Fig. 12, we have used the best parameter set for all algorithms as defined in the previous paragraph). In Fig. 13, we indicatively

present a smoke detection case in a video of Visor dataset (Fig. 13). As in the case of flame, one can combine the proposed method with other approaches aiming to model the spatiotemporal characteristic of smoke as we showed in [21], however, the development of an ad-hoc solution for smoke detection is beyond the scope of this paper.

(a)

Fig. 13. An example of smoke detection from Visor dataset.

B. *Dynamic Texture Categorization*

As was mentioned in the introduction of this section, the problems that we studied so far, i.e., flame and smoke detection, are binary classification problems, which require a sparse sampling approach. In this experimental study, our main goal is to evaluate the performance of our method in a multi-class classification problem using dense sampling. More specifically, for the evaluation of our method, we used one of the most popular datasets containing video sequences with dynamic textures, such as the DynTex Beta dataset [44]. The dataset consists of 162 video sequences classified in 10 different categories of dynamic textures, such as sea, vegetation, trees, flags, calm water, fountains, smoke, escalator, traffic and rotation (see Fig. 14).

Fig. 14. Screenshots from videos of Dyntex dataset containing various classes.

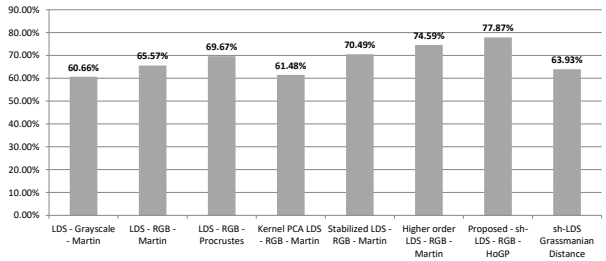

Fig. 15. Classification results of all methods using Dyntex dataset.

For the experimental results presented in Fig. 15, we used a training set consisting of four video sequences from each class, while the remaining video sequences were used for the evaluation of the algorithms. For comparing the algorithms, we used the optimum parameter set defined in the previous sub-sections. As we can clearly see in Fig. 15, the proposed method performs better than all the other state of the art approaches achieving improvements up to 3.28%. Moreover, the representation of video sequences as a cloud of points instead of a single Grassmannian point (last column of Fig. 15) improves the classification rate up to 13.94%. As shown in Fig. 16, which presents the confusion matrix for the proposed method, in most of the classes the classification rate is around 80%, with only those of flags and rotation presenting low rates due to the large intra-class variation, e.g., the rotation class contains videos of windmills, wheels etc..

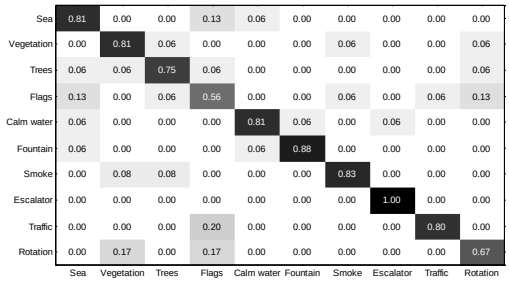

Fig. 16. Confusion matrix of the proposed method on Dyntex dataset.

C. *Human Action Recognition*

In all above cases, we dealt with dynamic textures analysis, which is a typical application domain of linear dynamical systems. In this section, we attempt to evaluate the performance of the proposed methodology in a different application domain, such as this of human action recognition studying two application scenarios: i) action recognition from depth sensor and ii) action recognition from video sequences.

1) *Action Recognition from depth sensors*

For the first scenario, we used three datasets with different characteristics containing skeletal data recorded using Microsoft Kinect sensor. More specifically, we initially used CERTH dataset [45] (see Fig.17), which contains a relative small number of actions, i.e., six different actions (bend forward, left kick, right kick, raise hands, hand wave, push with hands), performed by 6 subjects, each repeated 10 times (i.e., 360 actions in total). Then, we used the well-known G3D dataset [46], containing a large range of gaming actions, i.e., 20 gaming actions (punch right, punch left, kick right, kick left, defend, golf swing, tennis swing forehand, tennis swing backhand, tennis serve, throw bowling ball, aim and fire gun, walk, run, jump, climb, crouch, steer a car, wave, flap and clap), repeated 3 times by 10 subjects (i.e., 600 actions in total). Finally, the third dataset was the popular Microsoft Research Cambridge-12 Kinect gesture database (MSRC-12) [47], which contains a smaller number of actions, i.e., 12 actions (repeated 4-5 times per person), but with a high intra-class variation, i.e., 30 people. The MSRC-12 dataset is partitioned along different methods of instruction given to the subjects such as text and video. We used the part of the dataset where only video instructions were given (1608 sequences). All three datasets contain tracks of 20 skeleton joint positions.

To evaluate the performance of our method, we segmented the multidimensional signal into equally sized elementary segments using a sliding time window of 16 frames, which is a temporal length that gave us good results in dynamic texture analysis and also enables us to extract an adequate number of

sh-LDSs (for each elementary segment we created a tensor of size 3x16x20 i.e., 3 elements corresponding to the x,y and z coordinates, 16 is the temporal length and 20 the number of skeletal joints) for the creation of the histogram of HoGP descriptor. Moreover, in this way we can accomplish a better representation of human motion, as each elementary segment can be modeled better by a linear dynamical system than the whole non-linear sequence of data. For this reason, we provide experimental results based on the modeling of the whole signal (LDSs using Martin and Procrustes distance, Kernel PCA LDS, stabilized LDS, the generalized higher-order LDS with Martin distance and sh-LDS with Grassmanian distance) as well as on the modeling of the segmented signal using histogram approaches (LDS and sh-LDS with Martin distance). Similarly to the dynamic texture analysis, we provide an evaluation study with various LDS sizes (for all approaches) and codewords number (only for the three histogram approaches). As we can see in Fig. 18 and 19, the proposed method outperforms all other approaches providing the best results for LDS size equal to 3 (this dimension provides a computational cost reduction of around 90% as in the case of dynamic textures) and codebook size of 128.

Fig. 17. Screenshots from CERTH dataset a) raise hands, b) right kick, c) bend forward, d) push with hands, e) left kick and f) hand wave.

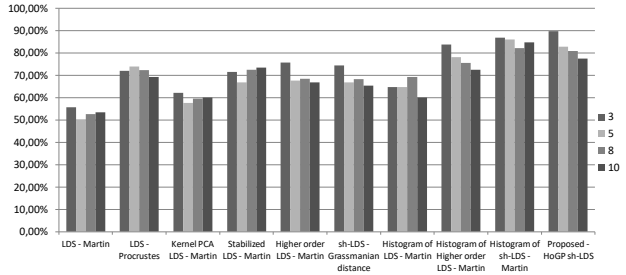

Fig. 18. Experimental results with different LDS sizes.

Table II presents the classification rates of the proposed method against eight LDS variations and four other state of the art algorithms, i.e., Dynamic Time Warping [48], Conditional Random Fields [49], Hidden Markov Models [50] and Restricted Boltzmann Machine [51] on the three datasets. For all algorithms, we used as input signal the joints coordinates, although one can also use other skeletal representations, as those proposed in [52], however, such a study is out of the scope of this paper. Moreover, the proposed method can be combined with other local or global models as in [53] or it can be used for the fusion of depth cameras and inertial sensors as in [54], however, again the goal of this paper is not to propose an ad-hoc motion recognition algorithm, but a general approach for the classification of multidimensional time-evolving data. In the experimental results presented in Table II, we used the same training and testing set for all algorithms in order to have a fair comparison (i.e., for CERTH dataset we used 6 instances of each action per subject for training and 4 instances for testing, for G3D dataset 1 instance of each action per subject used for training and 2 instances of each action for testing and finally for MSRC-12 37% of the dataset was used for training and 63% for testing). For the proposed method, we set the size of the observability matrix equal to 5, as in the case of dynamic texture, since it gave us again the best results, while for the Dynamic Time Warping algorithm, we initially adopted a K-medoid approach before the classification process in order to define the most representative motion of each class.

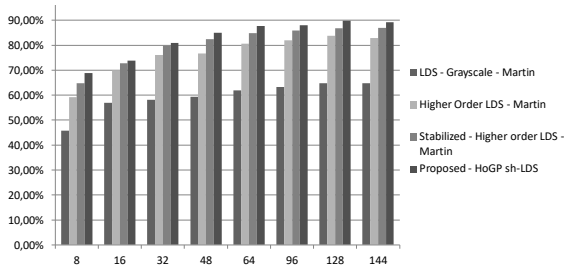

Fig. 19. Experimental results with various codebook sizes.

TABLE II
EXPERIMENTAL RESULTS FOR ALL DATASETS

	CERTH	G3D	MSRC12	Aver.
LDS - Martin	77.08%	50.00%	40.07%	55.72%
LDS - Procrustes	93.75%	65.25%	57.04%	72.01%
Kernel PCA LDS - Martin	90.97%	52.50%	43.08%	62.18%
Stabilized LDS - Martin	90.27%	64.50%	60.11%	71.63%
Higher order LDS - Martin	96.52%	63.50%	67.26%	75.76%
sh-LDS - Grassm.distance	93.75%	69.75%	59.72%	74.41%
Histogram of LDS - Martin	90.97%	53.25%	78.06%	74.09%
Histogram of shLDS - Martin	97.22%	84.25%	78.96%	86.81%
Proposed - sh-LDS - HoGP	**98.61%**	**90.75%**	**80.15%**	**89.84%**
Dynamic Time Warping	87.50%	57%	48.11%	64.20%
Conditional Random Fields	97.91%	69.25%	67.95%	78.37%
Hidden Markov Model	96.52%	77.40%	76.20%	83.37%
Restricted Boltzmann Machine	97.10%	84%	79.80%	86.97%

The last column of Table II presents the average classification rates on three datasets. As we can see, the proposed method outperforms all other approaches with the HoGP algorithm achieving improvements up to 3.03% from Martin distance (the average distance between sh-LDS-HoGP and sh-LDS-Martin) and up to 15.43% from a simple Grassmannian distance (the average distance between sh-LDS-HoGP and sh-LDS-Grassmann), while it also provides improvements up to 15.75% from histograms of traditional LDSs with Martin distance as the similarity metric.

2) Action Recognition from video sequences

The final scenario that we investigated was that of action recognition from video sequences. In this scenario, one can form the multidimensional signal using different types of features, e.g., raw pixels, silhouettes, shape features etc.. In this paper, we formed the multidimensional signal by using the elements of MBH (Motion Boundary Histograms) descriptor, which contains the relative motion between pixels. More specifically, for each MBH descriptor we formed a

multidimensional signal of size 8x3x4 and we represented a video sequence as a cloud of Grasmannian points (each one corresponding to a MBH descriptor). For the estimation of the MBH local descriptors, instead of using dense optical flow we applied sparse MPEG flow [55], which improves the speed of feature extraction and implies only minor reduction in classification performance. Table III presents experimental results with various codebook (CB) sizes for a number of descriptors using UCF sports dataset [56]. For the comparison of the descriptors, we used a standard histogram encoding (VLAD encoding or Fisher Vector representation can also be applied to improve performance as in [55]), the Leave-One-Out approach and the original sequences of the dataset, i.e., we did not add the flipped version of sequences as in [57].

TABLE III
EXPERIMENTAL RESULTS FOR ALL DESCRIPTORS IN UCF SPORTS DATASET

CB Size	HoGP (MBH)	HOF	HOG	MBHx	MBHy	MBH	HOG-MBH
16	**71.42%**	65.41%	69.17%	70.67%	72.18%	69.92%	63.90%
32	**81.19%**	66.91%	69.19%	71.42%	78.94%	69.94%	69.92%
64	**81.95%**	68.42%	73.68%	73.71%	77.44%	75.18%	71.42%
92	**77.44%**	67.66%	70.67%	71.33%	71.45%	71.42%	72.93%

TABLE IV
EXPERIMENTAL RESULTS IN HMDB51 DATASET

HoGP(MBH)	MBHx	MBHy	MBH
32.47%	27.95%	28.44%	28.79%

As we can see in Table III, HoGP outperforms all other descriptors regardless of the codebook size. The best experimental results for HoGP are produced for codebook size of 64, achieving improvements up to 6.77% compared to MBH. Table IV presents experimental results of HoGP against MBHx, MBHy and MBH descriptors (since they yielded the best results in UCF sports) in a larger dataset, namely HMDB51 [58]. For the experimental results in Table IV, we used codebook size equal to 64 for all descriptors, since this size produced the best results in Table III and we followed the same evaluation approach. As we can see HoGP outperforms again improving the performance of MBH descriptor up to 3.68%. We have to note here that one can form the multidimensional signal using other descriptors or improve the classification rate by combining HoGP with other spatio-temporal descriptors, however, this study is out of the scope of this paper.

VI. CONCLUSION

In this paper we presented a novel methodology for the modeling and classification of multidimensional time series by exploiting the correlation between the different channels of data and the geometric properties of the space in which the parameters of the descriptor lie. More specifically, we proposed a novel generalized form of a stabilized higher-order linear dynamical system (sh-LDS) and we introduced a new methodology, namely Histograms of Grassmannian Points (HoGP) for the classification of multidimensional time evolving data in various computer vision problems dealing with the analysis of dynamic scenes. As we showed in the experimental results, the proposed methodology improves the performance of LDSs in various application domains against a number of state of the art approaches. In the future, we aim to apply the proposed methodology to the classification of multimodal multidimensional data, such as those produced by multispectral imaging systems e.g., for dynamic texture analysis, or different motion sensing technologies for human action recognition.

ACKNOWLEDGEMENT

The research leading to these results has received funding from EC under grant agreement no. FP7-ICT-600676 "i-Treasures".

REFERENCES

[1] B. Boots, "Learning Stable Linear Dynamical Systems", [Online]. Avail.: https://www.ml.cmu.edu/research/dap-papers/dap_boots.pdf [Accessed 30 05 2016].

[2] R. Vidal and P. Favaro, "Dynamicboost: Boosting Time Series Generated by Dynamical Systems", IEEE Conf. Comp. Vision, 2007.

[3] R. Shumway and D. Stoffer, "An Approach to Time Series Smoothing and Forecasting Using the EM Algorithm", J. Time Series Analysis, vol. 3, no. 4, pp. 253-264, 1982.

[4] P.V. Overschee and B.D. Moor, "N4SID : Subspace Algorithms for the Identification of Combined Deterministic-Stochastic Systems", Automatica, vol. 30, pp. 75-93, 1994.

[5] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto, "Dynamic Textures", Int'l J. Computer Vision, vol. 51, no. 2, pp. 91-109, 2003.

[6] K. De Cock and B. De Moor, "Subspace angles between ARMA models", Syst. Control Lett., vol. 46, pp. 265–270, 2002.

[7] A. B. Chan and N. Vasconcelos, "Probabilistic kernels for the classification of auto-regressive visual processes", in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2005, pp. 846–851.

[8] S. V. N. Vishwanathan, A. J. Smola, R. Vidal, "Binet Cauchy kernels on dynamical systems and its application to the analysis of dynamic scenes", J. Comput. Vis., vol. 73, no. 1, pp. 95–119, 2007.

[9] R.J. Martin, "A metric for ARMA processes", Trans. Sig. Proc. vol. 48, no. 4, pp. 1164-1170, 2000.

[10] R. Duda, P. Hart, and D. Stork, "Pattern Classification", Wiley-Interscience, Oct. 2004.

[11] Y. Han, Y. Yang, Y. Yan, Z. Ma, N. Sebe, X. Zhou, Semi-Supervised Feature Selection via Spline Regression for Video Semantic Recognition, IEEE Trans Neural Networks and Learning Systems, Vol. 26, No. 2, Feb 2015, pp.252-64.

[12] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden Markov models for complex action recognition IEEE Conference on Computer Vision and Pattern Recognition., 1997, pp. 994–999.

[13] X. Zhuang, X. Zhou, M. Hasegawa-Johnson, T.S. Huang, Face Age Estimation Using Patch-based Hidden Markov Model Supervectors, 19th International Conference on Pattern Recognition, Florida, USA, Dec 8-11, 2008.

[14] X. Zhou, X. Zhuang, S. Yan, S. Chang, M. Hasegawa-Johnson, T. S. Huang. SIFT-Bag kernel for video event analysis, 6th ACM international conference on Multimedia, Oct. 2008.

[15] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas, "Conditional models for contextual human motion recognition" Comp. Vis. and Im. Under., vol. 104, no 2-3, pp. 210–220, 2006

[16] F. Caillette, A. Galata, and T. Howard, "Real-time 3-D human body tracking using learnt models of behaviour," Computer Vision and Image Understanding, vol. 109, no. 2, pp. 112–125, 2008.

[17] P. Turaga, R. Chellappa, A. Veeraraghavan, Advances in Video-Based Human Activity Analysis: Challenges and Approaches, Advances in Computers, vol. 80, 2010, pp. 237–290.

[18] R. Constantini, L. Sbaiz and S. Susstrunk, "Higher order SVD analysis for dynamic texture synthesis", IEEE Trans Image Processing, vol. 17, no. 1, pp. 42-52, Jan 2008.

[19] W. Guo, I. Kotsia, I. Patras, Tensor learning for regression, IEEE Trans. on Image Processing, Vol. 21, No. 2, pp. 816–827, 2012

[20] J. Zhang, Y. Hana, J. Jiang, "Tensor Rank Selection for Multimedia Analysis", Journal of Visual Communication and Image Representation, vol. 30, pp. 376–392, Jul. 2015

[21] K. Dimitropoulos, P. Barmpoutis and N. Grammalidis, "Higher Order Linear Dynamical Systems for Smoke Detection in Video Surveillance Applications", IEEE Trans. on Circuits and Systems for Video Technology, DOI: 10.1109/TCSVT.2016.2527340.

[22] M. Wang, Y. Gao, K. Lu, Y. Rui, "View-based Discriminative Probabilistic Modeling for 3D Object Retrieval and Recognition", IEEE Trans. on Image Proces., vol. 22, no.4, pp. 1395–1407, 2013.

[23] M. Wang, W. Li, D. Liu, B. Ni, J. Shen, S. Yan, "Facilitating Image Search With a Scalable and Compact Semantic Mapping", IEEE Trans. Cybernetics. Vol. 45, no. 8, pp. 1561-1574, Aug. 2015.

[24] A. Chan and N. Vasconcelos, "Classifying Video with Kernel Dynamic Textures", Proc. IEEE CVPR, pp. 1-6, 2007.

[25] R Chaudhry, A Ravichandran, G Hager, R Vidal, " Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions", IEEE Conf. Computer Vision and Pattern Recognition, pp. 1932-1939, 2009

[26] A. Ravichandran, R. Chaudhry and R. Vidal, "Categorizing dynamic textures using a bag of dynamical systems", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 2, pp. 342-353, February 2013.

[27] K. Dimitropoulos, P. Barmpoutis and N. Grammalidis, "Spatio-Temporal Flame Modeling and Dynamic Texture Analysis for Automatic Video-Based Fire Detection", IEEE Trans. Circuits and Systems for Video Tech., vol. 25, no. 2, pp. 339-351, Feb. 2015.

[28] P. Turaga, A. Veeraraghavan, A. Srivastava and R. Chellappa, "Statistical Computations on Grassmann and Stiefel Manifolds for Image and Video based Recognition", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 33, no. 11, pp. 2273-2286, 2011.

[29] Mehrtash T. Harandi, Conrad Sanderson, Sareh Shirazi, and Brian C. Lovell "Kernel analysis on Grassmann manifolds for action recognition", Pattern Recogn. Lett. vol. 34, no. 15 pp. 1906-15, 2013.

[30] C. T. Kuo, "Higher order SVD: theory and algorithms", 2013.

[31] N.L.C. Chui, J.M. Maciejowski, "Realization of stable models with subspace methods", Int. J. Automatica, vol. 32, no. 11, pp. 1587-1595, 1996.

[32] S. M. Siddiqi, B. Boots and G. J. Gordon, "A Constraint Generation Approach to Learning Stable Linear Dynamical Systems", in Proc. Advances in Neural Information Processing Systems 20, pp. 1329-1336, 2007.

[33] Arfken, G. "Gram-Schmidt Orthogonalization", in Mathematical Methods for Physicists, 3rd ed. Orlando, FL: Academic Press, pp. 516-520, 1985.

[34] J. Hamm and D. D. Lee, "Grassmann discriminant analysis: a unifying view on subspace-based learning", 25th International Conference on Machine Learning, pp. 376-383, 2008.

[35] L. Kaufman and P. Rousseeuw, "Finding Groups in Data: An Introduction to Cluster Analysis", Wiley, 1990.

[36] H. Karcher, " Riemannian center of mass and mollifier smoothing", in Communications on Pure and Applied Mathematics, vol. 30, no. 5, pp. 509 - 541, 1977.

[37] FIRESENSE, "FIRESENSE Project Protection of Cultural Heritage", [Online]. Available: http://www.firesense.eu/ [Accessed 30 05 2016].

[38] B. Ko, S. Ham and J. Nam, "Modeling and formalization of fuzzy finite automata for detection of irregular fire flames", IEEE Transactions on Circuits and Systems for Video Technology, vol. 21, no. 12, pp. 1903-1912, December 2011.

[39] Y. Chikuse, "Statistics on special manifolds", Lecture Notes in Statistics, Springer, New York, 2003

[40] P. Turaga, A. Veeraraghavan, R. Chellappa, "Statistical analysis on Stiefel and Grassmann manifolds with applications in computer vision", Computer Vision and Pattern Recognition, pp. 1-8, 2008.

[41] C. Vondrick, A. Khosla, T. Malisiewicz and A. Torralba, "HOGgles: Visualizing Object Detection Features", in International Conference on Computer Vision, Sydney, Australia, Dec. 2013.

[42] U. B. Toreyin, Y. Dedeoglu and E. A. Cetin, "Wavelet based real-time smoke detection in video", in EUSIPCO, 2005.

[43] P. Piccinini, S. Calderara and R. Cucchiara, "Releable smoke detection system in the domains of image energy and color", in IEEE International Conference on Image Processing, ICIP, 2008.

[44] R. Péteri, S. Fazekas and M. J. Huiskes, "DynTex: A Comprehensive Database of Dynamic Textures", in Pattern Recognition Letters, vol. 31, no. 12, pp. 1627-1632, 2010.

[45] K. Dimitropoulos, P. Barmpoutis, A. Kitsikidis, N. Grammalidis, "Extracting dynamics from multi-dimensional time-evolving data using a bag of higher-order Linear Dynamical Systems", Inter. Conf. on Computer and Vision Theory and Applications, Feb. 2016.

[46] V. Bloom, D. Makris, V. Argyriou, "G3D: A gaming action dataset and real time action recognition evaluation framework", IEEE CVPR, pp. 7-12, 2012

[47] S. Fothergill, H. M. Mentis, P. Kohli, S. Nowozin, "Instructing people for training gestural interactive systems", CHI, ACM, Joseph A. Konstan and Ed H. Chi and Kristina H, pp. 1737-1746, 2012

[48] G. Ten Holt, M. Reinders, and E. Hendriks, "Multi-dimensional dynamic time warping for gesture recognition", Conf. of the Advanced School for Computing and Imaging, 2007.

[49] Conditional Random Field. [Online]. Available: https://www.cs.ubc.ca/~murphyk/Software/CRF/crf.html [Accessed 30 05 2016].

[50] Hidden Markov Model. [Online]. Available: https://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html [Accessed 30 05 2016].

[51] Deep Neural Network. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/42853-deep-neural-network [Accessed 30 05 2016].

[52] R. Vemulapalli, F. Arrate and R. Chellapa, "Human action recognition by representing 3D skeletons as points in a Lie group", IEEE CVPR, pp. 588 - 595, 2014.

[53] S. Nie and Q. Ji, "Capturing global and local dynamics for human action recognition", International Conference on Pattern Recognition, 2014.

[54] C. Chen, R. Jafari, N. Kehtarnavaz, "Improving human action recognition using fusion of depth camera and inertial sensors", IEEE Trans. Human-Machine Syst., vol. 45, no. 1, pp. 51-61, Feb. 2015.

[55] V. Kantorov and I. Laptev, "Efficient feature extraction encoding and classification for action recognition" CVPR, Jun. 2014.

[56] UCF Sports Action Data Set. [Online]. Available: http://crcv.ucf.edu/data/UCF_Sports_Action.php [Accessed 20 09 2016].

[57] H. Wang, A. Klaser, C. Schmid, C. L. Liu. Dense trajectories and motion boundary descriptors for action recognition. IJCV, 2013.

[58] HMDB: a large human motion database. [Online]. Available: http://serre-lab.clps.brown.edu/resource/hmdb [Accessed 20 09 2016].

Kosmas Dimitropoulos received his B.Sc degree in Electrical and Computer Engineering from Democritus University and his Ph.D. degree in Applied Informatics from Macedonia University of Thessaloniki in 2001 and 2007 respectively. He is currently a post-doctoral research fellow at the Information Technologies Institute of the Centre for Research and Technology Hellas (ITI-CERTH) and a visiting lecturer at the University of Macedonia. His main research interests include computer vision, pattern recognition, 3D motion analysis from depth cameras, 3D graphics and visualization. He has participated in several European and national research projects and he has served as a regular reviewer for a number of international journals and conferences.

Panagiotis Barmpoutis received his B.Eng. & M.Eng. in Electrical and Computer Engineering from the Aristotle University of Thessaloniki in 2009. He also received his MSc in Forestry Informatics and his MSc in Medical Informatics from the Aristotle University of Thessaloniki in 2012 and 2013 respectively. From 2012, he is a Research Assistant in the Information Technologies Institute at Centre for Research and Technology Hellas (CERTH). His current research interests lie in the areas of computer vision and applications, real-time image processing, analysis and visualization, pattern recognition and machine learning.

Alexandros Kitsikidis received his B.Sc degree in Informatics from the Aristotle University of Thessaloniki in 2009. From 2012, he is a Research Assistant at the Information Technologies Institute of the Centre for Research and Technology Hellas (CERTH). His research interests include computer graphics, virtual and augmented reality applications, image processing and computer vision. Currently, he is working towards receiving the Master's Degree specializing in Digital Media.

Nikos Grammalidis is a Senior Researcher (Researcher Grade B) at the Information Technologies Institute - Centre of Research and Technology Hellas. He received the B.S. and Ph.D. degrees in Electrical and Computer Engineering from the Aristotle University of Thessaloniki, in 1992 and 2000, respectively. Prior to his current position, he was a researcher in 3D Imaging Laboratory at the Aristotle University of Thessaloniki. His main research interests include computer vision, signal, image and video processing, stereoscopy and multiview image sequence analysis and coding. Since 1992, he has been actively involved in more than 25 EC and National projects. He has served as a regular reviewer for a number of international journals and conferences.