Locally Time-Invariant Models of Human Activities using Trajectories on the Grassmannian

Pavan Turaga and Rama Chellappa
Center for Automation Research, University of Maryland, College Park, MD 20742
{pturaga,rama@umiacs.umd.edu}

Abstract

Human activity analysis is an important problem in computer vision with applications in surveillance and summarization and indexing of consumer content. Complex human activities are characterized by non-linear dynamics that make learning, inference and recognition hard. In this paper, we consider the problem of modeling and recognizing complex activities which exhibit time-varying dynamics. To this end, we describe activities as outputs of linear dynamic systems (LDS) whose parameters vary with time, or a Time-Varying Linear Dynamic System (TV-LDS). We discuss parameter estimation methods for this class of models by assuming that the parameters are locally time-invariant. Then, we represent the space of LDS models as a Grassmann manifold. Then, the TV-LDS model is defined as a trajectory on the Grassmann manifold. We show how trajectories on the Grassmannian can be characterized using appropriate distance metrics and statistical methods that reflect the underlying geometry of the manifold. This results in more expressive and powerful models for complex human activities. We demonstrate the strength of the framework for activity-based summarization of long videos and recognition of complex human actions on two datasets.

1. Introduction

Modeling and recognition of human activities from video is a problem with increasingly important applications including video surveillance, video summarization, anomaly detection and motion synthesis. Complex human activities have frequently been modeled as a composition of simpler events. In several domains, it has been observed that human activities are better described as a continuum of actions where the individual boundaries between actions are often blurry [25]. To draw a parallel to language processing, it has been long known in the speech community that words spoken in isolation sound quite different when spoken in continuous speech. This is commonly attributed to 'co-articulation' and 'assimilation' effects. Similarly, when actions appear in a connected form, it is hard to identify precisely where an action ends and where another begins. Consider the action shown in figure 1 (a) and a synthesized version which relies on finding segment boundaries and fitting models to each segment in figure 1 (b). As can be seen, segmentation followed by modeling causes abrupt changes to appear at segment boundaries during synthesis. This effect is also observed in sign-language where gestures are influenced by adjacent gestures [25], making segmentation and recognition difficult.

Activities may also be viewed from a stochastic process point of view. In this context, 'stationarity' or 'non-stationarity' is an important property of the stochastic process under consideration. Stationarity requires that the ensemble statistics of the process do not change with time. On the other hand, 'time-invariant' and 'time-varying' refer to the properties of the model used to describe a given stochastic process. A good discussion of the relation between stationary processes and time-invariant models is given in [3]. A key observation is that if a process is stationary, it can be well described by time-invariant models such as the Gauss-Markov model [18]. Now one might ask the question whether activities are stationary or non-stationary. Consider the common activities dataset of [23]. Each activity in the dataset contains 10 executions from 2 views. Considering each execution to be a realization of a random process $X(t)$, we compute the pdf of the random variable at each time instant $f_X(t)$, by fitting a parametric Gaussian estimated from the ensemble. If the activity is indeed stationary, then the pdf's at time-instants t and $t + \delta$ would be identical. We will answer the question using empirical estimates of KL divergence.

We computed the KL-divergence between the pdfs as a function of the lag δ averaged over all time-instants i.e. $KL_{avg}(\delta) = \frac{1}{M}\sum_{t=0}^{M-1} KL(f_X(t), f_X(t+\delta))$. Figure 2 shows how KL_{avg} varies with δ for different activities. As is evident, the statistical properties of the activity vary smoothly but significantly over time even for these simple actions. This suggests that complex human activities cannot

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

Figure 1. (a) Original sequence taken from the common activities dataset [23], (b) Synthesis by a sequence of linear dynamic models with boundaries shown by vertical yellow lines, (c) Synthesis by a continuous time-varying model. It can be seen that when actions are segmented and modeled using switching models, the synthesis results show abrupt changes in pose across boundaries whereas the time-varying model results in a much more natural evolution of poses.

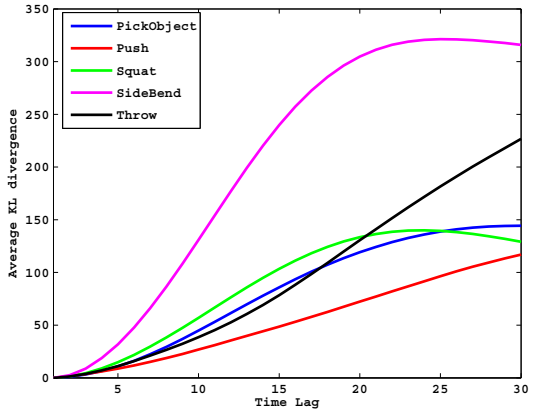

Figure 2. Illustration of how statistical properties change with time for 5 activities. The y-axis measures the KL divergence between ensemble statistics as a function of the time-lag. Figure best viewed in color.

be considered stationary stochastic processes. Indeed, in this paper, we consider human actions as quasi-stationary processes. To model such quasi-stationary processes, we notice that the plot in figure 2 reveals that we can assume local stationarity, since for small values of δ the statistical properties do not change significantly. Thus, it would suffice to fit locally time-invariant models, but allow the parameters of the model to vary with time. This observation forms the basis for the current work. Note that this approach is widely used in the speech processing community where speech signals are considered short-term stationary in windows of 20-40 milliseconds [6].

Related Work: Modeling of human actions is a well-studied problem and many methods have been proposed for modeling them. Some of the most popular models for atomic action classes are state-space models such as Hidden Markov models [27, 20] and Linear Dynamical systems [18, 9]. The underlying assumption of these models is that the dynamical process is stationary. Time-invariant models such as these are sufficient to characterize simple action classes such as bending, walking, running, pushing etc. To model more complex dynamics, several researchers have proposed using switching dynamical models [14, 5, 12, 11]. In these approaches a complex activity is broken down into simpler motion patterns and each motion pattern is modeled using a simple dynamical model such as an HMM or an LDS. The overall activity is then modeled by switching among a small set of dynamical systems. However, it has been observed in the analysis of sign-language that hand-gestures are influenced by adjacent gestures, which makes segmentation of primitive actions [25] difficult. We consider human actions as a continuum of dynamical processes, where the parameters change continuously over time as opposed to discrete jumps in time. We represent the LDS at each time-instant as a point on the Grassmann manifold. Then, the overall activity is considered as a trajectory on the Grassmann manifold. Time-varying linear dynamical processes have also been studied in the control literature where they are traditionally used as approximations to non-linear processes [8]. Modeling of time-invariant dynamical systems as points on the Grassmann manifold was considered by [22]. Tracking points on the Grassmann manifold by a Hidden Markov Model on the manifold was proposed by [19] in array-signal processing applications, where a constant velocity model is assumed on the manifold. In contrast to the generative approaches discussed above, there exist discriminative approaches for modeling human actions. An in-depth discussion of discriminative models is beyond the scope of this paper, and we refer the reader to [17, 10] and references therein.

Contributions: Our contributions in this paper are twofold. Firstly, we introduce a time-varying LDS (TV-LDS) model to describe complex activities. Then, we describe the TV-LDS model as a trajectory on the space of LDS models. Secondly, under local stationarity assumptions, we pose the learning and classification problems as trajectory modeling on the Grassmann manifold and derive methods consistent with the geometry of the Grassmann manifold for this task.

Organization of the paper: In section 2 we introduce the time-varying linear dynamic system for complex activities. Then in section 3 we describe how the model can be interpreted as a trajectory on the Grassmann manifold. In section 4, we review representation of points, distance metrics and statistical methods on the Grassmann manifold. Then, in section 5 we show how trajectories on the Grassmann manifold can be modeled. In section 6, we show experimental results and present conclusions in section 7.

2. Modeling of Complex Activities

An activity is considered as a complex evolution of poses which is governed by an underlying dynamic process. The underlying process is potentially highly non-

linear and time-varying. We model complex activities as outputs of a time-varying linear dynamical process. At each time-instant, we assume that the dynamical process is linear. We then allow the parameters of the LDS to vary at each time-instant. Let $f(t) \in \mathbb{R}^m$ denote the observations (flow/silhouette etc) at time-instant t. Then, the time-varying dynamical model is represented as

$$f(t) = C(t)z(t) + w(t), w(t) \sim N(0, R(t)) \quad (1)$$

$$z(t+1) = A(t)z(t) + v(t), v(t) \sim N(0, Q(t)) \quad (2)$$

where, $z(t) \in \mathbb{R}^d$ is the hidden state vector of dimension d, $A(t)$ is the time-varying transition matrix and $C(t)$ is the time-varying measurement matrix. $w(t)$ and $v(t)$ are noise components modeled as normal with 0 mean and covariance $R(t)$ and $Q(t)$ respectively. When the model parameters A, C, Q, R are constant, the model reduces to the well-known time-invariant LDS which has been successfully applied in several vision tasks [16, 18]. In summary, the model consists of a sequence of parameters: the measurement matrix $C(t)$ and the transition matrix $A(t)$ and the noise covariances $R(t), Q(t)$. Before we discuss the problem of parameter estimation, we show the strength of the model on the synthesis experiment described in section 1. The results of synthesis using a continuous time-varying model are shown in figure 1(c). It can be seen that the synthesized sequence exhibits a much more realistic evolution of poses.

2.1. Estimating the parameters

We first present a brief review of the parameter estimation problem for the time-invariant case before turning to the time-varying case.

The time-invariant case: Consider the time-invariant version of the model in equations (1) and (2).

$$f(t) = Cz(t) + w(t), w(t) \sim N(0, R) \quad (3)$$

$$z(t+1) = Az(t) + v(t), v(t) \sim N(0, Q) \quad (4)$$

For the time-invariant case, it is easily shown that there are infinitely many choices of parameters that give rise to the same sample path $f(t)$. Resolving this ambiguity requires one to impose further constraints and choose a canonical model. The conditions as proposed in [18] are that $m >> d, rank(C) = d$ and $C^TC = I$. The number of unknowns that need to be solved for are: $md - \frac{d(d+1)}{2}$ for C, d^2 for A, $\frac{d(d+1)}{2}$ for Q: resulting in $md + d^2$ unknowns (we have ignored the observation noise covariance as of now). For each observed frame we get m equations. Hence, $d+1$ linearly independent observations are sufficient to solve for the required parameters ($m(d+1) > md + d^2$ since $m >> d$).

The parameter estimates can be obtained in closed form using prediction error methods. Several estimation algorithms exist such as the ones described in [13] and [18]. We use the solution derived in [18] here. Let observations $f(1), f(2), \ldots f(\tau)$, represent the features for the frames $1, 2, ...\tau$. Let $[f(1), f(2), \ldots f(\tau)] = U\Sigma V^T$ be the singular value decomposition of the data. Then $\hat{C} = U, \hat{A} = \Sigma V^T D_1 V (V^T D_2 V)^{-1} \Sigma^{-1}$, where D_1 = [0 0;$I_{\tau-1}$ 0] and D_2 = [$I_{\tau-1}$ 0;0 0].

The time-varying case: Estimation of time-varying models for time-series have been studied in various domains such as speech processing, econometric data and communication channels. A commonly used assumption in these domains is that the time-varying AR (auto-regressive) and ARMA (auto-regressive moving average) parameters can be expressed as linear combinations of known deterministic functions of time such as the Fourier basis or the exponential basis [6]. Other approaches include Taylor-series expansions of the model parameters such as in [15] for econometric applications. Estimation of time-varying single-input single-output (SISO) AR models have been proposed by estimating an equivalent time-invariant single-input multiple-output (SIMO) process [21], and was applied for channel estimation in communication networks. These approaches are restricted to single-dimensional time-series data. Multi-dimensional time-varying dynamical models traditionally arise as a result of linearizing a non-linear dynamical system. In such cases, the time-varying parameters can be solved for analytically using Taylor series expansions around a 'nominal trajectory' [8]. However, in most practical applications including activity modeling, one does not know what the underlying non-linear equations are nor does one have the knowledge of a nominal trajectory. Recently, linear parameter varying (LPV) systems have been proposed to model time-varying processes. In these approaches, the time-varying model parameters are considered to be linear combinations of a small set of time-invariant parameters. The linear combination weights, also called the scheduling weights, change with time [7, 24]. However, identification of LPV systems is computationally very expensive [24]. In the following, we propose a computationally efficient and conceptually simple method to estimate the time-varying parameters of a dynamical system without making strong assumptions on the nature of the time-varying process.

To begin with, it is easily seen that even in the time-varying case there are infinitely many choices of the model parameters that can give rise to the same sample path $f(t)$. So, we impose the same set of conditions as in the time-invariant case i.e. $m >> d, rank(C(t)) = d$ and $C(t)^TC(t) = I$. Based on the analysis given above, there are $md + d^2$ unknowns for *each* time-instant and m equations per time-instant. Obviously this is an ill-posed problem since there are far more unknowns than there are equations. Hence, we impose another condition that the model parameters stay constant in local temporal neighborhoods.

The temporal neighborhood in which the parameters are assumed to stay constant should also ensure that $d+1$ linearly independent observations can be obtained within the neighborhood. In general, it cannot be guaranteed that a fixed $d+1$ sized neighborhood will satisfy this condition. However, in our experience we found that a neighborhood of size $1.5d - 2d$ was sufficient to meet this condition in most real-world human activities. Typically, d is of the order of $5-10$ and complex human activities extend to several hundred frames. It is reasonable to assume that in short windows of about $15-20$ frames the dynamics can be easily modeled by simple time-invariant dynamical processes.

We now have a sequence of dynamical systems which defines a trajectory on the space of LDS. Before we discuss how we model this trajectory, we first discuss the Grassmann manifold formulation of the LDS space.

3. Trajectories on the Model Space

For the time-invariant case, starting from an initial condition $z(0)$, it can be shown that the *expected* observation sequence is given by

$$E\begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ . \\ . \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ . \\ . \end{bmatrix} z(0) = O_\infty(M)z(0) \qquad (5)$$

Thus, the expected observation sequence generated by a time-invariant model $M = (A, C)$ lies in the column space S of the extended *observability* matrix given by $O_\infty(M) = [C^T, (CA)^T, (CA^2)^T, ...]^T$. In the time-varying case, we assumed that the model parameters stay constant in short temporal neighborhoods. Let the size of the temporal window be n. Thus, the n-length expected observation sequence generated by the model $M_t = (C_t, A_t)$ (model at time t) lies in the column space S_t of the *finite* observability matrix given by

$$O_n(M_t) = \left[\ C_t; C_tA_t; \ldots; C_tA_t^{n-1}\ \right] \qquad (6)$$

Thus, the time-varying model can be viewed as a sequence of subspaces S_t, where each subspace is spanned by the columns of the observability matrix at the corresponding time instant. Finite dimensional subspaces such as these can be identified as points on the Grassmann manifold. Thus, the sequence of subspaces can be mathematically expressed as a trajectory on the Grassmann manifold. We first provide the definition of the Grassmann manifold.

The Grassmann Manifold $G_{k,m-k}$ [2]: The Grassmann manifold $G_{k,m-k}$ is the space whose points are *k-planes* or *k*-dimensional hyperplanes (containing the origin) in $\mathbb{R}^m$. In the following, we review in brief appropriate distance metrics and statistical modeling methods on the Grassmann manifold and show how *trajectories* on the Grassmannian can be used for classification.

4. Distances and Statistics

To model and compare trajectories on the Grassmann manifold, we need to understand a) the representation of points, b) distance metrics and c) statistical models on the manifold. In this section, we provide a brief overview of each of these aspects. The Grassmann manifold $G_{m,k}$ is the space whose points are *k-planes* or *k*-dimensional hyperplanes (containing the origin) in $\mathbb{R}^m$. To each k-plane ν in $\mathbb{R}^m$, we can associate an $m \times k$ orthonormal matrix Y such that the columns of Y form an orthonormal basis for the plane. Note that there exist several choices for the basis Y. Thus, all the choices of basis vectors that span the same subspace need to be considered equivalent. To each k-plane ν in $G_{m,k}$ is associated an equivalence class of $m \times k$ matrices YR in $\mathbb{R}^{m\times k}$, for non-singular R, where Y is an orthonormal basis for the k-plane. This is also called the Procrustes representation. Alternately, one can define a unique projection matrix for the subspace given by $P = YY^T$ which projects points from the ambient Euclidean space onto the given subspace. In applications to human activities, the projection matrix representations leads to large computational overheads since it is a square $m \times m$ matrix. In practice, m is of the order of 10^3 or higher. Thus, we rely on the Procrustes representation of points which relies on storing only tall-thin $m \times k$ matrices.

Distance Metrics: The Grassmann manifold is endowed with a Riemannian structure that lends itself to computation of distances between points on the manifold via geodesics [4, 1]. Instead of geodesic computations, we adopt the Procrustes distance metric proposed in [2]. This choice results in efficient computation of the distance metrics and the class conditional probability density estimators. To each subspace, associate orthonormal bases Y_1 and Y_2. The squared Procrustes distance between the two subspaces is the smallest squared Euclidean distance between any pair of matrices in the corresponding equivalence classes. Hence,

$$d_p^2(S_1, S_2) = \min_R tr(Y_1 - Y_2R)^T(Y_1 - Y_2R) \qquad (7)$$

A closed-form solution is available for this minimization problem [2]. For the case where R varies over the space $R_{k,k}$ of all full rank $k \times k$ matrices, the distance is given by $d_p^2(S_1, S_2) = tr(I_k - A^TA)$, where $A = Y_1^TY_2$. The computational complexity involved in the above distance computation is $O(mk^2)$.

Statistical Modeling: There exist parametric probability densities such as the matrix Langevin and the matrix Bingham distributions [2] on the Grassmann manifold, which rely on the projection matrix representation of points. As discussed before, projection matrices are square $m \times m$ matrices where m can be quite large in practice. Hence, the parametric density form is computationally expensive. Alternately, given several examples from a class on the man-

ifold, the class conditional density can be estimated using Parzen windows. Using the Procrustes representation and distance metric, the density estimate is given by [2] as

$$\hat{f}(Y;M) = \frac{1}{n}C(M)\sum_{i=1}^{n} K[M^{-1/2}(I_k - Y_i^T Y Y^T Y_i)M^{-1/2}] \quad (8)$$

where $K(T)$ is the kernel function, M is a $k \times k$ positive definite matrix which plays the role of the kernel width or a smoothing parameter. $C(M)$ is a normalizing factor chosen so that the estimated density integrates to unity. The Y_is are orthonormal bases for the points on the Grassmann manifold. The matrix valued kernel function $K(T)$ is chosen to be $K(T) = exp(-tr(T))$ in this paper. The computational complexity involved in the above is $O(nmk^2)$.

5. Comparing sequences of Subspaces

Given a video of a long activity, first the time-varying model parameters $M_t = (A_t, C_t)$ are estimated using small temporal sliding-windows and the method described in section 2.1. Subsequently, for each window the observability matrix $O_n(M_t)$ is computed. Then for each observability matrix, an orthonormal basis is computed using standard SVD based algorithms. So, we now have a sequence of subspaces, or in other words a trajectory on the Grassmann manifold. To compare two subspace trajectories we propose two approaches.

Dynamic time warping: Dynamic time-warping (DTW) only requires an appropriate distance metric between points on the manifold. Given two complex activities and their corresponding subspace sequences $S_1(t)$ and $S_2(t)$, DTW tries to find a warping path $a(t)$ such that $S_1(t) = S_2(a(t))$. To solve the problem we can use any standard DTW algorithm.

Grassmann switching model: In the second approach, we parametrize the trajectory using a switching model akin to the HMM on the Grassmann manifold. Corresponding to an activity class C, suppose we are given M subspace sequences $\{S_i^C(t)\}_{i=1}^M$. We consider the dynamics to be described by a set of K hidden states $L^{(1)}, \ldots L^{(K)}$. The state at time t is denoted by $Q(t)$ and the observation at time t is denoted by $S(t)$. The overall model for the activity consists of the K hidden states, the intra-cluster pdfs $f(S(t)|Q(t) = L^{(i)})$, the transition probability matrix and the prior probability. In general, the Baum-Welch algorithm provides solutions for the above problems in a maximum likelihood sense. This requires one to have analytical expressions for the intra-cluster pdfs and the gradient of the likelihood of a sequence in terms of these parameters. In our case, we solve these problems in a much simpler, although sub-optimal way as follows. Given a sequence of subspaces $\{S_i^C(t)\}_{i=1}^M$, the following procedure is adopted to estimate the switching model.

1. Cluster the points into K clusters or hidden-states $L^{(1)}, \ldots L^{(K)}$.

2. Estimate a pdf within each cluster $f(S(t)|Q(t) = L^{(i)})$.

3. Estimate the transition probabilities $p(Q(t) = L^{(i)}|Q(t-1) = L^{(j)})$ between the clusters.

4. Estimate the prior probability $p(Q(0))$. Any of the distance metrics on the Grassmann manifold can be used to perform clustering. In our experiments, we used a spectral clustering algorithm – Normalized cuts – to get the clusters. Within each cluster, we use the non-parametric density estimate as described in section 4 to estimate the intra-cluster pdf. Once the clusters are found, we form the sequence of cluster labels corresponding to the sequence of subspaces. The sequence of labels is used to estimate the transition probabilities by bi-gram counts. Thus, we have now learnt a switching model on the Grassmann manifold for each activity class.

Given a new subspace sequence, we need a method to classify it into one of the action classes. In the case of standard HMMs, this problem is solved by the forward-backward algorithm and its variants. We use a simpler version that works much faster and using fewer computations. Given a sequence $S(t)$ and an activity model, we first assign each $S(t)$ into one of the clusters of the model. Let us denote by $Q(t)$ the sequence of cluster labels thus obtained. Then we compute the likelihood of the sequence as $p(Q(0))\prod_k f(S(k)|Q(k))p(Q(k)|Q(k-1))$. Though this is sub-optimal than the forward-backward algorithm, we found that we obtain significant computational advantages using these approximations.

Relation to Switching Linear Dynamical Systems: Switching linear dynamical models (SLDS) [14, 5, 12, 11] model a complex activity by breaking it down into simpler motion patterns where each motion pattern is modeled using a simple model such as an HMM or an LDS. The overall activity is then modeled by switching amongst a small set of dynamical systems. In the above Grassmann switching model, if we constrain the intra-cluster pdf to be $f(S(t)|Q(t) = L^{(k)}) = \delta(S(t) - \mu_k)$, where μ_k is the cluster center, then the Grassmann switching model reduces to the SLDS model. Thus, the SLDS model is a special case of the proposed Grassmann switching model. Further, in SLDS it is usually assumed that complex human actions can be separated into simpler motion patterns. However, we do not rely on segmentation of activities into primitive actions and thus our approach is applicable even in complex cases when segmentation is difficult.

6. Experiments

In this section we present experiments demonstrating the strength of the model for summarizing and recognizing complex activities. In the first experiment we show the results of summarizing a long video containing a complex activity – the game of Blackjack. For this, we used the dataset reported in [28]. The game of Blackjack con-

sists of a few elements such as dealing cards, waiting for bids, shuffling the cards etc. We try to estimate a Grassmann switching model for the entire video of Blackjack. The Grassmann switching model would then represent a 'summary' of the game, where the clusters of the model represent various elements of the game and the switching structure represents how the game progresses. This video consists of about 1700 frames. We extracted the motion-histogram features as proposed in [28] for each frame of the video. The time-varying model parameters are estimated in sliding windows of size 10. The dimension of the state vector is chosen to be $d = 5$. To estimate the Grassmann switching model for the game of Blackjack, we manually set the number of clusters to 5. In figure 3, we show an embedding of the video obtained from the model parameters using Laplacian eigenmaps. Each point corresponds to a time-invariant model parameter (A, C) pair or equivalently a point on the Grassmann manifold. Each cluster was found to correspond dominantly to a distinct element of the game as shown. The switching structure between the clusters is encoded in the transition matrix and is shown in figure 4. As can be seen the switching structure corresponds to a normal game of Blackjack. Since this is a data-driven procedure, it should be noted that the switching structure will not necessarily be the same for every individual Blackjack game. However, given two distinct Blackjack games we can now quantify the notion of how similarly the two games proceeded.

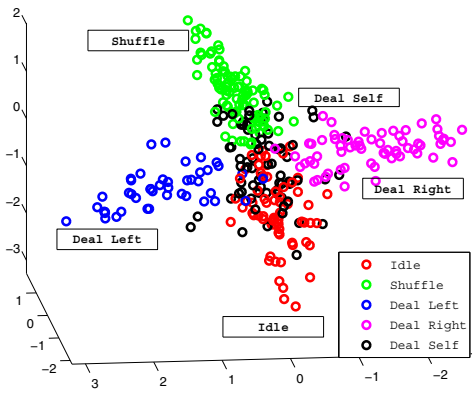

Figure 3. An embedding of the entire Blackjack video sequence. Figure best viewed in color.

In the next experiment, we took the common activities dataset described in [23] consisting of 10 simple actions – {Pick Object, Jog, Push, Squat, Wave, Kick, Side Bend, Throw, Turn around, Talk on cellphone}. Each action is performed 10 times each by the same actor under two different viewing angles separated by about 20°. We create more complex actions from this set. We divided the actions into two groups - the first group contains the first 5 actions, the second group contains the next 5 actions. Then, we created compound actions by taking one action

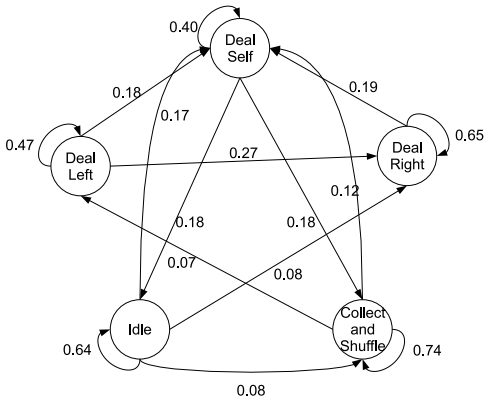

Figure 4. Estimated structure of the game of Blackjack. (For the sake of clarity arcs with low weights have not been shown).

from the first group and an action from the second group. Then, we swapped the two constituent actions. This causes the two resulting compound actions to share similar global second-order statistics (the mean and covariance). Thus, we have 10 compound actions as shown in table 1. To test the framework, we performed a leave-one-out testing where we trained on 9 executions and tested on the remaining execution. Both views were used in training as well as testing. Since the global second order-statistics of activities such as PickObject-Kick and Kick-PickObject etc are similar, time-invariant linear dynamic systems are expected to show confusion between them. The results of the recognition experiment are shown in table 1. As is evident, both the DTW based and the Switching model show 100% recognition since they account for the time-varying dynamics of the compound actions.

Activity Type	LDS	Grass. DTW	Grass. Switching model
PickObject - Kick	100	100	100
Kick - PickObject	50	100	100
Jog - SideBend	100	100	100
SideBend - Jog	50	100	100
Push - Throw	0	100	100
Throw - Push	100	100	100
Squat - TurnAround	100	100	100
TurnAround - Squat	0	100	100
Wave - TalkCellphone	50	100	100
TalkCellphone - Wave	50	100	100
Average	60%	100%	100%

Table 1. Recognition percentages on Compound actions

In the next experiment we performed a synthesis experiment on a skating dataset obtained from [26]. From a segment of video of about a 100 frames that contained fast skating actions as shown in figure 5 (a), a discrete-switching model and a time-varying model were estimated. The actions in the sequence exhibit co-articulation effects, where transitions between distinct poses contain intermediate poses that share the appearance of both the starting and the ending pose. The results of synthesis using the models are shown in figure 5. The experiment shows that the time-

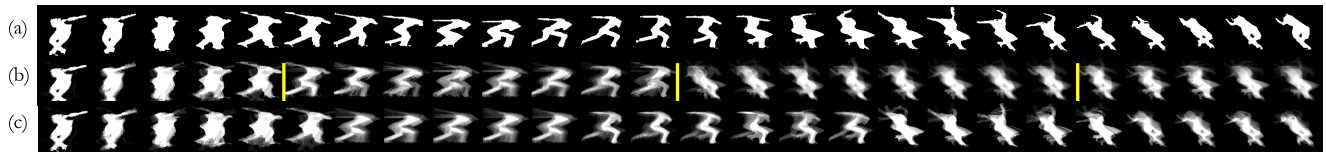

Figure 5. (a) Original skating sequence taken from [26], (b) Synthesis by a sequence of linear dynamic models with boundaries shown by vertical yellow lines, (c) Synthesis by a continuous time-varying model. It can be seen that synthesis results show abrupt changes in pose across boundaries whereas the time-varying model results in a smoother evolution of poses.

varying model can account for such co-articulatory effects and produce realistic looking sequences.

7. Discussion and Conclusion

We have presented a method to model time-varying dynamics for long videos of human activities. We assumed that the process is locally stationary and showed methods to learn the parameters under this assumption. Then, we discussed how the time-varying model can be viewed as a sequence of subspaces spanned by the columns of the time-varying observability matrix which was then interpreted as a trajectory on the Grassmann manifold. By exploiting the geometry of the Grassmann manifold, we proposed non-parametric (DTW) and parametric (switching model) methods to compare trajectories. Finally, we showed how the model is well-suited for summarizing long activity videos and recognizing complex actions.

References

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

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

[3] T. Claasen and W. Mecklenbrauker. On stationary linear time-varying systems. *IEEE Trans. on Circuits and Systems*, 29(3):169–184, 1982.

[4] A. Edelman, T. A. Arias, and S. T. Smith. The Geometry of Algorithms with Orthogonality Constraints. *SIAM Journal Matrix Analysis and Application*, 20(2):303–353, 1999.

[5] X. Feng and P. Perona. Human action recognition by sequence of movelet codewords. *3DPVT*, pages 717–721, 2002.

[6] M. Hall, A. V. Oppenheim, and A. Willsky. Time-varying parametric modeling of speech. *IEEE Conference on Decision and Control*, pages 1085–1091, 1977.

[7] L. H. Lee. Identification and Robust Control of Linear Parameter-Varying Systems. *PhD thesis, University of California at Berkeley, Berkeley, California*, 1997.

[8] L. Ljung. *System Identification Theory For the User*. PTR Prentice Hall, Upper Saddle River, N.J., 1999.

[9] M. C. Mazzaro, M. Sznaier, and O. Camps. A model (in)validation approach to gait classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(11):1820–1825, 2005.

[10] L. P. Morency, A. Quattoni, and T. Darrell. Latent-Dynamic Discriminative Models for Continuous Gesture Recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[11] B. North, A. Blake, M. Isard, and J. Rittscher. Learning and classification of complex dynamics. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(9):1016–1034, 2000.

[12] S. M. Oh, J. M. Rehg, T. Balch, and F. Dellaert. Learning and Inferring Motion Patterns using Parametric Segmental Switching Linear Dynamic Systems. *International Journal of Computer Vision*, 77(1-3):103–124, 2006.

[13] P. V. Overschee and B. D. Moor. Subspace algorithms for the stochastic identification problem. *Automatica*, 29(3):649–660, 1993.

[14] V. Pavlovic and J. M. Rehg. Impact of dynamic model learning on classification of human motion. *IEEE Conference on Computer Vision and Pattern Recognition*, 1:788–795, 2000.

[15] T. S. Rao. The fitting of nonstationary time-series models with time-dependent parameters. *Journal of the Royal Statistical Society B*, 32(2):312–322, 1970.

[16] P. Saisan, G. Doretto, Y. Wu, and S. Soatto. Dynamic texture recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[17] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Conditional Random Fields for Contextual Human Motion Recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 2:1808–1815, 2005.

[18] S. Soatto, G. Doretto, and Y. N. Wu. Dynamic textures. *IEEE International Conference on Computer Vision*, pages 439–446, 2001.

[19] A. Srivasatava and E. Klassen. Bayesian geometric subspace tracking. *Advances in Applied Probability*, 36(1):43–56, March 2004.

[20] T. Starner and A. Pentland. Real-time american sign language recognition from video using hidden markov models. *IEEE International Symposium on Computer Vision*, pages 265–270, 1995.

[21] M. Tsatsanis and G. Giannakis. Subspace methods for blind estimation of time-varying fir channels. *IEEE Trans. on Signal Processing*, 45(12):3084–3093, 1997.

[22] P. Turaga, A. Veeraraghavan, and R. Chellappa. Statistical Analysis on Stiefel and Grassmann Manifolds with Applications in Computer Vision. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[23] A. Veeraraghavan, R. Chellappa, and A. K. Roy-Chowdhury. The function space of an activity. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 959–968, 2006.

[24] V. Verdult and M. Verhaegen. Subspace identification of multivariable linear parameter-varying systems. *Automatica*, 38(5):805–814, 2002.

[25] C. Vogler and D. Metaxas. ASL Recognition Based on a Coupling Between HMMs and 3D Motion Analysis. *IEEE International Conference on Computer Vision*, pages 363–369, 1998.

[26] Y. Wang, H. Jiang, M. S. Drew, Z.-N. Li, and G. Mori. Unsupervised Discovery of Action Classes. *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[27] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time sequential images using hidden markov model. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 379–385, 1992.

[28] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 819–826, 2004.