Low Order Dynamics Embedding for High Dimensional Time Series

Fei Xiong, Octavia I. Camps, and Mario Sznaier*
Dept. of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115
http://robustsystems.ece.neu.edu

Abstract

This paper considers the problem of finding low order nonlinear embeddings of dynamic data, that is, data characterized by a temporal ordering. Examples where this problem arises include, among others, appearance–based multi–frame tracking, activity recognition from video and dynamic texture analysis/synthesis. Our main result is a semi-definite programming based manifold embedding algorithm that exploits the dynamical information encapsulated in the temporal ordering of the sequence to obtain embeddings with lower complexity that those produced by existing algorithms, at a comparable computational cost. In addition, the use of spatio-temporal information allows for minimizing the effects of outliers on the manifold structure and for handling fragmented sequences, where some of the data is missing, for instance due to occlusion.

1. Introduction

The problem of finding low complexity representations of high-dimensionality data is ubiquitous in computer vision and pattern recognition. Applications include, among others, appearance-based tracking [3, 8], activity recognition from video data [4], gait recognition[1], video and image inpainting [15], and dynamic texture analysis and synthesis [10]. In all cases, the key to obtain computationally tractable algorithms is to exploit the high degree of correlation of the original data to obtain low dimensional representations, which are then used as surrogates to solve the problem of interest. For example, in the context of multi-frame tracking of a target undergoing appearance changes, nonlinear manifold embedding allows for representing the evolution of all the pixels in the target as a trajectory in a low dimensional manifold, characterized by the evolution of just a few variables. Thus, predicting future target appearance entails predicting the values of these variables, followed by a nonlinear operation, rather than propagating the values of the (typically thousands) pixels involved.

Given its importance both for computer vision and machine learning, nonlinear manifold embedding has been a very active research area in the past decade, leading to a number of algorithms, including Locally Linear Embeddings (LLE) [13], semi-definite embedding (SDE)[16], Global coordination of local linear models (GCM) [12] and Dynamic Global Coordinate Model (DGCM) [9]. A common feature of these methods is that, while they exploit the local spatial topology of the data, they stop short of taking full advantage of the available dynamical information, encapsulated in its temporal ordering. Indeed, this information is typically discarded, for instance in the SDE method, or used in relatively simple models [9], that may fail to capture the underlying temporal dynamics. Thus, the resulting representations are not necessarily the most compact ones. In addition, neglecting dynamical information can lead to embeddings that are fragile in the presence of outliers or missing data. This phenomenon is illustrated in Fig. 1, obtained by applying the method in [16] to a boxing sequence from the KTH database [14]. As shown there, a single outlier (in this case due to camera jitter) can significantly alter the structure of the manifold.

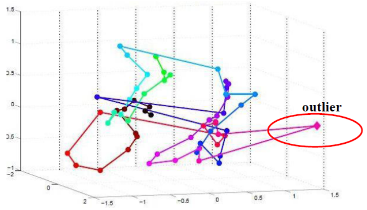


Figure 1: Effect of a single outlier on the manifold structure.

In this paper we propose a new manifold embedding algorithm for dynamical sequences that exploits both spatial and temporal information in order to obtain the simplest possible dynamical representation of the data. Briefly, the main idea is to model the data as the output of an underlying Wiener system of the form shown in Figure 2 consisting of the cascade of a piece-wise linear autoregressive model

*This work was supported in part by NSF grants IIS–0713003 and ECCS–0901433, AFOSR grant FA9550–09–1–0253, and the Alert DHS Center of Excellence under Award Number 2008-ST-061-ED0001.

(generating the manifold trajectories) and a possibly time-varying nonlinearity that maps the low dimensional manifold representation back to the original data space. As illustrated in the paper, constraining target manifolds to those spanned by feasible trajectories of the Wiener system endows the proposed method with robustness against outliers and allows for handling missing data. Further, in this context the complexity of the representation is given by a combination of the order (e.g the number of coefficients) of the regressor and the (local) dimension of the embedding manifold. By appealing to concepts from realization theory, the problem of minimizing this complexity can be reduced to a rank minimization form and efficiently solved used recently proposed convex relaxations [6], leading to an algorithm whose computational complexity is similar to that of the SDE method.

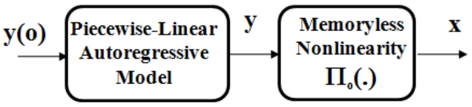


Figure 2: Wiener Model. Here $\mathbf{y} \in R^d$ and $\mathbf{x} \in R^D$ with $D >> d$. $\mathbf{y}(0)$ denotes the initial conditions of the autoregressive process.

The paper is organized as follows. Section 2 presents some background material, including a summary of the SDE algorithm and key results from realization theory and convex relaxations of rank required to develop the proposed embedding method, which is presented in detail in section 3. Section 4 presents experimental results and benchmarking using synthetic and real video data. Finally, section 5 summarizes our results and points out to directions for future research.

2. Notation and Background Results

In this section we present the notation used throughout the paper and summarize, for easy of reference, some key results from the literature required to develop the proposed embedding method.

2.1. Notation

$\mathbf{x}$	column vector or matrix valued variables.
$\mathbf{M} \succeq 0$	matrix $\mathbf{M}$ is positive semidefinite.

2.2. Semidefinite Embedding (SDE)

The dynamic embedding method that we propose in this paper is inspired by the Semidefinite Embedding method (SDE) [16] briefly described next. The main idea of this method is to find a possibly nonlinear mapping that preserves the local geometry while attempting to minimize the (local) dimension of the manifold. Specifically, given N points $\mathbf{X} \doteq \begin{bmatrix}\mathbf{x}_1 & \dots \mathbf{x}_N\end{bmatrix}$, define a neighborhood relationship matrix $\eta \in R^{N\times N}$ with entries $\eta_{ij} \in \{0,1\}$, where $\eta_{ij} = 1$ if and only if $\mathbf{x}_j$ is regarded as the neighbor of $\mathbf{x}_i$. Finally, denote by $\mathbf{Y} \doteq \begin{bmatrix}\mathbf{y}_1 & \dots \mathbf{y}_N\end{bmatrix}$ the low dimensional projections of the high dimensional data. In this context, the local geometry is preserved if, for every point $\mathbf{x}_i$, there exists a rotation, reflection and/or translation that maps $\mathbf{x_i}$ and its neighbors precisely onto $\mathbf{y}_i$ and its neighbors. Equivalently, for every pair (i,j) such that $\eta_{ij} = 1$, the following equation must hold

$$K_{ii} - 2K_{ij} + K_{jj} = L_{ii} - 2L_{ij} + L_{jj} \qquad (1)$$

where K_{ij} and L_{ij} denote the entries of the Grammian matrices associated with the data and its projections, e.g. $\mathbf{K} \doteq \mathbf{Y}^T\mathbf{Y}$ and $\mathbf{L} \doteq \mathbf{X}^T\mathbf{X}$. As pointed out in [16], maximizing the trace of $\mathbf{K}$ tends to minimize the dimension of the manifold where $\mathbf{Y}$ is embedded[1]. These observations allow for recasting the embedding problem into the following semi-definite optimization form:

$$\text{maximize} \quad \text{trace}(\mathbf{K})$$

subject to

1. $\mathbf{K} \succeq 0$.
2. $\sum_{i,j} K_{ij} = 0$.
3. $K_{ii} - 2K_{ij} + K_{jj} = L_{ii} - 2L_{ij} + L_{jj}$ for all (i,j) such that $\eta_{ij} = 1$ or $[\eta^T\eta]_{ij} = 1$

Once a solution $\mathbf{K}$ is found, the low dimensional projections can be recovered from its singular value decomposition: $\mathbf{Y} = \mathbf{D}^{\frac{1}{2}}\mathbf{V}^T = (\mathbf{U}\mathbf{D}^{\frac{1}{2}})^T$, where $\mathbf{K} = \mathbf{U}\mathbf{D}\mathbf{V}^T$. Note that, contrary to other embedding methods, the dimension of $\mathbf{Y}$ does not have to be specified a-priori. Rather, the SDE method detects the correct underlying dimensionality of the inputs from the singular value spectrum of $\mathbf{K}$.

2.3. Regressor Complexity and the Hankel Matrix

Consider a vector process described by an autoregressive model of the form:

$$\mathbf{y}_k = a_1\mathbf{y}_{k-1} + a_2\mathbf{y}_{k-2} + \ldots + a_n\mathbf{y}_{k-n} \qquad (2)$$

To every trajectory $\mathbf{y}_k$ of this model, one can associate a Hankel matrix defined as:

$$\mathbf{H}_\mathbf{y} \doteq \begin{bmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \cdots & \mathbf{y}_r \\ \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_{r+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_s & \mathbf{y}_{s+1} & \cdots & \mathbf{y}_{r+s} \end{bmatrix} \qquad (3)$$

[1] Intuitively, maximizing trace($\mathbf{K}$) tends to spread out the projections $\mathbf{y}_i$ as far as possible, thus flattening the manifold.

It is a well known result from realization theory [7, 11], that, under mild conditions, given a sequence of measurements $\{\mathbf{y}_k\}$ generated by (2), the order n of the model is related to the rank of the corresponding Hankel matrix by, $\text{rank}(\mathbf{H}_\mathbf{y}) = n$, provided that $r, s \geq n$.

2.4. Convex Relaxations of Rank Minimization Problems (RMP)

From section 2.3, it follows that minimal complexity models can be obtained by minimizing the rank of a matrix that depends linearly on the optimization variables. This observation plays a key role in obtaining low dynamic complexity embeddings. However, a potential difficulty here is that rank minimization problems are generically NP-hard. To circumvent this difficulty, we will exploit a class of recently proposed convex relaxations based on the fact that trace is the convex envelope of rank, that is, its tightest convex approximation [5]. This property allows for (approximately) solving rank minimization problems affine in the optimization variable by solving a semi-definite (hence convex) program. Specifically, let $\mathbf{M}(\mathbf{y})$ denote a matrix that depends affinely on the vector variable $\mathbf{y}$. Then, the problem $\min_\mathbf{y} \text{rank}\mathbf{M}(\mathbf{y})$ can be relaxed to:

$$\begin{array}{ll} \text{minimize}_{\mathbf{y},\mathbf{P},\mathbf{Q}} & \text{trace}(\mathbf{P}) + \text{trace}(\mathbf{Q}) \\ \text{subject to} & \begin{bmatrix} \mathbf{P} & \mathbf{M}(\mathbf{y}) \\ \mathbf{M}(\mathbf{y})^T & \mathbf{Q} \end{bmatrix} \succeq 0 \end{array} \tag{4}$$

where $\mathbf{P}$,$\mathbf{Q}$ are matrix slack variables. A further improvement on the solution to this problem can be obtained by using a method based on a re-weighted heuristic that seeks to iteratively solve

$$\begin{array}{ll} \text{minimize} & \text{logdet}(\mathbf{M} + \delta\mathbf{I}) \\ \text{subject to} & \mathbf{M} \succeq 0 \end{array}$$

where the function $\text{logdet}(\mathbf{M} + \delta\mathbf{I})$ is used as a smooth surrogate for $\text{rank}(\mathbf{M})$ and $\delta > 0$ is a regularization constant [6]. Considering the first order Taylor expansion of $\text{logdet}(\mathbf{M} + \delta\mathbf{I})$ around $\mathbf{M}_k$, the present best estimate for the solution leads to the following iterative algorithm:

$$\begin{array}{l} \min_{\mathbf{y},\mathbf{P}_{k+1},\mathbf{Q}_{k+1}} (\text{trace}((\mathbf{P}_k + \delta\mathbf{I})^{-1}\mathbf{P}_{k+1}) + \\ \text{trace}((\mathbf{Q}_k + \delta\mathbf{I})^{-1}\mathbf{Q}_{k+1})) \\ \quad \text{subject to} \quad \begin{bmatrix} \mathbf{P}_{k+1} & \mathbf{M}(\mathbf{y}) \\ \mathbf{M}(\mathbf{y})^T & \mathbf{Q}_{k+1} \end{bmatrix} \succeq 0 \end{array} \tag{5}$$

where $\mathbf{P}_k$, $\mathbf{Q}_k$ denote the solution obtained at the k^{th} iteration.

3. Low Order Dynamics Embedding (LDE)

The goal of this paper is to develop an algorithm specially tailored to dynamic data that minimizes the complexity of the resulting embeddings. As briefly outlined in the introduction, the key to accomplish this goal is to model the (temporally ordered) data as the output of an unknown Wiener system, and then search for embeddings that minimize the complexity of this description, measured in terms of both the dynamical order of the regressor and the dimension of the embedding manifold. From sections 2.2 and 2.3, it follows that, in principle, this can be accomplished by solving the following optimization problem:

$$\min_\mathbf{y} \text{ rank}(\mathbf{H}_y) - \lambda\text{trace}(\mathbf{K}) \tag{6}$$

$$\text{subject to:} \|y_i - y_j\|_2^2 = \|x_i - x_j\|_2^2 \tag{7}$$

$$\text{for all } \eta_{ij} = 1 \text{ or } [\eta^T\eta]_{ij} = 1$$

$$\mathbf{K} = \mathbf{Y}^T\mathbf{Y}, \ \mathbf{Y} \doteq \begin{bmatrix} \mathbf{y}_1 & \ldots \mathbf{y}_N \end{bmatrix} \tag{8}$$

Intuitively, the first term in the objective function seeks to minimize the complexity of the regressor, while the second, through the maximization of trace($\mathbf{K}$) seeks to minimize the dimension of the embedding manifolds. The constraints (7) impose that the embedding preserves the local spatial geometry. A difficulty here is that, due to the quadratic dependence on $\mathbf{y}_i$ of the objective function and the constraints, the overall problem is non-convex, even if using the rank relaxation discussed in section 2.4. In addition, restricting the embedding map to be a (local) isometry severely limits the degree of dimensionality reduction (intuitively, there are not enough degrees of freedom in the manifold to satisfy exactly all the high dimensional equality constraints). To circumvent these difficulties, in the sequel we will work with the inner products $k_{ij} \doteq \mathbf{y}_i^T\mathbf{y}_j$ and the matrix $\mathbf{G} \doteq \mathbf{H}_\mathbf{y}{}^T\mathbf{H}_\mathbf{y}$, rather than with $\mathbf{y}$ and $\mathbf{H}_\mathbf{y}$ (clearly $\mathbf{G}$ and $\mathbf{H}_\mathbf{y}$ have the same rank). In addition, rather than imposing an exact isometry, we will relax the constraint (7) to:

$$\left| \|y_i - y_j\|_2^2 - \|x_i - x_j\|_2^2 \right| \leq \epsilon \|x_i - x_j\|_2^2 \tag{9}$$

$$\text{for all } (i, j) \in \text{local neighborhood} \tag{10}$$

where ϵ is a design parameter. Defining the family of Gram matrices:

$$\mathbf{K}_{i,n} = \begin{bmatrix} \mathbf{y}_i^T\mathbf{y}_i & \mathbf{y}_i^T\mathbf{y}_{i+1} & \cdots & \mathbf{y}_i^T\mathbf{y}_{i+n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_{i+n}^T\mathbf{y}_i & \mathbf{y}_{i+n}^T\mathbf{y}_{i+1} & \cdots & \mathbf{y}_{i+n}^T\mathbf{y}_{i+n} \end{bmatrix} \tag{11}$$

and noting that $\mathbf{G} = \sum_{i=1}^n \mathbf{K}_{i,n+i-1}$ leads to the following problem (affine in k_{ij}, the elements of $\mathbf{K} \doteq \mathbf{K}_{1,2n}$):

$$\min_{k_{ij}} \{\text{rank}(\mathbf{G}) - \lambda\text{trace}(\mathbf{K})\} \tag{12}$$

$$\begin{aligned} &\text{subject to:} \\ &(1 - \epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \leq k_{ii} - 2k_{ij} + k_{jj} \\ &\qquad\qquad\qquad\qquad \leq (1 + \epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \end{aligned}$$

$$\mathbf{G} = \sum_{i=1}^n \mathbf{K}_{i,n+i-1} \ , \mathbf{K} \succeq 0 \ , \sum k_{ij} = 0 \tag{13}$$

where the additional constraint $\sum k_{ij} = 0$ guarantees that the embedding is translationally invariant. Note that in the problem above the constraint set and the second term in the objective function are affine in the optimization variables. Thus, using the convex relaxation (5) leads to the following algorithm:

Algorithm 1: Low Dimensional Dynamic Embedding (LDE)

```
1.- Solve (iteratively)

      $\min_{k_{ij}}$ trace$((\mathbf{P}_r + \delta\mathbf{I})^{-1}\mathbf{P}_{r+1})+$
      trace$((\mathbf{Q}_r + \delta\mathbf{I})^{-1}\mathbf{Q}_{r+1})) - \lambda$trace$(\mathbf{K})$
      subject to:
      $\begin{bmatrix} \mathbf{P}_{r+1} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{Q}_{r+1} \end{bmatrix} \succeq 0$
      $(1-\epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \le k_{ii} - 2k_{ij} + k_{jj}$
                              $\le (1+\epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|_2^2$
      $\mathbf{G} = \sum_{i=1}^{n} \mathbf{K}_{i,n+i-1}$ , $\mathbf{K} \succeq 0$ , $\sum k_{ij} = 0$

2.- Let $\mathbf{K}_o$ denote the optimal solution found in step 1.
    Perform an SVD $\mathbf{K} = \mathbf{UDV^T}$ and compute the manifold coordinates $\mathbf{Y} = \mathbf{D}^{\frac{1}{2}}\mathbf{V}^T$.
```

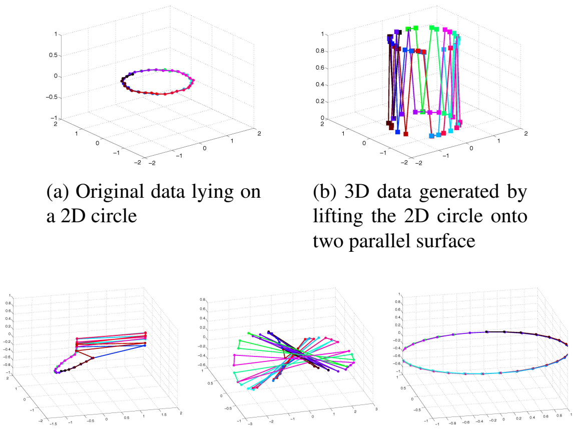
As in the SDE case, the rank of $\mathbf{K}$ indicates the embedding dimensionality. It is worth emphasizing that, when compared against the SDE algorithm described in section 2.2, the proposed algorithm requires adding a single semidefinite constraint (and the associated variables $\mathbf{P}$, $\mathbf{Q}$) to the optimization. In return, it minimizes not only the dimension of the embedding but also the complexity of the dynamic model required to represent the data.

4. Experimental Results and Benchmarking

In this section we apply the proposed algorithm to different embedding problems, involving both synthetic and real data and compare the results against those of obtained using the SDE [16] and dynamic global coordinate model (DGCM) [9] methods. We chose these two methods for benchmarking purposes since the SDE method was shown in [16] to outperform other static embedding approaches, while DGCM is one of the few methods that exploits dynamical information, albeit using an assumed, simple dynamics.

4.1. Embedding Fast Time Varying Data

This synthetic data example illustrates the role played by the temporal dynamics in obtaining smooth embeddings. The high dimensional data was generated by randomly lifting samples from a smooth 2D trajectory, in this case the circle shown in Fig. 3a to two parallel surfaces (Fig. 3b). The embeddings obtained applying the different methods to the lifted data are shown in Figures 3c–3e. As illustrated there, only the LDE method is capable of recovering the original smooth 2D trajectory. Note in passing that in this case the manifold obtained using SDE has dimension 3.



(a) Original data lying on a 2D circle

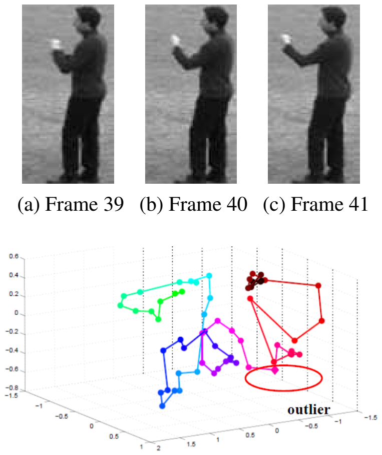
(b) 3D data generated by lifting the 2D circle onto two parallel surface

(c) The 2D manifold obtained using DGCM

(d) The manifold obtained using SDE.

(e) The 2D manifold obtained using LDE

Figure 3: Embedding fast time-varying data



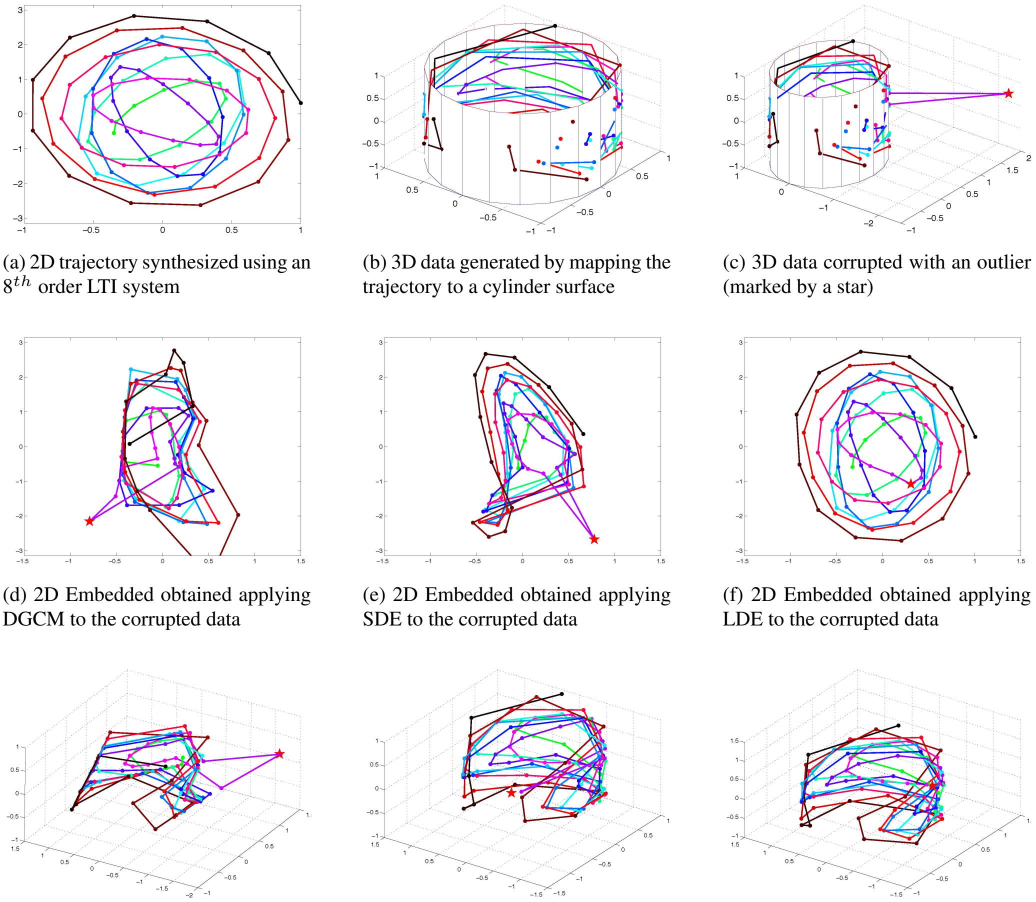
(a) Frame 39 (b) Frame 40 (c) Frame 41

(d) LDE embedding: the outlier has minimal effect on the overall manifold structure

Figure 4: Applying the LDE method to a sequence with an outlier (Frame 40).

4.2. Robustness to outliers

The next two examples illustrate the resiliency of the LDE method to outliers. Figure 4 shows the results of applying the LDE method to a boxing sequence from the KTH



(a) 2D trajectory synthesized using an 8^{th} order LTI system

(b) 3D data generated by mapping the trajectory to a cylinder surface

(c) 3D data corrupted with an outlier (marked by a star)

(d) 2D Embedded obtained applying DGCM to the corrupted data

(e) 2D Embedded obtained applying SDE to the corrupted data

(f) 2D Embedded obtained applying LDE to the corrupted data

(g) 3D trajectory reconstructed from the 2D DGCM embedding

(h) 3D trajectory reconstructed from the 2D SDE embedding

(i) 3D trajectory reconstructed from the 2D LDE embedding

Figure 5: Synthetic example illustrating the resiliency of the LDE method to outliers.

database, where the location of the actor in the 40th frame is shifted with respect to the location in the neighboring frames due to unintentional camera jitter. As shown there, the outlier has minimal influence on the manifold structure. On the other hand, as illustrated in Fig. 1, this single outlier substantially affects the manifold produced by the SDE method.

The next example considers synthetic data obtained by first generating a trajectory from an 8^{th} order LTI system that evolved in a 2D manifold, then lifting the data to the surface of a cylinder with unit radius, and finally adding an outlier, as shown in Figure 5b. Figures 5d–5f show the manifolds obtained using the different methods. As before, the embeddings produced by the SDE and DGCM are fragile to this single outlier, while the LDE embedding is largely unaffected.

4.3. Embedding sequences with missing data

Both the LDE and SDE methods can be easily extended to handle sequences with missing data. In the LDE case this can be accomplished by simply treating the projections $\mathbf{y}_j$ of the missing data as free variables by removing the constraints (7) for these cases. In the SDE case, the missing data can be recovered by first finding the embedding manifold and the corresponding manifold projections $\mathbf{y}_i$ of the available data, and then applying the inpainting procedure proposed in [2] to the sequence $\{\mathbf{y}_i\}$. Figure 6 shows qualitative results of applying these ideas to a sequence from the CMU mobo dataset. Table 1 summarizes quantitative results using the Walking (07_01.amc with missing frames 31 to 39) and Slowing Down (104_56.amc, with missing frames 51 to 59) sequences from the mocap dataset. In both cases, the missing frames were not used when computing the embeddings. As shown there, both the SDE and LDE methods do well for the walking sequence (with LDE outperforming SDE by 13%). On the other hand, for the slowing down sequence, LDE outperforms SDE by almost one order of magnitude. Briefly, since the walking sequence is almost periodic, considering only spatial information yields reasonably good interpolants, thus explaining the relatively good performance of SDE. On the other hand, for the non–periodic slowing down sequence, exploiting dynamical information is crucial to obtaining low-error interpolants. For comparison, we have also included the performance of a modified DGCM method, where missing data is handled by interpolating the manifold during each EM iteration. The low performance of the DGCM method is due to the fact that the simple first order dynamics it uses on the manifold are not rich enough to capture the temporal evolution of the data.

Table 1: Missing data reconstruction error

		DGCM	SDE	LDE
Walking	Mean Err.	56.51	21.91	**19.38**
Slowing down	Mean Err.	454.5	122.6	**38.0**

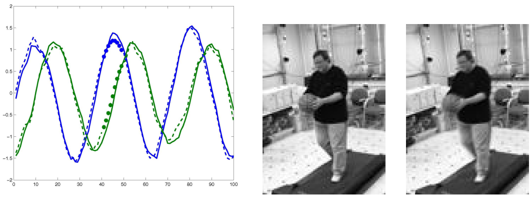

(a) (b) (c)

Figure 6: Using the LDE method to reconstruct missing frames 41-49. (a) Interpolation; dashed lines indicate the manifold obtained using 91 frames and rank minimization, the stars indicate interpolated data; solid lines are the ground truth. (b) Sample frame. (c) Reconstructed frame.

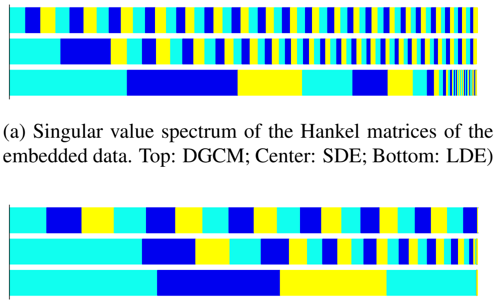
4.4. Dynamic Texture Reconstruction

Figure 7: Sample frames from the Flag sequence.

This example compares the performance of the DGCM, SDE and LDE algorithms for the problem of obtaining parsimonious representations of dynamic textures. Table 2 shows a comparison of the reconstruction error versus the order of the manifold dynamics, obtained when applying these methods to the flag video from the MIT temporal texture dataset (Figure 7). In all cases the original sequence was first projected to the manifold and then reconstructed using a radial basis function. When compared against DGCM, the LDE method is able to achieve a 11% improvement in the reconstruction error while reducing the dimension of the embedding manifold by almost an order of magnitude. When compared against SDE, the LDE method yields an improvement of at least 16% when using the same manifold dimension. Figure 8 sheds some insights into the reasons why LDE outperforms the other methods here. As shown there, by exploiting the temporal information, the SDE method is capable of finding an embedding where the matrix $\mathbf{K}_{LDE} = \mathbf{Y}_{LDE}^T \mathbf{Y}_{LDE}$ has rank 4 (hence only a 4-dimensional manifold is needed), compared against $\mathbf{K}_{SDE}$ that has roughly rank 10. Similarly, the Hankel matrices corresponding to the 3 methods have rank 6(LDE), 20 (SDE) and 27 (DGCM), indicating the need for dynamical models of these orders.

Table 2: Reconstruction error versus manifold dimension for the flag dynamic texture example

	Manifold Dimension	Error
DGCM	20	4.8145×10^6
LDE	20	3.3608×10^6
	10	4.0194×10^6
	4	4.3561×10^6
SDE	20	4.3325×10^6
	10	4.8099×10^6
	4	5.0646×10^6


(a) Singular value spectrum of the Hankel matrices of the embedded data. Top: DGCM; Center: SDE; Bottom: LDE)

(b) Singular value spectrum of the Grammian matrix of the embedded data. Top: DGCM; Center: SDE; Bottom: LDE)

Figure 8: Manifold and dynamic regressor dimensions for the flag example.

4.5. Appearance Prediction

This example illustrates the ability of the LDE method to extrapolate future values of the time series. In this experiment we considered 63 frames from a (downsampled by 4) jump–turn sequence from the CMU Motion Capture clips(91_46.amc). The goal is to use the first 50 frames to predict the values of the last 13. Applying the proposed LDE method leads to a one–dimensional manifold where the temporal evolution of the embedded data can be captured using a seventh order regressor. Similarly, using the SDE method leads to a one–dimensional manifold, but in this case the order of the required autoregressive model (estimated from the corresponding Hankel matrix) is 13. Table 3 shows the prediction error of these 2 methods, once the predicted manifold coordinates are mapped back to pixel space using a radial basis function. For comparison purposes we also show the results of the DGCM reconstruction. As illustrated there, the proposed LDE method yields a 50% reduction in the prediction error,while using a substantially simpler manifold model.

Table 3: Reconstruction error for the predicted frames, jump-turn sequence

	Manifold Dimension	Model Order	Error
DGCM	2	1	215.6
SDE	1	13	244.0
LDE	1	7	160.4

5. Conclusion

A key element in analyzing high-dimensional data is the exploitation of spatio/temporal correlations to obtain low dimensional, parsimonious representations. Existing nonlinear dimensionality reduction algorithm fully exploit the spatial topology, but most stop short fully of exploiting the temporal correlations that characterize a large portion of the data available in computer vision applications. Indeed, this information is typically discarded or used in relatively simple models that often fail to take full advantage of the underlying dynamics. In this paper we presented a new nonlinear embedding method, specially tailored to dynamic data that exploits both spatial and temporal information in order to obtain the simplest possible dynamical representation. As shown here, this can be accomplished by appealing to concepts from systems theory and convex optimization to reduce the problem to a computationally tractable convex optimization form. The advantages of the proposed algorithm vis-a-vis existing methods in terms of robustness, ability to reconstruct missing data and produce lower complexity embeddings were illustrated using both synthetic and real data. Research is currently underway seeking to extend these results to the case of time–varying manifolds and dynamics. Such a description is needed for instance when tracking a target that can undergo sudden appearance and motion changes.

References

[1] T. Ding. A robust identification approach to gait recognition. In *CVPR*, pages 1 –8, June 2008. 1

[2] T. Ding, M. Sznaier, and O. Camps. Rank minimization approach to video inpainting. In *ICCV*, Oct. 2007. 5

[3] A. Elgammal. Nonlinear generative models for dynamic shape and dynamic appearance. *2nd Int. Workshop on Generative Model-Based Vision*, 2004. 1

[4] A. Elgammal and C. Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *CVPR*, pages 681–688, 2004. 1

[5] M. Fazel, H. Hindi, and S. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *ACC*, volume 6, pages 4734 –4739 vol.6, June 2001. 3

[6] M. Fazel, H. Hindi, and S. Boyd. Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices. In *ACC*, volume 3, pages 2156 – 2162 vol.3, June 2003. 2, 3

[7] B. Ho and R. Kalman. Effective construction of linear, state-variable models from input/output functions. *Regelungstechnik*, 14:545–548, 1966. 3

[8] H. Lim, O. I. Camps, M. Sznaier, and V. Morariu. Dynamic appearance modelling for human tracking. In *CVPR*, pages 751–757, 2006. 1

[9] R. Lin, C. Liu, M. Yang, N. Ahuja, and S. Levinson. Learning nonlinear manifolds from time series. In *ECCV*, volume LNCS 3952, pages 245–256. Springer-Verlag, 2006. 1, 4

[10] C. Liu, R. Lin, N. Ahuja, and M. Yang. Dynamic textures synthesis as nonlinear manifold learning and traversing. In *BMVC*, 2006. 1

[11] M. Moonen, B. D. Moor, L. Vandenberghe, and J. Vandewalle. On- and off-line identification of linear state space models. *Int. J. of Control*, 49:219–232, 1989. 3

[12] S. Roweis, L. Saul, and G. Hinton. Global coordination of local linear models. *Advances in Neural Information Processing Systems*, 14:889–896, 2001. 1

[13] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *SCIENCE*, 290:2323–2326, 2000. 1

[14] C. Schldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *ICPR*, pages 32–36, 2004. 1

[15] M. Sznaier and O. I. Camps. A Hankel operator approach to texture modelling and inpainting. In *4th Int. Workshop on Texture Analysis and Synthesis*, volume 1, pages 125–130, 2005. 1

[16] K. Weinberger and L. Saul. Unsupervised learning of image manifolds by semidefinite programming. In *CVPR*, volume 2, pages II–988 – II–995 Vol.2, June 2004. 1, 2, 4