Manifold Learning for Object Tracking with Multiple Motion Dynamics

Jacinto C. Nascimento and Jorge G. Silva

Instituto de Sistemas e Robótica - Instituto Superior Técnico
Dept. of Electrical and Computer Engineering - Duke University*

Abstract. This paper presents a novel manifold learning approach for high dimensional data, with emphasis on the problem of motion tracking in video sequences. In this problem, the samples are time-ordered, providing additional information that most current methods do not take advantage of. Additionally, most methods assume that the manifold topology admits a single chart, which is overly restrictive. Instead, the algorithm can deal with arbitrary manifold topology by decomposing the manifold into multiple local models that are combined in a probabilistic fashion using Gaussian process regression. Thus, the algorithm is termed herein as *Gaussian Process Multiple Local Models* (GP–MLM).

Additionally, the paper describes a multiple filter architecture where standard filtering techniques, *e.g.* particle and Kalman filtering, are combined with the output of GP–MLM in a principled way. The performance of this approach is illustrated with experimental results using real video sequences. A comparison with GP–LVM [29] is also provided. Our algorithm achieves competitive state-of-the-art results on a public database concerning the left ventricle (LV) ultrasound (US) and lips images.

1 Introduction

There has been long standing interest in learning non-linear models to approximate high-dimensional data, and specifically in reducing the dimensionality of the data, while preserving relevant information. The scope of application is vast, including, *e.g.*, modeling dynamic textures in natural images, surface reconstruction from 3-D point clouds, image retrieval and browsing, and discovering patterns in gene expression data.

Consider the example of an image sequence. In the absence of features such as contour points or wavelet coefficients, each image is a point in a space of dimension equal to the number of image pixels. When facing an observation space of possibly tens or hundreds of thousands of dimensions, it is often reasonable to assume that the data is not dense in such a space and that many of the measured variables must be dependent with only a few free parameters that are embedded in the observed variables, frequently in a nonlinear way. Assuming that the number of free parameters remains the same throughout the observations, and also assuming spatially smooth variation of the parameters, we have geometric restrictions which can be well modeled as a manifold. Learning this

* This work was supported by project the FCT (ISR/IST plurianual funding) through the PIDDAC Program funds and by project "HEARTRACK" PTDC/EEA-CRO/098550/2008.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part III, LNCS 6313, pp. 172–185, 2010.

manifold is a natural approach to the problem of modeling the data, with the advantage of allowing *nonlinear* dimensionality reduction.

This paper proposes a new algorithm, named *Gaussian Process with Multiple Local Models* (GP–MLM), that applies manifold learning ideas to the problem of motion tracking, *e.g.*, in video sequences. The emphasis in motion tracking means that, unlike most manifold learning methods, the observations are assumed to be time-ordered. The proposed methodology addresses the problem of estimating unknown dynamics on an unknown manifold, from noisy observations. This leads to the simultaneous estimation of a nonlinear observation model and a nonlinear dynamical system - a nonlinear system identification type of problem, which has received some attention ([11,29,23]), but seldom in the context of manifolds, with a few recent exceptions [24]. While this problem is ill-posed (see *e.g.* [11]), it can be advantageous to exploit information that is common to both subproblems: the velocity vectors. Moreover, purely from a manifold learning point of view, GP–MLM addresses some limitations of existing methods, namely: (i) it is not limited to a simple coordinate chart - it can deal with arbitrary manifold topology through multiple local models; (ii) it provides a computationally efficient way to partition the manifold into multiple regions and compute the corresponding local parameterizations; (iii) it offers a principled way of combining the estimates from the multiple local models by using Gaussian process regression to compute the corresponding likelihoods. From a tracking perspective, it will be shown that GP–MLM can retrieve the contours with remarkable fidelity.

2 Background

Key concepts: A *manifold* [4] $\mathcal{M}$ is a set contained in $\mathbb{R}^m$, associated with a collection of p one-to-one continuous and invertible functions $\mathbf{g}_i : \mathcal{P}_i \rightarrow \mathcal{U}_i$, indexed by $i = 1, \ldots, p$ with overlapping domains $\mathcal{P}_i \subset \mathcal{M}$ such that $\mathcal{M}$ is covered by the union of the $\mathcal{P}_i$ and where each $\mathcal{U}_i \subset \mathbb{R}^n$. For points $\mathbf{y} \in \mathcal{P}_i \cap \mathcal{P}_j$ in the overlap between patches i and j, with images $\mathbf{x}_i$ and $\mathbf{x}_j$, it is possible to define a *transition function* $\Psi_{ij} : \mathbf{g}_i(\mathcal{P}_i \cap \mathcal{P}_j) \longrightarrow \mathbf{g}_j(\mathcal{P}_i \cap \mathcal{P}_j)$ which converts between the two local coordinate systems. See Fig. 1 for an illustration. Locally, $\mathcal{M}$ is "like" $\mathbb{R}^n$ and its *intrinsic dimension* is n. The $\mathbf{g}_i$ are called *charts*. It is assumed that $\mathcal{M}$ is *compact*, i.e., it can be covered with $p < \infty$ charts. The inverse mappings $\mathbf{h}_i = \mathbf{g}_i^{-1}$ are *parameterizations* of the manifold.

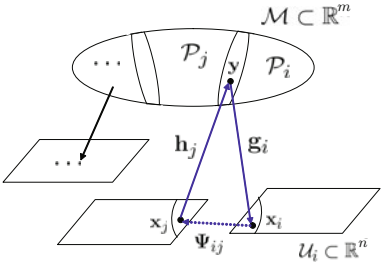

Fig. 1. A manifold and its charts

The $\mathcal{U}_i$ are *parametric domains* and the $\mathcal{P}_i$ are *patches*. Two charts $\mathbf{g}_i$ and $\mathbf{g}_j$ defined in the overlapping region $\mathcal{P}_i \cap \mathcal{P}_j$ should be compatible, that is, $\mathbf{g}_j^{-1}(\Psi_{ij}(\mathbf{g}_i(\mathbf{y}))) = \mathbf{y}$. For manifolds with arbitrary topology, there must be, in general, more than one chart and, therefore, more than one patch in order to maintain the one-to-one property.

The *tangent bundle* [4] of an n-dimensional manifold $\mathcal{M}$ is another manifold, $T(\mathcal{M})$, whose intrinsic dimension is $2n$ and whose members are the points of $\mathcal{M}$ and their tangent vectors. That is, $T(\mathcal{M}) = \{(\mathbf{y}, \mathbf{v}) : \mathbf{y} \in \mathcal{M}, \mathbf{v} \in T_{\mathbf{y}}(\mathcal{M})\}$ where $T_{\mathbf{y}}(\mathcal{M})$ is the tangent space of $\mathcal{M}$ at $\mathbf{y}$. It is readily apparent that $T_{\mathbf{y}}(\mathcal{M})$ is the set of possible velocity vectors of trajectories in $\mathcal{M}$ through $\mathbf{y}$. Therefore, any dynamic system defined in $\mathcal{M}$ must induce trajectories where both the velocities and their points of application belong to $T(\mathcal{M})$.

A *Gaussian process* [22] is a real-valued stochastic process $\{Y_{\mathbf{X}}\}_{\mathbf{x}\in\mathcal{X}}$, over an index set $\mathcal{X}$, where the joint probability density function for any finite set of indices $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ is Gaussian, with mean $\boldsymbol{\mu} \in \mathbb{R}^N$ and covariance $\mathbf{K} \in \mathbb{R}^{N\times N}$. Note that, in order to be a valid covariance matrix, $\mathbf{K}$ must be symmetric and positive semidefinite. This means that it can also be thought of as a valid *Mercer kernel* matrix. An attractive feature of Gaussian processes is that they allow the computation, in closed form, of probability densities in observation space.

Problem statement: Let $\mathbf{y}_{0:T-1} \equiv \{\mathbf{y}_t, t = 0, \ldots, T-1\}$, with discrete t and $\mathbf{y}_t \in \mathbb{R}^m$, be a trajectory. Let $\mathcal{Y} \equiv \{\mathbf{y}_{0:T_l-1}, l = 1, \ldots, L\}$ be a set of L such trajectories. It is assumed that the trajectories in $\mathcal{Y}$ lie close to an unknown manifold $\mathcal{M}$ of intrinsic dimension n (also unknown) embedded in $\mathbb{R}^m$, with $n < m$. Therefore, one or more lower dimensional representations $\mathcal{X}_i$ of the original set $\mathcal{Y}$ can be found, where each $\mathcal{X}_i \equiv \{\mathbf{x}_{0:T_l-1,i}, l = 1, \ldots, L\}$ represents all the trajectories in i-local coordinates, with $\mathbf{x}_{t,i} \in \mathbb{R}^n$. Being assumed compact, $\mathcal{M}$ can be charted by p charts, where p is unknown, and each $\mathcal{X}_i$ corresponds to one of the charts. It is intended to estimate $\mathcal{M}$ and identify the dynamics in the lower dimensional coordinates given by the charts of $\mathcal{M}$, assuming that the trajectories are generated by one or more discrete state space models of the form:

$$\mathbf{x}_{t,i} = \mathbf{f}_i(\mathbf{x}_{t-1,i}) + \boldsymbol{\omega}_{t,i} \tag{1}$$

$$\mathbf{y}_{t,i} = \mathbf{h}_i(\mathbf{x}_{t,i}) + \boldsymbol{\nu}_{t,i} \tag{2}$$

where $\boldsymbol{\omega}_{t,i}$ and $\boldsymbol{\nu}_{t,i}$ are noise variables. $\mathbf{h}_i$ is the i^{th} parametrization being used around $\mathbf{y}_t$, and $\mathbf{f}_i$ defines the dynamics. In summary, given $\mathcal{Y}$, we wish to learn the state model (2) and (1), thus capturing both geometrical and dynamical information about the trajectories.

Prior work: Several manifold learning algorithms have emerged in recent years. Recent advances include, on one hand, probabilistic methods such as the Generative Topographic Mapping (GTM) [2], Gaussian process related algorithms, such as Gaussian Process Latent Variable Models (GP-LVM) [18] and Gaussian Process Dynamical Models (GPDM) [29]; on the other hand, graph spectral methods such as ISOMAP [27], Locally Linear Embedding (LLE) [25], Laplacian [1] and Hessian Eigenmaps [10], as well as Semi-Definite Embedding [31,30].

Most methods assume that the manifold can be modeled using a single coordinate patch, an assumption that fails for manifolds with topologies as simple as a sphere. Also, spectral methods usually do not provide out-of-sample extension. Only a few methods, such as [5,19], attempt to deal with multiple charts without assuming p known somehow.

Estimating the intrinsic dimension n remains a challenge. The most common method [13] for estimating n is based on local Principal Component Analysis (PCA), relying on a threshold to select the n most significant eigenvalues of local covariance matrices. Other approaches can be found in [20,15] and the references therein. With either type of algorithms, the estimate often suffers from high variance and bias, as well as dependence on the unknown scale parameters for neighborhood analysis, as pointed out in [15]. Hence, dimensionality estimation continues to be a challenging problem, although some promising advances have recently been made using multiscale approaches [16].

Finally, while simultaneous dimensionality reduction and dynamical learning has received some attention [23,11,14], many of these approaches are not formulated in terms of manifolds. Some techniques that do explicitly use the manifold assumption are [24,21,12]. In [24], the manifold is modeled as a mixture of local linear hyperplanes (*i.e.*, factor analyzers), while we use instead a mixture of nonlinear GP regressors. In [21], a mapping from high-dimensional observations to latent states is estimated, both not the inverse. In [12] a manifold tracking method is used for learning nonlinear motion manifolds in the recovery of 3D body pose, but does not address the case when significant dynamics changes are observed in the video sequence (*i.e.*, multiple dynamics). Other methods that, like ours, are based on Gaussian Processes include [29,28]. However, [29] assumes one single chart and a priori fixed latent dimensionality, while [28] encourages certain topologies in a top-down manner, based on prior knowledge. Our approach also somewhat resembles, in spirit, the Spatial GPCA method [3], although Spatial GPCA operates at the pixel level rather than extracting contours and requires downsampling for computational reasons. Our main advantage resides in the fact that we perform dimensionality reduction, avoiding the need to downsample. In summary, our proposed method explicitly utilizes the manifold assumption, avoids the need to perform alignment of multiple local coordinate systems and maintains topological flexibility. To summarize, the following main differences should be considered: we consider arbitrary topologies with multiple nonlinear charts and multiple nonlinear dynamics, while existing methods consider either: (i) single nonlinear charts/dynamics [29]; (ii) multiple linear charts/dynamics [24]; or (iii) predefined topologies [28]. Besides, we do not marginalize over parameters and therefore can more easily perform out-of-sample prediction, as well as sequential state estimation, while GPDM [29] and [28] use batch inference.

3 GP–MLM Algorithm

The GP–MLM algorithm comprises the following steps: (i) estimation of intrinsic dimensionality and tangent subspaces; (ii) a nonparametric, nonlinear regression procedure for partitioning the manifold and learning the charts. Each of the steps is described here.

Intrinsic dimension: In the spirit of [13], GP–MLM addresses the problem of dimensionality estimation by automatically finding the "knee" of the eigenvalues $\lambda_1, \ldots, \lambda_m$ of the local covariance $\mathbf{S}_{\mathbf{y}_j} = \frac{1}{|\mathcal{B}_{\mathbf{y}_j,\epsilon}|-1} \sum_{\mathbf{y}_k \in \mathcal{B}_{\mathbf{y}_j,\epsilon}} (\mathbf{y}_k - \boldsymbol{\mu}_{\mathcal{B}_{\mathbf{y}_j,\epsilon}})(\mathbf{y}_k - \boldsymbol{\mu}_{\mathcal{B}_{\mathbf{y}_j,\epsilon}})^T$, using local PCA, but in GP–MLM this is done for *all* ϵ-local neighborhoods $\mathcal{B}_{\mathbf{y}_j,\epsilon}$ around each data point $\mathbf{y}_j$. For each neighborhood, the eigenvalue immediately before the greatest drop in value should correspond to the intrinsic dimension, estimated by $\widehat{n}_j \equiv \arg\max_{i=1,\ldots,m-1} |\lambda_{i+1} - \lambda_i|$. The global estimate is $\widehat{n} = \text{median}_{j=1,\ldots,N}(\widehat{n}_j)$, which is more robust than the mean. The advantage of this approach is that it takes advantage of the potentially large number of local PCA neighborhoods.

Temporal information is also used to improve the estimates of the tangent subspaces. We use the first differences $\Delta_{\mathbf{y}_t} = \mathbf{y}_t - \mathbf{y}_{t-1}$, together with the observations $\mathbf{y}_t$ for performing local PCA, by augmenting $\mathcal{B}_{\mathbf{y}_j,\epsilon}$ with $\mu_{\mathcal{B}_{\mathbf{y}_j,\epsilon}} + \Delta\mathbf{y}_k$, for $k = 1, \ldots, |\mathcal{B}_{\mathbf{y}_j,\epsilon}|$, with the neighborhood centers $\mu_{\mathcal{B}_{\mathbf{y}_j,\epsilon}}$ given by the sample means $\mu_{\mathcal{B}_{\mathbf{y}_j,\epsilon}} = \frac{1}{|\mathcal{B}_{\mathbf{y}_j,\epsilon}|} \sum_{\mathbf{y}_k \in \mathcal{B}_{\mathbf{y}_j,\epsilon}} \mathbf{y}_k$. Note that the velocities (of which the $\Delta_{\mathbf{y}_t}$ are rough estimates), applied at the neighborhood centers, must live on the corresponding tangent subspaces. This leads to an effective increase in the number of available points at each neighborhood, from $|\mathcal{B}_{\mathbf{y}_j,\epsilon}|$ points to $2|\mathcal{B}_{\mathbf{y}_j,\epsilon}|$ (or $2|\mathcal{B}_{\mathbf{y}_j,\epsilon}| - 1$ if either the first or last $\Delta_{\mathbf{y}_t}$ can not be computed).

Charts: At this stage, an estimate $\widehat{n}$ of the intrinsic dimension is available. The tangent bundle $T_{\mathcal{M}}$ can, if approximated by some finite set of $\widehat{n}$-dimensional tangent linear hyperplanes, form a convenient collection of local parametric domains upon which to map the manifold points. We partition $\mathcal{M}$ into overlapping patches $\mathcal{P}_1, \ldots, \mathcal{P}_p$, find p corresponding tangent hyperplanes, and estimate mappings back and forth between the patches and the hyperplanes. It is important to find a partition which facilitates subsequent estimation of the mappings. We follow the Tangent Bundle Approximation (TBA) approach proposed in [26] which is based on *principal angles*, a generalization of the concept of angle to linear subspaces.

The idea is not to allow the maximum principal angle between the tangent subspaces – spanned by matrices $\mathbf{V}_i$ and $\mathbf{V}_j$ of column eigenvectors found by local PCA on neighborhoods i and j – to vary more than a set threshold τ. The exact value of τ is not critical, as long as it is below $\frac{\pi}{2}$.

Patches are found by an agglomerative clustering procedure, *i.e.*, region growing. Each patch grows by appending all neighboring (within an ϵ radius) points where the tangent subspace does not deviate, in maximum principal angle, more than a set threshold from the tangent subspace at the initial seed. Any specific point may belong to more than one patch. The final result is a covering of $\mathcal{M}$ by a finite number, p, of overlapping patches. Within each patch, the curvature is controlled through τ, and the distance test ensures that each patch is a connected set. Subsequently, we find the best fitting hyperplane for each patch using PCA, providing local coordinate systems for different manifold regions. The collection of hyperplanes approximates the tangent bundle. Thus, PCA must be performed twice: first with local scope, in tight neighborhoods $\mathcal{B}_{\mathbf{x},\epsilon}$ around each point, so that the principal angles can be controlled *within the patch* during the partitioning procedure; and second, for all patch members, in order to find an overall hyperplane for charting and the corresponding coordinate system. If $\mathbf{S}_{\mathcal{P}_i}$ is the

covariance of the points in $\mathcal{P}_i$, i.e. $\mathbf{S}_{\mathcal{P}_i} = \frac{1}{|\mathcal{P}_i|-1}\sum_{\mathbf{y}_k \in \mathcal{P}_i}(\mathbf{y}_k - \boldsymbol{\mu}_{\mathcal{P}_i})(\mathbf{y}_k - \boldsymbol{\mu}_{\mathcal{P}_i})^T$, then, by performing the eigendecomposition $\mathbf{S}_{\mathcal{P}_i} = \mathbf{V}_{\mathcal{P}_i}\mathbf{D}_{\mathcal{P}_i}\mathbf{V}_{\mathcal{P}_i}^T$, where $\mathbf{V}_{\mathcal{P}_i}$ is the matrix whose columns are the eigenvectors of $\mathbf{S}_{\mathcal{P}_i}$ and $\mathbf{D}_{\mathcal{P}_i} = \text{diag}(\lambda_1, \ldots, \lambda_m)$, an orthonormal basis is found in the columns of $\mathbf{V}_{\mathcal{P}_i}$. Note that the patch mean $\boldsymbol{\mu}_{\mathcal{P}_i}$ does not, in general, coincide with the patch seed. The added computational burden of patch-wide PCA is negligible, compared to that of local PCA.

An important note is that GP–MLM (like TBA) does not guarantee that the number of patches is minimal - in fact, the followed approach usually leads to an overestimation of the number of patches needed to cover a manifold. On the other hand, it should also be noted that, since the principal angles only need to be computed between the data and the seeds, and not between all pairs of data points, the overall complexity of the partitioning algorithm is *not* quadratic in N, but rather it is $O(Np)$.

Gaussian process regression: Using the coordinate systems found above, and since there are no folds in any patch (thanks to the angular restriction), the regression problem associated with the charts is significantly simplified. From the previously obtained partition of the dataset into patches $\mathcal{P}_i$, with $i = 1, \ldots, p$, it is now intended to estimate the charts $\mathbf{g}_i(\mathbf{y})$. Let a particular training point $\mathbf{y}$, belonging to patch $\mathcal{P}_i$, be denoted $\mathbf{y} = [y_1 \ldots y_m]^T$, where $y_j, j = 1, \ldots, m$ refers to the j^{th} coordinate. Projecting $\mathbf{y}$ onto the subspace spanned by $\mathbf{V}_{\mathcal{P}_i}$ yields the i^{th} local representation $\mathbf{x}_i$. This can be done according to $\tilde{\mathbf{x}}_i = \mathbf{V}_{\mathcal{P}_i}^T(\mathbf{y} - \boldsymbol{\mu}_{\mathcal{P}_i})$ in which the intermediate quantity $\tilde{\mathbf{x}}_i$ simply corresponds to $\mathbf{y}$ in a new coordinate system with origin at $\boldsymbol{\mu}_{\mathcal{P}_i}$ and versors given by the columns of $\mathbf{V}_{\mathcal{P}_i}$; the following step is

$$\mathbf{x}_i = [\tilde{x}_{i,1} \ldots \tilde{x}_{i,n}]^T = \mathbf{g}_i(\mathbf{y}) \tag{3}$$

where $\mathbf{x}_i$ denotes a truncated version of $\tilde{\mathbf{x}}_i$ using only the first n components. This is the chart. The inverse mapping, that is, the parametrization $\mathbf{h}_i(\mathbf{x}_i)$ follows the expression

$$\mathbf{h}_i(\mathbf{x}_i) = \mathbf{V}_{\mathcal{P}_i}\begin{bmatrix}\mathbf{x}_i & \tilde{\mathbf{h}}_i(\mathbf{x}_i)\end{bmatrix}^T + \boldsymbol{\mu}_i \tag{4}$$

in which $\tilde{\mathbf{h}}_i$ must be estimated. The remaining $m - n$ components of $\tilde{\mathbf{x}}_i$ are approximated by $\tilde{\mathbf{h}}_i(\mathbf{x}_i)$, and thus the nonlinear character of the manifold is preserved. In the i^{th} local coordinates, the parametrization is $\mathbf{x}_i \rightarrow [\mathbf{x}_i \quad \tilde{\mathbf{h}}_i(\mathbf{x}_i)]^T$.

It is now necessary to estimate $\tilde{\mathbf{h}}$. For a particular $m - n$-dimensional vector $\tilde{\mathbf{x}}_i$, consider an independent Gaussian process for each scalar component $\tilde{x}_j$, dropping the j subscript of the j^{th} coordinate for conciseness – the exposition will proceed, without loss of generality, as if $m - n = 1$. The regression problem is that of estimating $\tilde{\mathbf{h}}_i$, from the set of available data $\tilde{\mathcal{X}}_{\mathcal{P}_i} = \{\tilde{x}_{k,i}\}_{k=1:|\mathcal{P}_i|}$ and the corresponding set of $|\mathcal{P}_i|$ local projections $\mathcal{X}_{\mathcal{P}_i} = \{\mathbf{x}_{k,i}\}_{k=1:|\mathcal{P}_i|}$, all collected in $\tilde{\mathbf{x}} \in \mathbf{R}^{|\mathcal{P}_i|}$ and $\mathbf{X} \in \mathbf{R}^{n \times |\mathcal{P}_i|}$ respectively. The estimate should be the one that best matches the model $\tilde{\mathbf{x}}_k = h_i(\mathbf{x}_{k,i}) + \omega_{k,i}$ with noise $\omega_{k,i} \sim \mathcal{N}(0, \sigma_i^2)$, $\forall_k$. It is assumed that the joint pdf of $\tilde{\mathbf{x}}$ is Gaussian, with zero mean (the data can be mean-subtracted) and with known covariance matrix $\mathbf{K} \in \mathbb{R}^{|\mathcal{P}_i| \times |\mathcal{P}_i|}$. With this assumption, it is possible to derive the conditional density $p(\tilde{\mathbf{x}}|\mathbf{X})$. Furthermore, for any new set of inputs $\mathbf{X}^\star$ outside of the training set, the conditional density $p(\tilde{\mathbf{x}}^\star|\mathbf{X}^*, \mathbf{X}, \tilde{\mathbf{x}})$ is given [22] by

$$\begin{aligned} p(y^\star|\mathbf{X}^*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(&\mathbf{K}(\mathbf{X}^*, \mathbf{X})\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{y}, \\ &\mathbf{K}(\mathbf{X}^*, \mathbf{X}^*) - \mathbf{K}(\mathbf{X}^*, \mathbf{X})\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{K}(\mathbf{X}, \mathbf{X}^*)). \end{aligned} \tag{5}$$

For constructing $\mathbf{K}$, we choose the RBF covariance function

$$k(\mathbf{x}_i, \mathbf{x}_j) = \theta_1 \exp(-\frac{1}{2\theta_2}\|\mathbf{x}_i - \mathbf{x}_j\|^2) + \delta_{ij}\theta_3 \tag{6}$$

and optimize the hyperparameters by maximizing the marginal likelihood, as proposed in [22].

4 Dynamical Learning Using the Manifold Model

We now extend GP–MLM to deal with the simultaneous estimation of the data manifold and dynamics. The idea is to start from the state model in (1), (2), assuming that, in the observation equation, $\mathbf{h}$ is given by the manifold model found by the GP–MLM and therefore fixed. We then tackle the following two subproblems: (i) Identification of the dynamics $\mathbf{f}$, given h; (ii) Estimation of the state at time t, given all information up to time t. The first subproblem is called system identification and is solved offline, as explained next.

System identification: We assume that the training trajectories have been mapped to low dimensional points $\mathbf{x}_{t,i}$ in patch $\mathcal{P}_i$, at instant t. For each i, we form training pairs $(\mathbf{x}_{t-1}, \mathbf{x}_t)$. The subscript i has been dropped for conciseness, since it will be assumed that the trajectory segment remains on patch i. This is no loss of generality, since in the case when the original high dimensional $\{\mathbf{y}_t\}_{t=0:T-1}$ crosses patches i and j (or more), this simply results in multiple trajectory segments, $\{\mathbf{x}_{t,i}\}_{t=0:T_i-1}$ and $\{\mathbf{x}_{t,j}\}_{t=0:T_j-1}$, which can be treated separately and which count towards the dynamics in patch $\mathcal{P}_i$ and $\mathcal{P}_j$ respectively.

The regression procedure aims at finding the best $\mathbf{f}_i$ that maps $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$ in patch $\mathcal{P}_i$, given the corresponding set $\mathcal{X}_i$ of trajectory segments pertaining to $\mathcal{P}_i$. The generative model is

$$\mathbf{x}_{t,i} = \mathbf{f}_i(\mathbf{x}_{t-1,i}) + \boldsymbol{\omega}_{t,i}. \tag{7}$$

In the case when the dynamics are linear, and dropping the i subscript, (7) turns into $\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \boldsymbol{\omega}_t$, with $\mathbf{A}$ a $n \times n$ matrix. When, additionally, the $\boldsymbol{\omega}_t$ are iid and Gaussian, then this is a thoroughly studied case; identification consists of estimating $\mathbf{A}$ from the pairs $(\mathbf{x}_{t-1}, \mathbf{x}_t)$, which can be done by the Least Mean Squares method.

When $\mathbf{f}$ is not a linear function of $\mathbf{x}$, then we propose a nonparametric approach, again based on Gaussian process regression using the RBF kernel (6).

As in the geometrical step, but now with training pairs $(\mathbf{x}_{t-1}, \mathbf{x}_t)$ arranged in matrices $\boldsymbol{\Xi}, \mathbf{X}$ defined as $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_{T-1}]$, $\boldsymbol{\Xi} = [\mathbf{x}_0, \ldots, \mathbf{x}_{T-2}]$, the regression procedure yields, for any new $\mathbf{x}^\star_{t-1}$, Gaussian conditional densities $p(\hat{x}_t^{(i)}|\mathbf{x}^\star_{t-1}, \boldsymbol{\Xi}, \boldsymbol{\xi}^{(i)}) = \mathcal{N}(\mu_{x_t^{(i)}}, \sigma^2_{x_t^{(i)}})$, for all $i = 1, \ldots, n$ components of $\hat{\mathbf{x}}_t$ and with $\boldsymbol{\xi}^{(i)} \in \mathbb{R}^{(T-1)}$ equal to the i-th column of $\mathbf{X}^T$.

Filtering: The second subproblem is one of *filtering*. It is not desirable in general to use one single observation to obtain the state, because simply inverting the observation equation (2) ignores the temporal dependence between successive data points. The

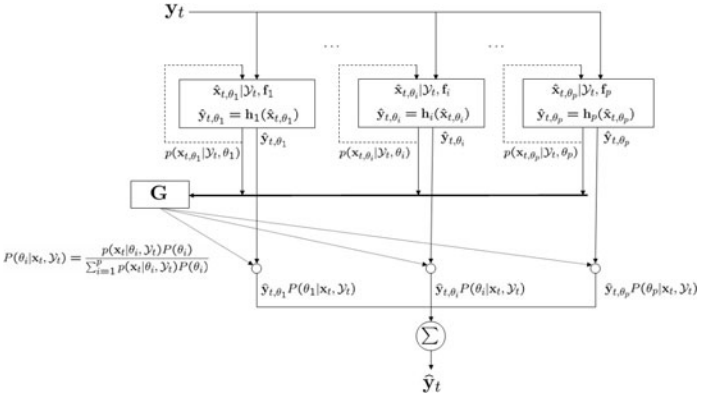

Fig. 2. Block diagram of the mixture architecture for combining the local dynamic models

correct procedure is to estimate the state, at each instant t using information about the whole trajectory up to time t. This can be done online by a variety of filtering methods.

Note that GP–MLM is a multiple-model framework; thus, we employ one filter for each patch, using different dynamics, observation models and coordinate systems. This means that a procedure for combining the local estimates is required. Fig. 2 illustrates how this is performed. Essentially, we make use of the predictive variance from each local GP in order to compute patch posterior probabilities (mixture weights) inexpensively, *i.e.*, we set

$$P(\theta_i|\mathbf{x}, \mathcal{Y}_t) \propto p(\mathbf{x}|\theta_i, \mathcal{Y}_t). \tag{8}$$

The mixture weights provided by block $\mathbf{G}$ take the different dynamics into account. Different strategies are possible: a "winner-take-all" rule, where only the output of the model with the highest posterior probability is used, or a "blending" rule, where the weighted average using all models is computed. In this paper we present results using Kalman and particle filtering with the above mentioned rules.

5 Experimental Results

This section presents an experimental evaluation of GP–MLM in several data sequences. The evaluation is done in two main situations: first, two ultrasound sequences of the left ventricle (LV) of the heart, aiming at estimating the endocardium boundary. In both, the object of interest undergoes changing motion dynamics. For all experiments, three identification strategies are compared: (i) linear first order; (ii) linear second order and (iii) Gaussian process (GP) first order. In the second experiment, lip sequences are considered. Two situations are presented: (i) speaking, and (ii), singing, where in the latter the lips boundary exhibits a higher deformation. An objective evaluation is conducted for all the experiments using several metrics proposed in the literature.

Heart tracking: This example consists of two ultrasound (US) images sequence. Each US image displays a cross section of the left ventricle (LV) in the long-axis. The length of the sequences is: 490 frames (26 cardiac cycles) and 470 frames (19 cardiac cycles).

The heart motion is described by two dynamics: an expansion motion that occurs in diastole phase, and a contraction motion that characterizes the systole phase. To represent the boundary of the LV, 21 contour points are used, which would require thousands of manual clicks, if we were to obtain ground-truth by hand. Instead, an automatic procedure is used [17]. The MMDA (*Multiple Model Data Association*) tracker is robust with respect to outliers and capable of coping with different, abrupt motion dynamics. Thus, we measure the performance of the GP–MLM with the respect to the MMDA tracking output, which we treat as ground truth.

In this study, we go further in the attempt to find the best technique (*i.e.* Kalman vs particle filtering; "winner-take-all" vs "blending" rules); at the same time we hope to demonstrate the superiority of the non-linear GP $1st$ order model. To attain this goal an objective evaluation between the MMDA contour estimates (taken as gold contours) and the GP–MLM estimates is provided; several metrics proposed in the literature for contours comparison are used. To accomplish this, a comparison between the contour estimates provided by MMDA tracker (*i.e.* the ground-truth) and the GP–MLM estimate is conducted. Five metrics are used in these tests: Hammoude distance (HMD) [6]; average distance (AV); Hausdorff distance (HDF); Mean sum of Square Distances (MSSD); Mean Absolute Distance (MAD) (as in used in [9]); and the DICE metric. Next, we briefly describe them.

Let $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_x}\}$, and $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{N_y}\}$, be two sets of points obtained by sampling the estimated contour and the reference contour. The smallest distance from a point $\mathbf{x}_i$ to the curve $\mathcal{Y}$ is

$$d(\mathbf{x}_i, \mathcal{Y}) = \min_j ||\mathbf{y}_j - \mathbf{x}_i|| \tag{9}$$

This is known as the distance to the closest point (DCP). The average distance between the sets $\mathcal{X}$, $\mathcal{Y}$ is

$$d_{\mathrm{AV}} = \tfrac{1}{N_x} \textstyle\sum_{i=1}^{N_x} d(\mathbf{x}_i, \mathcal{Y}) \tag{10}$$

where N_x is the length of the $\mathcal{X}$ The Hausdorff distance between both sets is defined as the maximum of the DCP's between the two curves

$$d_{\mathrm{HDF}}(\mathcal{X}, \mathcal{Y}) = \max\Big(\max_i\{d(\mathbf{x}_i, \mathcal{Y})\}, \max_j\{d(\mathbf{y}_j, \mathcal{X})\}\Big) \tag{11}$$

The Hammoude distance is defined as follows [6]

$$d_{\mathrm{HMD}}(\mathcal{X}, \mathcal{Y}) = \frac{\#((R_{\mathcal{X}} \cup R_{\mathcal{Y}}) - (R_{\mathcal{X}} \cap R_{\mathcal{Y}}))}{\#(R_{\mathcal{X}} \cup R_{\mathcal{Y}})} \tag{12}$$

where $R_{\mathcal{X}}$ represents the image region delimited by the contour $\mathcal{X}$, similarly for $R_{\mathcal{Y}}$.

To define MSSD [7] and MAD [8] distances, let us consider the tracked sequence S_i with m contours $\{c_1, c_2, ..., c_m\}$, where each jth contour c_j has n points $\{(x_{j,1}, y_{j,1})$,-$(x_{j,2}, y_{j,2}), ..., (x_{j,n}, y_{j,n})\}$, the distances of sequence S_i from other version of the sequence S_i^r (which is the ground truth) are

$$d_{\mathrm{MSSD_i}} = \tfrac{1}{m} \textstyle\sum_{j=1}^{m} \tfrac{1}{n} \sum_{k=1}^{n} ((x_{j,k} - x_{j,k}^r)^2 + (y_{j,k} - y_{j,k}^r)^2) \tag{13}$$

$$d_{\mathrm{MAD_i}} = \tfrac{1}{m} \textstyle\sum_{j=1}^{m} \tfrac{1}{n} \sum_{k=1}^{n} \sqrt{(x_{j,k} - x_{j,k}^r)^2 + (y_{j,k} - y_{j,k}^r)^2} \tag{14}$$

The overall performance measure for a particular method is the averaged distance on the whole test set of L sequences:

$$d_{\mathrm{MSSD}} = \frac{1}{L}\sum_{i=1}^{L} d_{\mathrm{MSSD_i}}, \qquad d_{\mathrm{MAD}} = \frac{1}{L}\sum_{i=1}^{L} d_{\mathrm{MAD_i}}$$

The DICE metric is also used, which is the mean perpendicular distance between estimated contour and the ground-truth contour. We compute the average metric distance for all points in the curve as follows

$$d_{\mathrm{DICE}} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{y}_i\|\mathbf{n}_i \tag{15}$$

where $\mathbf{n}_i$ defines the normal vector at point i.

Table 1 left, lists the MSE for the three identification strategies for each path found by GP–MLM. In both sequences the GP consistently provide the best results comparing with the remaining strategies. In these experiments, the data was split in two disjoint training/test sets (50% for training and testing).

Objective evaluation: Table 1 shows the fidelity in the representation of the LV contour obtained in the two US sequences. These values correspond to the mean values of the metrics. From this table and in both sequences and for the majority of the measures, the best values are obtained when ones used particle filtering with the "blending" rule. Although, the particle filtering with the "blending" rule provides the best results, what is important to stress is that any tracking method can be incorporated in the framework and the manifold is always well estimated.

In this study we carried out an additional experiment, we varied the number of frames used in training-testing sets for both sequences, more specifically, we varied the number of training images from 25%, 50% and 75%. Table 2 shows the Hammoude distance using the particle filter with the blending rule (similar behavior is observed of the other tracking versions). From the Table 2, what it is interesting to note is that changing the number of training-test images, the manifold is always well estimated for both sequences, where a slight and negligible increase of this metric is shown.

Table 1. MSE for the three identification strategies obtained in both US sequences: linear $1st$ and $2nd$ order models and a non-linear GP model (left); objective evaluation considering five metrics. The mean values are shown for the two US sequences(right).

	Sequence # 1 MSE		
Patch #	Linear 1^{st} order	Linear 2^{nd} order	GP 1^{st} order
1	4.7826	6.9604	**1.1440**
2	2.5327	1.7007	**0.4164**
3	4.8318	4.4788	**0.4199**
4	7.1060	1.7813	**0.3520**
5	2.0454	4.2491	**0.4662**
	Sequence # 2 MSE		
Patch #	Linear 1^{st} order	Linear 2^{nd} order	GP 1^{st} order
1	5.8521	5.3898	**0.5788**
2	5.9573	3.6770	**0.1379**
3	4.8241	4.5712	**0.4720**
4	6.0968	4.9661	**2.6763**

		d_{HMD}	d_{AV}	d_{HDF}	d_{MSSD}	d_{MAD}	d_{DICE}
Seq. 1	KF - WTA	0.14	3.08	5.48	13.23	3.09	2.52
	KF - BLD	0.14	3.08	5.47	13.20	3.09	2.53
	PF - WTA	**0.09**	2.12	3.86	7.42	2.17	1.79
	PF - BLD	**0.09**	**2.02**	**3.63**	**6.22**	**2.04**	**1.70**
Seq. 2	KF - WTA	0.11	2.73	4.80	10.66	2.79	2.00
	KF - BLD	0.11	2.81	4.89	11.33	2.89	2.04
	PF - WTA	**0.08**	1.76	3.70	4.92	1.78	**1.59**
	PF - BLD	**0.08**	**1.74**	**3.64**	**4.81**	**1.75**	**1.59**

Table 2. Hammoude metric for two US sequences, varying the number of training images

d_{HMD}	25%	50%	75%
Seq. 1	0.063	0.088	0.093
Seq. 2	0.077	0.081	0.090

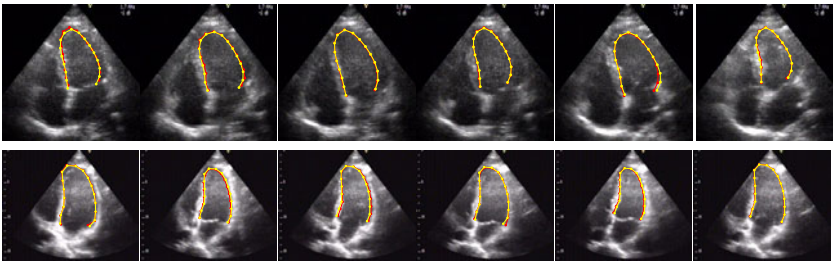

Fig. 3. GP–MLM tracking estimates (yellow line), superimposed with MMDA tracker (red line) taken as gold contours. First sequence (top row), and second sequence (bottom row).

Fig. 3 shows some snapshots for both LV sequences. The manifold results are shown in yellow solid lines, and the ground truth is (output of the MMDA tracker).

Lip tracking: The second example consists of lip tracking in two different situations: speaking and singing. We show results in seven speaking sequences and three singing sequences. In the speaking sequences, each one has about 80 images, while in the singing case the sequence are a bit longer (100 images). Comparing to the previous example, the nonrigid object (lip boundary) exhibits an higher variability in the shape, specially when a person is singing.

From this point on, and due to the lack of space we present the results using particle filtering with the blending rule (other alternatives are, of course, possible to use as previously illustrated).

In the following, the training and testing mechanism follows a leave-one-out strategy (this can be also used in the case of the LV tracking, but there was no need to do this due to the large extension of the LV sequences).

Table 3 (left) shows the results obtained for the speaking case. It can be seen that the framework proposed herein maintain comparable results as in the previous case. Recall that the Hammoude metric (XOR pixel wise operation between the ground truth and the manifold estimates) is always below 15%. Comparing to the results obtained for the singing sequences (see right of the Table 3), we see that a small decrease on this distance, and the small increase of the metrics which penalizes maximum local distances. This is somehow expected, since in this case, a large and sudden changes in the lips boundary may be obtained in consecutive frames. For instance, in Fig. 5 (top row) the $2nd$, $3rd$ and $7th$, $8th$ frames are consecutive in the video frame. These correspond to difficult situations where the GP–MLM is able to produce good results.

We also compare the GP-MLM approach with the *Gaussian Process Latent Variable Model* GPLVM. [1] To perform the comparison, we first used the reconstruction parameters

[1] The code is available from the authors at `http://www.cs.man.ac.uk/~neill/gplvm/`

Table 3. Average distances and metrics obtained using the GP–MLM, for speaking sequences (left) and singing sequences (right)

	Speaking Sequences					
	d_{HMD}	d_{AV}	d_{HDF}	d_{MSSD}	d_{MAD}	d_{DICE}
Seq1	0.08	2.89	5.80	12.58	3.06	2.14
Seq2	0.11	3.68	7.33	22.44	4.07	3.29
Seq3	0.15	4.62	10.29	48.78	5.69	4.26
Seq4	0.09	3.74	7.93	39.99	4.18	3.04
Seq5	0.14	4.36	8.62	35.19	4.71	3.91
Seq6	0.08	3.23	6.86	15.31	3.33	2.53
Seq7	0.10	3.67	8.08	23.65	3.93	3.02

	Singing Sequences					
	d_{HMD}	d_{AV}	d_{HDF}	d_{MSSD}	d_{MAD}	d_{DICE}
Seq1	0.16	5.19	10.82	68.62	6.60	4.52
Seq2	0.14	4.31	9.07	71.53	5.24	4.19
Seq3	0.14	4.95	10.07	54.79	5.48	4.33

Fig. 4. GP–MLM tracking estimates for seven speaking sequences shown in red dots

of the GPLVM (see [18] for details). We then applied the GPLVM (as we do for the GP-MLM) using the particle filtering with the blending rule for contour tracking. We illustrate the results by showing the Hammoude distance provided by both methods. As previously, this metric is computed between the GP-MLM contour estimates with the output of the MMDA (taken as the ground-truth); and the GPLVM estimates with the MMDA. From the Table 4, we can see that comparable results are achieved. Recall that, for sequences having a higher deformation (see the results in the singing sequences) the GP-MLM exhibits good results.

Fig. 5. GP–MLM tracking estimates for three singing sequences shown in red dots

Table 4. Comparison between the tracked contours provided by GP-MLM and the GPLVM in terms of the Hammoude distance. The mean values of the distance are shown for each sequence.

	Left Ventricle		**Speaking**							**Singing**		
GP-MLM	**0.088**	**0.081**	0.104	**0.079**	0.142	**0.092**	0.150	**0.084**	**0.113**	**0.157**	**0.141**	**0.145**
GPLVM	0.091	0.088	**0.091**	0.081	**0.112**	0.095	**0.140**	**0.084**	0.127	0.177	0.151	0.156

6 Conclusions

A novel method for manifold learning has been proposed in this paper. This framework employs a local and probabilistic approach to learn a geometrical model of the manifold and thus reduce the dimensionality of the data. The GP-MLM uses the Gaussian process regression as a way to find continuous patches. The decomposition of the patches renders GP-MLM more flexible when dealing to arbitrary topology. A framework was proposed for probabilistically combining the local patch estimates, based on the output of Gaussian process regression. The optimization of the Gaussian process hyperparameters is accomplished via standard gradient descent, which offers a suitable and effective tool for model selection. Dynamical system identification and recursive state estimation are tackled by using the multiple local models returned by the manifold learning step. Identification is accomplished via Gaussian process regression. A filter bank architecture (which uses the learned dynamics) was also developed, both for Kalman and particle filters. A systematic comparative evaluation in several sequences was conducted, combining both filtering techniques with different gating strategies. The experimental evaluation provided indicates that the performance of the GP-MLM provides good results and it is competitive with the GPLVM approach.

Issues for future research include reducing the number of patches, as well as a way to compute the scale parameter ϵ. Reliable estimation of the intrinsic manifold dimension also remains a difficult challenge, on its own right. Robust statistics may be a fruitful direction of research for this problem.

References

1. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15, 1373–1396 (2003)
2. Bishop, C., Svensen, M., Williams, C.: GTM: The generative topographic mapping. Neural Computation 10, 215 (1998)
3. Ravichandran, A., Vidal, R., Halperin, H.: Segmenting a Beating Heart Using PolySegment and patial GPCA. In: IEEE International Symposium on Biomedical Imaging, pp. 634–637 (2006)
4. Boothby, W.M.: An Introduction to Differential Manifolds and Riemannian Geometry. Academic Press, London (2003)
5. Brand, M.: Charting a manifold. In: NIPS, p. 15 (2002)
6. Hammoude, A.: Computer-assited endocardial border identification from a sequence of two-dimensional echocardiographic images, PhD (1988)
7. Akgul, Y., Kambhamettu, C.: A coarse-to-fine deformable contour optimization framework. IEEE Trans. PAMI. 25(2), 174–186 (2003)

8. Mikić, I., Krucinki, S., Thomas, J.D.: Segmentation and tracking in echocardiographic sequences: Active contours guided by optical flow estimates. IEEE Trans. Med. Imag. 17(2), 274–284 (1998)
9. Comaniciu, D., Zhou, X., Krishnan, S.: Robust real-time tracking of myocardial border:an information fusion approach. IEEE Trans. Med. Imag. 23(7), 849–860 (2004)
10. Donoho, D.L., Grimes, C.: Hessian eigenmaps: new locally linear embedding techniques for high-dimensional data. Tech. Report TR-2003-08 (2003)
11. Doretto, G., Chiuso, A., Wu, Y., Soatto, S.: Dynamic textures. IJCV (2003)
12. Elgammal, A., Lee, C.-S.: Nonlinear manifold learning for dynamic shape and dynamic appearance. CVIU 106, 31–46 (2007)
13. Fukunaga, K., Olsen, D.R.: An algorithm for finding intrinsic dimensionality of data. In: IEEE Trans. on Computers (1971)
14. Julosky, A., Weiland, S., Heemels, M.: A Bayesian approach to identifications of hybrid systems. IEEE Trans. on Automatic Control 10, 1520–1533 (2005)
15. Levina, E., Bickel, P.J.: Maximum likelihood estimation of intrinsic dimension. In: NIPS (2004)
16. Little, A., Jung, Y.-M., Maggioni, M.: Multiscale Estimation of Intrinsic Dimensionality of Data Dets. AAAI, Menlo Park (2009)
17. Nascimento, J.C., Marques, J.S.: Robust shape tracking with multiple models in ultrasound images. IEEE Trans. on Image Proc. 17(3), 392–406 (2008)
18. Neil, L.: Probabilistic non-linear principal component analysis with gaussian process latent variable models. J. of Machine Learning Research 7, 455–491 (2005)
19. Raginsky, M.: A complexity-regularized quantization approach to nonlinear dimensionality reduction. In: IEEE Int. Symp. on Info. Theory (2005)
20. Raginsky, M., Lazebnik, S.: Estimation of intrinsic dimensionality using high-rate vector quantization. In: NIPS, vol. 18 (2005)
21. Rahimi, A., Recht, B.: Unsupervised regression with applications to nonlinear system identification. In: NIPS, vol. 19 (2007)
22. Rasmussen, C., Williams, C.: Gaussian Processes for Machine Learning. MIT Press, Cambridge (2005)
23. Roweis, S.T., Ghahramani, Z.: An EM algorithm for identification of nonlinear dynamical systems. Kalman Filtering and Neural Networks (2000)
24. Li, R., Tian, T.-P., Sclaroff, S.: Simultaneous Learning of Nonlinear Manifold and Dynamical Models for High-dimensional Time Series. In: ICCV (2007)
25. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326 (2000)
26. Silva, J., Marques, J., Lemos, J.M.: Non-linear dimension reduction with Tangent Bundle Approximation. In: ICASSP (2005)
27. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323 (2000)
28. Urtasun, R., Fleet, D.J., Geiger, A., Popovic, J., Darrell, T., Lawrence, N.: Topologically-Constrained Latent Variable Models. In: ICML (2008)
29. Wang, J.M., Fleet, D.J., Hertzmann, A.: Gaussian process dynamical models. In: NIPS, pp. 1441–1448. MIT Press, Cambridge (2005)
30. Weinberger, K., Saul, L.: Unsupervised learning of image manifolds by semidefinite programming. IJCV 70(1), 77–90 (2006)
31. Weinberger, K., Sha, F., Saul, L.: Learning a kernel matrix for nonlinear dimensionality reduction. In: ICML, pp. 839–846 (2004)