Manifold Learning for Object Tracking With Multiple Nonlinear Models

Jacinto C. Nascimento, *Member, IEEE*, Jorge G. Silva, Jorge S. Marques, and João M. Lemos

***Abstract*— This paper presents a novel manifold learning algorithm for high-dimensional data sets. The scope of the application focuses on the problem of motion tracking in video sequences. The framework presented is twofold. First, it is assumed that the samples are time ordered, providing valuable information that is not presented in the current methodologies. Second, the manifold topology comprises multiple charts, which contrasts to the most current methods that assume one single chart, being overly restrictive. The proposed algorithm, Gaussian process multiple local models (GP–MLM), can deal with arbitrary manifold topology by decomposing the manifold into multiple local models that are probabilistic combined using Gaussian process regression. In addition, the paper presents a multiple filter architecture where standard filtering techniques are integrated within the GP–MLM. The proposed approach exhibits comparable performance of state-of-the-art trackers, namely multiple model data association and deep belief networks, and compares favorably with Gaussian process latent variable models. Extensive experiments are presented using real video data, including a publicly available database of lip sequences and left ventricle ultrasound images, in which the GP–MLM achieves state of the art results.**

***Index Terms*— Manifold learning, multiple dynamics, tangent bundle, tracking.**

I. Introduction

THE proliferation of very large sets of high-dimensional data poses an increasing challenge in terms of both computational complexity and of generalization ability of automated learning procedures. There is a need for algorithms capable of reducing the data dimensionality while preserving relevant information. The scope of application is vast, *e.g.* in computer vision, scene reconstruction from multiple views [1] and modeling dynamic textures in natural images [2]; computer graphics, surface reconstruction from 3D point clouds [3]; multimedia, image retrieval and browsing [4]; bioinformatics, discovering patterns in gene expression data [5], [6].

This paper proposes a new algorithm - GP–MLM, applied to the problem of motion tracking 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. As a particular case, it can also be used for the more narrow purpose of manifold learning without dynamics.

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 [2], [7], [8], but seldom in the context of manifolds, with a few recent exceptions [9]. While this problem is ill-posed (see *e.g.* [2]), 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.

The paper is organized as follows: Section II defines the problem formulation. Section III reviews relevant prior work. Section IV describes GP-MLM in the static case, including: estimation of intrinsic dimensionality and tangent subspaces, using local geometric and velocity information; soft partitioning of the manifold into patches; and a non-parametric, sparse and nonlinear regression procedure learning the charts. Section V extends GP-MLM to the dynamical case, presenting methods for system identification and establishing a multiple filter bank architecture for state estimation that integrates geometry and dynamics. Experimental results are presented in Section VI, including synthetic examples and extended number of real video sequences. It is shown that our method can retrieve the contours provided by established trackers such as the MMDA (*Multiple Model Data Association*) [10] and by the DBN (*Deep Belief Network*) [11], [12], which are used as ground truth, and compares favorably with other manifold methods such as GPLVM [13], [14].

Manuscript received September 4, 2012; revised November 26, 2013; accepted January 6, 2014. Date of publication January 29, 2014; date of current version February 25, 2014. This work was supported in part by the FCT under Projects PEst-OE/EEI/LA0009/2013, HEARTRACK-PTDC/EEACRO/103462/2008, ARGUS - PTDC/EEACRO/098550/2008, and in part by INESC-ID Funds under Contract PEst-OE/EEI/LA0021/2013. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Stefano Tubaro.

J. C. Nascimento and J. S. Marques are with the Instituto de Sistemas e Robótica, Instituto Superior Técnico, Lisboa 1049-001, Portugal (e-mail: jan@isr.ist.utl.pt; jsn@isr.ist.utl.pt).

J. G. Silva is with the SAS Institute, Cary, NC 27513 USA (e-mail: jorge.gomes.da.silva@gmail.com).

J. M. Lemos is with INESC-ID, Instituto Superior Técnico, Lisboa 1049-001, Portugal (e-mail: jlml@inesc.pt).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2014.2303652

A shorter version of this work appears in [15], but in this paper we provide a more comprehensive literature review, explanations, and a much extended experimental evaluation.

II. Problem Statement

The GP-MLM is based on the *tangent bundle* concept. The Tangent bundle [16] 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:

$$T(\mathcal{M}) = \{(\mathbf{y}, \mathbf{v}) : \mathbf{y} \in \mathcal{M}, \mathbf{v} \in T_{\mathbf{y}}(\mathcal{M})\} \tag{1}$$

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 $T(\mathcal{M})$. Any particular function $\mathbf{f}(\mathbf{y}) = \mathbf{v}$, which given a point of application $\mathbf{y}$ provides a velocity vector $\mathbf{v} \in T_{\mathbf{y}}(\mathcal{M})$, is called a *section* of $T(\mathcal{M})$. Another, better known designation is *vector field*.

Now, 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 discrete, vector-valued time series, henceforth called a trajectory, eventually obtained by sampling a continuous process. Let $\mathcal{Y} \equiv \{\mathbf{y}_{0:T_l-1}, l = 1, \ldots, L\}$ be a set of L trajectories. It is assumed that, apart from observation noise, 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 an atlas with p charts, where p is unknown, and each $\mathcal{X}_i$ corresponds to one of the charts.

The goal is to estimate $\mathcal{M}$ and identify the dynamics in the lower dimensional coordinates given by the charts of $\mathcal{M}$. It is assumed that the trajectories are generated by one or more discrete state space models of the form:

$$\mathbf{x}_{t,i} = \mathbf{f}_i(\mathbf{x}_{t-1,i}) + \boldsymbol{\omega}_{t,i} \tag{2}$$

$$\mathbf{y}_{t,i} = \mathbf{h}_i(\mathbf{x}_{t,i}) + \boldsymbol{\nu}_{t,i} \tag{3}$$

where $\boldsymbol{\omega}_{t,i}$ and $\boldsymbol{\nu}_{t,i}$ are random variables with unknown distributions. Here, $\mathbf{h}_i$ is the i^{th} parametrization being used around $\mathbf{y}_t$, that therefore can be obtained as the inverse of the appropriate chart $\mathbf{g}_i$ found in the manifold learning step, and $\mathbf{f}_i$ is a vector field that defines the dynamics. Note that, in the limit as the sampling frequency increases, $\mathbf{f}_i(\mathbf{g}_i(\mathbf{y}))$ must belong to the tangent space of $\mathcal{M}$ at $\mathbf{y}$, $T_{\mathbf{y}}(\mathcal{M})$, while $\mathbf{x}_i$ must belong to the parametric domain of $\mathbf{h}_i$ and, as such, $\mathbf{x}_i \in \mathbb{R}^n$. Note also that, due to observation noise and model uncertainty, the observations will not, in general, lie exactly on the manifold. In summary, the problem can be formulated as:

- Given $\mathcal{Y}$, find an approximation of $T(\mathcal{M})$ or, more precisely, find the intrinsic dimension n, build a partition into p patches (p must also be estimated), estimate the p charts $\mathbf{g}_i$ and their domains $\mathcal{P}_i$, obtain the lower dimensional trajectories $\mathcal{X}_i$;
- Given $\mathcal{Y}$ and the estimate of $T(\mathcal{M})$, estimate dynamical functions $\mathbf{f}_i$;
- Given a new observation $\mathbf{y}$, and assuming each patch $\mathcal{P}_i$ is defined by parameters $\boldsymbol{\theta}_i$, estimate the conditional probabilities $P(\boldsymbol{\theta}_i|\mathbf{y})$ so that the local models can be combined through a partition of unity.

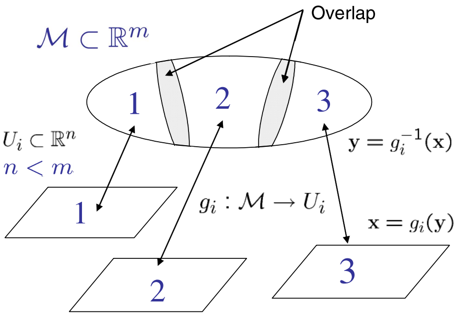

Overlap
$\mathcal{M} \subset \mathbb{R}^m$
1
2
3
$U_i \subset \mathbb{R}^n$
$n < m$
$\mathbf{y} = g_i^{-1}(\mathbf{x})$
$g_i : \mathcal{M} \to U_i$
$\mathbf{x} = g_i(\mathbf{y})$
1
2
3

Fig. 1. Manifold $\mathcal{M}$ associated with a collection of one-to-one continuous invertible functions $\mathbf{g}_i : \mathcal{P}_i \to \mathcal{U}_i$ (the charts) where $\mathcal{P}_i \subset \mathcal{M}$ are the patches.

If this can be done, then the geometrical and dynamical information are captured and most of the stated objectives can be satisfied. Of course, this is an ill-posed problem. Assumptions such as the smoothness of $\mathbf{f}$ and $\mathbf{g}$, the reconstructibility of the state $\mathbf{x}$ from the observations and restrictions to the distributions of $\boldsymbol{\omega}$ and/or $\boldsymbol{\nu}$ must be considered in order to make the problem tractable Fig. 1 shows an illustration.

III. Prior Work

A. Spectral and Probabilistic Methods—Relationship With Kernels

Several manifold learning algorithms have emerged in recent years, mainly concerned with nonlinear dimensionality reduction, which is closely related to the problem of feature extraction [17]. Other work in the field includes, on one hand, probabilistic methods such as the Generative Topographic Mapping (GTM) [18], Gaussian process related algorithms, such as Gaussian Process Latent Variable Models (GP-LVM) [13], [14] and Gaussian Process Dynamical Models (GPDM) [7]; on the other hand, graph spectral methods such as ISOMAP [19], Locally Linear Embedding (LLE) [20], Laplacian [21] and Hessian Eigenmaps [22], as well as Semi-Definite Embedding [23]–[25].

Spectral methods are closely related to spectral clustering [26], which is a technique for clustering and graph partitioning. The name "spectral" stems from the fact that these methods compute the eigen-spectrum (eigenvalues and corresponding orthonormal basis of eigenvectors) of an appropriately constructed square matrix of either similarities of dissimilarities between all training points. This matrix is, in turn, obtained from a graph over the data samples. The details of constructing the graph and, subsequently, the (dis)similarity matrix depend on the specific algorithm. Typically, and depending on the method, either the top or the bottom eigenvectors give the new coordinates of the training points in low dimensional space. ISOMAP, LLE, Laplacian/Hessian Eigenmaps and SDE are spectral methods.

Probabilistic methods, on the other hand, focus on optimizing quantities, such as marginal likelihoods, that depend on probability density functions of the data, $\mathcal{Y}$ given the low dimensional representations, $\mathcal{X}$ and the mapping between them, which usually depends on parameters $\boldsymbol{\theta}$. Examples include GTM and GPLVM. Within this framework, it is possible to start from the density $p(\mathcal{Y}|\mathcal{X}, \boldsymbol{\theta})$ and do one of two things: integrating out $\mathcal{X}$ and optimizing $p(\mathcal{Y}|\boldsymbol{\theta})$ as a function of $\boldsymbol{\theta}$, like GTM, or integrating out $\boldsymbol{\theta}$ and optimizing $p(\mathcal{Y}|\mathcal{X})$ with respect to $\mathcal{X}$, which is done in GP-LVM.

An alternative classification of existing algorithms as either *local* or *global* is also possible, as in [27]. Global methods attempt to parameterize the manifold in a way that preserves distances (measured on the manifold) between all points, nearby or faraway. The definition of "nearby" and "faraway" varies from one specific algorithm to another. For example, ISOMAP is a global method. On the other hand, some other methods, such as LLE, focus mostly on preserving the distances between nearby points - these are local methods. Such algorithms rely on the divide-and-conquer approach, turning one difficult problem into many easier ones. It is, however, more common to adopt the spectral/probabilistic division.

One common trend to both types of methods is that they can be thought of as kernel methods. Spectral algorithms construct symmetrical similarity/dissimilarity matrices that can be transformed into kernels, while probabilistic methods (namely those based on Gaussian processes) assume *a priori* covariance matrices, that are valid Mercer kernels. In other words, spectral methods can be used to learn kernel matrices, that can then be used either for performing kernel PCA (KPCA) [28], as previously noted in [28], or else as covariance matrices by probabilistic methods, *e.g.* Gaussian processes, often with better results than, *e.g.*, the standard Gaussian kernel. A thorough reinterpretation of graph spectral methods as kernel PCA can be found in [29].

B. Charting and Topology

Most existing manifold learning methods assume that the manifold can be modeled using a single coordinate patch, *i.e.* only one mapping $\mathcal{Y} \longrightarrow \mathcal{X}$, an assumption that fails for manifolds with topologies as simple as a sphere, that must be covered by more than one chart, *i.e.* an *atlas*. Furthermore, graph spectral methods usually do not provide a continuous description - they yield a point-to-point correspondence between observations in high dimensional space and their lower dimensional representations, which means that additional processing is required if one intends to perform an out-of-sample extension. Only a few methods attempt to deal with multiple charts. In [9], [30], and [31], 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 [32], a mapping from high-dimensional observations to latent states is estimated, but not the inverse. In [33], a manifold tracking method is used for learning nonlinear motion manifolds in the recovery of 3D body pose, although it 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 [7], [34]. However, [7] assumes one single chart and a priori fixed latent dimensionality, while [34] encourages certain topologies in a top-down manner, based on prior knowledge. 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.

Regarding estimation of the number p of local models, in [30] multiple linear hyperplanes are combined – one for each observation, which can be problematic. A more parsimonious model, based on a Gaussian mixture model learned through Expectation–Maximization (EM), is proposed in [35], where the number of charts is estimated and an asymptotic analysis is also presented for the Gaussian mixture case. More recently, [31] uses a Dirichlet process model in order to nonparametrically learn the number of components in a mixture of linear factor analyzers. We use a soft-clustering procedure that employs *subspace angles* [36] as an intra-cluster similarity criterion.

C. Intrinsic Dimension

Estimating the intrinsic dimension n remains a challenge. On one hand, it is an inherently local problem, in the sense that one should attempt to find n such that $\mathbb{R}^n$ is a good approximation of the data in a neighborhood of each $\mathbf{y}$. On the other hand, a global consensus must be found regarding the value of n. Due to noise, irregular sampling and border effects, local estimates may be wrong. The most common method [37], [38] for estimating n is based on local PCA, relying on a threshold to select significant eigenvalues λ_i of the local covariance matrix

$$\mathbf{S}_{\mathbf{y}_j} = \frac{1}{|\mathcal{B}_{\mathbf{y}_j,\epsilon}| - 1} \sum_{\mathbf{y}_k \in \mathcal{B}_{\mathbf{y}_j,\epsilon}} (\mathbf{y}_k - \boldsymbol{\mu}_{\mathcal{B}_{\mathbf{y}_j,\epsilon}})(\mathbf{y}_k - \boldsymbol{\mu}_{\mathcal{B}_{\mathbf{y}_j,\epsilon}})^T, \quad (4)$$

in an ϵ-radius neighborhood $\mathcal{B}_{\mathbf{y}_j,\epsilon}$ around $\mathbf{y}_j$. The mean of the neighborhood is $\boldsymbol{\mu}_{\mathcal{B}_{\mathbf{y}_j,\epsilon}}$, which does not necessarily coincide with $\mathbf{y}_j$, since this may be noisy. The cardinality $|\mathcal{B}_{\mathbf{y}_j,\epsilon}|$ is the number of points in the neighborhood. The number of significant λ_i is an estimate of n. Naturally, the unknown scale parameter ϵ strongly influences the method. Indeed, this is the case with all currently available intrinsic dimension estimation procedures.

Other algorithms are based on different approaches, such as vector quantization, maximum likelihood under a Poisson model, finding the correlation dimension or the ratio of growth of the number of nearest neighbors versus the neighborhood size (see [39], [40] and the references therein). With either type of algorithms, for high levels of noise, and particularly for a high ($> 5 - 6$) intrinsic dimensionality, it is unlikely that the estimate will be very accurate or stable. For example, applying the eigenvalue algorithm described in [37] to gene expression data [6], the estimated intrinsic dimensionality varies in the interval 15-25, which indicates high variance. Bias is also a problem with most methods, as pointed out in [40]. Hence, dimensionality estimation continues to be a challenging problem, although some promising advances have recently been made using multiscale approaches [41].

D. Landmark Selection

The complexity of most manifold learning algorithms is, in general, dependent not only on the dimensionality but also on the number of observations, N. An important example is the ISOMAP, where the computational cost is $O(N^2)$, which has motivated the L-ISOMAP variant [27] that uses a randomly chosen subset of the points as *landmarks* (L is for Landmark). Besides the issue of complexity, sparse models also tend to have better generalization capability. Therefore, it can be advantageous to employ sparse regression techniques when estimating the charts.

E. The Dynamical Case

Manifold learning for the dynamical case has not yet received much attention, although some related work does exist, namely [8], that presents an EM algorithm for simultaneous estimation of the hidden states and the unknown dynamics, based on an Extended Kalman Filtering (EKF) formulation and Radial Basis Function (RBF) fitting. The problem is not, however, formulated in terms of manifolds, and no consideration is given to issues of dimensionality reduction. A similar approach, related to the problem of modeling dynamical textures in Computer Vision, is given in [2]. Here, a linear autoregressive dynamical model is used in low-dimensional space, after performing dimensionality reduction. The hidden states and the parameters of the AR model are estimated, once again, using EM. The differences to [8] are: (i) dimensionality reduction is explicitly considered, albeit limited to standard techniques such as PCA and wavelet decomposition - the problem is also not formulated in terms of manifold learning; (ii) more stringent conditions are imposed on the dynamics, which are not only restricted to be linear but also to follow a controllable canonical form. The only way that nonlinear systems can be handled by this technique is through unloading any nonlinearity to the dimensionality reduction step. For the purpose of modeling dynamical textures, this still yields good results, since it is shown that the probability density of the state perturbation is more critical for realistic results than the dynamic update function itself. It is also shown that this density should be non-Gaussian. An exponential family is used in this case.

In control literature, [42] have proposed a procedure designed for hybrid systems, using a piecewise ARX (AutoRegressive with eXogenous inputs) model formulation. For each region of the input space, an ARX model must be estimated. Estimation of the unknown discrete subsystem evolution is cast as a classification problem, for which a Bayesian-inspired algorithm is proposed. The relevant probability density functions are approximated by using particle filtering. However, the number of regions is assumed known. A hard partitioning is used, assuming linear separability of the regions (although by using a weighing function the case of non-linearly separable regions is addressed). Also, like [2], within each region the ARX formulation is linear. Dimensionality reduction is not considered. Nevertheless, there are interesting parallels between a trajectory crossing multiple patches of a manifold and a system switching between multiple modes in a hybrid system formulation.

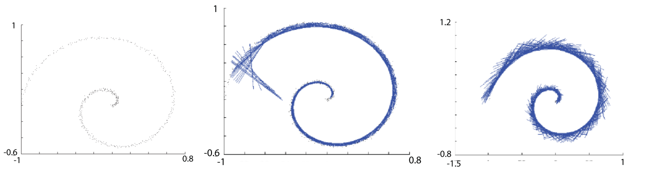


Fig. 2. Using robust estimation to improve the tangent subspaces. Left to right: noisy spiral data; least squares estimates; RANSAC estimates.

IV. GP-MLM Algorithm

A. Intrinsic Dimension

In the spirit of [37], GP-MLM addresses the problem of dimensionality estimation by automatically finding the "knee" of the eigenvalues $\lambda_1, \ldots, \lambda_m$ 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

$$\widehat{n} = \text{median}_{j=1,\ldots,N}(\widehat{n}_j). \tag{5}$$

The median – which is more robust than the mean – for all eigenvalue plots is therefore used as the estimate of n. 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. It is possible to use the first differences

$$\Delta_{\mathbf{y}_t} = \mathbf{y}_t - \mathbf{y}_{t-1}, \tag{6}$$

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

$$\mu_{\mathcal{B}_{\mathbf{y}_j,\epsilon}} = \frac{1}{|\mathcal{B}_{\mathbf{y}_j,\epsilon}|} \sum_{\mathbf{y}_k \in \mathcal{B}_{\mathbf{y}_j,\epsilon}} \mathbf{y}_k. \tag{7}$$

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).

The quality of the tangent subspace estimates is crucial to the success of GP-MLM. One of the main factors affecting the reliability of the tangent subspaces, in the presence of noise, is the neighborhood size. A too small radius will lead to poor tangent subspace estimates due to an insufficient amount of points, while a too large radius will, on the other hand, lead to the inclusion of points from different folds, due to curvature. To gain some tolerance to this effect, it is possible to use a robust statistics approach. In Fig. 2, an example involving a 1D spiral embedded in $\mathbb{R}^2$ is shown. The noise standard deviation is $\sigma = 0.1$ and the neighborhood size is set to $k = 60$, *i.e.* the 60 nearest neighbors are used. With unmodified least squares, near the outer arm of the spiral, it can be seen that the tangent subspaces are affected by points from the inner fold. On the other hand, with robust estimation – in this case the RANSAC algorithm [43] – this phenomenon disappears. Other possible robust algorithms include MINPRAN or NAPSAC [44].

B. Charts

A manifold with arbitrary topology requires, in general, more than one chart, *i.e.* an atlas. 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. This section describes how to partition $\mathcal{M}$ into overlapping patches $\mathcal{P}_1, \ldots, \mathcal{P}_p$; how to find p corresponding tangent hyperplanes; and how to estimate mappings back and forth between the patches and the hyperplanes. First, it is important to find a partition that facilitates subsequent estimation of the mappings. For that, it would be convenient if a simple projection operation could allow a one-to-one mapping between each hyperplane and the corresponding manifold patch. Fortunately, this can be ensured by controlling the curvature within each region for which purpose the concept of *principal angles* is used. The q principal angles between subspaces spanned by the columns $\mathbf{u}$ and $\mathbf{v}$ of some generic matrices $\mathbf{A}$ and $\mathbf{B}$, respectively, are defined, as in [36], by

$$\cos\theta_k = \frac{|\mathbf{u}_k\mathbf{A}^T\mathbf{B}\mathbf{v}_k|}{\|\mathbf{A}\mathbf{u}_k\|\,\|\mathbf{B}\mathbf{v}_k\|} \tag{8}$$

with $k = 1, \ldots, q = \dim(\mathbf{A}) = \dim(\mathbf{B})$ and, for $k > 1$, subject to recursively defined constraints for $\mathbf{u}_k$ and $\mathbf{v}_k$, the k-th columns of $\mathbf{A}$ and $\mathbf{B}$:

$$\mathbf{u}_i^T\mathbf{A}^T\mathbf{A}\mathbf{v} = \mathbf{v}_i^T\mathbf{B}^T\mathbf{B}\mathbf{v}_k \tag{9}$$

for $i = 1, 2, \ldots, k-1$. For the manifold partitioning problem, the matrices $\mathbf{A}$ and $\mathbf{B}$ correspond to the matrices $\mathbf{V}_i$ and $\mathbf{V}_j$ of column eigenvectors found by local PCA on neighborhoods i and j. The intended one-to-one behavior can be achieved by not allowing the maximum principal angle between tangent subspaces (obtained according to the procedure in IV-A) to vary more than a set threshold τ. The exact value of τ is not critical, as long as it is below $\frac{\pi}{2}$. An efficient algorithm for the computation of the principal angles, that also circumvents numerical problems for small angles, is given in [36]. The maximum of θ_k is used as GP-MLM's intra-patch similarity measure.

Patches are found by an agglomerative soft clustering procedure, *i.e.* region growing. Each patch grows by appending all neighboring (within an ϵ radius) points where the normal subspace does not deviate, in maximum principal angle, more than a set threshold from the tangent subspace at the initial seed. Any specific data point may belong to more than one patch. The pseudo-code in Algorithm 1 illustrates the procedure.

Algorithm 1 Region Growing

```
$i \leftarrow 0$
while $\mathcal{M}$ not covered do
    $i \leftarrow i + 1$
    Start new patch $\mathcal{P}_i$
    $\mathbf{y}_0 \leftarrow$ random seed chosen among points not attributed to any patch
    $\mathbf{V}_0 \leftarrow$ tangent subspace basis at $\mathbf{y}_0$ found by PCA in $\mathcal{B}_{\mathbf{y},\epsilon}$
    while not all $N$ points visited do
        $\mathbf{d} \leftarrow$ distances between all unattributed points and all points in $\mathcal{P}_i$
        $\mathbf{y}_* \leftarrow$ choose unattributed point with minimum $\mathbf{D}$
        $\mathbf{V}_* \leftarrow$ tangent subspace at $\mathbf{y}_*$
        $\theta_1, \ldots, \theta_n \leftarrow$ principal angles between $\mathbf{V}_0$ and $\mathbf{V}_*$
        if $\max_{k=1,\ldots,n} \theta_k < \tau$ and $\min \mathbf{d} < \epsilon$ then
            append $\mathbf{y}_*$ to $\mathcal{P}_i$
        end if
    end while
end while
```

The final result is a covering of $\mathcal{M}$ by a finite number, p, of overlapping patches. Within each patch, the tangent subspaces do not deviate more than τ from the tangent subspace at the seed, 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 best hyperplane for the entire patch, in a least squares sense, is spanned by the n largest eigenvectors returned by patch-wide PCA, where n is estimated by the procedure in IV-A. Each patch is thus associated to an hyperplane, and 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.*

$$\mathbf{S}_{\mathcal{P}_i} = \frac{1}{|\mathcal{P}_i| - 1}\sum_{\mathbf{y}_k \in \mathcal{P}_i}(\mathbf{y}_k - \boldsymbol{\mu}_{\mathcal{P}_i})(\mathbf{y}_k - \boldsymbol{\mu}_{\mathcal{P}_i})^T, \tag{10}$$

then, by performing the eigendecomposition

$$\mathbf{S}_{\mathcal{P}_i} = \mathbf{V}_{\mathcal{P}_i}\mathbf{D}_{\mathcal{P}_i}\mathbf{V}_{\mathcal{P}_i}^T, \tag{11}$$

where $\mathbf{V}_{\mathcal{P}_i}$ is the matrix whose columns are the eigenvectors of $\mathbf{S}_{\mathcal{P}_i}$ and $\mathbf{D}_{\mathcal{P}_i} = \operatorname{diag}(\lambda_1, \ldots, \lambda_m)$, 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 remark is that GP-MLM 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 $O(Np)$.

C. 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 $\mathbf{y} = [y_1 \dots y_m]^T$, where y_j, $j = 1, \dots, m$ refers to the j^{th} coordinate. Recall that a matrix $\mathbf{V}_{\mathcal{P}_i}$ of eigenvectors and a mean vector $\boldsymbol{\mu}_{\mathcal{P}_i}$ are available. 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

$$\tilde{\mathbf{x}}_i = \mathbf{V}_{\mathcal{P}_i}^T (\mathbf{y} - \boldsymbol{\mu}_{\mathcal{P}_i}) \tag{12}$$

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

$$\mathbf{x}_i = [\tilde{x}_{i,1} \dots \tilde{x}_{i,n}]^T = \mathbf{g}_i(\mathbf{y}) \tag{13}$$

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 parameterization $\mathbf{h}_i(\mathbf{x}_i)$ follows the expression

$$\mathbf{h}_i(\mathbf{x}_i) = \mathbf{V}_{\mathcal{P}_i} \left[x_1 \dots x_n \;\; \tilde{\mathbf{h}}_i(\mathbf{x}_i) \right]^\top + \boldsymbol{\mu}_i \tag{14}$$

in which $\tilde{\mathbf{h}}_i$ must be estimated. The remaining $m - n$ components of $\tilde{\mathbf{x}}_i$ could have been set to zero, which would yield a piecewise linear approximation of $\mathcal{M}$. Since, however, they are approximated by $\tilde{\mathbf{h}}_i(\mathbf{x}_i)$, the curvature, and thus the nonlinear character of the manifold, is preserved. In the i^{th} local coordinates, the parameterization is

$$\mathbf{x}_i \to [\mathbf{x}_i \quad \tilde{\mathbf{h}}_i(\mathbf{x}_i)]^T. \tag{15}$$

It is now necessary to estimate $\tilde{\mathbf{h}}$. Among the large variety of existing nonlinear regression techniques, Gaussian process regression offers the feature of allowing the analytical computation of posterior probability densities in data space. 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 functions $\tilde{\mathbf{h}}_i$ in the model

$$\tilde{x}_{k,i}^{(m)} \approx \tilde{\mathbf{h}}_i(\mathbf{x}_{k,i}), \tag{16}$$

where $k = 1, \dots, |\mathcal{P}_i|$. It is assumed that $\tilde{h}_i$ admits a Gaussian process prior [45]

$$\tilde{h}_i \sim \mathcal{GP}(\mathbf{0}, k(\cdot, \cdot)) \tag{17}$$

with mean function zero and covariance function $k(\cdot, \cdot)$.

When conditioned on the data $\tilde{\mathbf{z}} = \{\tilde{x}_{k,i}^{(m)}\}_{k=1:|\mathcal{P}_i|}$ and $\mathbf{X} = \{\mathbf{x}_{k,i}\}_{k=1:|\mathcal{P}_i|}$, the posterior for $\tilde{h}_i$ is

$$\begin{aligned} p(\tilde{h}_i(\mathbf{x}_{k,i})|\mathbf{z}, \mathbf{X}) = \mathcal{N}(&\mathbf{K}(\mathbf{x}_{k,i}, \mathbf{X})\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{z}, \\ &\mathbf{K}(\mathbf{x}_{k,i}, \mathbf{x}_{k,i}) - \mathbf{K}(\mathbf{x}_{k,i}, \mathbf{X})\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{K}(\mathbf{X}, \mathbf{x}_{k,i})). \end{aligned} \tag{18}$$

Note that $\mathbf{z}$ is a vector of size $|\mathcal{P}_i|$, $\mathbf{X}$ is a matrix of size $n \times |\mathcal{P}_i|$ and $K(\mathbf{A}, \mathbf{B})$ denotes the matrix resulting from evaluating the covariance function $k(\cdot, \cdot)$ between rows of $\mathbf{A}$ and $\mathbf{B}$.

For the covariance function $k(\cdot, \cdot)$, we choose the Gaussian radial basis function (RBF) defined by

$$k(\mathbf{x}_k, \mathbf{x}_l) = \theta_1 \exp(-\frac{1}{2\theta_2} \|\mathbf{x}_k - \mathbf{x}_l\|^2) + \delta_{kl}\theta_3 \tag{19}$$

and optimize the hyperparameters $\theta = (\theta_1, \theta_2, \theta_3)$ by maximizing the marginal likelihood, as proposed in [45].

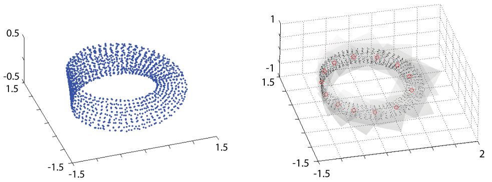


Fig. 3. Left: points on a Möbius strip. Right: patch seeds (red circles) and charts (grey surfaces). GP-MLM can handle non-orientable manifolds.

As an example of the results from partitioning and regression see Fig. 3, which shows a Möbius strip, a non-orientable manifold, *i.e.*, a manifold where there is no uniquely defined normal vector field. This fact poses no problem to GP-MLM.

D. Conditional Densities from Gaussian Process Regression

An important issue that must be faced is that of estimating the conditional densities from Gaussian process regression. Since the latent variable is $\mathbf{x} \in \mathbb{R}^n$ and the function being estimated is $\widetilde{\mathbf{h}}(\mathbf{x}) \in \mathbb{R}^{m-n}$, it can be seen that the pairs for regression in GP-MLM are not $(\mathbf{x}, \mathbf{y})$, but rather $(\mathbf{x}, \widetilde{\mathbf{h}}(\mathbf{x}))$. In consequence, the conditional density returned from Gaussian process regression is in fact $p(\widetilde{\mathbf{h}}(\mathbf{x})|\mathbf{x}, \theta_i)$, which has Lebesgue measure in $\mathbb{R}^{m-n}$, and not in $\mathbb{R}^m$. That is, only the density of the *orthogonal* component of $\mathbf{y}$, conditioned on the tangent component, and on the patch, is returned in local coordinates. This is equivalent to $p(\widetilde{\mathbf{h}}(\mathbf{x})|\mathbf{x}, \theta_i)$.

In order to tackle with this issue and compute posterior probabilities for the patches, *i.e.* $P(\theta_i|\mathbf{y})$, the likelihood $p(\mathbf{y}|\theta_i)$ can be expressed in a way that accounts for the tangent and orthogonal components as follows

$$p(\mathbf{y}|\theta_i) = p\left(\left[\begin{matrix} \mathbf{x} \\ \widetilde{\mathbf{h}}(\mathbf{x}) \end{matrix} \right] \middle| \mathbf{x}, \theta_i \right) p(\mathbf{x}|\theta_i) \tag{20}$$

However, the term $p(\mathbf{x}|\theta_i)$ is not possible to compute. Still, an estimate of how likely some $\mathbf{x}$ is, given the patch $\mathcal{P}$, can be obtained. Recall that, the pdf $p(\widetilde{\mathbf{h}}(\mathbf{x})|\mathbf{x})$ is highly peaked in regions where there are many training points, and flat in empty regions. If we take the peak of this pdf, then $q(\mathbf{x}, \theta_i) = \max p(\widetilde{\mathbf{h}}(\mathbf{x})|\mathbf{x}, \theta_i)$ can be used as a surrogate for $p(\mathbf{x}|\theta_i)$. While $q(\mathbf{x}, \theta_i)$ is not a pdf (*i.e.*, not normalized), it will be assumed that

$$q(\mathbf{x}, \theta_i) \propto p(\mathbf{x}|\theta_i) \tag{21}$$

and the desired posterior can be approximated by

$$\begin{aligned} P(\theta_i|\mathbf{y}) &= \frac{p(\mathbf{y}|\theta_i)P(\theta_i)}{\sum_{j=1}^{P} p(\mathbf{y}|\theta_j)P(\theta_j)} \\ &\approx \frac{p(\widetilde{\mathbf{h}}|\mathbf{x}, \theta_i)q(\mathbf{x}, \theta_i)P(\theta_i)}{\sum_{j=1}^{P} p(\widetilde{\mathbf{h}}|\theta_j)q(\mathbf{x}, \theta_j)P(\theta_j)}. \end{aligned} \tag{22}$$

V. Dynamical Learning

This section extends GP–MLM to deal with the simultaneous identification of the data manifold and dynamics in the lower dimensional space. We take advantage of the manifold model in two ways: (i) although the charting process does not necessarily yield the state vector, it provides low dimensional observables, making it easier to identify the dynamics and estimate the true state; (ii) the filtered trajectories are restricted to remain on the estimated manifold, making the tracking process more robust against gross outliers in observation space.

Estimating the dynamics $\mathbf{f}$, the observation model $\mathbf{h}$ and the state vector $\mathbf{x}_t$ from the data $\mathbf{y}_t$ are ill-posed inverse problems. Indeed, there is a large number of combinations that can "explain" the data sequence. Our approach is to start from the state model given by (2,3) and assume that, in the observation equation, $\mathbf{h}$ is fixed and given by the manifold model found by GP–MLM. We adopt a two-stage procedure:

- Identification of the dynamics $\mathbf{f}$;
- Estimation of the state $\mathbf{x}_t$, given all the information up to t, *i.e.*, $\mathbf{y}_{0:t}$

The first subproblem is called *system identification* in control literature, and is solved offline, whilst the second is the *state estimation* that we now explain.

A. 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 $\{\mathbf{x}_{t,j}\}_{t=0:T_j-1}$, that 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

$$\mathbf{x}_{t,i} = \mathbf{f}_i(\mathbf{x}_{t-1,i}) + \boldsymbol{\omega}_{t,i}. \tag{23}$$

In the case when the dynamics are linear, and dropping the i subscript, (23) 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 $(\mathbf{x}_{t-1}, \mathbf{x}_t)$, that 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 (19).

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

B. State Estimation

The second subproblem is the causal estimation of the state, or *filtering*. One should incorporate all the available information, namely from the previous time steps, into a probabilistic reasoning. In this paper, we include experimental evaluations using Kalman filtering (linear and Gaussian case) and particle filtering (nonlinear and non Gaussian case) using sequential importance resampling [47].

Note that GP–MLM is a multiple-model framework in the sense that one filter per patch is used. Moreover, we use different dynamics, different observation models as well as different coordinate systems. This means that procedures for transferring the state between patches and for combining the local estimates are required. Fig. 4 illustrates how this is accomplished.

To do this, 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

$$P(\theta_i|\mathbf{x}, \mathcal{Y}_t) \propto p(\mathbf{x}|\theta_i, \mathcal{Y}_t). \tag{24}$$

The mixture weights provides by the block $\mathbf{G}$ take the different dynamics into account. Two different strategies are tested: a "winner-take-all" rule, where only the output of the model with the highest posterior probability is used, and a "blending" rule, where the weighted average using all models is computed.

Notice that the term blending also arises in the state estimate tracking formulations. For instance the JPDAF [46] uses notion of the combined innovation, computed over the several detected measurements detected at a given time instant as the weighted sum of the individual innovations, that is, this formulation weights the influence of the various candidate measurements.

VI. Result

This section presents an experimental evaluation of the proposed method in several data sequences. We consider four main situations: (i) an example containing a sequence of a bouncing ball; (ii) two ultrasound sequences of the left ventricle (LV); (iii) 10 lip sequences including speaking and singing cases. In the LV and lip (speaking and singing cases), we use two state-of-the-art trackers (the MMDA and DBN trackers)[1] as a way to automatically provide the ground-truth. Also, we provide an objective evaluation for all data sequences.

A. Bouncing Ball Sequence

A sequence of a bouncing ball (termed here as *BouncingBall*) containing 180 frames is used. To represent the boundary of the ball, 20 contour points are used, which would require a total 3600 manual clicks, if we were to obtain ground truth by hand. In this example, the MMDA is used to automatically produce the ground truth. The state vector $\mathbf{x} \in \Re^D$, with $D = 2$, meaning that it contains the rigid parameters for the translation.

Fig. 5(a) shows the overlapped ball positions provided by the MMDA throughout the ball motion. The ball motion is as follows. First, the ball moves in a *up-down* motion [Fig. 5(b) and (c)], followed by (*right-left*) dynamics [Fig. 5(d) and (e)]. In this example, the motion is fully

[1] We refer to the interested reader to [10]–[12] for a more in depth review.

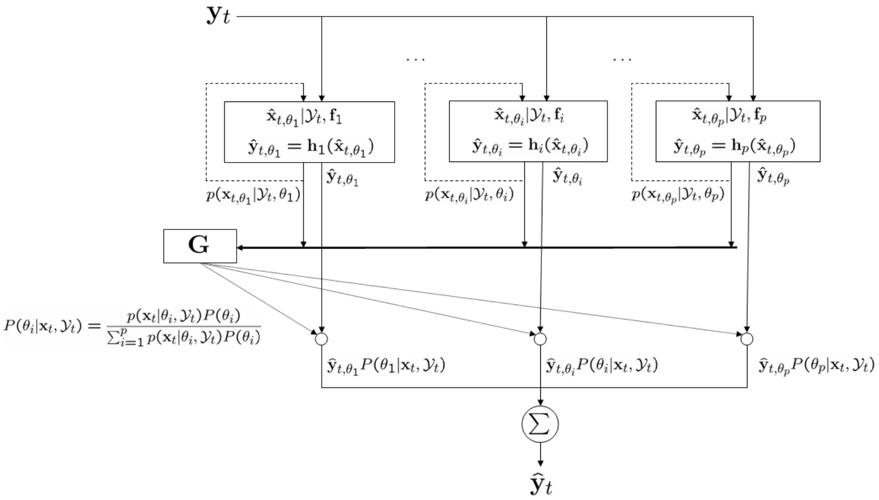

$\mathbf{y}_t$
$\hat{\mathbf{x}}_{t,\theta_1}|\mathcal{Y}_t, \mathbf{f}_1$
$\hat{\mathbf{y}}_{t,\theta_1} = \mathbf{h}_1(\hat{\mathbf{x}}_{t,\theta_1})$
$\hat{\mathbf{x}}_{t,\theta_i}|\mathcal{Y}_t, \mathbf{f}_i$
$\hat{\mathbf{y}}_{t,\theta_i} = \mathbf{h}_i(\hat{\mathbf{x}}_{t,\theta_i})$
$\hat{\mathbf{x}}_{t,\theta_p}|\mathcal{Y}_t, \mathbf{f}_p$
$\hat{\mathbf{y}}_{t,\theta_p} = \mathbf{h}_p(\hat{\mathbf{x}}_{t,\theta_p})$
$p(\mathbf{x}_{t,\theta_1}|\mathcal{Y}_t, \theta_1)$
$\hat{\mathbf{y}}_{t,\theta_1}$
$p(\mathbf{x}_{t,\theta_i}|\mathcal{Y}_t, \theta_i)$
$\hat{\mathbf{y}}_{t,\theta_i}$
$p(\mathbf{x}_{t,\theta_p}|\mathcal{Y}_t, \theta_p)$
$\hat{\mathbf{y}}_{t,\theta_p}$
G
$P(\theta_i|\mathbf{x}_t, \mathcal{Y}_t) = \frac{p(\mathbf{x}_t|\theta_i, \mathcal{Y}_t)P(\theta_i)}{\sum_{i=1}^{p} p(\mathbf{x}_t|\theta_i, \mathcal{Y}_t)P(\theta_i)}$
$\hat{\mathbf{y}}_{t,\theta_1} P(\theta_1|\mathbf{x}_t, \mathcal{Y}_t)$
$\hat{\mathbf{y}}_{t,\theta_i} P(\theta_i|\mathbf{x}_t, \mathcal{Y}_t)$
$\hat{\mathbf{y}}_{t,\theta_p} P(\theta_p|\mathbf{x}_t, \mathcal{Y}_t)$
Σ
$\hat{\mathbf{y}}_t$

Fig. 4. Block diagram of the mixture architecture for combining the local dynamic models.

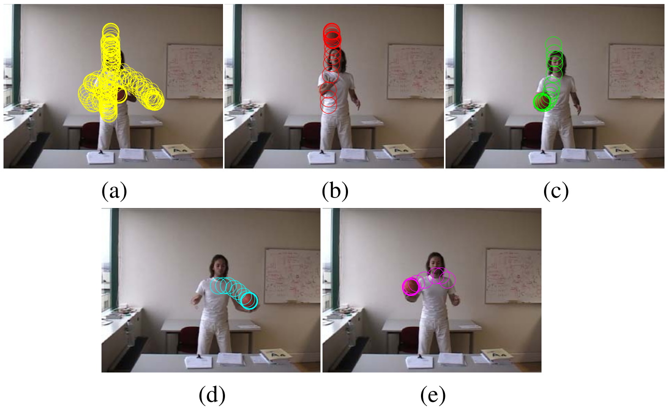


Fig. 5. MMDA tracked results for the sequence *BouncingBall*: (a) overlapped ball positions throughout the sequence, (b) up and (c) down motion dynamics; left (d) right (e) motion dynamics. The ball was thrown three times in (b), (c), and four times in (d), (e).

described by four dynamics. Here, the ball was thrown three consecutive times in the *up-down* motions, followed by another four times in the *left-right* motions.

From the successive ball positions shown in Fig. 5, it becomes clear that the projection onto the image plane of the ball is not a manifold, because the trajectories of the ball self-intersect. As such, one should avoid the use of simple $x-y$ coordinates of the object. Instead, if one takes not only the positions but also the velocities, more specifically, instead of representing $\mathbf{y} = [x, y]^T$, one considers the observation vector $\mathbf{y} = [\dot{x}, \dot{y}, x, y]^T$, that is, the vertical velocity and vertical position, then the self intersection disappears and the closed curve becomes once again a manifold.

The resulting 4D vectors are then used as input for the GP-MLM algorithm, and a manifold model is then estimated. In this example, the algorithm found the intrinsic dimension of one with 10 estimated patches.

Table I shows the error for three identification strategies, namely, 1st and 2nd order linear dynamical model and GP non-linear dynamical model. It is shown that the GP 1st order exhibits the best performance in both examples.

TABLE I

MEAN SQUARE ERROR (MSE) ON *Boucing Ball* SEQUENCE USING THREE DIFFERENT STRATEGIES: LINEAR 1ST AND 2ND ORDER MODELS AND A NON-LINEAR GP MODEL

	Case # 2 MSE		
Patch #	Linear 1^{st} order	Linear 2^{nd} order	GP 1^{st} order
1	0.0031	0.0849	**0.0003**
2	0.2857	0.0400	**0.0010**
3	**0.0005**	0.0054	0.0007
4	0.0016	**0.0012**	0.0064
5	0.0181	0.0046	**0.0001**
6	0.0005	0.0245	**0.0003**
7	0.0001	0.0346	**0.0000**
8	0.0068	0.0016	**0.0000**
9	0.0072	0.0010	**0.0000**
10	0.0010	0.0004	**0.0000**

The next section provides an extended comparison between the Kalman Filter (KF) and particle filter (PF) using the “blending” (BLD) and “winner-take-all” (WTA) rules.

B. Ultrasound Sequences of the LV

The second example consists of two ultrasound images sequences (*heart*1 and *heart*2 sequences). In this section we present results from the two trackers, MMDA [10] and DBN [11], [12], which allow to testify the performance of the GP-MLM when several ground-truth (*i.e.*, inputs) are provided to the algorithm.

Two LV sequences from two patients are used in these tests. Their length (in frames) is: 490 (26 cardiac cycles), 470 (19 cardiac cycles) frames; 20 points of a quadratic B-spline are used to describe the contour of the LV.

In these examples, the heart motion is described by two main dynamics: an expansion motion that occurs in the diastole phase, and a contraction motion that characterizes the systole phase. In this study we go further in the attempt to find which is the best technique (*i.e.*, KF vs PF; WTA vs BLD rules); at the same time we hope to demonstrate the superiority of the non-linear GP 1st order model as in the previous example. To attain this goal an objective evaluation between the trackers contour estimates (taken as reference contours)

TABLE II

MSE FOR THE THREE IDENTIFICATION STRATEGIES: LINEAR 1ST AND 2ND ORDER MODELS AND A NON-LINEAR GP MODEL: (a) MMDA AND (b) DBN TRACKERS

	Sequence # 1 MSE		
Patch #	Linear 1^{st} order	Linear 2^{nd} order	GP 1^{st} order
1	46.16	61.98	**0.42**
2	6.95	7.77	**1.36**
3	7.66	8.44	**1.34**
	Sequence # 2 MSE		
Patch #	Linear 1^{st} order	Linear 2^{nd} order	GP 1^{st} order
1	3.99	3.67	**0.32**
2	19.76	54.69	**0.03**

(a)

	Sequence # 1 MSE		
Patch #	Linear 1^{st} order	Linear 2^{nd} order	GP 1^{st} order
1	12.37	17.13	**1.94**
2	**0.05**	0.59	0.74
3	9.62	19.98	**3.99**
4	6.93	7.33	**1.70**
	Sequence # 2 MSE		
Patch #	Linear 1^{st} order	Linear 2^{nd} order	GP 1^{st} order
1	48.24	53.97	**3.78**
2	138.19	169.05	**95.00**

(b)

and the GP-MLM estimates is provided; several metrics proposed in the literature for contours comparison are used.

These experiments are conducted as follows. From the ground truth contours automatically provided (by MMDA or DBN tracker) we take 30% of the frames for each case (*e.g.* each sequence) for training purposes. This allow us to identify the object dynamics (see Section V-B), *i.e.* first and second order linear models as well as non-linear GP dynamical model. In the test stage, we apply the filtering techniques (*e.g.* using KF or PF) taking the best patch response (WTA rule) or joining all the patches estimates (BLD rule).

Table II lists the MSE for both trackers and for the three identification strategies for each patch found by the GP-MLM. Most of the time, in both sequences, and from both ground truth trackers, the GP consistently provides the best results comparing with the remaining strategies.

Table III shows the results obtained using different combinations of filters vs. rules. It can be seen that the particle filtering[2] with the "blending" rule provides the best results for both sequences and trackers.

1) Objective Evaluation: To evaluate the performance of the algorithm we compare the contour estimates provided by both the MMDA and DBN trackers (*i.e.*, the ground truth or reference contours) and the GP-MLM estimates. Five error metrics were used in these tests: (i) Hammoude metric [48] (d_{HMD}); (ii) Average metric (d_{AV}); (iii) Hausdorff metric (*cf.* [49]) (d_{HDF}); iv) Mean Sum of Squared metric (d_{MSSD}) (*cf.* [50]) (v) Mean Absolute metric (d_{MAD}) (*cf.* [51]); and (vi) Average Perpendicular metric (d_{AVP}) (used in MICCAI challenge).[3]

TABLE III

MSE USING TWO ALGORITHMS: KALMAN FILTER AND PARTICLE FILTER. THE RESULTS ARE SHOWN FOR "BLENDING" AND "WINNER-TAKE-ALL" RULES: (a) MMDA AND (b) DBN TRACKERS

	Sequence # 1 MSE	
Gating	Kalman Filtering	Particle Filtering
Winner-take-all	450.99	64.58
Blend	440.10	**62.19**
	Sequence # 2 MSE	
Gating	Kalman Filtring	Particle Filtering
Winner-take-all	330.43	89.78
Blend	321.50	**89.31**

(a)

	Sequence # 1 MSE	
Gating	Kalman Filtering	Particle Filtering
Winner-take-all	441.87	131.98
Blend	394.98	**108.49**
	Sequence # 2 MSE	
Gating	Kalman Filtring	Particle Filtering
Winner-take-all	441.88	112.10
Blend	394.99	**97.14**

(b)

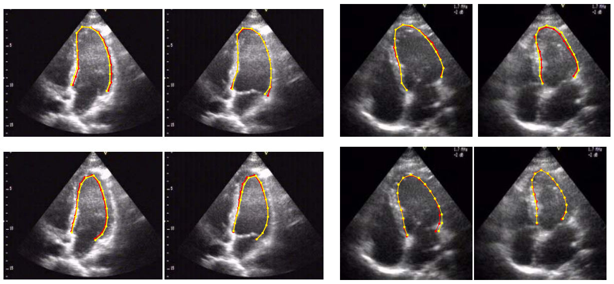


Fig. 6. GP–MLM tracking estimates (yellow line), superimposed with MMDA in red line (top row) and with the DBN in red line (bottom row) taken as gold contours. Sequence *heart*1 and *heart*2 are shown in four right and four left images, respectively, (each column in the figure corresponds to the same frame).

Table IV shows the fidelity in the representation of the LV contour obtained in the two US sequences. The input sets (contours) for each metric is the output of the GP-MLM and the contours (ground truth) provided by MMDA or DBN trackers. These values correspond to the mean values of the metrics described above. From Table IV we conclude that, in both sequences, the best tracking performance is obtained when the particle filtering with the "blending" rule is used. Although, the blending rule with particle filter provides the best results, we stress that, for these examples, GP-MLM tracking performance is indeed remarkable for both techniques and rules (see the small and similar values obtained for all metrics). Fig. 6 shows some images illustrating the contour estimates of the GP-MLM, overlapped with the ground-truth of the trackers.

C. Lip Tracking

This section illustrates the GP–MLM performance in lip tracking. We present results of the proposed approach in two different data sets.

Comparing to the LV example, the nonrigid object (lip boundary) exhibits a higher variability in the shape, specially when a person is singing.

[2] In all experiments we used a number of 500 particles.

[3] We do not give analytical details concerning the above metrics, since they are classical and commonly used metrics for contour comparison (ie, easily found elsewhere).

TABLE IV

MSE FOR THE TWO LV US SEQUENCES, USING TWO ALGORITHMS: KALMAN FILTERING AND PARTICLE FILTERING. THE RESULTS ARE SHOWN FOR "BLENDING" (BLD) AND "WINNER-TAKE-ALL" (WTA) RULES. THE METRICS ARE SHOWN FOR THE GROUND TRUTH OF THE MMDA AND THE DBN TRACKERS

		MMDA						DBN					
	Rule	d_{HMD}	d_{AV}	d_{HDF}	d_{MSSD}	d_{MAD}	d_{AVP}	d_{HMD}	d_{AV}	d_{HDF}	d_{MSSD}	d_{MAD}	d_{AVP}
Seq1	KF - WTA	0.09	2.70	4.79	10.40	2.70	2.45	0.14	3.08	5.48	13.23	3.09	2.52
	KF - BLD	0.09	2.70	4.79	10.35	2.71	2.44	0.14	3.08	5.47	13.20	3.09	2.53
	PF - WTA	**0.06**	1.58	3.14	3.88	1.58	1.77	**0.09**	2.12	3.86	7.42	2.17	1.79
	PF - BLD	**0.06**	**1.56**	**3.09**	**3.72**	**1.56**	**1.75**	**0.09**	**2.02**	**3.63**	**6.22**	**2.04**	**1.70**
Seq2	KF - WTA	0.09	2.33	3.94	7.83	2.36	1.79	0.11	2.73	4.80	10.66	2.79	2.00
	KF - BLD	0.09	2.32	3.92	7.76	2.35	1.77	0.11	2.81	4.89	11.33	2.89	2.04
	PF - WTA	**0.08**	1.93	3.29	5.51	1.94	**1.69**	**0.08**	1.76	3.70	4.92	1.78	**1.59**
	PF - BLD	**0.08**	**1.92**	**3.28**	**5.48**	**1.93**	**1.69**	**0.08**	**1.74**	**3.64**	**4.81**	**1.75**	**1.59**

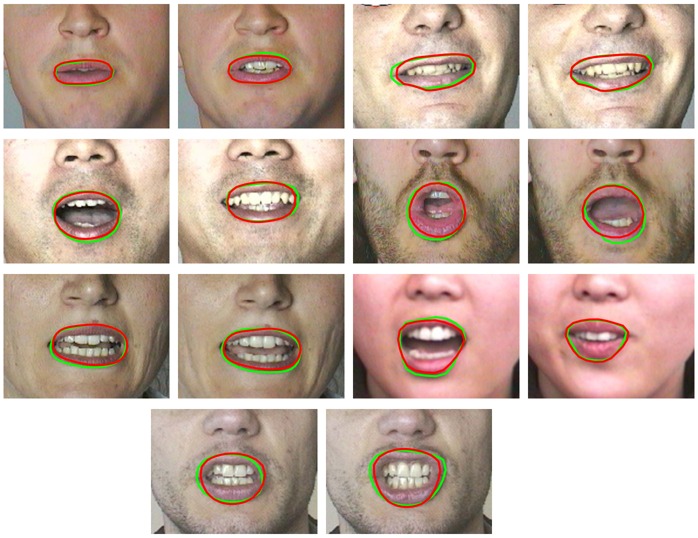


Fig. 7. GP–MLM tracking estimates for seven speaking sequences shown in red. The results are shown with the MMDA ground truth in green.

TABLE V

AVERAGE DISTANCES ERROR AND METRICS OBTAINED USING THE GP–MLM, FOR SPEAKING SEQUENCES (TOP) AND SINGING SEQUENCES (BOTTOM)

		Speaking Sequences Metrics				
	Tracker	d_{HMD}	d_{AV}	d_{HDF}	d_{MAD}	d_{AVP}
Seq1	MMDA	0.08	2.53	4.57	2.60	1.86
	DBN	0.08	2.89	5.80	3.06	2.14
Seq2	MMDA	0.13	4.68	9.15	5.96	3.59
	DBN	0.11	3.68	7.33	4.07	3.29
Seq3	MMDA	0.15	5.68	10.07	6.27	4.78
	DBN	0.15	4.62	10.29	5.69	4.26
Seq4	MMDA	0.12	4.45	8.97	5.23	3.67
	DBN	0.09	3.74	7.93	4.18	3.04
Seq5	MMDA	0.08	3.09	5.58	3.39	2.28
	DBN	0.14	4.36	8.62	4.71	3.91
Seq6	MMDA	0.09	3.61	6.96	3.99	2.91
	DBN	0.08	3.23	6.86	3.33	2.53
Seq7	MMDA	0.12	4.54	8.39	5.04	3.76
	DBN	0.10	3.67	8.08	3.93	3.02
		Singing Sequences Metrics				
	Tracker	d_{HMD}	d_{AV}	d_{HDF}	d_{MAD}	d_{AVP}
Seq1	MMDA	0.15	5.23	10.76	6.84	4.55
	DBN	0.15	5.19	10.82	6.60	4.52
Seq2	MMDA	0.15	4.73	10.18	6.45	4.22
	DBN	0.14	4.31	9.07	5.24	4.19
Seq3	MMDA	0.13	4.23	7.20	5.32	3.21
	DBN	0.14	4.95	10.07	5.48	4.33
		Emotion Sequences Metrics				
	Ground Truth	d_{HMD}	d_{AV}	d_{HDF}	d_{MAD}	d_{AVP}
Seq1-10	Manual	0.10	1.44	3.00	1.54	2.62

From this point on, and to avoid an exponential growth of the results, we present the results using the 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.

Fig. 7 shows in red dots the GP–MLM estimates overlapped with the MMDA ground truth.[4]

Table V (top) shows the results obtained for the speaking case, which are consistent to 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 bottom of the Table V), we see that a small decrease on this metric, and the small increase of the metrics which penalizes maximum local distances are obtained. This is somehow expected, since in this case, a large and sudden changes in the lip boundary may in close frames of the sequence. For instance, in Fig. 8 the $1st$, $2nd$ columns and the $3rd$ and $4th$ columns shows closed (consecutive or two frames difference) frames in the sequence. These correspond to difficult situations where the GP–MLM is able to produce remarkable results. Notice that, the Hammoude metric error remains in the same range, despite the larger shape variations in the singing lip

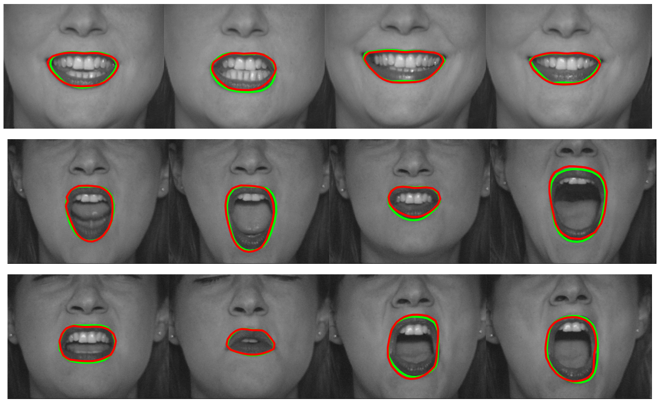


Fig. 8. GP–MLM tracking estimates (red dots) for three singing sequences shown in red dots. The comparison is illustrated with the MMDA ground-truth (green dots).

[4] Here we do not show the results obtained when the DBN is used since visually they are very quite similar

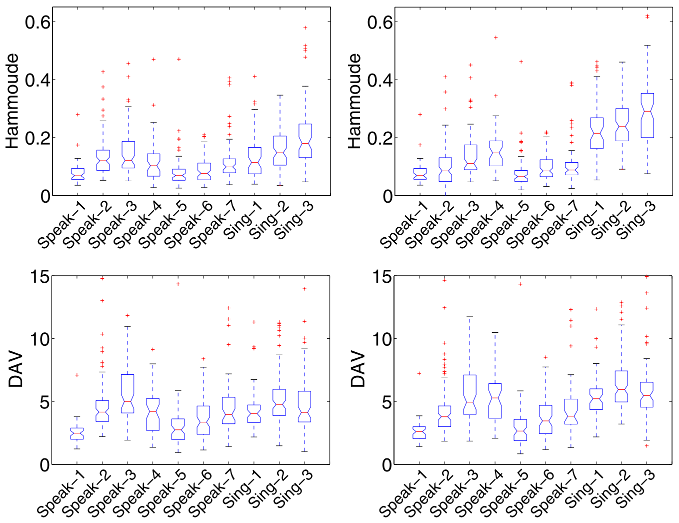


Fig. 9. Hammoude (top) and average (bottom) metrics provided by the GPMLM (left) and GPLVM (right). The x-labels from speak-1 to speak-7, corresponds to the sequences shown in Fig. 7 from the top to the bottom row, respectively. The x-labels from sing-1 to sing-3, correspond to the sequences shown in Fig. 8, from top to bottom row, respectively.

sequences (see top and bottom panel of the Table V). To our knowledge this is the first *low-dimensional* tracker capable of coping with large shape deformation in the deformable model.

VII. Comparison to the State-of-the-Art

In this section we include a comparison with the GP-LVM method originally proposed in [14].[5] A quantitative comparison between the GPMLM and GPLVM using the lips sequences shown in Section VI-C and the metrics detailed in Sec. VI-B.1 is conducted.

A. GPLVM

As in the GPMLM, the GPLVM is based on Gaussian process regression. Although, unlike the general formulation of the regression problems, in which the training pairs $\{\mathbf{x}_n, \mathbf{y}_n\}$, with $\mathbf{x}_n \in \mathbb{R}^q$ and $\mathbf{y}_n \in \mathbb{R}^m$, are available and a mapping function $\mathbf{f}$ such that $\mathbf{y}_n = \mathbf{f}(\mathbf{x}_n)$ is sought, in the GPLVM framework the positions $\mathbf{x}_n$ are initially unknown and should be estimated along with the mapping from $\mathbf{x}_n$ to $\mathbf{y}_n$. Thus, instead of marginalizing over latent variables $\mathbf{x}$ to find the linear mapping (like PCA), the GPLVM marginalizes over mapping functions and then optimize the latent positions $\mathbf{x}_n$.

To perform a comparison with the GPLVM in the context of tracking, we follow the work proposed by R. Urtasun and her colleagues [53], where a dynamic MAP estimation framework based on a GPLVM using a independent image-based tracker is used (we refer to the reader [53] for an in depth review).

Fig. 9, shows the results of the two manifold learning algorithms for tracking the lip sequences. The top and bottom rows of the figure show the results concerning the Hammoude and average metrics, respectively. 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 this figure, we conclude that both algorithms perform quite well in this problem exhibiting comparable results. We see that for the speak sequences the range of the two metrics are the same (the best values alternate between the two methods). Recall however that, for sequences having a higher deformation (see the results in the singing sequences) the GP-MLM compares favorably. We see that in the case of singing sequences the GPLVM provides higher results, i.e. $d_{\text{HMD}} > 0.2$ and d_{AV} slightly above 5 pixels.

VIII. Conclusion

A novel method for manifold learning has been proposed. The 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 is proposed for probabilistically combining the local patch estimates, based on the output of Gaussian process regression.

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 demonstrates that quite remarkable and competitive results are achieved.

References

[1] R. Hartley and A. Zisserman, *Multiple View Geometry*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[2] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto, "Dynamic textures," *Int. J. Comput. Vis.*, vol. 51, no. 2, pp. 91–109, 2003.

[3] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuezle, "Surface reconstruction from unorganized points," in *Proc. SIGGRAPH*, 1992, pp. 71–78.

[4] X. He, W.-Y. Ma, and H.-J. Zhang, "Learning an image manifold for retrieval," in *Proc. ACM Int. Conf. Multimedia*, New York, NY, USA, 2004, pp. 17–23.

[5] P. Baldi and G. Hatfield, *DNA Microarrays and Gene Expression*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[6] H. Lähdesmäki, O. Yli-Harja, W. Zhang, and I. Shmulevich, "Intrinsic dimensionality in gene expression analysis," in *Proc. GENSIPS*, Newport, Rhode Island, USA, 2005, pp. 1–2.

[7] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models," in *Proc. Adv. NIPS*, 2005, pp. 1441–1448.

[8] S. T. Roweis and Z. Ghahramani, "An EM algorithm for identification of nonlinear dynamical systems," in *Kalman Filtering and Neural Networks*, S. Haykin, Ed. New York, NY, USA: Wiley, 2000.

[9] R. Li, T.-P. Tian, and S. Sclaroff, "Simultaneous learning of nonlinear manifold and dynamical models for high-dimensional time series," in *Proc. 11th IEEE ICCV*, Oct. 2007, pp. 1–8.

[10] J. C. Nascimento and J. S. Marques, "Robust shape tracking with multiple models in ultrasound images," *IEEE Trans. Image Process.*, vol. 17, no. 3, pp. 392–406, Mar. 2008.

[11] G. Carneiro and J. C. Nascimento, "Multiple dynamic models for tracking the left ventricle of the heart from ultrasound data using particle filters and deep learning architectures," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 2815–2822.

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

[12] G. Carneiro, J. C. Nascimento, and A. Freitas, "The segmentation of the left ventricle of the heart from ultrasound data using deep learning architectures and derivative-based search methods," *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 968–982, Mar. 2012.

[13] L. Neil, "Gaussian process models for visualisation of high dimensional data," in *Proc. NIPS*, vol. 17. 2004, pp. 329–336.

[14] L. Neil, "Probabilistic non-linear principal component analysis with Gaussian process latent variable models," *J. Mach. Learn. Res.*, vol. 6, pp. 1783–1816, Nov. 2005.

[15] J. C. Nascimento and J. G. Silva, "Manifold learning for object tracking with multiple motion dynamics," in *Proc. ECCV*, 2010, pp. 172–185.

[16] W. M. Boothby, *An Introduction to Differential Manifolds and Riemannian Geometry*. New York, NY, USA: Academic, 2003.

[17] K. Torkkola, "Feature extraction by non-parametric mutual information maximization," *J. Mach. Learn. Res.*, vol. 3, pp. 1415–1438, Mar. 2003.

[18] C. Bishop, M. Svensen, and C. Williams, "GTM: The generative topographic mapping," *Neural Comput.*, vol. 10, no. 1, pp. 215–234, Jan. 1998.

[19] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[20] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[21] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.

[22] D. L. Donoho and C. Grimes, "Hessian eigenmaps: New locally linear embedding techniques for high-dimensional data," Dept. Statist., Stanford University, Stanford, CA, USA, Tech. Rep. TR-2003-08, 2003.

[23] K. Q. Weinberger and L. K. Saul, "Unsupervised learning of image manifolds by semidefinite programming," in *Proc. IEEE Comput. Soc. Conf. CVPR*, Jul. 2004, pp. 988–995.

[24] K. Q. Weinberger, F. Sha, and L. K. Saul, "Learning a kernel matrix for nonlinear dimensionality reduction," in *Proc. 21st ICML*, 2004, pp. 839–846.

[25] K. Q. Weinberger and L. K. Saul, "Unsupervised learning of image manifolds by semidefinite programming," *Int. J. Comput. Vis.*, vol. 70, no. 1, pp. 77–90, 2006.

[26] N. Cristianini, J. Shawe-Taylor, and J. Kandola, "Spectral kernel methods for clustering," in *Proc. Adv. NIPS*, vol. 14. 2001, pp. 649–655.

[27] V. de Silva and J. B. Tenenbaum, "Global versus local methods in nonlinear dimensionality reduction," in *Proc. Adv. NIPS*, vol. 15. 2002, pp. 721–728.

[28] B. Schölkopf, A. J. Smola, and K.-R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, 1998.

[29] J. Ham, D. D. Lee, S. Mika, and B. Schölkopf, "A kernel view of the dimensionality reduction of manifolds," in *Proc. 21st ICML*, 2004, pp. 369–376.

[30] M. Brand, "Charting a manifold," in *Proc. NIPS*, vol. 15. 2002, pp. 961–968.

[31] M. Chen, J. Silva, J. Paisley, C. Wang, D. Dunson, and L. Carin, "Compressive sensing on manifolds using a nonparametric mixture of factor analyzers: Algorithm and performance bounds," *IEEE Trans. Signal Process.*, vol. 58, no. 12, pp. 6140–6155, Dec. 2010.

[32] A. Rahimi and B. Recht, "Unsupervised regression with applications to nonlinear system identification," in *Proc. NIPS*, 2007, pp. 1113–1120.

[33] A. Elgammal and C.-S. Lee, "Nonlinear manifold learning for dynamic shape and dynamic appearance," *Comput. Vis. Image Understand.*, vol. 106, no. 1, pp. 31–46, 2007.

[34] R. Urtasun, D. J. Fleet, A. Geiger, J. Popovic, T. J. Darrel, and N. L. Lawrence, "Topololically-constrained latent variable models," in *Proc. ICML*, 2008, pp. 1–8.

[35] M. Raginsky, "A complexity-regularized quantization approach to nonlinear dimensionality reduction," in *Proc. IEEE Int. Symp. Inf. Theory*, Adelaide, Australia, Sep. 2005, pp. 352–356.

[36] A. Bjorck and G. H. Golub, "Numerical methods for computing angles between linear subspaces," *Math. Comput.*, vol. 27, no. 123, pp. 579–594, 1973.

[37] K. Fukunaga and D. R. Olsen, "An algorithm for finding intrinsic dimensionality of data," *IEEE Trans. Comput.*, vol. C-20, no. 2, pp. 176–183, Feb. 1971.

[38] J. Bruske and G. Sommer, "Intrinsic dimensionality estimation with optimally topology preserving maps," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 5, pp. 572–575, May 1998.

[39] M. Raginsky and S. Lazebnik, "Estimation of intrinsic dimensionality using high-rate vector quantization," in *Proc. Adv. NIPS*, vol. 18. 2005, pp. 1105–1112.

[40] E. Levina and P. J. Bickel, "Maximum likelihood estimation of intrinsic dimension," in *Proc. Adv. NIPS*, 2004, pp. 777–784.

[41] A. V. Little, J. Lee, Y. M. Jung, and M. Maggioni, "Estimation of intrinsic dimensionality of samples from noisy low-dimensional manifolds in high dimensions with multiscale SVD," in *Proc. IEEE 15th Workshop SSP*, Sep. 2009, pp. 85–88.

[42] A. Julosky, S. Weiland, and M. Heemels, "A Bayesian approach to identifications of hybrid systems," *IEEE Trans. Autom. Control*, vol. 50, no. 10, pp. 1520–1533, Oct. 2005.

[43] M. A. Fishler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[44] D. Myatt, P. Torr, S. Nasuto, J. Bishop, and R. Craddock, "NAPSAC: High noise, high dimensional robust estimation—It's in the bag," in *Proc. BMVC*, 2002, pp. 458–467.

[45] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2005.

[46] C. Rasmussen and G. D. Hager, "Probabilistic data association methods for tracking complex visual objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 560–576, Jun. 2001.

[47] A. Doucet, N. de Freitas, and N. Gordon, *Sequencial Monte Carlo Methods in Practice*. New York, NY, USA: Springer-Verlag, 2001.

[48] A. Hammoude, "Computer-assited endocardial border identification from a sequence of two-dimensional echocardiographic images," Ph.D. dissertation, Univ. Washington, Seattle, WA, USA, 1988.

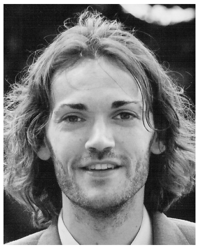
[49] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using Hausdorff distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 850–863, Sep. 1993.

[50] Y. Akgul and C. Kambhamettu, "A coarse-to-fine deformable contour optimization framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 2, pp. 174–186, Feb. 2003.

[51] I. Mikić, S. Krucinki, and J. D. Thomas, "Segmentation and tracking in echocardiographic sequences: Active contours guided by optical flow estimates," *IEEE Trans. Med. Imag.*, vol. 17, no. 2, pp. 274–284, Aug. 1998.

[52] D. MacKay, "Introduction to Gaussian processes," in *Neural Networks and Machine Learning*, C. Bishop, Ed. Norwell, MA, USA: Kluwer, 1998, pp. 133–166.

[53] R. Urtasun, D. J. Fleet, A. Herzmann, and P. Fua, "Priors for people tracking from small training sets," in *Proc. 10th IEEE ICCV*, Oct. 2005, pp. 403–410.



Jacinto C. Nascimento (S'00–M'06) received the Electrical Engineering degree from Instituto Superior de Engenharia de Lisboa in 1995, and the M.Sc. and Ph.D. degrees from Instituto Superior Técnico (IST), Technical University of Lisbon, in 1998 and 2003, respectively. Currently, he is an Assistant Professor with the Informatics and Computer Engineering Department, IST, and a Researcher with the Institute for Systems and Robotics. He has published 30 publications in international journals and 80 in conference proceedings (many of which of the IEEE), has served on program committees of many international conferences, and has been a reviewer for several international journals. His research interests include statistical image processing, pattern recognition, machine learning, medical imaging analysis, video surveillance, and general visual object classification.

Jorge G. Silva received the Electrical Engineering, M.Sc., and Ph.D. degrees in electrical and computer engineering from Instituto Superior Técnico, Lisbon, Portugal, in 1993, 1999, and 2007, respectively. He was an Adjunct Professor with Instituto Superior de Engenharia de Lisboa, Lisbon, from 1996 to 2007. From 2007 to 2012, he was a Research Scientist with Duke University, Durham, NC, USA. Since 2012, he has been a Senior Research Statistician with the SAS Institute, Cary, NC, USA. His research interests include statistical models applied to large-scale problems, e.g., manifold learning, social networks, and recommender systems.

Jorge S. Marques received the Electrical Engineering, Ph.D., and Aggregation degrees from the Technical University of Lisbon, Portugal, in 1981, 1990, and 2002, respectively. Currently, he is an Associate Professor with the Electrical and Computer Engineering Department, Instituto Superior Técnico, Lisbon, and a Researcher with the Institute for Systems and Robotics. He has published over 150 papers in international journals and conferences and he is the author of the book *Pattern Recognition: Statistical and Neural Methods* (IST Press, 2005). He was the Co-Chairman of the IAPR Conference IbPRIA in 2005, President of the Portuguese Association for Pattern Recognition from 2001 to 2003, and Associate Editor for the *Statistics and Computing Journal* (Springer). His research interests are in the areas of statistical image processing, shape analysis, and pattern recognition.

João M. Lemos received the Ph.D. degree from Instituto Superior Técnico (IST), Technical University of Lisbon, Portugal, in 1989, after extensive periods of research work at the University of Florence, Italy, and a period of experimental work at the Department of Chemical Engineering, Imperial College of Science, Technology and Medicine, London, U.K., and became an Associate Professor in 1997. Currently, he is a Full Professor of systems decision and control with IST and Researcher of INESCID, where he leads the Control of Dynamic Systems Group since 1990. He has been a Chairman of the Department of Electrical and Computer Engineering, IST, and Coordinator of the Postgraduation Programme in electrical and computer engineering; and since 2007, he has been the Chairman of the Scientific Board of INESC-ID. He has co-authored over 40 journal papers, ten book chapters and 180 conference papers, and supervised 13 Ph.D. thesis and 27 M.Sc. dissertations. His research interests are in the area of computer control, including adaptive and predictive control, control and estimation with multiple models, distributed control, modeling and identification, and process monitoring. He has been involved both either as responsible or participant, in projects concerning control applications to several types of industrial processes, large scale boilers in thermoelectric plants, solar collector fields and furnaces, biomedical systems (general anaesthesia, HIV 1 therapy, biochip temperature control), automotive systems, and water delivery canals.