A GEOMETRIC APPROACH TO MOTION TRACKING IN MANIFOLDS[1]

J. G. Silva[*], J. S. Marques[†], J. M. Lemos[‡]

[*]*ISEL, R. Conselheiro Emídio Navarro, 1949-019 Lisboa, Portugal*
jgs@isel.ipl.pt

[†]*IST/ISR, Av. Rovisco Pais 1949-001, Lisboa, Portugal*
jsm@isr.ist.utl.pt

[‡]*INESC-ID/IST, R. Alves Redol, 9, 1000-029 Lisboa, Portugal*
jlml@inesc.pt

Abstract: In many multi-dimensional tracking problems, the quantities of interest are restricted to a manifold in observation space. Learning the manifold shape is a necessary step for dimensionality reduction, which in turn allows faster and more robust tracking performance. For manifolds with arbitrary topology, learning the shape from noisy scattered data is not trivial. This paper presents a geometric approach that is valid for arbitrary manifold dimension and topology. An approximation of the tangent bundle is computed by region growing, making it possible to estimate a set of manifold charts. A tracking algorithm which takes advantage of the geometric information thus found is also presented.

Keywords: Multi-dimensional, Modeling, Target tracking.

1. INTRODUCTION

Many tracking problems in control and computer vision, involving observations in high-dimensional spaces, can have their complexity reduced by incorporating geometric information in the tracking approach.

Assuming the observed trajectory to lie in a manifold, embedded in a high dimensional space, and to be corrupted by noise, the dimensionality reduction provided by using inherent geometric constraints is expected to result in higher robustness and smaller computational load (Marques *et al.*, 1999). This paper presents a tracking algorithm which makes use of such geometric information. It is an extension of previous work on 1-D manifolds (Silva *et al.*, 2000).

The algorithm described here is exemplified for a 2-torus in 3-D space, but it is valid for other topologies and higher dimensions, both of the manifold and the embedding space. The manifold is assumed to be orientable and compact, that is, to have a unit normal vector field and to allow covering by a finite number of charts. It may or not have a boundary.

The tangent bundle of the manifold is the set of tangent hyperplanes at all manifold points (O'Neill 1997). There are infinitely many such points and tangent hyperplanes.

The main, novel idea contained in the present work is to use a region growing approach in the non-trivial task of approximating the tangent bundle of the manifold by a finite number of tangent hyperplanes, when the manifold has arbitrary topology. This simplifies estimation of the charts.

An overview of the algorithm is given in the next section. The following sections describe the major steps in detail and present experimental results, as well as conclusions.

2. OVERVIEW

Beginning with a set of scattered, noisy points in observation space, used as a training set, it is

[1] This work was partially supported by FCT POCTI, under project 37844.

intended to track trajectories that evolve in the same manifold as the training set.

An example training set is shown in Figure 1, where several noisy observations lie on a torus.

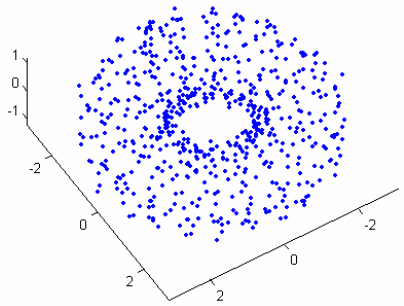

Fig. 1. Example training set: noisy observations **y** on a torus.

The steps involved in the algorithm are the following:

- Approximating the tangent bundle of the manifold;
- Finding local parametrizations and charts;
- Performing tracking in the lower dimensional parametric domains.

Each of these steps is addressed in detail in the following sections.

The manifold M is modelled through a set of diffeomorphisms $g_i : U_i \subset R^n \rightarrow R^m$, where n is the manifold dimension, m is the dimension of the embedding space and the U_i are open sets. These functions are *charts* of the manifold. Collectively, their overlapping images $g_i(U_i)$, also called *patches*, cover the manifold. Being diffeomorphisms, the charts also admit inverses, g_i^{-1}, which are called *parametrizations*.

First, the unit normal vector field of the whole manifold is estimated by computing, for each data point, the smallest eigenvectors of the local covariance matrix.

Next, the set of overlapping patches is found. This is done by region growing, until all data points belong to at least one patch. Each patch grows by appending all neighbouring points where the normal vector field does not deviate, in angle, more than a set threshold from the normal at the initial seed. In general, a given data point may belong to more than one patch. This yields local coordinate systems, by using using Principal Component Analysis (PCA) to find the best fitting hyperplane for each patch.

Charts are then estimated by thin-plate spline approximation, using the coordinate systems found above. Since the normal was only allowed to change direction up to a specified angular limit, it is guaranteed that the charts are bijective. They are also differentiable, because of thin-plate spline properties, so they are diffeomorphisms, as intended.

Tracking is then performed in the chart domains. The estimated trajectories can be lifted back to the embedding observation space by using the charts.

3. TANGENT BUNDLE APPROXIMATION

In this step, hyperplanes are found from scattered data points $\mathbf{y}_i=(y^1,...,y^m)$, allowing locally flat descriptions of the manifold.

When flattening a manifold with arbitrary topology, it is, in general, necessary to partition the manifold into more than one patch to avoid the so-called cartographer's dillema, which is due to metric distortion. Note that in some situations, such as the torus example, which has zero Gaussian curvature, one single patch would be theoretically enough. However, the present algorithm is intended for a broader class of problems.

Each patch can be associated to an hyperplane, and the collection of hyperplanes will be an approximation to the tangent bundle. The hyperplanes provide local coordinate systems valid in different regions. The best fitting hyperplane for each patch, in a least squares sense, is spanned by the the n largest eigenvectors returned by the PCA procedure, performed on all patch member points.

In order to make chart estimation easier, at a later stage, it is required for simple projection to give a one-to-one mapping between the hyperplane and the corresponding manifold region. This can be ensured by not allowing the hypersurface normal to vary more than a set threshold τ, in angle. It is thus necessary to compute the normals.

Normals are computed by visiting all data points and, for each one, finding the m-n smallest eigenvectors (smaller than a threshold) of the covariance in a neighbourhood of radius ε, as shown in Figure 2. It is assumed that the data are sufficiently dense to leave no gaps greater than δ and observation noise has standard deviation nowhere greater than σ. The radius ε is chosen to account for both δ and σ.

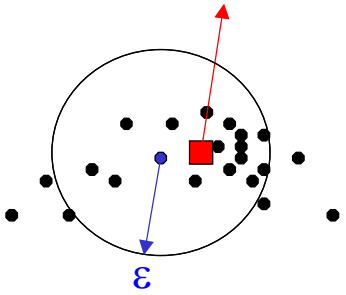

Fig. 2. Surface normal from PCA in a neighbourhood.

Since the data points are noisy, the point of application of the normal is set to the mean of the neighbourhood, which is a form of low-pass filtering.

The eigenvectors thus returned have arbitrary orientations, so it would be necessary ro revisit all data points to maintain consistency between nearby normals, which is a NP-complete problem (Hoppe 1994). However, since it is just necessary to compute angles, direction is all that matters and orientation can be discarded. After normals are found, region growing takes place as following:

while M not covered
  P = new patch
  y_0 = choose a new seed from data points not in any patch
  n_0 = normal at y_0
  while NOT all points visited
    y_1 = choose nearest neighbor
    n_1 = normal at y_1
    if angle(n_0,n_1) < τ AND distance(y_0,y_1) < ε
      append y_1 to P
    end if
  end while
end while

The end result is a covering of *M* by a finite number, *p*, of overlapping patches. Within each patch, the normal doesn't deviate more than τ, and the distance test ensures that each patch is a connected set.

4. CHARTS AND PARAMETRIZATION

It is desirable that the manifold charts be diffeomorphisms, so that smoothness is ensured. There are many alternatives for non-linear function approximation that meet these requirements. The results presented here were obtained using thin-plate splines, which are described in (Duchamp 2002).

In short, a thin-plate spline *g* is the function that minimizes the weighted sum

$$\rho E(\tilde{g}) + (1-\rho) R(\tilde{g}) \qquad (1)$$

where ρ is a smoothing parameter and

$$E(\tilde{g}) = \sum_j (y_j - \tilde{g}(x_j))^2 \qquad (2)$$

$$R(\tilde{g}) = \int_{R^n} tr(H_g^2)\, dx_1 \ldots dx_n \qquad (3)$$

denote an error measure and a roughness measure, respectively. H_g is the Hessian of $\tilde{g}$. The *l*-degree solution comes in the form

$$\tilde{g}(x) = p(x) + \sum_j^{l-k} a_j \Psi(x_j - c_j) \qquad (4)$$

where $p(\boldsymbol{x})$ is a polynomial term involving *k* of the a_j coefficients, while the remaining *l-k* coefficients multiply the radial basis functions ψ, which are given by

$$\Psi(x_j - c_j) = \left\| x_j - c_j \right\|^2 \log\left(\left\| x_j - c_j \right\|^2 \right), \quad (5)$$

with centres c_j.

Having previously partitioned the manifold *M* in *p* patches, it is possible to find *p* charts $g_i(\boldsymbol{x})$ in the form given by (4), followed by a translation and a rotation.

Figure 3 shows some of the charts estimated in this fashion.

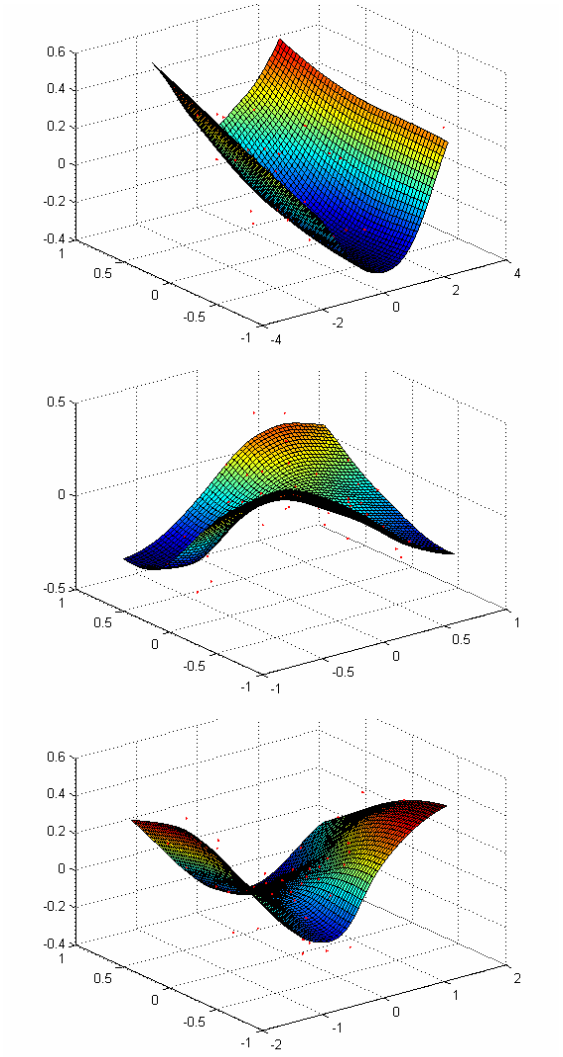

Fig. 3. Some of the charts obtained by thin-plate spline approximation. The graphs represent the functions, $\tilde{g}(x)$, which is this case are scalar, since they were obtained for a 2-torus embedded in 3-D space.

To summarize, with points $\boldsymbol{y}=(y_1,\ldots,y_m)$ belonging to a given patch *i*, and having previously performed

PCA, a matrix V_i of eigenvectors and a mean vector μ_i are available. Projecting $\boldsymbol{y}$ on the hyperplane associated with patch i is a matter of computing

$$\tilde{x} = V_i^T \left(y - \mu_i \right) \tag{6}$$

$$x = \begin{bmatrix} \tilde{x}_1 \\ \vdots \\ \tilde{x}_n \end{bmatrix}. \tag{7}$$

Equation (6) is an isometry, in this case, a translation followed by a rotation, while (7) describes simple projection.

As for the the chart g_i, which allows the inverse mapping of (7), it follows the expression

$$g_i(x) = V_i \begin{bmatrix} x_1 \\ \vdots \\ x_n \\ \tilde{g}(x) \end{bmatrix} + \mu_i \tag{8}$$

The remaining m-n components of $\tilde{x}$, instead of being set to zero, which would yield a rough, piecewise linear approximation of M, are thus preserved. Locally, the manifold parametrization is

$$x \to \begin{bmatrix} x & \tilde{g}(x) \end{bmatrix}^T \tag{9}$$

with $\tilde{g}$ given by (4).

5. TRACKING

At this stage, an observed trajectory can be projected onto the previously found hyperplanes, and all tracking can be done in n-D instead of m-D.

In order to track the projected trajectories, dynamic models are needed. A linear, discrete state model is used that includes dynamic and observation noise, both assumed Gaussian for simplicity.

It must be stressed that, while the original, m-D observations $\boldsymbol{y}_k$ are non-linear functions of $\boldsymbol{x}_k$, it is assumed that $\boldsymbol{x}$ itself is directly observed, by projection of $\boldsymbol{y}$, so it is possible to use a linear observation model.

A time-varying Kalman filter is used to estimate the trajectory from the noisy observations, and lifting back to the manifold with charts g allows recovery of the m-D trajectory.

Motion is descibed by the following stochastic difference equation:

$$s_{k+1} = As_k + Bw_k \tag{10}$$

where k denotes time, s is the state vector and the scalar w is dynamic noise. The observation equation is

$$o_k = Cs_k + Dn_k \tag{11}$$

where o is the observation vector (made equal to $\boldsymbol{x}_k$) and n is sensor noise. A white-noise acceleration model (Kalata, 1984) is followed, and the same dynamics are assumed for all patches. The A, B, C and D matrices are, therefore, constant.

Finally, since the $\mathbf{y}_k$ are projected onto p different hyperplanes, there are p different observations at time k. The problem of how to select the best hyperplane is solved by nearest neighbour classification. The training data point nearest to $\boldsymbol{y}_k$ is found (a computationally expensive procedure) and, among the patches it may belong to, the one that yields the least squared reconstruction error, that is

$$e_k = \left\| y_k - g_i(x_k) \right\|^2, \tag{12}$$

is selected.

6. RESULTS

For the experiment described in this section, the purpose is two-fold: it is intended to approximate a synthetically generated manifold, and also to recover an experimental trajectory, generated on the same manifold and deliberately corrupted with additive Gaussian noise.

The synthetic manifold is, in this case, a 2-torus. The learning dataset is a collection of points that lie on the surface as seen above, in Figure 1.

The results from the approximation step can also be seen above, in Figure 3. At this stage, the charts become available.

The next step in the experiment is motion recovery. A synthetic trajectory was generated, independently from the training set, according to a simple dynamic behaviour:

$$\begin{bmatrix} \theta_{k+1} \\ \varphi_{k+1} \end{bmatrix} = \begin{bmatrix} \theta_k \\ \varphi_k \end{bmatrix} + u_k + w_k, \tag{13}$$

including a constant velocity u_k term and Gaussian dynamic noise w_k. The trajectory, as illustrated in Figure 4, is a slightly noisy straight line in (θ, φ) space, which wraps around at angles $-\pi$ and π.

Through use of toroidal coordinates

$$
\begin{aligned}
r &= 2 + \cos(\varphi) \\
x &= r\cos(\theta) \\
y &= r\sin(\theta) \\
z &= \sin(\varphi)
\end{aligned}
\quad , \qquad (14)
$$

the synthetic trajectory was transported to the manifold, and Gaussian observation noise was finally also added. The time sequence of resulting data points, **y**=(x, y, z), constitutes the $\{ \boldsymbol{y}_1, \ldots, \boldsymbol{y}_k, \ldots \}$ experimental trajectory.

Tracking was performed after projection of the observations on the appropriate planes, learned during tangent bundle approximation. The resulting projected motion is shown in Figure 5.

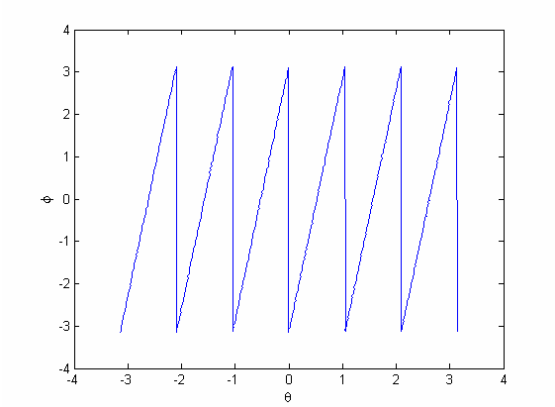

Fig. 4. Synthetic trajectory in (θ, φ), - π < θ, φ < π. The φ period is six times shorter than the θ period.

After lifting to observation space (using the charts), the accurately estimated trajectory can be seen in Figures 6 and 7, overlayed on the experimental one.

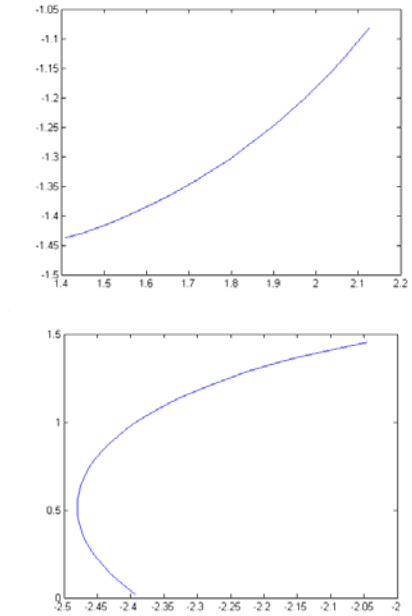

Fig. 5. Parts of the experimental trajectory, projected into some of the estimated planes.

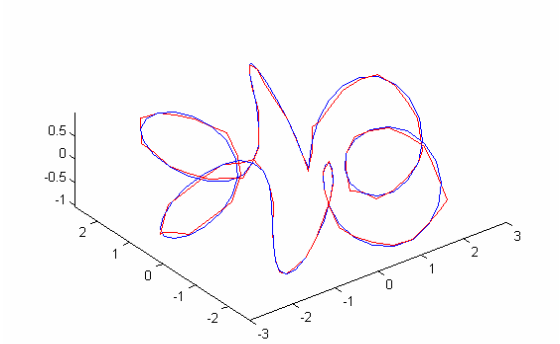

Fig. 6. Tracking results in the observation space. The experimental and estimated trajectories are blue and red, respectively.

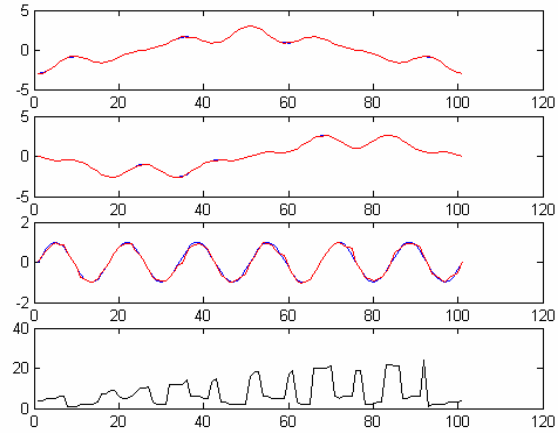

Fig. 7. Time plots of the (*x*,*y*,*z*) experimental trajectory coordinates (blue) and the estimated trajectory coordinates (red). Below, the index of the nearest hyperplane at each instant.

The dynamic model parameters for expressions (10) and (11), with 3-D data points being in this case projected onto 2-D planes, are

$$
A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \qquad (14)
$$

and the state vector is $s = \begin{bmatrix} x_1 & x_2 & dx_1 & dx_2 \end{bmatrix}^T$, with dx_1 and dx_2 denoting the increments in the x_1 and x_2 directions respectively, while the observation model parameters are

$$
C = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}, \quad D = 1. \qquad (15)
$$

7. CONCLUSIONS

This paper presents a method for manifold learning, applicable to trajectory tracking in *n*-D manifolds

embedded in *m*-D space, that relies on geometric information to reduce complexity. The manifold learning results are promising and make accurate trajectory tracking simpler to achieve.

An important direction of future work is modelling non-linear dynamics, with possibly distinct behaviours in different manifold regions, instead of the present method of using the white-noise acceleration model in all patches. A non-linear observer approach is being considered, rather than any form of Extended Kalman Filtering.

Also, a way to combine a probabilistic model with the current geometric model is being studied. This is needed in order to detect outliers and to select the best hyperplane for projection in a principled way, without having to use the computationally expensive nearest neighbour approach. One possible solution is to use mixtures of Gaussians on the hyperplanes, using Expectation-Maximization (EM) to estimate the parameters. This is simpler, both computationally and in terms of convergence, if done in the lower dimensional space, rather than directly in the observation space.

REFERENCES

Duchamp, T. and W. Stuetzle (2002). Spline Smoothing on Surfaces (preprint), To appear in: *Proceedings of ACM Siggraph*.

Hoppe, H. (1994). *Surface reconstruction from unorganized points* (PhD dissertation). University of Washington.

Kalata, P. (1984). The tracking index: A generalized parameter for α-β and α-β-γ trackers, *IEEE Transactions on Aerospace and Electronic Systems,* pp. 174-182.

Marques, J., J. M. Lemos and A. Abrantes (1999). Estimation of random trajectories on manifolds: Application to object tracking, In: *Proceedings of the IEEE International Conference on Image Processing,* pp. 103-107.

O'Neill, B. (1997). *Elementary Differential Geometry,* Academic Press.

Silva, J., J. Marques and J. M. Lemos, (2000). Robust motion tracking on one-dimensional manifolds, In: *Proceedings of the 4th Portuguese Conference on Automatic Control,* pp. 279-283.