ELSEVIER
Signal Processing ▮▮▮ (▮▮▮▮) ▮▮▮–▮▮▮
SIGNAL PROCESSING
www.elsevier.com/locate/sigpro

A HMM approach to the estimation of random trajectories on manifolds ☆

Jorge S. Marques[a,b,*], João M. Lemos[a,c], Arnaldo J. Abrantes[d]

[a] *Instituto de Sistemas e Robotica, Av. Rovisco Pais, 1049-001 Lisbon, Portugal*
[b] *Instituto Superior Técnico, Av. Rovisco Pais, 1049-001 Lisbon, Portugal*
[c] *INSEC, Portugal*
[d] *ISEL, Portugal*

Received 9 June 2000

Abstract

Dynamic image analysis requires the estimation of time-varying model parameters (e.g., shape coefficients). This can be seen as states of a dynamic model which are restricted to a subset of Euclidean space. This paper describes an algorithm for the estimation of the state evolution on manifolds exploiting three sources of information: the manifold geometry, the motion model and the sensor model. Examples are provided to illustrate the performance of this method in situations where classic procedures cannot perform well. © 2002 Published by Elsevier Science B.V.

Keywords: Image analysis; Object tracking; Hidden Markov models; Manifold estimation

1. Introduction

Dynamic image analysis requires the estimation of time-varying model parameters (e.g., shape coefficients). This can be seen as states of a dynamic model which are restricted to a subset of Euclidean space. Typical examples are the estimation of objects motions (e.g., cars) or the evolution of objects shapes (e.g., mouth contour, heart cavities) from video sequences. Hereafter, state is to be understood in this way. A common factor present in most estimation problems is the fact that unknown variables exist in high-dimensional spaces but they cannot take arbitrary values. Instead, they are usually restricted to smooth subsets (e.g., surfaces) (Fig. 1). These subsets often have a complex structure and must be estimated from the observation data. The following examples illustrate this point. Consider the problem of visual tracking of cars in a lane. A simple prototype situation is considered below in this paper. Since the lane geometry will enforce the trajectory, this is a valuable information for reducing the computational load of the search algorithm and furthermore enhances the robustness of the position estimation. Classical algorithms i.e., which do not restrict the state variables to a manifold, are more prone to yield instabilities in the tracking error. This drawback is even more serious when the observations of the car position are drawn from

☆ This work was partially supported by PRAXIS XXI under project EEI/12050/1998 (Tracking of Moving Objects).

* Corresponding author. Instituto de Sistemas e Robotica, Instituto Superior Técnico, Torre Norte, AV. Rovisco Pais, 1049-001 Lisbon, Portugal. Tel.: +351-1-310-0000; fax: +351-1-314-5843.
E-mail addresses: jsm@isr.ist.utl.pt (J.S. Marques), jlml@inesc.pt (J.M. Lemos), aja@cedet.isel.pt (A.J. Abrantes).

0165-1684/02/$ - see front matter © 2002 Published by Elsevier Science B.V.
PII: S0165-1684(02)00242-6

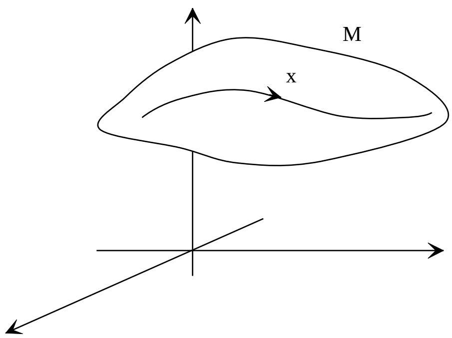

Fig. 1. Trajectory on a manifold.

omnidirectional sensors which measure only the distance to the target. An example is given below. Where these examples refer to the actual position of the moving object constrained to a manifold, one can also think of its motion parameters also being constrained. For instance, let the car velocity be modeled as the low-pass filtering of white noise with a given bandwidth (BW). If BW slowly varies in time, being constrained to some subset of values, the methods to be considered in this paper may also be applied with advantage.

In static problems (e.g., in image reconstruction and object recognition) attempts have been made for incorporating known restrictions in the estimation process [6,11,13]. Two types of constraints are usually considered: hard constraints which lead to the use of constrained optimization methods (e.g., POCS [1,7]) or soft constraints based on the use of regularization techniques or prior information [12]. In [6] three-dimensional object views are represented as a linear combination of eigen images multiplied by appropriate coefficients. Although the number of coefficients is very high, the number of degrees of freedom is much smaller i.e., when the view changes the coefficients describe a trajectory on a low dimension manifold. Advantage is taken of this fact for object recognition.

In dynamic scene analysis, state constraints play an even more important role for two main reasons: (i) they significantly improve the trajectory estimates (the improvement being often dramatic, e.g., a nonobservable system may become observable if appropriate restrictions are used) and (ii) they allow one to formulate the estimation problem in a lower dimension subspace (the dimension of the working subspace depends on the manifold dimension and not on the data dimension). Both effects are instrumental for achieving good results and should be considered in the design of trajectory estimation algorithms [3,2]. In [3] geometric restrictions are used for lip tracking. Although lips are represented by 40 control points belonging to a space of dimension 80, by exploiting the constraints in the control point movement, estimation has only to be performed in a space of dimension 5. Other examples are provided by in the analysis of Human gestures. In [2] gestures are described by a one-dimensional manifold denoted as *principal curve*.

In this paper, by considering a general framework for trajectory estimation on manifolds, a specific algorithm to solve this problem under a general hypothesis is proposed. By relying on discrete approximation and hidden Markov model (HMM) techniques, an algorithm for trajectory estimation on manifolds is derived. Illustrative examples are presented.

2. Estimation framework

The problem is stated as follows: Let x be an unknown trajectory defined in a manifold $\mathscr{M} \subset R^n$. The trajectory x is to be retrieved from a sequence of nonlinear and noisy observations y. It will be assumed that x is a realization of a stochastic process defined on the manifold and y consists of nonlinear and noisy observations of x values; these processes are characterized by a motion model $p(x_t/x_{t-1})$ and by a sensor model $p(y_t/x_t)$ that have to be estimated from the data.

The overall solution to this problem involves three steps: (i) manifold learning; (ii) motion and sensor model learning; and (iii) trajectory retrieval. The first steps are performed off line while the last step is performed on line. It will be assumed that a set of (x, y) sequences is known forming a training set employed in steps (i) and (ii).

In the first step the manifold is estimated from known x trajectories (Fig. 2). The manifold is segmented into a set of regions and each region is described by a parametric function defined on a local coordinate system: a hyperplane is fitted to each region defining a low dimension subspace of independent variables; the projection error is not neglected, being modeled as a dependent variable.

In the second step, two models, a motion model and a sensor model, are estimated from the data. The

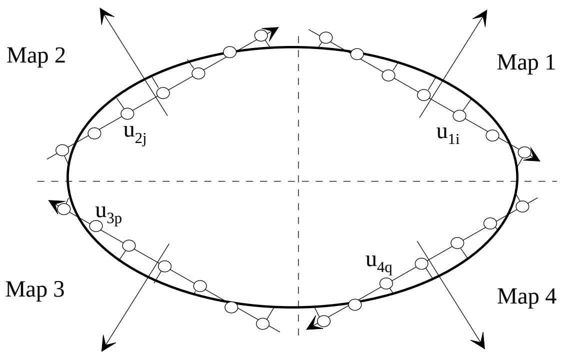

Fig. 2. Manifold model.

motion model defines the allowed transitions between points in the manifold and the corresponding probability associated with the transitions. Three approaches can be considered for describing the motion of a point on the manifold: a continuous time approach, a discrete time approach with continuous state and a discrete approach based on time and state discretization. The first approach requires the use of a stochastic differential equation defined on the manifold. This topic has been addressed in the controls literature [4,5] and can be pursued in the context of trajectory estimation. The second approach leads to the use of difference equations. The third approach is based on discrete time and discrete state models, discrete Markov models being an appealing solution.

A sensor model defining the probability distribution of the observed data for each manifold point is also needed and has to be estimated from the data. The previous steps concern manifold learning and motion/sensor learning. The third step is the estimation of trajectories on the manifold assuming that the manifold and the motion/sensor models are known. The solution of this problem depends on the models considered. The continuous time approach is based on stochastic differential equations, leading to the use of nonlinear observers on manifolds and will not be pursued here. A second approach based on the discrete state formulation is adopted in this paper, leading to a set of HMMs (a model per manifold region) linked by transitions among different regions.

In the example considered in Fig. 2 it is assumed that a point moves clockwise with a limited velocity. This results in a Markov chain with transitions to one of two consecutive states (see Fig. 3). This is just an

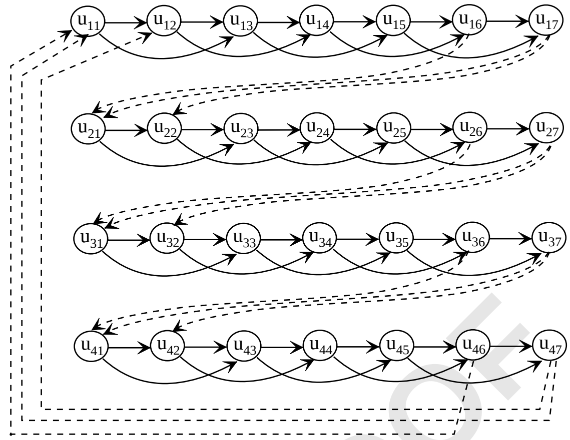

Fig. 3. Motion model: local and global Markov models.

example of a Markov chain which can arise in such a type of problems. Other structures of the Markov chain modeling motion may be considered.

3. The discrete manifold analysis (DMA) algorithm

This section describes an algorithm for the estimation of trajectories on manifolds. This algorithm, denoted as DMA, provides a solution for the three problems described before: manifold learning; motion/sensor model learning and trajectory retrieval.

3.1. Manifold learning

The DMA algorithm splits the manifold into M regions and approximates each region by a set of centroids (Fig. 2). Centroid computation is performed in a low dimension hyperplane estimated from the data.

Let X be a set of manifold points (training set). It will be assumed that the elements of X are realizations of a random variable whose distribution is approximated by a mixture of Gaussians

$$p(x) = \sum_{k=1}^{M} c_k N(x; \mu_k, R_k), \tag{1}$$

where M is the number of modes and $N(x; \mu, R)$ denotes a normal density function with mean μ and covariance matrix R. Under this hypothesis the expectation-maximization method (EM) is used to estimate the means μ_i and covariances R_i of the Gaussian modes [9] according to

Fuzzy classification (E-step):

$$\pi_k(x) = \frac{\hat{c}_k N(x; \hat{\mu}_k, \hat{R}_k)}{\sum_{i=1}^{M} \hat{c}_i N(x; \hat{\mu}_i, \hat{R}_i)}. \tag{2}$$

Update (M-step):

$$\hat{c}_k = \frac{1}{\#X} \sum_{x \in X} \pi_k(x), \tag{3}$$

$$\hat{\mu}_k = \frac{1}{\#X \hat{c}_k} \sum_{x \in X} \pi_k(x) x, \tag{4}$$

$$\hat{R}_k = \frac{1}{\#X \hat{c}_k} \sum_{x \in X} \pi_k(x)(x - \hat{\mu}_k)(x - \hat{\mu}_k)^{\mathrm{T}}, \tag{5}$$

where $\#X$ denotes the cardinality of set X. All data points contribute to estimate the means and covariances associated to all mixture modes but with different weights. Both steps are recursively computed until convergence is achieved. The number of modes, M, is either assumed a priori known or estimated from the data by a suitable method, e.g., minimum description length (MDL) [10]. The EM algorithm allows one to split the manifold into M disjoint regions, $\mathcal{M}_i$, according to a set of discriminant functions defined by the squared Mahalanobis distance

$$d(x, \mu_i)^2 = (x - \mu_i)^{\mathrm{T}} R_i^{-1} (x - \mu_i). \tag{6}$$

For the sake of describing the manifold one should define local coordinates in each region. This can be done by applying principal component analysis (PCA) [9] to each region, using the second order statistics μ_i, R_i. The PCA defines an orthogonal basis in each region associated with the eigenvectors of the covariance matrix. Each point $x \in \mathcal{M}_i$ is given by

$$x = \mu_i + V_i x_i, \tag{7}$$

where μ_i is the mean of the ith region, V_i is a $n \times n$ matrix whose columns are the eigenvectors of the covariance matrix and

$$x_i = V_i^{\mathrm{T}} (x - \mu_i) \tag{8}$$

are the local coordinates of x on $\mathcal{M}_i$.

If the smallest eigenvalues of the covariance matrix are close to zero an approximation may be considered in which their corresponding eigenvectors are removed from V_i. In this case, V_i becomes a $n \times m$ matrix and equation $\hat{x} = \mu_i + V_i x_i$ defines the orthogonal projection of x on the hyperplane spanned by the columns of V_i.

The hyperplane dimension (number of basis functions) depends on the eigenvalues of the covariance matrix. To keep the mean square error small, only the axes corresponding to small eigenvalues can be discarded. The mean square error, E^2, is given by

$$E^2 = \sum_{i=m+1}^{n} \lambda_i. \tag{9}$$

Fig. 4a shows the original data points on a parabolic surface (the surface is assumed to be unknown). Using the EM method and PCA, this set is decomposed in disjoint classes, each one being approximated by a local hyperplane (Fig. 4b). The data points are projected into the closest local hyperplane, their local coordinates being computed according to (8) (see Fig. 4c). The data points are projected into the local hyperplane and their local coordinates are computed. The data inside each region is approximated by a set of prototypes (centroids) obtained by the k-means algorithm [9]. The set of centroids associated to all the regions is denoted by U. The DMA algorithm adopts a discrete representation of the manifold, each manifold region being approximated by a set of prototypes $U_i = \{u_{i1}, \ldots, u_{in_i}\}$ expressed in local coordinates.

3.2. Motion and sensor model learning

A set of local Markov models is used to describe state trajectories inside the manifold regions. These local models are integrated in a global Markov model by considering trajectory transitions across the region borders. In this framework, trajectories are sequences of points on the manifold, corresponding to centroids on the hyperplanes.

Let $x = (x(1), \ldots, x(N)), x(t) \in M$, be a trajectory on the manifold and $u = (u(1), \ldots, u(N)), u(t) \in U$ a sequence of centroids obtained by projecting each manifold point, $x(t)$, onto the closest hyperplane and approximating the projected point by the closest prototype (Fig. 5).

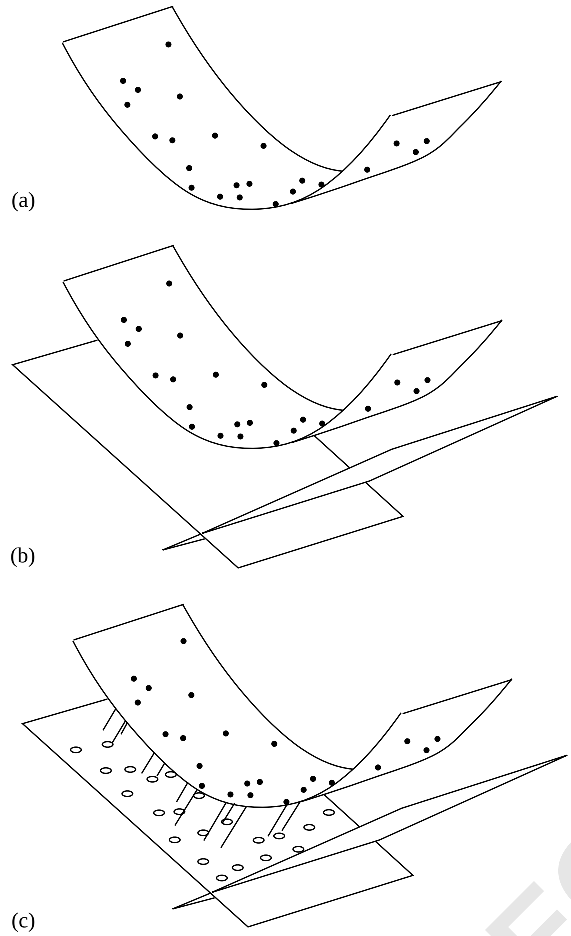

Fig. 4. Manifold learning: (a) original data points; (b) approximation of the manifold by local hyperplanes using EM (shifted downwards); (c) representation of projected data by centroids on the hyperplane.

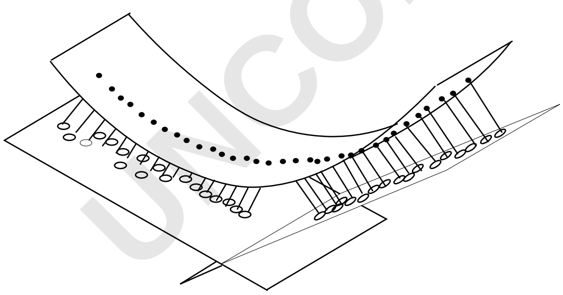

Fig. 5. Motion model: state trajectory and orthogonal projection and approximation by the nearest centroids.

It will be assumed that u is a random process which satisfies the Markov property of first order (P stands for probability)

$$P(u(t)/u(t-1),\ldots,u(1)) = P(u(t)/u(t-1)). \quad (10)$$

The u trajectory inside a region M_i takes values in U_i and is described by a local Markov chain with transition matrix $A^{ii} = (a^{ii}_{kl})$,

$$a^{ii}_{kl} = P(u(t) = u_{ik}/u(t-1) = u_{il}). \quad (11)$$

The u sequence is denoted as the *discrete state sequence*. Repeating this procedure for all manifold regions leads to M local Markov chains which define the motion models inside the regions.

To deal with transitions between two regions $M_i, M_j,\ i \neq j$, cross transition matrices $A^{ij} = (a^{ij}_{kl})$,

$$a^{ij}_{kl} = P(x(t) = u_{ik}/x(t-1) = u_{jl}) \quad (12)$$

are considered. The cross-transition matrices link all the local Markov models into a global model (Fig. 3) with transition matrix defined by

$$A = \left[\begin{array}{c|c|c|c} A^{11} & A^{12} & \ldots & A^{1M} \\ \hline \vdots & \vdots & \vdots & \vdots \\ \hline A^{M1} & A^{M2} & \ldots & A^{MM} \end{array}\right]. \quad (13)$$

In general A is a stochastic matrix whose entries may all be nonnegative. However, A is actually a sparse matrix in many situations (e.g., when a motion is being considered as in the example of Fig. 3) since most of the cross transitions have zero probability. This property stresses the physical meaning of the local Markov chains which describe the motion inside the manifold regions. Although the motivation for building matrix A stems from time and state discretization of a diffusion model which represents the motion considered, it is remarked that the methods described in this paper apply to general Markov chains. It should also be mentioned that the complexity of matrix A should be low enough so as to allow its estimation from available data.

In this paper, matrix A is estimated by computing the relative frequencies of all possible state transitions, a procedure justified by the strong law of large numbers. This amounts to assume available a sufficiently rich set of trajectories covering the whole manifold. The estimates converge almost surely to the true value as the number of points tends to infinity. To estimate the manifold sequence a sensor model is required. Let

$y = (y(1), \ldots, y(N))$ be an observation sequence. It will be assumed that $y(t)$ is an instantaneous measurement of $u(t)$ being characterized by the probability function $p(y(t)/u(t))$ which has to be estimated from the data.

The motion model $P(u(t)/u(t-1))$ and the sensor model $p(y(t)/u(t))$ define a hidden Markov model.

3.3. Trajectory retrieval

It is not usually possible to estimate $u(t)$ from a single observation $y(t)$ since in this case it is not possible to exploit the dynamical properties of the state trajectory u. The estimation of the manifold trajectory x is performed in two steps: (i) estimation of u and (ii) projection of u onto the manifold. Projection is performed by (7). Therefore, only the first step has to be addressed. This is a well-known problem in HMM theory which can be solved either by using the forward–backward algorithm or the Viterbi algorithm [8]. The first is used when an on-line estimate of the state variable is required, relying only on current and past observations while the Viterbi algorithm provides the optimal state trajectory assuming that all observations (including future observations) are known.

4. Experimental results

Hereafter, the DMA algorithm is evaluated with both synthetic and real data. Three examples will be described to illustrate the concepts presented in the paper.

4.1. Example 1

Fig. 6 shows the retrieval of trajectories on a sphere from incomplete and noisy measurements. In this example, the motion is a known Markov chain on the sphere and the observations are obtained by $y = [1\ -11]x + v$ where v is white Gaussian noise, i.e., y is a linear combination (projection onto a line) of the Cartesian coordinates of the point x on the sphere, corrupted by additive noise.

Since the observation model is noninvertible, the state cannot be recovered from a single measurement. Each observation defines a plane which intersects the sphere at an infinite number of points, located along a circle. To recover the x trajectory it is necessary to use information about the motion dynamics and the manifold geometry.

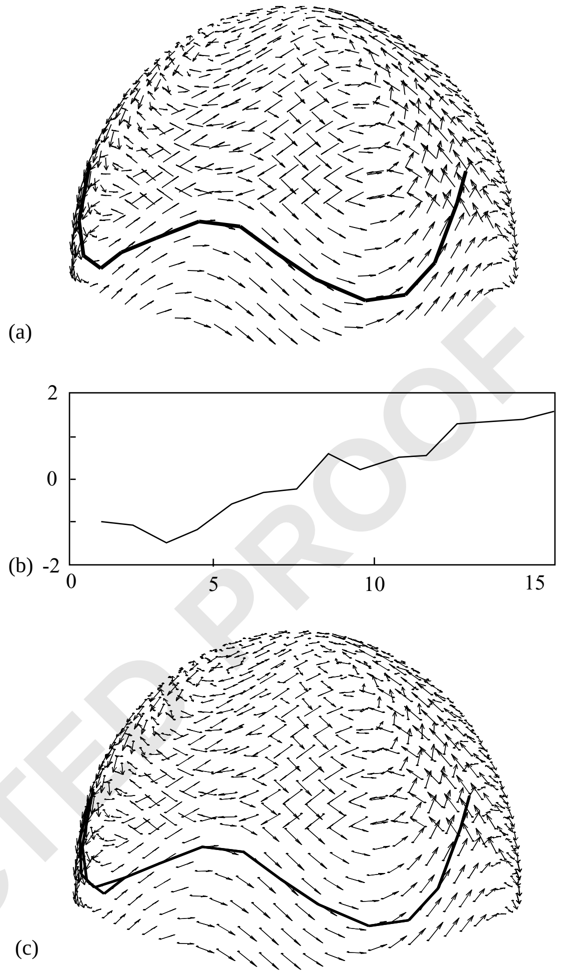

Fig. 6. Trajectory estimation on manifolds: (a) true trajectory (b) noisy observations (c) retrieved trajectory (arrows define the motion field; the starting point of each arrow corresponds to a centroid).

In this example, a velocity field was defined on the sphere. The velocity field is used to compute the allowed transitions among the centroids. Fig. 6a shows the centroids and the velocity field defined on the sphere surface. A state trajectory generated by the HMM is also displayed. Figs. 6b and c show the observation sequence associated to the trajectory and the trajectory estimate obtained by the DMA algorithm.

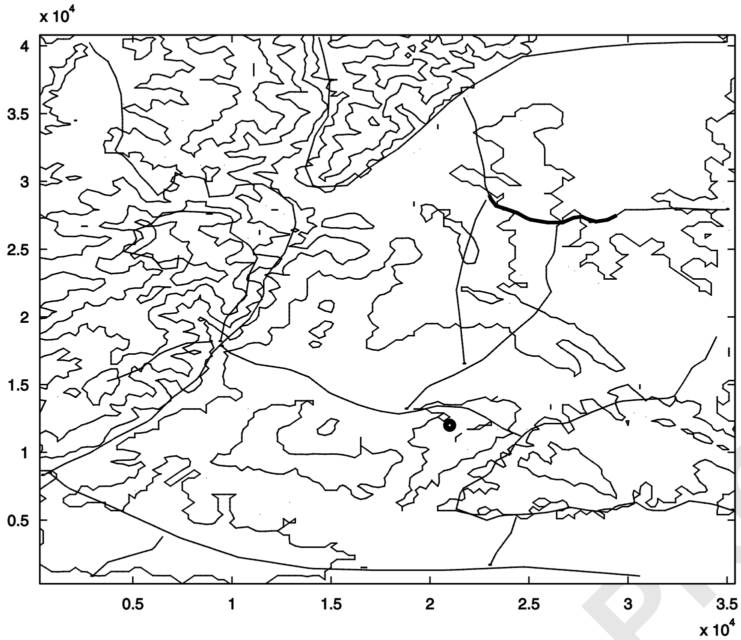

Fig. 7. Digital elevation map and road network (the car trajectory is displayed in bold and the receiver is identified by a circle).

A perfect reconstruction was achieved (the right centroids were selected), except at the first two instants of time. We stress that this problem cannot be solved by standard estimation methods which do not restrict the state to lie on the manifold.

4.2. Example 2

The second example considers the problem of estimating the position of a vehicle moving along trajectories in a three-dimensional manifold given by the roads in a map. It is assumed that only the distance to the receiver is known. Since the road map is assumed to be known, the first step of the DMA is not actually used in the example. Fig. 7 shows the manifold contour map together with the traject to be detected (bold line). The units in both scales are [m]. From the point located with a dot, radar observations are made according to the model

$$y_t = \cos(\omega d_t) + w_t, \tag{14}$$

where y_t is the signal measured at time t, $\omega = 0.02$, d_t is the distance to the observation point and w_t is white noise, $w_t \tilde{N}(0, \sigma^2 = 0.05)$.

Fig. 8 shows the estimated path of the vehicle yielded by the DMA algorithm.

4.3. Example 3

The third example illustrates the performance of the DMA algorithm in a tracking experiment. Fig. 9 shows the results of the DMA algorithm in the estimation of a slot-race car trajectory on a video sequence of 593 images obtained at 12 fps. For this sake a background image is subtracted from each new frame. The image difference is then compared to a threshold and the number of active pixels in the vicinity of each centroid is computed. Let y_t denote the tth image. It is assumed that

$$p(y_t/u_k) = \begin{cases} cn_k, & 0 \leqslant n_k \leqslant L^2, \\ 0, & \text{otherwise}, \end{cases} \tag{15}$$

where n_k is the number of active pixels in a $L \times L$ square region centered at u_k.

Figs. 9a and d show four consecutive images extracted from this sequence. Due to motion, the car is

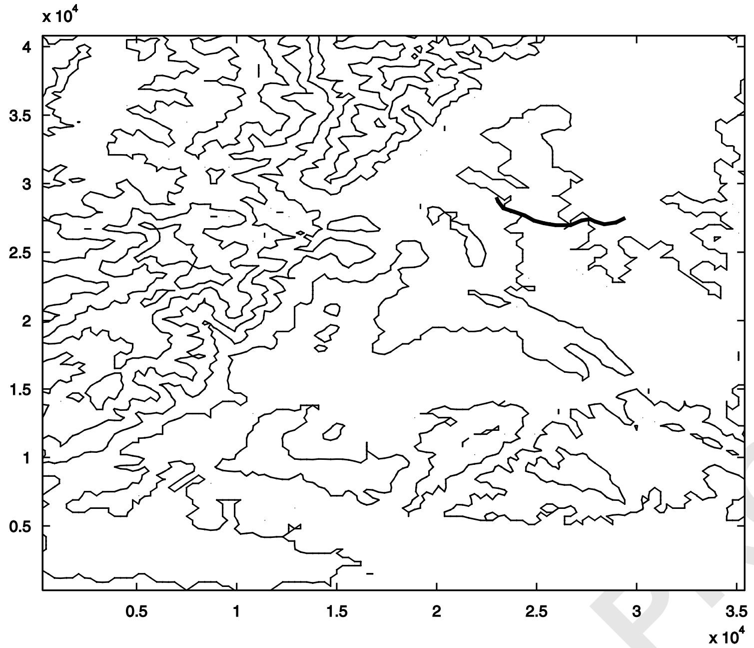

Fig. 8. Estimated trajectory.

not always easy to detect. This difficulty will be overcome by using the manifold and the motion restrictions. Fig. 9e shows the car trajectories used to train the model as well as the local frames computed by the DMA algorithm. The estimation of the car position in the training phase was performed manually. However, trajectory retrieval experiments use automatic image processing techniques.

The HMM parameters (transition matrix and sensor model) were estimated from the training set by the strong law of large numbers. Fig. 9f shows the tracking results obtained with the DMA algorithm for a sequence of 7 consecutive images. Since all the frames are known, the Viterbi algorithm was used for trajectory retrieval. The true locations are also displayed for comparison. Good results were achieved even when the car is occluded by the bridge.

None of the problems described in this section can be solved without the geometric information learned from the training set in an early stage of the procedure and defining the manifold on which the state vector is assumed to exist. This information is embodied in the centroid positions.

5. Conclusion

This paper exploits geometric restrictions for solving estimation problems in dynamic scene analysis. Although the primary motivation stems from image processing problems, the methods described may also be used in relation to control systems where parameters slowly move on a manifold due to changes in plant operating condition. An algorithm is proposed to estimate unknown state trajectories in manifolds. This algorithm denoted as discrete manifold analysis (DMA) allows to use the available information about the manifold geometry, as well as the motion dynamics and the sensor model. The manifold is split into disjoint regions, each of them being approximated by a hyperplane. This allows to reduce the dimensionality of the unknown data and to simplify the description and estimation of the state trajectories. A set of local hidden Markov models (HMM) is used to represent the state trajectories inside each region and the observation sequence. To allow transitions between different regions, a global HMM is defined by completing the local descriptions, valid inside the manifold regions, with transition models. The state trajectory is

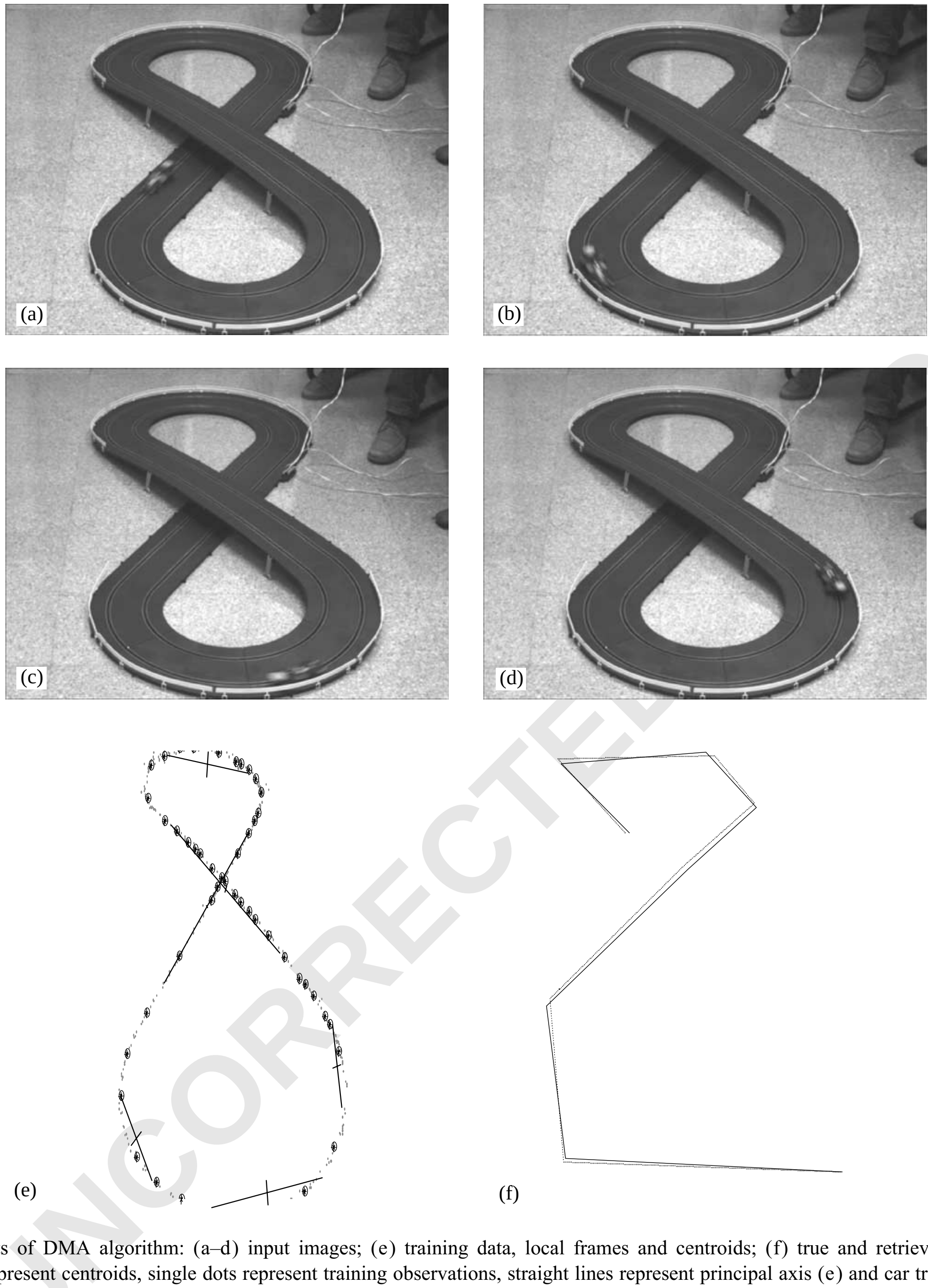

Fig. 9. Results of DMA algorithm: (a–d) input images; (e) training data, local frames and centroids; (f) true and retrieved trajectories (black dots represent centroids, single dots represent training observations, straight lines represent principal axis (e) and car trajectories (f)).

recursively estimated by dynamic programming. The experimental results presented in the paper show the ability of the DMA algorithm to estimate state trajectories on manifolds, exploiting the information on the manifold geometry as well as the motion and sensor information.

To sum up, the advantage of the proposed approach comes from: (1) the use of hyperplanes for manifold approximation; (2) discretization, which allows for the possibility of using the nonlinear HMM description.

Acknowledgements

The authors thank the anonymous reviewers for the valuable comments which helped to improve the paper.

References

[1] J. Biemond, R. Lagendijk, R. Mersereau, Iterative methods for image deblurring, Proc. IEEE (1990).
[2] A.F. Bobick, A.D. Wilson, A state-based approach to the representation and recognition of gesture, Trans. Pattern Anal. Mach. Intell. (December 1997) 1325–1337.
[3] C. Bregler, S. Omohundro, Nonlinear manifold learning for visual speech recognition, Internat. Conf. Comput. Vision (1995) 494–499.
[4] J. Clark, An introduction to stochastic differential equations on manifolds, in: Geometric Methods on Systems Theory, Reidel, Dordrecht, 1973.
[5] R. Marino, P. Tomei, Nonlinear Control Design, Prentice-Hall, Englewood Cliffs, NJ, 1995.
[6] H. Murase, S. Nayar, Visual learning and recognition of 3-D objects from appearance, Internat. J. Comput. Vision 14 (1995) 5–24.
[7] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, Numerical Recipes in C, Cambridge University Press, Cambridge, 1994.
[8] L. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE. (1989) 257–286.
[9] B. Ripley, Pattern Recognition and Neural Networks, Cambridge University Press, Cambridge, 1996.
[10] J. Rissanen, Universal coding, information, prediction and estimation, IEEE Trans. Inform. Theory (1984).
[11] R. Sara, R. Bajcsy, Fish-scales: representing fuzzy manifolds, Internat. Conf. Comput. Vision (1998).
[12] R. Szeliski, Bayesian Modeling of Uncertainty in Low-Level Vision, Kluwer Academic Publishers, Dordrecht, 1989.
[13] N. Winters, S. Santos-Victor, Omni-directional visual navigation, Internat. Symp. Intell. Robotic Systems, 1999.