Modeling and Classifying Human Activities from Trajectories Using a Class of Space-Varying Parametric Motion Fields

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

Abstract—Many approaches to trajectory analysis, such as clustering or classification, use probabilistic generative models, thus not requiring trajectory alignment/registration. Switched linear dynamical models (e.g., HMMs) have been used in this context, due to their ability to describe different motion regimes. However, these models are not suitable for handling space-dependent dynamics that are more naturally captured by nonlinear models. As is well known, these are more difficult to identify. In this paper, we propose a new way of modeling trajectories, based on a mixture of parametric motion vector fields that depend on a small number of parameters. Switching among these fields follows a probabilistic mechanism, characterized by a field of stochastic matrices. This approach allows representing a wide variety of trajectories and modeling space-dependent behaviors without using global nonlinear dynamical models. Experimental evaluation is conducted in both synthetic and real scenarios. The latter concerning with human trajectory modeling for activity classification, a central task in video surveillance.

Index Terms—EM algorithm, hidden Markov models, parametric models, trajectories, vector fields.

I. INTRODUCTION

Activity analysis has always been the foci of research in surveillance systems. More specifically, object motion trajectory-based recognition has aroused significant interest, due to its large spectrum of applications, not only in automatic video surveillance, but also in sign languages data segments, car navigation systems (CNS) and sports video trajectory analysis.

In this work we will focus on activity classification from trajectories performed by pedestrians in far field surveillance settings, i.e. in environments where the camera covers a wide area and it is not possible to extract detailed shape information about the object (i.e. pedestrians). The model herein proposed is a class of parametric motion fields with a switching mechanism. Among the possible patterns descriptors, the trajectory is one of most discriminant features [1]–[3] which well matches long range setup requirements. Recent advances in object tracking make possible to obtain reliable and long trajectories from video data. The object trajectories are a rich source of information to account for simultaneous behaviors, occurring at the same time in a given surveillance scenario and allow to analyze their behavior and identify suspicious or abnormal movements patterns that deviate from normal cases.

A generative model for trajectory modeling based on switched motion fields was recently proposed in [2]. The model comprises estimation methods that are able to learn a small set of motion fields describing typical behaviors of the objects. It is assumed that each trajectory is driven by one motion field at a given time instant, the so-called active field. Switching among active fields may occur at any time instant and position. The class of parametric motion fields proposed in [2], is rich enough to describe a wide variety of behaviors and simple enough to be learned from experimental data using closed form expressions.

This paper addresses the problem of trajectory model estimation as well as classification in far field surveillance scenarios with a fixed network camera. The proposed method represents the person trajectory as a concatenation of segments each of them generated by a simple parametric motion field. Given the video data of a parking lot or university campus we wish to estimate a small number and meaningfull velocity fields which explain most of the activity performed in that place. Activity classification is performed in a second stage using the information about the motion models which were activated. We also illustrate that the obtained parametric models, although simple, allow to obtain a flexible representation of complex pedestrian’s trajectories.

II. RELATED WORK

The Computer vision community has devoted an increasing interest to human activity analysis. Previous work concerning activity analysis can be framed depending on: (i) the nature of the approach, i.e. parametric or non-parametric, (ii) the use of a tracking module, (iii) short range or long range/far field settings, (iv) type of features used and (v) camera settings.

Previous work devoted to activity analysis can be roughly divided in two categories: parametric and non-parametric approaches [4]. Parametric approaches are either rule based or learned using supervised learning techniques, usually comprising visual features, such as, position velocity and appearance [5], [6]. Non-parametric approaches learn the activity patterns from statistical properties of the observed data, not explicitly defining the activity model [7]–[10].
Concerning the tracking issue, some approaches model the activities by directly extracting motion and appearance features from the video sequence [8], [11]–[13], without relying on the use of a tracking module. Nevertheless, the majority of the approaches assume that the objects are first detected and tracked throughout the sequence [7], [14]–[16] and the activities are then modeled as sequences of objects movements.

Short range interaction or far field settings also characterize the nature of the methods. In the first, the camera is close to people, where the person’s silhouette occupies significant part of the image region, typically more than 10%. In the second (i.e., long-range), the contour of the person occupies a small image region, typically less than 5%. Depending on which setting is being used, the type of features changes accordingly. Thus, in short-range scenarios, detailed measurements of human gestures and pose can be extracted. Several types of features have been used to characterize the human body in these scenarios: the head and hands positions [17], appearance models [18], texture [19], blobs [20], skeleton [21], silhouettes [22] or articulated models, taking into account geometric/motion restrictions [23]–[25]. In the second (i.e., far field), the camera covers a wide area and typically the captured video is of low resolution and poor quality. This precludes the computation of sophisticated features such as the above mentioned. Under this difficulty, only objects position are recorded through time along tracks, which are denoted as trajectories.

As already mentioned, in wide area surveillance settings it is not possible to obtain a detailed description of the observed persons, thus most methods rely solely on trajectories (usually of the center of “image mass” of the persons) obtained by tracking algorithms. Different trajectory analysis problems (such as classification and clustering) have been addressed using pairwise (dis)similarity measures between trajectories; these include Euclidean [26] and Hausdorff [27], [28] distances, and dynamic time warping [29]. Another class of approaches assumes that trajectories are produced by probabilistic generative models [30], [1], [31], [32], usually of the hidden Markov model (HMM) type. These approaches have the important advantage of not requiring alignment/registration of the trajectories being compared; moreover, they provide a solid probabilistic inference framework, based on which model parameters may be obtained from observed data.

The plethora of human activity recognition methods can also be divided in three categories, depending on the adopted camera setup. This comprises: (i) single-view, (ii) single-view/view-invariant and (iii) multi-view settings [33], [34]. The majority of human motion recognition methods that have been proposed use a single view fixed camera (see for instance [35]–[39]). All of these approaches are short range. However, they require the same camera view during both training and testing phases and if the person under study is captured from a different view the performance of these algorithms may decrease. To overcome this limitation, several works come up with single-view view invariant representation [40]–[43] or fully view-invariant multi-camera setups [44]–[46].

The model proposed in this paper can be classified in this framework. First, it is a parametric model. In fact, a new class of space-variant switched parametric motion fields model is proposed. Several types of object motion (translation, Euclidean similarity, rigid body and affine motion) are considered. The activity performed by a single object (i.e., pedestrian) is tracked and separated from other co-occurring activities, and the features related to the activity can be incorporated in the obtained tracks. The tracks are the consecutive mass-center positions of the detected bounding boxes. The images are acquired in a far field setup using a network of video surveillance cameras which are continuously recording the scene. More specifically, a network fixed camera located at IST university campus is used. Thousands of images were acquired corresponding to several hours of recording.

This work provides a valuable extension regarding the work in [47]. The improvements are as follows:

1) The vector fields used in [47] to describe object’s trajectories are constant and do not depend on the pedestrian position in the image. This is a heavy constraint especially when the model deals with complex motion patterns, jeopardizing the classification/recognition tasks. In the present work, space-variant vector fields are proposed which leads to flexible models and an expressive representation of complex trajectories.

2) Model switching is now described in a better way. In [47] switching between different vector fields is allowed and the switching probabilities are assumed to be constant in the whole image domain. This means that switching from one motion regime to another motion regime does not depend on the target position. The is obviously too restrictive. For example, think about a cross between two streets. Switching probabilities at the cross should be different than along the streets. This is now considered in this paper. Switching probabilities depend not only on the current active field but also on the pedestrian position. This allows a much better description of nonstationary trajectories in far-field surveillance scenarios.

3) Also, one of the limitations in [47] is that the model parameters are estimated in two stages: (i) a low level stage in which the translations as well as the covariances are estimated via EM with a MML criterion as proposed in [48]. Then, a second run of the EM, the high level stage, is necessary to obtain the transition matrices which are constant in the image region. Here, we propose a framework where all the model parameters are jointly estimated by the EM method with following advantages: (i) the transition matrices and the vector fields are both space-variant in the image domain and (ii) a family of a class of parametric motion models can be successfully described within this framework.

4) Finally the results are applied more general experimental setup, containing new far-field surveillance scenarios.

This paper is organized as follows. Section III describes the generative model. Section IV presents the family of the switched parametric models, that are the original contribution of the paper. Section VI explains how the parametric models are learned using the EM algorithm. Section VII
presents experimental validation in both synthetic and real data. Section VIII concludes the paper.

III. GENERATIVE MOTION MODEL

For the sake of simplicity, we assume that objects (pedestrians) may move freely in the image domain. We also ignore the discrete nature of digital images and model the object position at time $t$ by a vector $x_t$ in $\mathbb{R}^2$.

Let $T = \{T_1, \ldots, T_K\}$, with $T_k : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, for $k \in \{1, \ldots, K\}$, be a set of $K$ vector (velocity, i.e., displacement in one time unit) fields. The velocity vector at point $x \in \mathbb{R}^2$ of the $k$-th field is denoted as $T_k(x)$. At each time instant, one of these velocity fields is active, i.e., is driving the motion. Formally, each object trajectory is generated according to

$$x_t = x_{t-1} + T_k(x_{t-1}) + w_t, \quad t = 2, \ldots, L$$

where $k \in \{1, \ldots, K\}$ is the label of the active field at time $t$, $w_t \sim \mathcal{N}(0, \sigma^2_k I)$ is white Gaussian noise with zero mean and variance $\sigma^2_k$ (which may be different for each field), and $L$ is the length (number of points) of the trajectory. We denote by $\sigma = \{\sigma_1, \ldots, \sigma_K\}$ the set of noise variances. The initial position follows some distribution $p(x)$. According to the model just described, the conditional probability density of a trajectory $x = (x_1, \ldots, x_L)$, given the sequence of active models $k = \{k_1, \ldots, k_L\}$ is

$$p(x|k, T) = p(x_1) \prod_{t=2}^{L} \mathcal{N}(x_t|x_{t-1} + T_{k_t}(x_{t-1}), \sigma^2_{k_t} I)$$

The sequence of active fields $k = \{k_1, \ldots, k_L\}$ is modeled as a realization of a first order Markov process, with some initial distribution $P(k_1)$, and a space-varying transition matrix $P(k_t = j | k_{t-1} = i, x_{t-1}) = B_{ij}(x_{t-1})$, where $B : \mathbb{R}^2 \rightarrow \mathbb{R}_{+}^{K \times K}$ is a field of stochastic matrices,

$$B(u) = \begin{bmatrix}
B_{11}(u) & \cdots & B_{1K}(u) \\
\vdots & \ddots & \vdots \\
B_{K1}(u) & \cdots & B_{KK}(u)
\end{bmatrix} \label{eq:B}
$$

such that $\sum_j B_{ij}(u) = 1$, for any $u$ and any $i$. This model allows the switching probability to depend on the location of the object. The matrix-valued field $B$ can also be seen as a set of $K^2$ fields with values in $[0, 1]$, under the above mentioned constraint.

The joint distribution of a trajectory and the underlying sequence of active regions, is given by

$$p(x, k|T, B, \sigma) = p(x_1)P(k_1) \prod_{t=2}^{L} p(x_t|k_t, x_{t-1}, k_{t-1})$$

$$= p(x_1)P(k_1) \prod_{t=2}^{L} p(x_t|x_{t-1}, k_{t-1})P(k_t|k_{t-1}, x_{t-1}). \label{eq:joint}$$

Of course, $P(k_t|k_{t-1}, x_{t-1})$ is a function of $B$, $p(x_t|x_{t-1}, k_t)$ is a function of $T$ and $\sigma_k$, and $p(x_t, k_t|x_{t-1}, k_{t-1})$ is a function of $T$, $B$, and $\sigma_k$.

IV. PARAMETRIC MOTION FIELDS

For surveillance applications, the proposed model has the advantage of being intuitive and interpretable, since each velocity field can be observed and describes a different type of motion in the scene. Furthermore, this information can be used, for example, by the manager of a public area to characterize the typical ways in which people move in that area.

V. SPACE-VARYING SWITCHING MATRIX FIELD

The previous fields generate segments of trajectories. However, we allow switching among different motion fields. In fact, the active model label is a Markov process with a space-varying switching matrix $B(x)$. This is a field of stochastic matrices which will be learned from the data.

### Table I

<table>
<thead>
<tr>
<th>Class of Parametric Motion Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Type</td>
</tr>
<tr>
<td>-------------</td>
</tr>
<tr>
<td>Translation</td>
</tr>
<tr>
<td>Rigid body</td>
</tr>
<tr>
<td>Euclidean similarity</td>
</tr>
<tr>
<td>Affine</td>
</tr>
</tbody>
</table>


We represent the field of stochastic matrices using a set of scalar basis functions, \( \psi_m(x) : \mathbb{R}^2 \to \mathbb{R} \), for \( m = 1, \ldots, M \),

\[
   
   \mathbf{B}(x) = \sum_{m=1}^{M} b^{(m)} \psi_m(x) 
   \]

(7)

where each “coefficient” \( b^{(m)} \in \mathbb{R}^{K \times K} \) is a stochastic matrix, i.e., for any \( m = 1, \ldots, M \), and any \( p, q = 1, \ldots, K \),

\[
   \sum_{k=1}^{K} b_{p,k}^{(m)} = 1 \quad \text{and} \quad b_{p,q}^{(m)} \geq 0
   \]

(8)

Of course, we must guarantee that this representation preserves the stochastic nature of matrix \( \mathbf{B}(x) \). A sufficient condition for all the entries of the expansion to be non-negative is that \( \psi_m(x) \geq 0 \), for all \( x \) and all \( m = 1, \ldots, M \). Moreover, since each \( b^{(m)} \) is a stochastic matrix,

\[
   1 = \sum_{k=1}^{K} \sum_{m=1}^{M} b_{p,k}^{(m)} \psi_m(x) = \sum_{m=1}^{M} \psi_m(x) \sum_{k=1}^{K} b_{p,k}^{(m)} = \sum_{m=1}^{M} \psi_m(x).
   \]

In conclusion, \( \mathbf{B}(x) \), as given by (7), is a stochastic matrix if the basis functions verify the following conditions: at any point \( x \), \( \psi_m(x) \geq 0 \), for all \( i \); and

\[
   \sum_{m=1}^{M} \psi_m(x) = 1.
   \]

These two conditions are known as partition of unity property and are satisfied by B-spline basis functions [50], of which bilinear interpolating functions are a particular simple case, adopted in this paper.

Specifically, the image domain \( \mathcal{L} = [0, 1]^2 \) is discretized using a uniform grid with step \( \Delta \) and nodes \( x_m = (x_m, y_m) \). The basis function \( \psi_m(x) \) centered at the node \( x_m \) is given by

\[
   \psi_m(x) = \begin{cases} 
   |x - x_m| \cdot |y - y_m| / \Delta^2, & \text{if } |x - x_m| < \Delta \wedge |y - y_m| < \Delta \\
   0 & \text{otherwise}
   \end{cases}
   \]

If we plot this function it looks like a small pyramid located at each grid node.

VI. LEARNING THE VECTOR FIELDS VIA EM

We now address the problem of estimating the set of velocity fields \( \mathcal{T} \), the field of transition matrices \( \mathbf{B} \) and the set of noise variances \( \sigma \) from a set of \( S \) observed (independent) trajectories \( \mathcal{X} = \{x_1^{(j)}, \ldots, x_S^{(j)}\} \), where \( x_1^{(j)}(x_1^{(j)}, \ldots, x_S^{(j)}) \) is the \( j \)-th observed trajectory, assumed to have length \( L_j \). Naturally, we assume that the corresponding set of sequences of active fields, \( \mathcal{K} = \{k_1^{(j)}, \ldots, k^{(j)}\} \), is not observed (it is hidden).

A. Estimation Criterion: Marginal MAP

The fact that the active field labels \( \mathcal{K} \) are missing suggests using the EM algorithm to find a marginal maximum a posteriori (MMAP) estimate of \( \theta \) under some prior \( p(\theta) = p(T)p(B)p(\sigma) \); formally,

\[
   \hat{\theta} = \arg \max_{\theta} \left[ \sum_{K} \prod_{j=1}^{S} p(x_1^{(j)}, k^{(j)} | \theta) \right] p(\theta)
   \]

(9)

where \( p(\theta) \) is the prior on the parameters. Notice that the sum with respect to \( K \) is over all possible sequences of labels of adequate length, so this sum cannot be explicitly computed since it involves an exponential number of terms. Next, we will derive the complete likelihood which characterizes the EM algorithm, which solves (9) without explicitly performing this marginalization. We assume that the initial distributions \( p(x_1) \) and \( P(k_1) \) are known, thus do not have to be estimated; this assumption could be easily relaxed.

B. Complete Likelihood

The EM algorithm is based on the conditional expectation of the complete log-likelihood

\[
   \log p(\mathcal{X}, \mathcal{K} | \theta) = \sum_{j=1}^{S} \log p(x_1^{(j)}, k^{(j)} | \theta)
   \]

(10)

where each term \( p(x_1^{(j)}, k^{(j)} | \theta) \) is given by (3). As is common and convenient in dealing with missing labels, we use binary indicator variables, defined as follows: each label \( k_t^{(j)} \in \{1, \ldots, K\} \) (the active field at time \( t \) of trajectory \( j \)) is represented by a binary vector \( y_t^{(j)} = (y_t^{(j)}, \ldots, y_{t,K}^{(j)}) \in \{0, 1\}^K \), where \( y_t^{(j)} = 1 \Leftrightarrow k_t^{(j)} = 1 \). With this notation, the complete log-likelihood becomes

\[
   \mathcal{L} = \log p(\mathcal{X}, \mathcal{K} | \theta)
   \]

\[
   = C + \sum_{j=1}^{S} \sum_{t=2}^{L_j} \sum_{l=1}^{K} \log \mathcal{N}(x_t^{(j)} | x_{t-1}^{(j)}, \theta) + T_t(x_t^{(j)}, \sigma_l^2 I)
   \]

\[
   + \sum_{j=1}^{S} \sum_{t=2}^{L_j} \sum_{l=1}^{K} \sum_{g=1}^{K} y_{t-1,g}^{(j)} y_{t,l}^{(j)} \log B_{g,l}(x_{t-1}^{(j)})
   \]

(11)

where \( C = \sum_{j=1}^{S} \log p(x_1^{(j)}) + \log P(k_1^{(j)}) \) is a constant.

C. EM Algorithm

The E-step aims at computing the conditional expectation of the complete log-likelihood given in (11), given the current estimates of the parameters \( \hat{\theta} \), i.e., \( Q(\hat{\theta} | \theta) = \mathbb{E} \left[ \mathcal{L}(\mathcal{X}, \mathcal{K} | \theta) \right] \). This leads to the computation of the conditional expectations with respect of missing binary indicators \( y_t^{(j)} \) and switching indicators \( y_{t-1,g}^{(j)} y_{t,l}^{(j)} \). These probabilities are obtained by a modified forward-backward procedure [51]. Recall that the transition matrix is not constant, but depends on the trajectories.

In the M-step, the field and parameter estimates are updated according to

\[
   \hat{\theta}_{\text{new}} = \arg \max_{\theta} Q(\theta | \hat{\theta}) + \log p(\theta)
   \]

(12)

In this section we study this maximization in detail, as well as the adopted priors, by looking separately at the maximization with respect to each component of \( \theta = (\mathcal{T}, \mathbf{B}, \sigma) \).
1) Updating $\hat{\theta}$: For $\sigma$ we adopt flat priors, i.e., we are looking for usual maximum likelihood noise variance estimates. Computing the partial derivative of $Q(\theta; \hat{\theta})$ with respect to each component $\sigma_k^2$ of $\sigma$, and equating to zero, we obtain

$$
(\hat{\sigma}_k^2)_{\text{new}} = \frac{\sum_{j=1}^{S} \sum_{i=2}^{L_j} \tilde{y}_{i,k}^{(j)} (x_{i-1}^{(s)} - x_{i-1}^{(s)})^2}{\sum_{j=1}^{S} \sum_{i=2}^{L_j} \tilde{y}_{i,k}^{(j)}}
$$

for $k = 1, \ldots, K$.

2) Updating $\hat{\mathbf{B}}$. The parametric motion fields estimates depend on the model being used (see Table I). Details concerning the estimation of the motion fields can be consulted in Appendix.

D. Updating $\hat{\mathbf{B}}$

Using the representation described in Section V, we change the problem of estimating $\mathbf{B}$ into the problem of estimating a set of stochastic matrices $\mathbf{B} = \{b^{(1)}, \ldots, b^{(M)}\}$, by maximizing $Q(\theta; \hat{\theta}$, under the constraint (8). Inserting (7) into (11), and dropping all terms that do not depend on $\mathbf{B}$ (equivalently, on $\hat{\beta}$), the objective function can be written as

$$
\mathcal{E}(\mathbf{B}) = \sum_{j=1}^{S} \sum_{i=2}^{L_j} \sum_{k=1}^{K} \sum_{g=1}^{L_i} \log \sum_{m=1}^{M} b_{g,l}^{(m)} \varphi_m(x_{i-1}^{(s)}) \tag{14}
$$

where $\varphi_m(x_{i-1}^{(s)}) = \sum_{q=1}^{Q} b_{p,q}^{(m)} \psi_q(y_{i-1}^{(s)})$. Equation (14) should be maximized under the constraint in (8). The Lagrangian for this constrained problem is

$$
\mathcal{E}(\mathbf{B}) + \sum_{m=1}^{M} \sum_{p=1}^{K} \lambda_{mp} \left( \sum_{k=1}^{K} b_{p,q}^{(m)} - 1 \right)
$$

where the $\lambda_{mp}$ are Lagrange multipliers. Since it is not possible to analytically solve the system of (non-linear) equations resulting from equating the gradient of this Lagrangian to zero, we directly attack the constrained problem, using the gradient projection (GP) algorithm [52].

The two main ingredients of the GP algorithm are the computation of the gradient of the objective function and the projection onto the constraint set. Concerning the gradient, it is simple to compute the partial derivatives of $\mathcal{E}(\mathbf{B})$ with respect to the $b_{g,l}^{(m)}$, which are given by

$$
\frac{\partial \mathcal{E}(\mathbf{B})}{\partial b_{g,l}^{(m)}} = \sum_{j=1}^{S} \sum_{i=2}^{L_j} \tilde{y}_{i,k}^{(j)} \varphi_m(x_{i-1}^{(s)}) B(x_{i-1}^{(s)}) \psi_q(y_{i-1}^{(s)})
$$

and constitute the elements of the gradient. The second ingredient is the projection of a matrix onto the set of stochastic matrices. Notice that this projection is equivalent to projecting each row of the matrix onto the probability simplex; for this purpose, we use a fast $O(K \log K)$ algorithm which was recently proposed [53].

In summary, $\mathbf{B}_{\text{new}}$ is obtained by minimizing $\mathcal{E}(\mathbf{B})$ under the constraints in (8), using the GP algorithm, with the projection step carried out by the algorithm described in [53]. An alternative approach consists of using the natural gradient in the Riemannian space of stochastic matrices as proposed in [54]. A comparison between both approaches is outside the scope of this paper.

VII. EXPERIMENTAL RESULTS

This section presents experimental results illustrating the performance of the trajectory model with multiple parametric motion fields. This section is split into two parts: (i) trajectory modeling and (ii) activity recognition based on the pedestrian trajectories. In the first part, we illustrate the performance of the proposed model with multiple motion fields applied to synthetic data and to real data. The real data was obtained in two different surveillance setups. Then, in the second part, we address the classification of trajectories in both setups.

A. Modeling Trajectories

1) Synthetic Data: The first synthetic example is shown in Fig. 1. This figure shows 100 trajectories generated by model (1), using a mixture of 5 known vector fields. This example simulates a roundabout with four entries and exits, combining linear and circular motions. In this problem, no attempt is made to automatically determine the number of motion models. We assume that this information is known beforehand.

Fig. 2 shows the motion fields estimated by the EM method. Two parametric models were considered: the rigid body model (top row) and the affine model (bottom row). Both models are capable of correctly estimating the motion fields. The rigid model needs 5 motion fields to represent all the trajectory displacements while the affine model needs a smaller number of fields (three) and provides a more compact description. We stress that these motion field estimates were obtained with no a priori knowledge concerning which model is active in each trajectory, at each time instant.

Fig. 2 allows a qualitative evaluation of the model accuracy. However, an objective evaluation is needed. In order to assess the model performance, we will measure its ability in predicting the target position one step ahead in time. Since the trajectory is generated according to dynamic model (1),

![Fig. 1. Synthetic trajectories for the roundabout problem.](image-url)

![Fig. 2. Estimated fields for the roundabout example using rigid body (top row) and affine mode (bottom row). We assume that the number of vector fields is known.](image-url)
the predictor and the prediction error are given by

\[
\hat{x}_t = x_{t-1} + T_k(x_{t-1})
\]

(15)

\[
\hat{e}_t = x_t - x_{t-1} - T_k(x_{t-1})
\]

(16)

These equations assume that the active field \( k_t \) is known. Since we do not know this information in practice, we will select the error with smallest norm (ideal switching).

This leads us to define a signal-to-noise-ratio (SNR) measure given by

\[
\text{SNR} = 10 \log_{10} \left( \frac{\sum_{t=2}^{L} \|x_t - x_{t-1}\|^2}{\sum_{t=2}^{L} \min_k \|x_t - x_{t-1} - T_k(x_t)\|^2} \right)
\]

(17)

Next, we will study the performance of the multiple motion field model when we vary some of the parameters. First we will consider the effect of the input noise variance \( \sigma^2 \) on the synthesized trajectories and model estimates. Fig. 3 shows trajectories generated with three values of the noise variance \( \sigma^2 \in \{0.001, 0.005, 0.01\} \). The uncertainty increases when the variance increases.

We trained multiple motion field models for the three testing conditions using both the rigid body model (5 fields) and the affine model (3 fields). The SNR results are shown in Table II. As expected, both methods decrease their performance with the increase of the input noise variance. Both models achieve comparable SNR scores.

Another important issue concerns the estimation of the field of switching matrices. This field is represented in a non-parametric way using a regular grid: we estimate a switching matrix for each grid node using the EM algorithm. The grid size is therefore a key parameter which influences the complexity and accuracy of the switching matrix estimation.

We performed several tests to assess the role of the grid size and in each case we varied the input noise variance as before. We assumed that the grid has \((2S_g + 1) \times (2S_g + 1)\) nodes with \(S_g \in \{1, \ldots, 9\} \). We run the EM algorithm for each grid size, using 4 different initializations and computed the average SNR.

Figure 4 shows the average SNR results for different values of \( \sigma^2 \) and for different grid sizes. As expected, the SNR tends to increase when the grid size increases. However the improvement is small and this is surprising since we would expect a strong influence of the grid size on the performance of the method.

This raises one question: why do we obtain good SNR results with small grid sizes (e.g. 3 x 3 nodes)? Under this condition the switching matrix cannot be well estimated, since we have a very limited capability to represent spatial changes in the image domain (i.e. low resolution grid). However, notice that the metric (17) is insensitive to the switching mechanism. In fact, we are computing the smallest prediction error (not the most probable one), and we are basically measuring the accuracy of the best motion field for each motion displacement. The conclusion is: motion fields are well estimated even when the switching matrix is represented in a very coarse way.

Fig. 4 can also be used to select the best parameters. In the case of the rigid body model (Fig. 4 left), the best results are achieved for \( S_g = 7 \) (grid size 15 x 15) when \( \sigma^2 = 0.001 \) and \( S_g = 5 \) (grid size 11 x 11) when \( \sigma^2 = 0.005 \). However, this requires a high computational cost. As the dynamical noise increases the SNR differences among the grid dimension tend to be less noticeable as shown by the dotted line.

In the case of affine model (Fig. 4 right), we also observe an increase of the SNR values for higher dimensions of the grid. A good tradeoff is \( S_g = 5 \), i.e. grid of 11 x 11 nodes. This value will be used in experiments with real data in Section VII-B.1.

2) Real Data: We now consider the application of the proposed model in two real scenarios (see Fig. 5, with trajectories superimposed): (i) IST campus (Lisbon), and the (ii) UPC campus (Barcelona)\(^1\). In both cases, the images were acquired using a fixed camera.

In the IST campus 1 hour of recording (taken in different days) allowed to obtain \( \approx 6.9 \times 10^4 \) frames. In this case a local camera was used with frame rate 25 images/sec. In the UPC scenario, the images were obtained from a remote and fixed network camera located at UPC campus. Thousands of images (\( \approx 1.3 \times 10^5 \)) were acquired corresponding to 4 hours of recording at approximately frame rate 10 images/sec\(^2\).

Before applying the proposed model to estimate a set of

\(^1\)UPC images were acquired in the scope of the EU project FP6-EU-IST-045062 URUS - Ubiquitous Networking Robotics in Urban Settings.

\(^2\)This low frame rate is due to the limitations of the cameras network and due to the fact that it is not intended to store much data.
motion fields, we need to extract the trajectories from the video sequences by tracking the pedestrians. For that purpose, we used the Lehigh omnidirectional tracking system (LOTS) to detect regions, followed by region association, as in [55]. The trajectories are then projected onto a view orthogonal to the ground plane (the so called bird’s eye view) to enforce viewpoint invariance. This is done using a projective transformation (homography) from the image onto a plane parallel to the ground. The number of the trajectories are 150 (for the IST campus) and 270 (for the UPC scenario).

We used the model proposed in this paper to extract a set of parametric motions fields describing the human activities in each of these scenarios. For the sake of simplicity, we used a subset of the whole set of trajectories and applied the EM algorithm described in Section VI. Furthermore, we assume that the number of models is known. This restriction will be relaxed later in the next section when we apply the model in activity classification.

For the sake of simplicity and illustration purposes we used a subset of trajectories as depicted in Fig. 6. This allows to obtain a smaller set of estimated motion fields. We present results concerning both rigid body and affine models for the two scenarios.

Fig. 7 shows the motion fields estimates obtained for the IST scenario using the rigid body model. In this example we used 4 motion models since they are enough to obtain an acceptable representation of all the observed trajectories. The final estimates depend on the EM initialization. In fact, there are several configurations of the motion fields which explain the observed trajectories and may appear as final estimates.

Fig. 8 shows the motion field estimates obtained using the affine model. In this case we assumed that motion could be described by two motion fields. Comparing both figures, we conclude that the motion field shown in Fig. 8 (a) roughly corresponds to the motion fields displayed in Fig. 7 (b), (d). Similarly, the motion field shown in Fig. 8 (b) corresponds to the fields of Fig. 7 (a), (c).

The same procedure is conducted for the second scenario. Fig. 9 shows the results for the UPC scenario using three models. On the left image we can see an “up” motion field, a “circular” motion field (Fig. 9 (b)), and “down-left” motion field (Fig. 9). Fig 10, shows the results on the same set of trajectories, but now using the rigid body. As previously, the same motion fields are obtained, although differently organized. The motion fields in Fig 10 (a) correspond to the ones obtained in Fig. 9 (a), (b); the motion fields in Fig 10 (b) correspond to the obtained motion in Fig. 9 (b), (c).

As previously, we also compute the SNR for both models using the fields shown the Figs. 7, 9, 8 and 10. The results are given in Table III. Both models exhibit quite similar performance, and the SNR results are comparable with the
ones shown in Table II (third line) for the case of significantly deformed trajectories.

**B. Classifying Human Activities**

This section addresses the application of the multiple vector fields in human activity classification. Fig. 11 shows the same set of trajectories (see Fig. 5) but each trajectory has a color label according to the trajectory-class, or activity.

The following class of trajectories are considered in each scenario:

1) **IST campus:** In this scenario the following class of trajectories (activities) are observed (shown in Fig. 5 (left)): (i) entering the central building (pink lines - diagonally up direction), (ii) crossing downwards (green lines - diagonally down direction), (iii) crossing upwards (cyan lines - diagonally up direction) (iv) walking along (yellow lines - down direction) and (v) crossing straight (red lines).

2) **UPC campus:** The trajectory classes shown in Fig. 5 (right) are: $a_1 \rightarrow$ walking and stepping down the stairs (pink lines - from up-left to bottom right direction) $a_2 \rightarrow$ walking along (yellow lines - down direction) $a_3 \rightarrow$ crossing and stepping down the stairs, (green lines - left to right direction), $a_4 \rightarrow$ pass diagonally, (red lines - left to right direction) and $a_5 \rightarrow$ turning the campus (cyan lines - clockwise direction).

We will now discuss the trajectory classification algorithm. The generative model introduced in Section III can be cast into a maximum a posteriori classifier. For that purpose, we split the set of trajectories according to the corresponding activities. This leads to a classified training set

$$T = \{X^{(a)}, a = 1, \ldots, A\},$$

where $X^{(a)}$ is the set of trajectories associated to activity $a$ and $A$ denotes the number of different activities. Then, we estimate $A$ generative models $\{\hat{\theta}^{(a)}, a = 1, \ldots, A\}$, each of them associated to a different activity, using the EM algorithm described in Section VI-C. Finally, given a new trajectory $x = (x_1, \ldots, x_L)$, the MAP activity classifier is given by

$$\hat{a}(x) = \arg \max_{a \in \{1, \ldots, A\}} p(x|\hat{\theta}^{(a)})P(a)$$

where $P(a)$ is the prior probability for activity $a$ (here we assume $p(a) = 1/A$), and $p(x|\hat{\theta}^{(a)})$ is the probability density function of the trajectory $x$ under the model with parameters $\hat{\theta}^{(a)}$, which is computed using the forward backward procedure.

In this study, we first determine the parameters to be used in the algorithms. We assume that the training stage should be as automatic as possible. In addition we would like to have a fast training stage, depending on a small number of trajectories. Fast training allows us to efficiently test many different configurations for the model parameters such as: number of velocity fields, type of parametric motion models, grid resolution, noise variances and initialization of the switching matrix. All these parameters will be tested and all possible configurations will be considered.

1) **Discriminative Parameters Selection:** As already described in Section IV, the model parameters that characterize the multiple vector field model are defined by the triplet $\theta = (T, \mathbf{B}, \sigma)$, which includes the parameters of the motion fields $T$, noise standard deviations $\sigma$ and space-varying switching matrices $\mathbf{B}$. Besides this set of parameters, we also have to specify the number of motion models $M$. Opposing with Section VII-A where we heuristically chose the number of motion fields for illustration purposes, now this parameter has to be automatically selected instead. To accomplish this, the underlying assumption used in this paper is that, we make use of the knowledge that the obtained model is going to be used for a specific task, in this case, a classification task. Particularly, pedestrian motion is characterized by trajectories which are modeled by vector fields. The idea, is thus, to select the generative model that achieves the best classification performance. This reasoning
is also applied for the remaining parameters in $\theta$. Thus, the parameter selection is discriminative since their choice will lead to the best classification accuracy.

We vary the number of motion vector fields $K$ from 1 to 8. For each value of $K$, we define an interval noise variance $\sigma^2 = \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$. The diagonal elements of the transition matrix are set to $b_{ii} = 1 - c$, with $c = (K - 1)/10$ while the non-diagonal elements are set to $b_{ij} = (1 - b_{ii})/(K - 1)$ initially allowing all kinds of transitions between vector fields at all space positions. The vector field $T$ is random in the interval $[-0.1, 0.1]$. Therefore, a velocity is randomly chosen and is equal for every nodes in the grid.

From the extensive experiments done with this model, we noticed that the initialization of the vector field $T$ is the issue that most influences the EM convergence. For this reason, we performed four different initializations of $T$.

We have also tried different values for the transition matrix. However, from our experiments, we did not observe significant changes in the algorithm performance unless $c$ is significantly reduced and becomes close to zero (e.g., $c = (K - 1)/100$). In this case, the classification performance decreases and the classifiers becomes more unstable for some of the parametric models. We have therefore considered these initialization values for $c$. Table IV summarizes different initializations for each parameter considered.

Using the above setup, we performed a total of 256 experiments (1024 runs) for each type of parametric model using a validation test. This set contains approximately 35% of the trajectories in both scenarios. We kept in mind that the validation set must contain samples of all the class-trajectories. Figs. 12, 13, 14 and 15 report the mean classification scores obtained in this study for the four parametric motion models: translation, rigid body, Euclidean similarity and affine. In each case, two figures are presented corresponding to two different initializations of matrix $B$ in all grid nodes. The first image (top row of the images) corresponds to $c = (K - 1)/10$ and the second image (bottom row of the figures) corresponds to $c = (K - 1)/100$. Next, we show the results obtained for all models.

To obtain the parameter values we took just three training trajectories for each pedestrian activity. This number of trajectories was chosen since it is the minimum value that provides a classification score above 80%. More specifically, a decrease of 10% and 14% was obtained using two and one training trajectories per class, respectively. Also, using more than three training trajectories per class leads to no significant classification improvement.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of models $K$</td>
<td>${1, 2, \ldots, 8}$</td>
</tr>
<tr>
<td>Noise variance $\sigma^2$</td>
<td>${10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}}$</td>
</tr>
<tr>
<td>Translation $t$ of the vector field $T$</td>
<td>random in $[-0.1, 0.1]$</td>
</tr>
<tr>
<td>Transition matrix $B$</td>
<td>$b_{ii} = 1 - \frac{K - 1}{10}$, $b_{ij} = \frac{1 - b_{ii}}{K - 1}$</td>
</tr>
</tbody>
</table>

Fig. 12. (a) Classification performance for the translation model in the UPC campus sequences for several initializations of the covariance matrix: red (one model), green (two models), blue (three models), yellow (four models), pink (five models), cyan (six models), black (seven models), and brown (eight models). (b) Box plot results for the classification for different number of models. Initialization of the transition matrix (see also Table IV) $b_{ij} = 1 - \frac{K - 1}{100}$ (top row), and $b_{ij} = 1 - \frac{K - 1}{10}$ (bottom row).

Fig. 13. (a) Classification performance for the rigid body in the UPC campus sequences for several initializations of the covariance matrix: red (one model), green (two models), blue (three models), yellow (four models), pink (five models), cyan (six models), black (seven models), and brown (eight models). (b) Box plot results for the classification for different number of models. Initialization of the transition matrix (see also Table IV) $b_{ij} = 1 - \frac{K - 1}{100}$ (top row), and $b_{ij} = 1 - \frac{K - 1}{10}$ (bottom row).
Fig. 14. (a) Classification performance for the Euclidean similarity body in the UPC campus sequences for several initializations of the covariance matrix: red (one model), green (two models), blue (three models), yellow (four models), pink (five models), cyan (six models), black (seven models), and brown (eight models). (b) Box plot results for the classification for different number of models. Initialization of the transition matrix (see also Table IV) $b_{ii} = 1 - \frac{K}{100}$ (top row), and $b_{ii} = 1 - \frac{K}{1000}$ (bottom row).

Fig. 15. (a) Classification performance for the affine body in the UPC campus sequences for several initializations of the covariance matrix: red (one model), green (two models), blue (three models), yellow (four models), pink (five models), cyan (six models), black (seven models), and brown (eight models). (b) Box plot results for the classification for different number of models. Initialization of the transition matrix (see also Table IV) $b_{ii} = 1 - \frac{K}{10}$ (top row), and $b_{ii} = 1 - \frac{K}{100}$ (bottom row).

To obtain the results in the Table V we run the EM algorithm 8 times, keeping the 4 best estimates (the mean estimates over these best estimates are shown).

As a final remark, and to take in consideration the parameter $\mathbf{T}$, in each run of the EM a different random initializations
for this parameter is used (as we did in the discriminative procedure).

C. Comparison With Other Related Methods

This section provides a comparison between the parametric models herein proposed and the switched dynamical hidden Markov model (SD-HMM) proposed in [47]. The comparison is made in terms of: (i) performance accuracy, (ii) number of motion models $K$ and (iii) running time complexity.

1) Performance Accuracy: As mentioned in Section II, the SD-HMM is characterized by the following two main stages: (i) the estimation of the (low level) parameters of the model, i.e. mean and covariance matrices of the normal distributions, which are estimated via EM with a minimum message length (MML) criterion that allows to automatically discover the model order $K$ of the mixture, and (ii) the estimation of the switching matrices using a class-dependent HMM, that requires a second use of the EM. The first step in (i) does not depend on the class activity, since all the translations and covariances are shared among all classes. In (ii) the high level parameters of the model (i.e. transition matrices) are estimated, being dependent on the class-activity. One particularity of the SD-HMM is that, in the first stage aforementioned, four different options are made available concerning the structure of the covariance matrices. The options are the following: (1) free covariances, (2) diagonal covariances, (3) common covariance for all components and (4) common diagonal covariances for all components.

As previously, and for the sake of a fair comparison, a 5-fold-cross-validation is also adopted, where all the training class trajectories are presented in each fold as in the previous experiment. Again, we run the EM eight times (concerning the above two stages), and keep the four best scores for each option when evaluating the performance accuracy. Taking the overall mean performance of the methods (i.e. the average of the diagonal entries of the confusion matrix) the following values are obtained for the different operating modes: (1) 75.83%, (2) 77.96%, (3) 79.49%, (4) 62.52%.

Table VI discriminates the results obtained only for the best option (3rd option) in the SD-HMM. It is seen the difficulty of this method to discern between the activities having similar motion. More specifically, the algorithm does not provide high accuracy between the activity $a_1 \rightarrow$ walking and stepping down the stairs and $a_4 \rightarrow$ pass diagonally. This happens, since these two activities have the same “shape”, where in both, the pedestrians follow an up-left to bottom-right direction (notice the pink and red lines in Fig. 11 right). The only difference resides in the image region. Indeed, this is the major roadblock of this approach, i.e. the switching matrix is not space-dependent. This jeopardizes the performance of the SD-HMM in these situations. Similar behavior happens within the above activities and the activity $a_3 \rightarrow$ crossing and stepping down the stairs that also has similar motion, i.e. from left to the right direction with a smaller slope (see green lines in Fig. 11 right). Comparing with the results in Table V these ambiguities are overpassed thanks to the space-variant feature of the switching mechanism. Also, note that a 100% accuracy is achieved for both methods in the activity $a_5 \rightarrow$ turning the campus. This happens since this activity “flows” in a clockwise direction which is completely different from the other activities in the UPC scenario.
2) **Number of Models K**: The performance accuracy described in the previous sub-section VII-C.1, was accomplished varying the number of models in the interval $K = \{1, \ldots, 8\}$ for the SD-HMM method. The first stage of the SD-HMM, is able to automatically provide the most suited number of the low level dynamical models from the above interval. Since we go through a 5-fold-cross-validation, the number $K$ may vary for each fold. We computed the statistics of this parameter for all the operating modes of the SD-HMM. In Fig. 16 each box concerns the uncertainty about the median over the four runs of the EM. We concluded that the proposed framework (in all of the parametric models) is able to produce higher accuracy scores with less number of models (specially for the Euclidean and affine models where only $K = 4$ models are used).

3) **Running Time Complexity**: We also computed the running time spent of both methodologies to produce the estimates. In the proposed framework we used 20 iterations for the EM, whilst we find acceptable using 10 iterations for the EM in the SD-HMM approach. Table VII summarizes (third column) the obtained results. We conclude that the proposed framework is also competitive regarding this issue. Recall that, the per-iteration running time of the translation ($\approx 1.17$ sec.) and rigid body ($\approx 1.15$ sec.) models are comparable with the SD-HMM ($\approx 1.06$ sec.). However, the Euclidean ($\approx 0.80$ sec.) and affine ($\approx 0.78$ sec.) motion models exhibit faster running times, since they use a smaller number of models $K$ (i.e. $K = 4$ instead of $K = 6$). This faster computation is due to the complexity of the approach which is quadratic in the number of the grid nodes $n$, and linear in the number of the motion models $K$. More specifically, the complexity of the proposed approach comprises the following operations: (i) computation of the vector fields $T$ with the complexity $O_T(dDK)$ where $d$ stands for two-dimension vector, $D$ the dimension of the parametric model and $K$ the number of models; (ii) computation of the covariance matrices (assumed with a diagonal structure) with complexity $O_\sigma(n^2dK)$ and (iii) the computation the stochastic transition matrix $B$ requiring a complexity of $O_B(n^2K \log K)$. From the above, the overall complexity is $O_\theta(n^2K(1 + \log K))$.

4) **Concluding Remarks**: More than trying to figure out what is the best parametric model, the most important conclusion is that the class of the proposed parametric models are capable to provide quite acceptable results at classifying pedestrian’s trajectories using simple motion models which depend on a small number of parameters. Furthermore, we avoid the need of computing the vector fields in every grid nodes, by assuming a parametric based motion. Such assumption does not affect the performance accuracy on pedestrian’s trajectory classification, and most of all, this strategy alleviates the computational effort, since the complexity is reduced. From the Table VII the Euclidean and affine models are able to produce quite similar performance with less number of models $K$, regarding the translation and rigid body models. This happens, since the former models allow for a more compactness description. From the comparison provided with the SD-HMM, we conclude that the family of parametric models are competitive in terms of performance accuracy, number of motion models and running time complexity.

### VIII. Conclusion

In this paper, we have presented an approach to modeling trajectories, using parametric motion models based on multiple vector fields, and applied it in a surveillance task. In the proposed model, a trajectory is driven, at each instant of time, by one of a set of vector fields. Switching between fields is controlled by a space-dependent probabilistic mechanism (modeled as a field of stochastic matrices). This approach allows representing a wide variety of trajectories exhibiting space dependent behaviors, without resorting to non-linear dynamical models (which are very hard to estimate). We have presented an EM algorithm to estimate the underlying motion fields along with the space-dependent switching probabilistic model. The estimates are based on finite-dimensional parameterizations of all the fields. Almost all the update equations of the EM algorithm have simple closed form expressions. Experiments have testified for the the ability of the proposed EM algorithm to estimate the motion and switching fields from observed trajectories. Experiments with synthetic and real data have shown that complex trajectories can be generated using
a set of simple parametric motion fields depending on a small number of parameters.

APPENDIX

PARAMETRIC MOTION FIELD ESTIMATION

The motion fields are estimated maximizing the following function (introduced in (11), middle row)

$$\mathcal{F} = C + \sum_{j=1}^{K} \sum_{l=1}^{L} y_{l,j} \log N(x_{l,j}, x_{l-1,j}^{(i)} + T_{i}(x_{l-1,j}^{(i)}), \sigma_{l}^2 I).$$

For the sake of simplicity, we just consider a single trajectory and a single field without loss of generality. Replacing the expressions of the parametric motion field and the normal density function in (19) we obtain

$$\mathcal{F}(A, t) = \sum_{t=2}^{L} \omega_{t} \| x_{t} - Ax_{t-1} - t \|^2$$

where $A$ is a $2 \times 2$ matrix and $t$ is a translation vector.

A. Estimation of $t$

Computing the derivative of $\mathcal{F}$ with respect to $t$ and equating to zero, we obtain

$$\frac{\partial \mathcal{F}(A, \hat{t})}{\partial t} = 0 \rightarrow \sum_{t=2}^{L} \omega_{t} (x_{t} - Ax_{t-1} - \hat{t}) = 0$$

$$\times \sum_{t=2}^{L} \omega_{t} (x_{t} - Ax_{t-1}) = \sum_{t=2}^{L} \omega_{t} x_{t}.$$ (21)

Therefore,

$$\hat{t} = \bar{x}_{t} - A\bar{x}_{t-1}.$$ (22)

where $\bar{x}_{t}$ and $\bar{x}_{t-1}$ are the weighted average of the $x_{t}$ and $x_{t-1}$, respectively. The weighted average (of a variable $u$) is defined as $\bar{u} = (\sum_{t=2}^{L} \omega_{t})^{-1} \sum_{t=2}^{L} \omega_{t} u$.

B. Estimation of $A$

The estimation of the matrix $A$, depends on the parametric model being used, as shown in Table I. Matrix $A$ is either an (i) identity matrix, (ii) rotation matrix, (iii) Euclidean similarity or (iv) arbitrary matrix.

We first replace in $\mathcal{F}$ the estimated value of the translation $\hat{t}$ obtained in the previous section

$$\mathcal{F}(A, \hat{t}) = \sum_{t=2}^{L} \omega_{t} \| x_{t} - \hat{x}_{t} - A(x_{t-1} - \hat{x}_{t-1}) \|^2$$

$$= \sum_{t=2}^{L} \omega_{t} \| \bar{x}_{t} - A\bar{x}_{t-1} \|^2.$$ (23)

where we define for a given variable $u_{t}$, $\bar{u}_{t} = u_{t} - \bar{u}$.

1) Translation Model: In this model, the matrix $A$ is the identity matrix, no estimation is required.

2) Affine Model: In this case, there is no constraints on $A$, thus

$$\frac{\partial \mathcal{F}(A, \hat{t})}{\partial A} = 0$$

$$\sum_{t=2}^{L} \omega_{t} (x_{t} - 2x_{t}^{T} Ax_{t-1} + x_{t-1}^{T} A^{T} Ax_{t-1}) = 0.$$ (25)

After straightforward calculation\(^5\) we obtain,

$$A \sum_{t=2}^{L} \omega_{t} \bar{x}_{t-1} x_{t-1}^{T} = \sum_{t=2}^{L} \omega_{t} \bar{x}_{t-1} x_{t-1}^{T}.$$ (26)

Defining

$$R_{ij} = \frac{1}{L-1} \sum_{t=2}^{L} \omega_{t} \bar{x}_{t-1} x_{t-1}^{T}.$$ (27)

we rewrite (26) as

$$A = R_{01}(R_{11})^{-1}.$$ (28)

3) Euclidean Similarity: In this case, we have the following restriction

$$A = \begin{bmatrix} a & b \\ -b & a \end{bmatrix}.$$ (29)

the cost function can be written as follows

$$\mathcal{F}(A, \hat{t}) = \sum_{t=2}^{L} \omega_{t} \| x_{t} - Ax_{t-1} \|^2 = \sum_{t=2}^{L} \omega_{t} \| \bar{x}_{t} - M(\bar{x}_{t-1}) \theta \|^2$$

where $\theta = [a, b]^{T}$ and

$$M(u) = \begin{bmatrix} u_{x} & u_{y} \\ u_{y} & -u_{x} \end{bmatrix}.$$ (31)

where the subscripts $x, y$ are the components of the vector $u$.

A necessary condition for optimality is

$$\frac{\partial \mathcal{F}(A, \hat{t})}{\partial \theta} = 0 \rightarrow \sum_{t=2}^{L} \omega_{t} M(\bar{x}_{t-1})^{T} (\bar{x}_{t} - M(\bar{x}_{t-1}) \theta) = 0.$$ (32)

Since $M^{T}(u)M(u) = \| u \|^{2} I$, with $I$ the identity matrix, we obtain

$$\left( \sum_{t=2}^{L} \omega_{t} \| \bar{x}_{t-1} \|^2 \right) \theta = \sum_{t=2}^{L} \omega_{t} M(\bar{x}_{t-1})^{T} \bar{x}_{t}.$$ (33)

Using the definition in (27) we can write

$$\text{tr}(R_{11}) \theta = \text{tr}(R_{01}^{T} 1)$$ (34)

where $1 = [1]^{T}$. Finally, we arrive at

$$\theta = \left( \text{tr}(R_{11}) \right)^{-1} R_{01}^{T} 1.$$ (35)

4) Rigid Body: In this model, the matrix $A$ is a rotation matrix, i.e. a unitary matrix where $A^{T} = A^{-1}$ with the additional constraint, det($A$) = 1, which preserves the axes orientation. The estimation of $A$ with the above constraints is obtained using singular value decomposition (SVD), $R_{01} = U D V^{T}$. The best estimate of the matrix rotation is simply

$$R = U V^{T}.$$ (36)

\(^5\)Recall that $\frac{\partial}{\partial A} \text{tr}(AXB) = A^{T} B^{T}$ and $\frac{\partial}{\partial A} \text{tr}(B^{T} X A X B) = 2XBB^{T}$. 
Jacinto C. Nascimento (S’00–M’06) received the E.E. degree from the Instituto Superior de Engenharia de Lisboa, Lisbon, Portugal, in 1995, and the M.Sc. and Ph.D. degrees from the Instituto Superior Técnico (IST), Technical University of Lisbon, Lisbon, in 1998 and 2003, respectively.

He is currently a Principal Researcher of an FCT project with the Institute for Systems and Robotics, IST. He has authored or co-authored over 90 papers in international journals and conference proceedings (many of which in the IEEE). His current research interests include statistical image processing, pattern recognition, machine learning, medical imaging analysis, video surveillance, general visual object classification.

Dr. Nascimento was on the program committees of many international conferences and is a reviewer of several international journals.

Jorge S. Marques received the E.E. and Ph.D. degrees and the Aggregation Title from the Technical University of Lisbon, Portugal, in 1981, 1990, and 2002, respectively.

He is currently an Associate Professor with the Electrical and Computer Engineering Department, Instituto Superior Técnico, Lisbon, Lisbon, where he is a Researcher with the Institute for Systems and Robotics. He has authored or co-authored over 150 papers in international journals and conferences, and has authored a book entitled *Pattern Recognition: Statistical and Neural Methods* (IST Press, 2005, 2nd ed., in Portuguese). His current research interests include statistical image processing, shape analysis, and pattern recognition.

Dr. Marques was the Co-Chairman of the IAPR Conference IbPRIA 2005, the President of the Portuguese Association for Pattern Recognition from 2001 to 2003, and an Associate Editor of the *Statistics and Computing Journal* (Springer).

João M. Lemos received the Ph.D. degree from Instituto Superior Técnico (IST), Technical University of Lisbon, Lisbon, Portugal, in 1989, where he was involved in research with the University of Florence, Florence, Italy, and the Department of Chemical Engineering, Imperial College of Science, Technology and Medicine, London, U.K.

He is currently a Full Professor (Professor Catedrático) of systems decision and control with IST and a Researcher with INESC-ID, where he has been leading the Control of Dynamic Systems Group since its creation in 1990. He is the Chairman of the Department of Electrical and Computer Engineering, IST, the Coordinator of the post-graduation program in electrical and computer engineering, and has been the Chairman of the Scientific Board of INESC-ID since 2007. He was an Associate Professor in 1997. He has co-authored 40 papers in journals (ISI referenced), 10 book chapters, and 180 conference papers (peer reviewed). He has supervised 13 Ph.D. thesis (completed) and 27 M.Sc. dissertations. His current research interests include computer control including adaptive and predictive control, control and estimation with multiple models, distributed control, modeling and identification, and process monitoring.

Dr. Lemos has either led or been a participant in projects on control applications to several types of industrial processes, large-scale boilers in thermoelectric plants, solar collector fields and furnaces, biomedical systems (general anesthesia, HIV 1 therapy, biochip temperature control), automotive systems, and water delivery canals.