

Clustering of Gaussian Random Vector Fields in Multiple Trajectory Modelling

Miguel Barão
CIMA, University of Évora
INESC-ID Lisboa
Portugal
Email: mjsb@uevora.pt

Jorge S. Marques
Institute for Systems and Robotics (ISR/IST), LARSyS
Instituto Superior Técnico, University of Lisboa
Portugal
Email: jsm@isr.ist.utl.pt

Abstract—This paper concerns the estimation of multiple dynamical models from a set of observed trajectories. It proposes vector valued gaussian random fields, representing dynamical models and their vector fields, combined with a modified k-means clustering algorithm to assign observed trajectories to models. The assignment is done according to a likelihood function obtained from applying the random field associated to a cluster, to the data. The algorithm is shown to have several advantages when compared with others: 1) it does not depend on a grid, region of interest, grid resolution or interpolation method; 2) the estimated vector fields has an associated uncertainty which is given by the algorithm and taken into account. The paper presents results obtained on synthetic trajectories that illustrate the performance of the proposed algorithm.

I. INTRODUCTION

In many practical problems one is interested in finding models describing some observed data that enable some form of generalization and predictions to be done. One such example is in traffic surveillance where cars move in particular patterns determined by road geometry, traffic rules and driving behavior. In these kind of problems, one objective is to check if something abnormal is happening. This can be achieved by having models describing normal behavior, and then check observations against these models to measure how predictable or unpredictable the behavior is. Models can be parameterized by hand, but that involves a high labour cost and any change in the road or in traffic rules would require human intervention to re-parameterize the models. A better alternative is use machine learning to automatically discover the models from the data with minimum human intervention. The latter approach is pursued here.

As input data, it is assumed that a set of trajectories are readily available, one for each vehicle, represented as sequences of points in a 2-dimensional space. The generation of these trajectories is not considered here. It could, *e.g.*, be a pre-existing video processing system converting image sequences to trajectories or have GPS equipped vehicles recording their locations at regular time intervals.

The main objective is to learn a set of dynamical models, in the form of vector fields, that best describe the observed trajectories. Several works in the literature have considered this or similar problems and have proposed different kinds of solutions.

One approach is to learn a small set of vector fields and allow switching among them so that several kinds of trajectories can be described by fewer models with switching in appropriate places. This technique has been used in [Nascimento et al., 2013], [Nascimento et al., 2014], [Nascimento et al., 2015]. In the aforementioned works, the switching mechanism is space dependent and described by a field of Markov matrices, where each matrix provides the transition probabilities between the models at a particular point in space. Since the active models are not observed (only the trajectory points), these are considered hidden states. The problem is then formulated as an Hidden Markov Model (HMM) where the switching probabilities are estimated using the Expectation-Maximization (EM) algorithm. The algorithms are further complicated since in each step of the EM, the expectation step requires a forward-backward algorithm to assign trajectory points to models and the maximization step requires a full iterative optimization problem on a probability space to be solved.

A different approach, which aims to lower computational complexity, is to consider a set of dynamical models where each trajectory is to be described by exactly one model, as in [Ferreira et al., 2013]. In that work, a new modified k-means clustering algorithm is proposed so that each cluster is represented by the trajectories that belong to the cluster, instead of using a “centroid trajectory”. The usual euclidean distance between a data point and the centroid of a cluster is replaced by a similar measure between the trajectory velocities and the velocities of the vector field estimated by the cluster. The algorithms obtained are purely deterministic and solutions are found by iteratively assigning trajectories and solving least-squares problems.

In this paper, a new solution based on random vector fields is proposed. Other works have also used random fields with success for the purpose of obtaining vector fields from data, *e.g.*, [Kim et al., 2011] and [Barão and Marques, 2017]. The idea here is to estimate multiple vector fields that best describe the data, without switching between fields, as done in [Ferreira et al., 2013], but taking advantage of the stochastic nature of the random field methodology. As a result of this approach, the solution yields several advantages over the other methods: 1) the random field approach can be seen as a bayesian approach

where a prior random field is conditioned on the data to provide a resulting field; 2) the predicted velocity vectors are multivariable gaussians where the mean vectors provide the expected velocity field and covariance matrices provide the associated uncertainty; 3) it does not require a grid, so that there are no grid resolution issues or artifacts due to the interpolation method used. The method is illustrated in synthetic trajectories similar to ones used for benchmarking in other works.

The paper is divided in four main sections. Section II provides background so that the paper is more self-contained: basic random field theory is provided in section II-A and the k-means algorithm is explained in section II-B. Section III formulates the main problem and proposes the algorithm based on mixing the k-means and the random field. Section III-A shows how random fields are used to predict velocity fields and section III-B shows how the clustering algorithm works. Finally, section IV provides simulation examples using synthetic data and section V presents concluding remarks.

II. BACKGROUND

A. Gaussian random fields

Suppose that a regular grid of $N \times N$ points is available and that a random variable X_{ij} is placed at each node. In general, the set of random variables is described by their joint probability distribution $p(X_{11}, X_{12}, \dots, X_{NN})$. Now, assuming that two variables at positions (n, m) and (r, s) are observed to be x_{nm} and x_{rs} , then the probability distribution for the remaining ones can updated to $p(X_{11}, \dots, X_{NN} \mid X_{nm} = x_{nm}, X_{rs} = x_{rs})$, where the observed variables are removed from the variables on the left. This very basic fact is at the core of many learning algorithms. In particular, if the variables are jointly gaussian, conditioning on observed variables yields a closed form solution for the mean and covariance of the remaining ones.

Now suppose that instead of a grid of points, a region with uncountable many points is considered. In this case, there are uncountable many random variables, one at each point of the 2-dimensional space. It is no longer possible to write a probability distribution, as done above for the grid, and a new construction is required. This is where random fields enter the scene. A random field is a generalization that allows uncountable many points in some manifold \mathcal{M} instead of a finite number of points. A particular case of a random field is a stochastic process, where the set of points (manifold) is usually the real line, used for the time parameter. See [Rasmussen and Williams, 2006] for an introduction to random fields.

This work will be dealing with 2-dimensional image spaces where, at each point of the space, a random velocity vector is assigned. Furthermore, it shall be assumed that for any set of points in the image, the corresponding random velocity vectors are jointly gaussian and satisfy the conditions of the Kolmogorov extension theorem [Billingsley, 1995]. This kind of formulation is usually called a *vector valued gaussian random field*.

The gaussian random field is defined by a mean function $m(\cdot)$ and a cross-covariance function $K(\cdot, \cdot)$ also known as kernel. The mean function assigns a velocity vector $m(x)$ to each point x in the image, while the kernel $K(x, x')$ provides the cross-covariance matrix between the velocity vectors at points x and x' . In what follows, $T(x)$ denotes the random velocity vector at the point x .

Considering a selection of two distinct points x_1 and x_2 , their respective random vectors $T_1 \triangleq T(x_1)$ and $T_2 \triangleq T(x_2)$ are jointly characterized by the multivariable gaussian distribution

$$\begin{bmatrix} T_1 \\ T_2 \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} m_1 \\ m_2 \end{bmatrix}, \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \right), \quad (1)$$

where the more compact notation $m_i \triangleq m(x_i)$ and $K_{ij} \triangleq K(x_i, x_j)$ was used.

When an observation $T_1 = V$ is done, where V is a concrete velocity, the random vector T_2 at x_2 is updated to a new gaussian distribution

$$p(T_2 \mid T_1 = V) = \mathcal{N}(m_2^*, K_{22}^*), \quad (2)$$

with a new mean m_2^* and covariance matrix K_{22}^* given by

$$m_2^* = m_2 + K_{21}K_{11}^{-1}(V - m_1), \quad (3)$$

$$K_{22}^* = K_{22} - K_{21}K_{11}^{-1}K_{12}. \quad (4)$$

The application of this result in a more general setting is straightforward. Replace $T_1 = V$ by the training set encompassing all the observed data, and x_2 by the set of points of interest for prediction. The resulting formula has the exact same structure as (3)-(4), the only difference being the higher dimensionality of the matrices and vectors involved.

B. k-means algorithm

The k-means algorithm aims to partition a dataset into k non-overlapping clusters so that similar data is assigned to the same cluster.

The algorithm starts by an initial assignment of data points to clusters, e.g., random assignment, and then proceeds by alternating the following two steps:

- 1) Compute the ‘‘centroids’’ of the clusters;
- 2) Assign data points to the clusters.

The centroid of a cluster represents the cluster itself and is usually computed as the center of mass of the data points assigned to that cluster. The assignment of data points to clusters is done so that the distance between the point and the chosen centroid is minimum among all centroids.

While many variants of the algorithm exist, the basic algorithm described above works reasonably well in practice. It converges to a local minimum when the Euclidean distance is used, but can fail to converge when other distance functions are used. Since it does not guarantee a global optimum, many implementations resort to running it multiple times from different initial cluster configurations and then select the best one.

The version implemented in this paper differs in several respects from the basic version above and is described next.

III. PROBLEM FORMULATION AND SOLUTION

In this paper, it is assumed that a set of trajectories $x(t)$ were generated by a set of N unknown velocity fields $T_n(x)$, $n = 1, \dots, N$. Our aim is to learn each of these fields from the observed trajectories. Since it is unknown the field to which a trajectory belongs, a clustering process is simultaneously employed that makes a hard assignment of trajectories to clusters based on a likelihood function. Then, a random field corresponding to each cluster is re-estimated whenever the trajectories assigned to a cluster change.

It is assumed that S continuous trajectories are sampled to yield multiple sequences $(x_0, x_1, \dots, x_{L_s})$ of $L_s + 1$ points, $s = 1, \dots, S$. The time index is normalized here to whole numbers. The normalized velocities can then be computed by the differences $v_t = x_{t+1} - x_t$, so that there are L_s velocities (v_0, \dots, v_{L_s-1}) based at the points (x_0, \dots, x_{L_s-1}) .

A. Gaussian random field prediction

Let \mathbf{X} and \mathbf{V} be two $L \times 2$ matrices containing the observed positions and corresponding velocities, organized so that each row corresponds to a point of the trajectory. Then, it is possible to predict the velocities at arbitrary points using equations (3) and (4). The following assumptions are made to the gaussian random field:

- 1) The random vector field has zero mean everywhere, $m(x) = [0 \ 0]$, so that the mean at a number of selected points yields a matrix

$$\mathbf{m} = \begin{bmatrix} 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}. \quad (5)$$

- 2) Given any two points x_i and x_j , the covariance matrix between their respective random velocity vectors T_i and T_j is isotropic and therefore the covariance matrix can be written as $\mathbf{K}_{ij} = k_{ij}\mathbf{I}_{2 \times 2}$, where $k_{ij} \triangleq k(x_i, x_j)$ is a scalar function that depends only on the distance between the two chosen points x_i and x_j , and $\mathbf{I}_{2 \times 2}$ is the identity matrix.

The isotropic assumption allows the representation of cross-covariances to be greatly simplified by using the reduced covariance matrix

$$\mathbf{K} = \begin{bmatrix} k_{11} & \cdots & k_{1r} \\ \vdots & & \vdots \\ k_{r1} & \cdots & k_{rr} \end{bmatrix}. \quad (6)$$

The full matrix would then given by the Kronecker product

$$\mathbf{K} \otimes \mathbf{I}_{2 \times 2} = \begin{bmatrix} \mathbf{K}_{11} & \cdots & \mathbf{K}_{1r} \\ \vdots & & \vdots \\ \mathbf{K}_{r1} & \cdots & \mathbf{K}_{rr} \end{bmatrix}. \quad (7)$$

- 3) The kernel function $k(\cdot, \cdot)$ is a positive decreasing function depending only on the euclidean distance $\|x_1 - x_2\|$

between the two points. Common examples of kernel functions are of the form

$$k(x_1, x_2) \triangleq \exp(-\alpha\|x_1 - x_2\|^\beta), \quad (8)$$

where the parameter α adjusts the decay rate. When the exponent $\beta = 1$, the kernel is known as the Ornstein-Uhlenbeck kernel function, and when $\beta = 2$ it is known as the squared exponential function and produces more smooth predictions.

These three assumptions imply a condition of strong stationarity in space, which makes practical sense since there are no particular regions of interest.

Given the observed trajectory points, collected in a matrix \mathbf{X} , and the coordinates of interest for prediction \mathbf{X}^* , the covariance matrix can be computed using (8) which, when written in partitioned form, is given by

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{xx} & \mathbf{K}_{x^*} \\ \mathbf{K}_{x^*} & \mathbf{K}_{**} \end{bmatrix}, \quad (9)$$

where the subscripts x and $*$ denote respectively the part of the observed data and the points where prediction is to take place.

The velocity vectors can now be predicted using (3)-(4):

$$\mathbf{m}^* = \mathbf{K}_{*x}\mathbf{K}_{xx}^{-1}\mathbf{V} \quad (10)$$

$$\mathbf{K}_{**}^* = \mathbf{K}_{**} - \mathbf{K}_{*x}\mathbf{K}_{xx}^{-1}\mathbf{K}_{x^*} \quad (11)$$

where the zero matrix \mathbf{m} was dropped from equation (10).

B. Modified k-means clustering

While the standard k-means algorithm has an explicit representation of clusters by a parameter vector — its centroid — in this work each cluster is implicitly represented by the set of trajectories assigned to it. This idea, also explored in [Ferreira et al., 2013] in a different setting, is that the set of trajectories in a cluster defines a corresponding gaussian random vector field. Then, an arbitrary trajectory can be compared against this random field to measure its “cost”.

Let \mathbf{X}_n and \mathbf{V}_n denote the points and velocities of all trajectories assigned to cluster n , appropriately concatenated into single matrices, and let $(\mathbf{X}_s, \mathbf{V}_s)$ denote an arbitrary trajectory. Then, using the random field generated by the cluster, a prediction can be performed at the points \mathbf{X}_s of the trajectory using equations (10) and (11) to yield predicted velocities \mathbf{T}_s^* as random vectors. The predicted velocities are Gaussian with mean \mathbf{m}_s^* and covariance matrix \mathbf{K}_{ss}^* .

The cost function basically measures how well the trajectory $(\mathbf{X}_s, \mathbf{V}_s)$ is predicted by the field and is defined by the log-likelihood

$$L_n(\mathbf{X}_s, \mathbf{V}_s) = \text{Tr}((\mathbf{V}_s - \mathbf{m}_s^*)(\mathbf{K}_{ss}^*)^{-1}(\mathbf{V}_s - \mathbf{m}_s^*)^T) \quad (12)$$

of the cluster given the trajectory. This function effectively replaces the usual euclidean distance from a data point to the centroid of a cluster. The clustering algorithm then works by iteratively assigning trajectories to the clusters having lower cost.

Let $\tau : \{1, \dots, S\} \rightarrow \{1, \dots, N\}$ denote a cluster assignment function, which assigns trajectories to clusters, and its inverse $\tau^{-1}(n)$ gives the set of trajectories that belong to a particular cluster n . The algorithm then iterates the following two steps:

- 1) Compute a $N \times S$ log-likelihood matrix \mathbf{L} . This process uses the current assignment function τ so that, for each trajectory s and cluster n , the log-likelihood matrix components $\mathbf{L}_{ns} \triangleq L_n(\mathbf{X}_s, \mathbf{V}_s)$ are given by:

$$\mathbf{m}_s^* = \mathbf{K}_{sn} \mathbf{K}_{nn}^{-1} \mathbf{V}_s, \quad (13)$$

$$\mathbf{K}_{ss}^* = \mathbf{K}_{ss} - \mathbf{K}_{sn} \mathbf{K}_{nn}^{-1} \mathbf{K}_{sn}^T, \quad (14)$$

$$\mathbf{L}_{ns} = \text{Tr}((\mathbf{V}_s - \mathbf{m}_s^*)(\mathbf{K}_{ss}^*)^{-1}(\mathbf{V}_s - \mathbf{m}_s^*)^T). \quad (15)$$

The matrix \mathbf{K}_{nn} is the cross-covariance matrix between trajectories in $\tau^{-1}(n)$ and \mathbf{K}_{sn} is the cross-covariance matrix between trajectory s and the ones in $\tau^{-1}(n)$.

- 2) Update clusters. The assignment function τ is updated according to

$$\tau(s) \triangleq \arg \min_n L_n(\mathbf{X}_s, \mathbf{V}_s), \quad (16)$$

where the log-likelihood function L_n uses the assignment function τ from the previous iteration.

The two steps are repeated until the cluster assignment function τ converges. Since the assignment function τ only has a finite number of possibilities, either the algorithm converges or enters into a cycle. In either case it is interrupted.

IV. EXAMPLES

To illustrate the algorithm several sets of trajectories were generated, moving horizontally/vertically, and moving circularly in clockwise and counter-clockwise directions. The algorithm was applied to this set of trajectories with a squared-exponential kernel with parameter $\alpha = 3$.

Although not required by the algorithm, a grid of 11×11 nodes was generated and prediction performed at those points for illustrative purposes only. The images show, at each node, the mean vector as a yellow arrow and the uncertainty, measured by the log-trace of the covariance matrix, as gray scale in background. Whiter areas mean lower uncertainty and are associated with larger amounts of data in those regions.

Figure 1 illustrates the application of the algorithm when the number of clusters corresponds exactly to the number of models used to generate the data. In this case, a global optimal solution was found with all the trajectories correctly assigned to clusters, and the fields well estimated. This is not always the case, as the algorithm can become stuck in local minima.

Figure 2 illustrates a different situation when, for the same data, the number of clusters allowed is inferior to the number of models generating the data. In this case, the algorithm assigned trajectories to their closest match, so that linear trajectories are aligned with the rotational direction of the circular trajectories in their neighborhood.

V. CONCLUSIONS

In this paper, a new combination of gaussian random fields and a k-means clustering algorithm was proposed. The aim was to build dynamical models describing a set of trajectories observed in a 2-dimensional space. The observed trajectories can be thought as describing the movement of vehicles, but the same principle applies to moving people, animals or any other object. The output of the algorithm is a set of models obtained by first clustering the trajectories and then model each cluster by a vector valued gaussian random field that represents its own velocity field.

Using random fields shows some advantages over other methods. It does not depend on an ad-hoc selection of grid and interpolation method for the generation of a vector field, as done in other works, and is sufficiently flexible to represent many nonlinear systems. It also includes in a natural way the description of all the uncertainties involved.

A modified k-means algorithm was also proposed to take care of the assignment of trajectories to fields. While the original algorithm uses a distance function and the notion of centroid to represent a cluster, here a modification of the algorithm was used where no centroid was explicitly used and the distance was replaced by a cost function related to the likelihood of the assignments. This way, clusters were defined implicitly by the trajectories that make part of the cluster itself.

The combined use of gaussian processes and the k-means algorithm raised some difficulties related to the high flexibility with which the gaussian random field can explain data and the fact that the k-means may not converge to a global optimum. This drawback was somewhat mitigated by selecting a high correlation function (kernel) even for distant points in space which proved to be enough for the algorithm to work on most situations.

The main drawback of the proposed algorithm is that the random fields depend on all the observed data collected, as opposed to just grid nodes as in other methods. This limits the possibility of the algorithm to be used online without implementing some form of forgetting in order to keep the complexity constant.

ACKNOWLEDGMENTS

This work was partially supported by FCT (Fundação para a Ciência e Tecnologia), grants PTDC/EEIPRO/0426/2014, UID/CEC/50021/2013 and UID/EEA/50009/2013.

REFERENCES

- [Barão and Marques, 2017] Barão, M. and Marques, J. S. (2017). Gaussian random vector fields in trajectory modelling. In *Irish Machine Vision and Image Processing Conference - IMVIP*.
- [Billingsley, 1995] Billingsley, P. (1995). *Probability and Measure*. John Wiley and Sons.
- [Ferreira et al., 2013] Ferreira, N., Klosowski, J. T., Scheidegger, C. E., and Silva, C. T. (2013). Vector fields k-means: Clustering trajectories by fitting multiple vector fields. In B. Preim, P. R. and Theisel, H., editors, *Eurographics Conference on Visualization*, volume 32.
- [Kim et al., 2011] Kim, K., Lee, D., and Essa, I. (2011). Gaussian process regression flow for analysis of motion trajectories. In *IEEE International Conference on Computer Vision*.

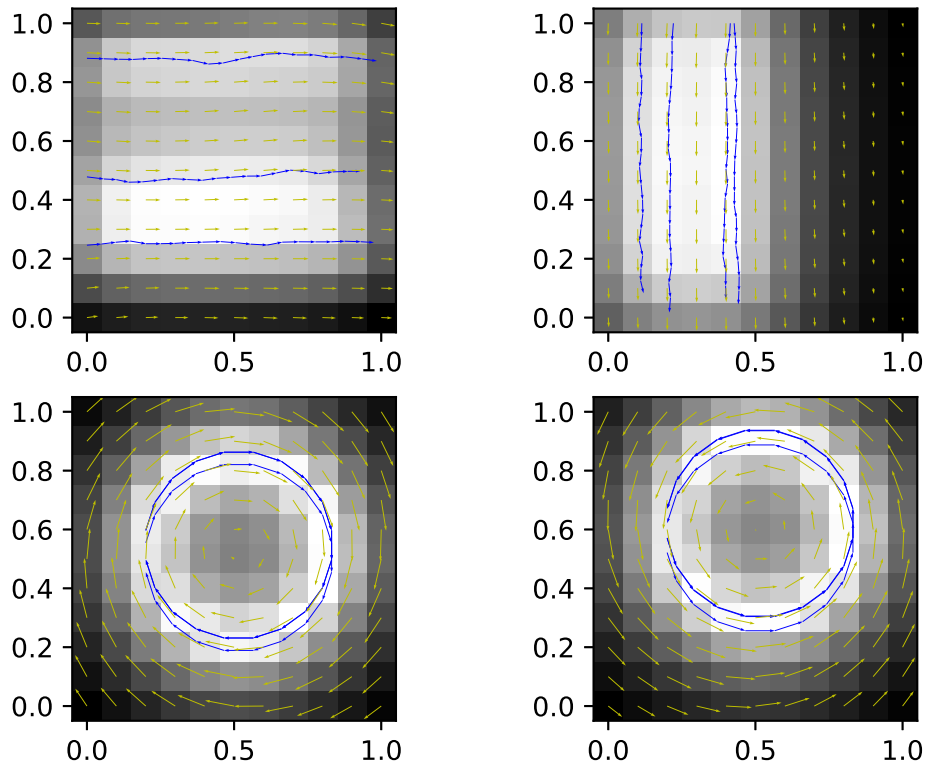


Fig. 1. Clustering of trajectories generated by 4 different models into 4 clusters.

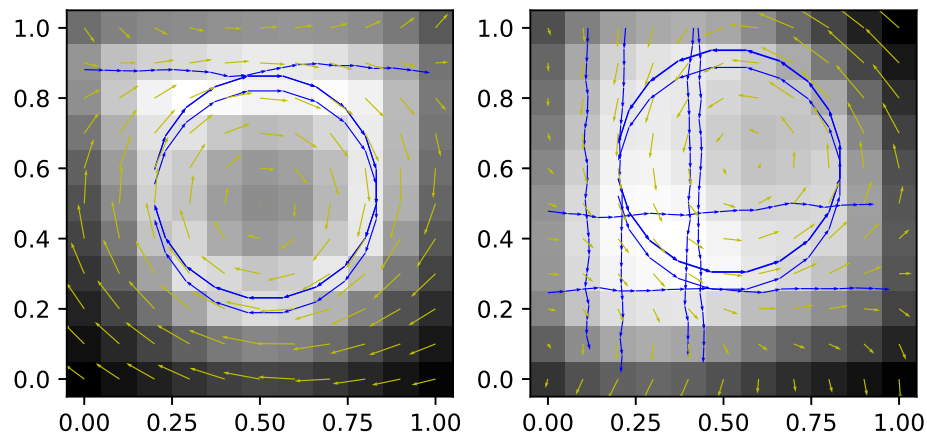


Fig. 2. Clustering of trajectories generated by 4 different models into just 2 clusters.

- [Nascimento et al., 2014] Nascimento, J. C., Barão, M., Marques, J. S., and Lemos, J. M. (2014). Information geometric algorithm for estimating switching probabilities in space-varying HMM. *IEEE Transactions on Image Processing*, 23(12):5263–5273.
- [Nascimento et al., 2015] Nascimento, J. C., Barão, M., Marques, J. S., and Lemos, J. M. (2015). An information geometric framework for the optimization on discrete probability spaces: Application to human trajectory classification. *Neurocomputing*, 150:155–162.
- [Nascimento et al., 2013] Nascimento, J. C., Figueiredo, M. A. T., and Marques, J. S. (2013). Activity recognition using mixture of vector fields. *IEEE Transactions on Image Processing*, 22(5):1712–1725.
- [Rasmussen and Williams, 2006] Rasmussen, C. E. and Williams, C. (2006). *Gaussian Processes for Machine Learning*. MIT Press.