

GROUPING MOTION TRAJECTORIES

Samuel Pachoud, Emilio Maggio, Andrea Cavallaro

School of Electronic Engineering and Computer Science - Queen Mary, University of London (UK)

ABSTRACT

We present a method to group trajectories of moving objects extracted from real-world surveillance videos. The trajectories are first mapped into a low dimensionality feature space generated through linear regression. Next the regression coefficients are clustered by a Gaussian Mixture Model initialized by K-means for improved efficiency. The model selection problem is solved with Bayesian Information Criterion that penalizes models with high complexity. We demonstrate the proposed approach on both synthetic and real-world scenes. Experimental results show that the proposed clustering method outperforms K-means and mixture of regression models, while also reducing the computational complexity compared to the latter.

Index Terms— Surveillance video, object trajectories, clustering

1. INTRODUCTION

The automated analysis of surveillance video and remote sensing data generates large amounts of trajectories that describe behaviors of moving objects. Accumulating and analyzing trajectory information over time offers valuable information on trends (dominant behaviors) and abnormal movements (outliers) that cannot be uncovered by a human operator alone. Fig. 1 shows an example of accumulated trajectory information that is clustered to identify groups of objects with similar behaviors. The process of characterizing object behaviors can be divided into two main steps, namely feature representation and feature clustering.

In this paper we present a trajectory analysis method that uses a trajectory *representation* based on linear regression coefficients. The subsequent analysis is performed by GMM *clustering* initialized by K-means for improved efficiency. GMM is applied directly on the regression coefficients. The automatic selection of the number of clusters is performed using the Bayesian Information Criterion (*BIC*).

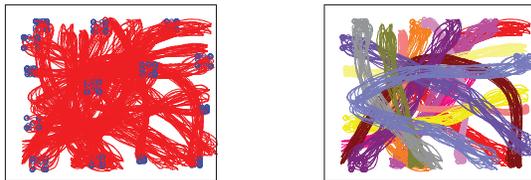


Fig. 1. Example of trajectory clustering. (Left) Accumulated object trajectory information over time. (right) Clusters representing the dominant object behaviors

The paper is organized as follows. Sec. 2 discusses the existing literature for trajectory analysis. Sec. 3 describes the trajectory representation and the clustering algorithm used in the proposed method. Experimental results and evaluation are presented in Sec. 4. Finally in Sec. 5 we draw the conclusions.

2. PREVIOUS WORK

The natural feature space where trajectories lie has a variable dimensionality as targets are observable for different time-spans and measurements are affected by background clutter and sensor noise. To address this problem, similarity measures that enable the comparison of vectors with different dimensionality can be used. The Hausdorff distance [1] and the Longest Common Sub-Sequence (LCSS) [2] are examples of distances used to compute trajectory similarity. However, these measures do not explicitly model noise, thus resulting in poor performances with noisy data [3]. Other methods exist that reduce the dimensionality of the data while filtering the noise and preserving important trajectory information. Dimensionality reduction can be obtained by resampling the trajectories via interpolation to a predefined number of data-points [4]. In such a way only positional information is analyzed, not accounting for higher order statistics. A more descriptive alternative is to represent the main features of a trajectory using Principal Component Analysis (PCA) [5]. Polynomial approximation has also proved to be effective in smoothing trajectories and in reducing the dimensionality. In particular, Chebyshev polynomials [6] and regression coefficients can be used [7]. Using Hidden Markov Models (HMM), each trajectory is projected into the feature space defined by the hidden parameters [8, 9]. Recently Trajectory Directional Histograms (TDH) have been proposed to represent the statistic directional distribution [4], and to complement the information from resampled trajectories. A performance comparison between different distances and representations applied to clustering showed a good performance of PCA in terms of computational efficiency, accuracy and robustness to noise [3]. Hausdorff and LCS instead performed poorly on the testing datasets.

Once trajectories are mapped onto an appropriate feature space, clustering is applied to organize the data into meaningful structures. Neural networks have been used to learn the main activity patterns in a scene [6, 10]. A Gaussian Mixture Model (GMM) can be used to model the intra-cluster trajectory variability [7]. Otherwise, trajectory similarity in the feature space can be used in hierarchical clustering [11]. Starting from single trajectory entities, larger clusters are formed through merging (bottom-up) [2] or the complete set is iteratively split into smaller clusters (top-down) using graph cuts [4]. Recently, to alleviate the constraints of standard partitional methods (e.g., gaussianity or linearity of the model), spectral methods have been applied to the analysis of trajectories [9, 12]. A summary of algorithms for trajectory analysis and their characteristics is presented in Tab. 1.

Table 1. Summary of trajectory analysis algorithms and their application (NS: number of scenarios; NT: number of trajectories used in the experiments)

Ref.	Representation	Distance	Algorithm	Scenarios	NS	NT
[1]	Native 2D time series	Hausdorff distance	Graph cuts	Outdoor surveillance	2	n.a.
[4]	Resampled trajectories and directional histograms	Euclidean and Bhattacharyya	Graph cuts (Dominant-set clustering)	Traffic monitoring	1	1200
[2]	Native 2D time series	LCSS	Hierarchical clustering	Outdoor surveillance	1	30
[5]	Trajectory segmentation+PCA	Euclidean	Similarity-based querying	Synthetic data	n.a.	n.a.
[3]	PCA	Spectral clustering	Euclidean	Outdoor surveillance + sign language	2	130
[6]	Chebyshev polynomials	Euclidean	Neural network (SOM)	Indoor dataset	1	220
[10]	Quantized flow vectors	Euclidean	Neural network (leaky neurons)	Outdoor surveillance	1	n.a.
[7]	Regression coefficients	Gaussian Mixture Models		Hand tracking data	1	20
[9]	Hidden Markov Models	Mutual fitness score	Spectral clustering	Highway + traffic monitoring	2	n.a.
[12]	Heuristics based pre-processing	Euclidean	Two-layer spectral clustering	Traffic monitoring	1	467

3. GROUPING TRAJECTORIES

3.1. Trajectory representation

The output of a tracking algorithm can be expressed as a 2-D time series $\{(u_i, v_i)\}_{i=1, \dots, t}$ that represents the displacement of an object in the image plane. We represent each dimension independently (1-D time series) with a set of coefficients from a linear regression. Let $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_j, \dots, \mathbf{y}_N\}$ be a set of N 1-D time series. Each time series j has length n_j with measurements observed at the points (or times) \mathbf{x}_j . The regression of \mathbf{y}_j on \mathbf{x}_j [7] is

$$\mathbf{y}_j = \mathbf{X}_j \boldsymbol{\beta}_j + \mathbf{e}_j, \quad (1)$$

where $\mathbf{e}_j \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$ models the error, and the $n_j \times (P + 1)$ regression matrix \mathbf{X}_j is a P -th order Vandermonde matrix [7] defined as

$$\mathbf{X}_j = \begin{bmatrix} 1 & x_{j1} & x_{j1}^2 & \cdots & x_{j1}^P \\ 1 & x_{j2} & x_{j2}^2 & \cdots & x_{j2}^P \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & x_{jn_j} & x_{jn_j}^2 & \cdots & x_{jn_j}^P \end{bmatrix}. \quad (2)$$

$\boldsymbol{\beta} = [\beta_0 \ \beta_1 \ \cdots \ \beta_P]^T$ is a vector composed of the unknown regression coefficients. In case of Gaussian error the optimal solution is obtained through least squares. The value of P gives the order of the regression model and defines the dimensionality of the feature space. The choice of P depends on the desired level of descriptiveness and it is application dependent. For our outdoor surveillance scenarios $P = 2$ (i.e., a second order polynomial) is appropriate. Intuitively this means that a trajectory is defined by position (β_0), speed (β_1), and acceleration (β_2). Therefore each trajectory is mapped into a 6-D parameter space, with 3 coefficients for each coordinate of the Cartesian space (Fig. 2).

3.2. Clustering

Previous approaches define the trajectories as individual time series generated from a finite mixture model consisting of the linear regression components (MRM) [7]. Each cluster k is described by a set of regression coefficients $\hat{\boldsymbol{\beta}}_k$ and by the error σ_k^2 . The resulting GMM model is

$$p(\mathbf{y}_j | \boldsymbol{\Theta}) = \sum_{k=1}^K \alpha_k \mathcal{N}(\mathbf{y}_j | \mathbf{X}_j \hat{\boldsymbol{\beta}}_k, \sigma_k^2 \mathbf{I}). \quad (3)$$

where $\boldsymbol{\Theta} = \{\alpha_1, \dots, \alpha_K, \hat{\boldsymbol{\beta}}_1, \dots, \hat{\boldsymbol{\beta}}_K, \sigma_1^2, \dots, \sigma_K^2\}$ are the parameters of the mixture. The intra-cluster variability is modeled by σ_k^2 , which represents the trajectory spread in the image plane. Note

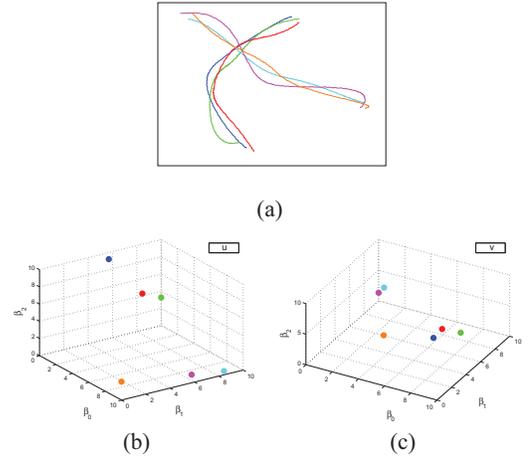


Fig. 2. Trajectory representation with second-order linear regression. (a) Input trajectories. (b - c) Regression space, visualization of the trajectory coefficients (color-coded) associated with the horizontal and vertical coordinates, respectively

that σ_k^2 does not provide information on which regression coefficients generates the discrepancy from the average trajectory. As we are interested in modeling the variations of the single regression parameters, we approximate each trajectory with a different regression coefficient β_j (see Eq. (1)), computed via least squares. Then we model the trajectory generation in the space of the regression coefficients as

$$p(\boldsymbol{\beta}_j | \boldsymbol{\Phi}) = \sum_{k=1}^K \alpha_k \mathcal{N}(\boldsymbol{\beta}_j | \hat{\boldsymbol{\beta}}_k, \Sigma_k). \quad (4)$$

Now $\boldsymbol{\Phi} = \{\alpha_1, \dots, \alpha_K, \hat{\boldsymbol{\beta}}_1, \dots, \hat{\boldsymbol{\beta}}_K, \Sigma_1, \dots, \Sigma_K\}$ defines the parameters of the mixture (as $\boldsymbol{\Theta}$ in Eq. (3)). Each cluster is defined by a centroid, $\hat{\boldsymbol{\beta}}_k$, and a $(P + 1) \times (P + 1)$ covariance matrix, Σ_k . Given the number of cluster, K , and the set of trajectories, Y , the set of parameters, $\boldsymbol{\Phi}$, maximizing the likelihood is selected via EM. In the following, we will refer to this solution as EM-GMM.

3.3. Initialization and model selection

The final result provided by EM-GMM depends on the *initialization* of the parameters. A common method for initialization is to draw the

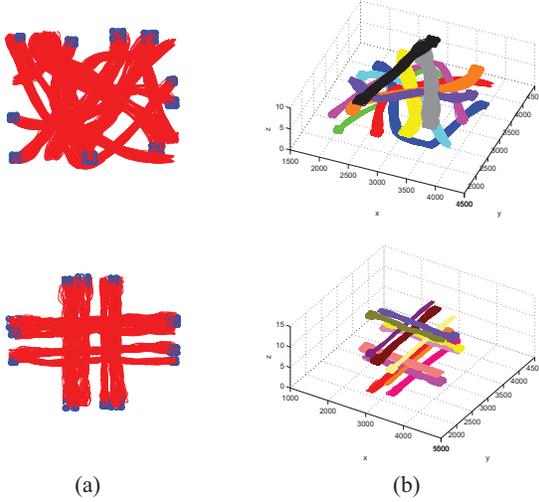


Fig. 3. Example of clustering results on synthetic data. (a) Input trajectories; (b) clustering results using EM-GMM with K-means initialization

starting membership values from an exponential distribution. However, in this case the EM procedure is likely to terminate in a local minimum of the likelihood. Multiple runs of EM-GMM with different initializations are necessary to better explore the parameter space. In order to reduce the complexity and the iterations of EM-GMM, we use a less complex algorithm to select more promising locations of a sub-space of Φ . First multiple runs of K-means are performed. Next the partition with the smallest residual is selected and the centroids are used as starting point for EM-GMM. This operation speeds up the convergence of the algorithm as multiple runs of K-means are less time-consuming than GMM runs.

The automatic selection of the number of clusters K (*model selection*) is performed using the Bayesian Information Criterion (*BIC*) [13]. *BIC* is a likelihood-based method that penalizes models with high complexity, depending on the number of parameters necessary to describe the model. Let $\mathcal{L}(Y, \Phi_K)$ be the likelihood, after convergence, associated to the cluster parameters Φ_K . *BIC* chooses the value K that maximizes

$$BIC(P) = \log \mathcal{L}(Y, \Phi_K) - \frac{M}{2} \log(N), \quad (5)$$

where $M = K - 1 + K(P + 1)(2 + P/2)$ is the number of independent parameters in Φ . The likelihood is ideally increasing with the model complexity; the second term of the r.h.s. of Eq. (5) introduces a penalty that increases with the complexity as well, thus favoring models with a lower number of parameters. Although we also tested V-fold cross validation [14], the increased computational complexity necessary to test multiple data partitions was not compensated by significant performance improvements.

Fig. 3 shows sample results of EM-GMM with K-means initialization. EM-GMM with *BIC* correctly estimates the number of clusters and their elements. In particular, the trajectory dataset in the second row is composed of 12 sets of trajectories: 8 sets representing slower objects (e.g., pedestrians) and 4 sets representing faster objects (e.g., vehicles). Although some groups of trajectories are almost completely overlapping, the proposed algorithm separates

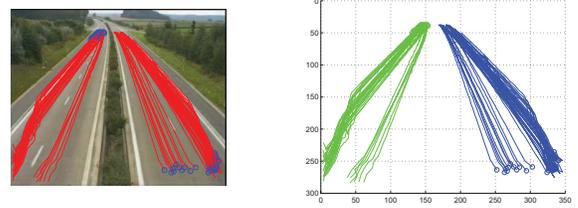


Fig. 4. Trajectory clustering results for S1. (Left) Trajectory dataset; (right) the two dominant object behaviors are recognized by EM-GMM

different object behaviors based on their speed (this information is encoded in the first order regression coefficients).

4. EXPERIMENTAL RESULTS

In this section we demonstrate the proposed method on real-world surveillance scenarios. Two standard datasets are used, namely a highway monitoring scenario (here referred to as S1) from the MPEG-7 dataset, and a urban traffic surveillance scenario (here referred to as S2) from the VACE dataset. In S1 (80 trajectories) the goal is to form two clusters representing the two directions of the highway. In S2 (158 trajectories) we want to split the data into six clusters, two clusters for the vehicles in the road and four clusters for the pedestrians on the sidewalks. Fig. 4 and Fig. 5 show the results obtained by EM-GMM.

We compare the proposed clustering method (EM-GMM) with two other clustering algorithms, namely K-means and MRM [7]. For a fair comparison, the model selection step is disabled in order to evaluate clustering only. The number of clusters K is given as input to the three methods. Given the output set of clusters O and the labeled ground-truth set G , the correctness percentage $C(O, G)$ over a set of N trajectories is defined as

$$C(O, G) = \frac{100}{N} \sum_{i=1}^K \max_{j=1 \dots K} (|o_i \cap g_j|), \quad (6)$$

where $o_i \in O$ is a cluster in the output set, and $g_j \in G$ is a cluster in the ground truth set. To achieve independence from the initialization, each algorithm is run 20 times and the average of $C(\cdot)$ over the runs is computed. Tab. 2 shows the performance evaluation comparison according to Eq. (6). To test the robustness of the algorithms the two datasets were corrupted with Gaussian and uniform random noise. The values of σ and the amplitude of the uniform noise showed in Tab. 2 are percentages of the horizontal coordinate range. Furthermore, to evaluate when trackers work at different frame-rates, new datasets with sub-sampled original data are used. The results on S1 show that the three algorithms achieve satisfactory performance on simple datasets (the two directions are correctly detected). However, EM-GMM clearly exhibits better performance than K-means and MRM in more complex scenarios (S2). It is also noticeable that the representation based on regression parameters is robust to data noise and sub-sampling. In both scenarios data corruption does not significantly reduce the clustering performance.

Regarding the computational complexity, EM-GMM takes 5.9s to perform 20 runs on S2 (Matlab implementation on a Pentium4 3.2GHz), while MRM and K-means complete the same task in 19.9s

Table 2. Performance comparison between clustering methods. To evaluate robustness we add noise to and we subsample the input data (Subsampling Rate (SSR)).

Methods			Gaussian noise					Uniform noise					SSR	
			1%	2%	3%	4%	5%	1%	2%	3%	4%	5%	2	3
EM-GMM	S1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	S2	99.4	99.4	99.4	94.9	93.0	82.9	97.5	98.7	98.1	98.1	98.7	97.5	99.4
K-means	S1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	S2	79.3	77.5	76.3	80.6	79.2	79.8	77.8	77.4	79.8	78.4	80.6	84.2	81.1
MRM [7]	S1	99.9	99.8	99.8	99.8	100.0	99.9	99.7	99.8	100.0	99.9	99.6	100.0	100.0
	S2	88.9	84.5	87.2	88.2	87.6	88.9	89.1	85.8	83.8	88.1	89.0	87.7	88.8

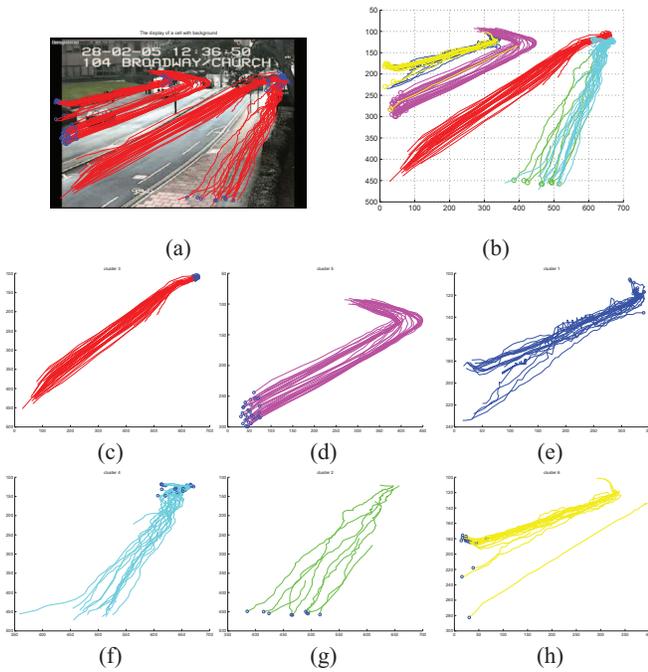


Fig. 5. Trajectory clustering results for S2. (a) Trajectory dataset; (b) trajectory clustering result using EM-GMM; (c)-(h) the main behaviors recognized in the scenario corresponding to four groups of pedestrians walking on the sidewalks (in two directions) and two groups of vehicles

and 2.0s, respectively. EM-GMM is more efficient than MRM because of the different computations needed to estimate the likelihood. MRM computes the log-likelihood in the Cartesian space summing up all the trajectories data points, whereas EM-GMM computes the likelihood on the low-dimensionality space of the regression coefficients.

5. CONCLUSIONS

We presented an algorithm for clustering object trajectories from surveillance video. Trajectories are represented by the coefficients of a linear regression and clustering is performed via a Gaussian Mixture Model initialized by K-means. The selection of the number of clusters is automated using the Bayesian Information Criterion.

The algorithm was validated on real-world surveillance scenarios and compared with alternative clustering approaches. Experimental results showed that the proposed method outperforms both K-means and a mixture of linear regression model [7]. Furthermore the computational complexity of the complete method is reduced by 70% compared with the classical mixture of regression model. Current work includes the use of homography transformation to account for scene perspective in order to cluster the trajectories on a top-view plane.

6. REFERENCES

- [1] I. N. Junejo, O. Javed, and M. Shah, “Multi feature path modeling for video surveillance,” in *ICPR*, 2004, pp. 716–719.
- [2] D. Buzan, S. Sclaroff, and G. Kollios, “Extraction and clustering of motion trajectories in video,” in *CVPR*, 2004, pp. 521–524.
- [3] Z. Zhang, K. Huang, and T. Tan, “Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes,” in *ICPR*, Sept. 2006, pp. 1135–1138.
- [4] X. Li, W. Hu, and W. Hu, “A coarse-to-fine strategy for vehicle motion trajectory clustering,” in *ICPR*, 2006, pp. 591–594.
- [5] F.I. Bashir, A.A. Khokhar, and D. Schonfeld, “Segmented trajectory based indexing and retrieval of video data,” in *ICIP*, 2003, pp. 623–626.
- [6] A. Naftel and S. Khalid, “Motion clustering using spatiotemporal approximations,” in *IMSA*, 2005, pp. 207–212.
- [7] S. Gaffney and P. Smyth, “Trajectory clustering with mixtures of regression models,” in *KDDM*, 1999, pp. 63–72.
- [8] F. Porikli, “Trajectory distance metric using Hidden Markov Model based representation,” in *ECCV*, 2004.
- [9] F. Porikli, “Trajectory pattern detection by HMM parameter space features and eigenvector clustering,” in *ECCV*, 2004.
- [10] N. Johnson and D. Hogg, “Learning the distribution of object trajectories for event recognition,” *IVC*, vol. 14, no. 8, pp. 609–615, 1996.
- [11] S. C. Johnson, “Hierarchical clustering schemes,” *Psychometrika*, vol. 2, pp. 241–254, 1967.
- [12] Z. Fu, W. Hu, and T. Tan, “Similarity based vehicle trajectory clustering and anomaly detection,” in *ICIP*, 2005, pp. 602–605.
- [13] G. Schwarz, “Estimating the dimension of a model,” *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.
- [14] M. Stone, “Cross-validation choice and assessment of statistical predictions,” *JRSS*, vol. B-36, pp. 111–147, 1974.