UNSUPERVISED FUZZY CLUSTERING FOR TRAJECTORY ANALYSIS

*Nadeem Anjum and Andrea Cavallaro**

Multimedia and Vision Group - Queen Mary, University of London
Mile End Road, E1 4NS London, United Kingdom
Email: {nadeem.anjum,andrea.cavallaro}@elec.qmul.ac.uk

ABSTRACT

We propose an unsupervised fuzzy approach for motion trajectory clustering. The proposed approach is divided into three main steps: first Mean-shift is used for local mode seeking by analyzing trajectory data over multiple feature spaces. This step generates a set of tentative clusters. Next, adjacent clusters are combined by analysing the cluster attributes across all feature spaces. Sparse clusters are finally considered as generated by outlier object behaviors and then removed. The performance of the proposed algorithm is evaluated on real outdoor video surveillance scenarios with standard data-sets and it is compared with state-of-the-art techniques.

Index Terms— Video surveillance, Mean-shift, clustering, object trajectories.

1. INTRODUCTION

Object trajectory analysis is an important step in applications like video surveillance, automotive systems, medical screening and autonomous robotic systems. Trajectory analysis, for example based on clustering, helps in defining events of interest and in identifying anomalies. Trajectory clustering classifies trajectory data-sets into *homogeneous* groups using an appropriate trajectory modeling. This is achieved by first transforming the trajectories into an appropriate *feature space* and then defining an efficient *distance measure* between trajectories in the feature space.

Porikli [1] uses a supervised *Hidden Markov Model* (HMM) to represent each trajectory into a feature space formed by hidden parameters. This framework can accurately measure coordinate, orientation, and speed similarity of trajectory pairs. Breitenbach *et al.* [2] present a semi-supervised technique based on a *consistency method* that learns intrinsic structures of a trajectory and then uses the information to detect anomalous events in a scene. Bashir *et al.* [3] use Principal Component Analysis (PCA) to represent trajectories by reducing their dimensionality for indexing and retrieval of video data. Antonini *et al.* [4] use Independent Component Analysis (ICA) to model trajectories for pedestrians counting in a video sequence. Li *et al.* [5] have recently proposed Trajectory Directional Histograms (TDH) to represent the statistical directional distribution, and to complement the information from resampled trajectories for vehicle motion trajectory clustering.

After transforming the trajectories into an appropriate feature space, the next step is to organize the data into clusters based on some homogeneity criteria (distance measure). The most common *homogeneity* criteria are conventional (normalized) distance measures [6, 7]. Mean, Maximum, Minimum, modified Hausdorff-type distances ([8]) and Longest Common Subsequences (LCSS) ([9]) are also popular similarity measures for trajectory clustering. Zhang *et al.* [10] compared different similarity measures and feature space representations and found that a combination of PCA with the Euclidean distance outperforms other techniques.

The trajectory clustering approaches presented so far are either *supervised* (e.g., they need training for the estimation of the model parameters) or *semi-supervised* (e.g., they require a priori knowledge of the number of clusters). In this paper, we proposed an *unsupervised fuzzy clustering* approach that uses Mean-shift over a multi-feature space representation. Moreover, as the trajectory clustering results obtained from the independent feature spaces have variable degree of belongingness, we analyze the consistency of behavior of the trajectories across all feature spaces to obtain the final clustering. A cluster merging and a cluster removal procedure are also included in the approach to refine the clustering results without requiring the prior knowledge of the number of clusters.

The paper is organized as follows. Sec. 2 presents the proposed clustering technique. The experimental results and a comparison with the state of the art are discussed in Sec. 3. Finally, in Sec. 4 we draw the conclusions.

2. FUZZY CLUSTERING USING MEAN-SHIFT (FMS)

Let a trajectory T_j be represented as $T_j = \{(x_j^i, y_j^i); i = 1, \ldots, N_j\}$, where (x_j^i, y_j^i) is the estimated position of the target in the image plane and N_i is the number of trajectory points. Each trajectory needs to be transformed in appropriate feature spaces before clustering. Let $F_m(.)$ be a transformation functions defined as $F_m(T) \rightarrow \Psi_m$, with $m = 1, ..., M$. The transformation $F_m(.)$ maps each trajectory to a *d-dimensional* feature space, Ψ_j, with $j = 1, ..., J$. We use Ψ_1, the space spanned by the first two components of the trajectory data obtained through *PCA*, and Ψ_2, the space spanned by the *average velocity vector* of each trajectory. After transforming the trajectories into the feature spaces, we analyze the trajectory data using Mean-shift in each space to seek the local modes and generate the clusters. Initially, the mode seeking process starts by fixing first trajectory as a seed point; once the Mean-shift process converges to the local mode all the points within the bandwidth of the kernel are assigned to that mode. The assigned points are not considered for future iterations. The next seed point is selected randomly from the unprocessed points and the process terminates when all points assigned to a corresponding local mode. The details of the Mean-shift procedure itself are given in next section.

2.1. Mean-shift clustering

Mean-shift is a clustering technique that climbs the gradient of a probability distribution to find the nearest dominant mode or peak

*The authors acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPSRC), under grant EP/D033772/1

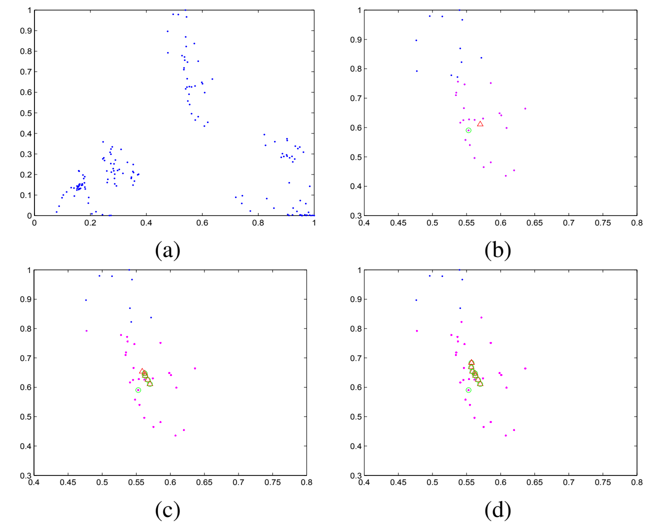

Fig. 1. Example of Mean-shift clustering. Points represent trajectories (Key: blue: unprocessed points; magenta: points within kernel bandwidth; green: mode-seeking path; triangles mode at each iteration)(a) Initial trajectory representation. (b) - (d) Results of the 1^{st}, 5^{th}, and 8^{th} iteration of the mode seeking procedure (zoom)

([11]). Let $\chi_l \in \Psi_j$; $l = 1, ..., L$ be a set of L data points. The multivariate density estimator $\hat{f}(x)$ is defined as

$$\hat{f}(x) = \frac{1}{Lh^d} \sum_{l=1}^{L} K\left(\frac{x - \chi_l}{h}\right), \tag{1}$$

where h is the bandwidth and *K(.)* is a kernel, defined as

$$K(x) = \begin{cases} \frac{1}{2V_d}(d+2)(1 - x^T x) & \text{if } x^T x < 1 \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

with V_d representing the volume of a *d-dimensional* sphere. The density gradient estimate of the kernel can be written as

$$\hat{\nabla} f(x) = \nabla \hat{f}(x) = \frac{1}{Lh^d} \sum_{l=1}^{L} \nabla K\left(\frac{x - \chi_l}{h}\right). \tag{3}$$

Equation (3) can be re-written as

$$\hat{\nabla} f(x) = \frac{d+2}{h^d V_d} \left(\frac{1}{L_c} \sum_{\chi_l \in S(x)} (\chi_l - x) \right), \tag{4}$$

where *S(x)* is a hypersphere of radius h, with volume $h^d V_d$, centered in x and containing L_c data points. The Mean-shift vector $\zeta_h(x)$ is defined as

$$\zeta_h(x) = \frac{1}{L_c} \sum_{\chi_l \in S(x)} (\chi_l - x), \tag{5}$$

and, using Eq. (4), we can express $\zeta_h(x)$ as

$$\zeta_h(x) = \frac{h^d V_d}{d+2} \frac{\hat{\nabla} f(x)}{\hat{f}(x)}. \tag{6}$$

The output of the Mean-shift procedure is the set of data points associated to each mode. This process is illustrated in Fig. 1.

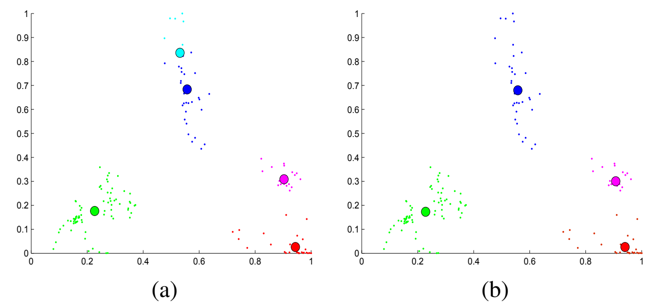

Fig. 2. Example of Cluster Merging (CM). (a) Initial trajectory clustering result (5 clusters); (b) final clustering result after CM (4 clusters)

2.2. Fuzzy cluster analysis and refinement

To refine the clustering results, we apply a Cluster Merging (CM) procedure that fuses two adjacent clusters if the density modes are sufficiently close. The *proximity condition* is defined by the 10% of the kernel bandwidth h. Each trajectory may have a different degree of belongingness to more than one cluster, in multiple feature spaces. To obtain the final clustering, each trajectory is assigned to a particular cluster if its belongingness is consistent across all feature spaces.

Let ξ_k and ξ_{k+1} be the number of clusters in Ψ_k and Ψ_{k+1}, with $\xi_k \leq \xi_{k+1}$ (note that different feature spaces may generate different numbers of clusters). Also, let $C_i^k \in \Psi_k$ and $C_j^{k+1} \in \Psi_{k+1}$ be clusters in the respective feature spaces. The next step is to find the correspondence among the clusters found in feature spaces. Let $\hat{\nu}$ be the index of the cluster in Ψ_{k+1} that has the maximum correspondence with the i^{th} cluster of Ψ_k, i.e.:

$$\hat{\nu} = argmax(C_i^k \cap C_j^{k+1}), \tag{7}$$

with $j = 1, ..., \xi_{k+1}$. Let the cluster $B_i = C_i^k \cap C_{\hat{\nu}}^{k+1}$ contain the overlapping elements in $C_{\hat{\nu}}^{k+1}$ and C_i^k. If $\{ \Delta \}_i^q$, with $q = 1, 2$, represents non-overlapping elements, then $\Delta_i^1 = C_i^k - (C_i^k \cap C_{\hat{\nu}}^{k+1})$ and $\Delta_i^2 = C_{\hat{\nu}}^{k+1} - (C_i^k \cap C_{\hat{\nu}}^{k+1})$, with Δ_i^1 and Δ_i^2 forming new independent clusters.

Finally, CM is applied again to merge adjacent clusters based on the *proximity condition* and the modes associated to too few data points are considered outliers. The outlier condition is set as the 5% of the maximum peak in the dataset. An example of CM result is shown in Fig. 2

3. RESULTS

We demonstrate the proposed clustering approach on three real outdoor traffic scenes and compare the results with those obtained with state-of-the-art methods[1]. The following test sequences are used: $S1$, a highway surveillance sequence from the *MPEG-7* dataset (134 trajectories); $S2$ and $S3$, from *VACE* [12] dataset (47 and 159 trajectories, respectively). All sequences are captured at 25 Hz. The number of clusters for each sequence (ground truth), γ, is: $\gamma(S1) = 2$, the $\gamma(S2) = 3$, $\gamma(S3) = 4$. The *accuracy*, λ, of a clustering result is calculated as

$$\lambda = \begin{cases} \frac{1}{N} \sum_{i=1}^{\gamma} (P_i) & \text{if } \tau > \gamma \\ \frac{1}{N} \sum_{i=1}^{\tau} (P_i) & \text{otherwise} \end{cases}, \tag{8}$$

[1]High resolution figures with the results and additional demo videos are available at http://www.elec.qmul.ac.uk/staffinfo/andrea/traj.html

where N is the total number of trajectories in a dataset, P is the number of trajectories correctly clustered in a cluster C_m, and τ is the total number of clusters estimated by the algorithm. If $\tau > \gamma$, then, after sorting the P for all τ clusters in descending order, we select the top γ clusters to calculate the accuracy.

We compare the proposed method with the Coarse-to-Fine clustering technique (CF) ([5]). Moreover, we modified the *second* stage of CF by replacing trajectory interpolation based distance measure with the Longest Common Subsequences LCSS ([9]) to improve its performance with real data. We refer to this modified technique as Modified Coarse-to-Fine clustering (MCF).

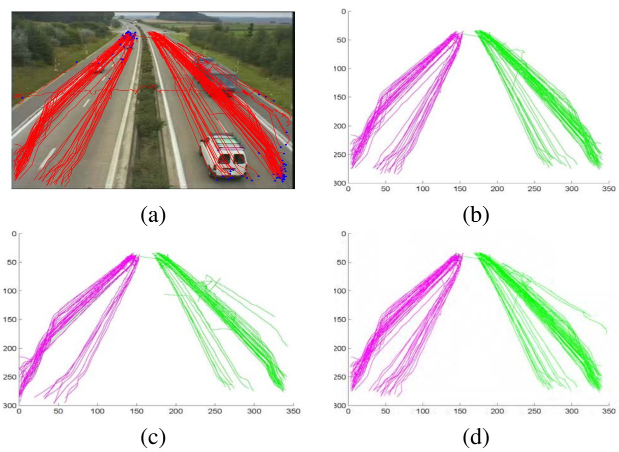

Fig. 3. Comparison of trajectory clustering results on the dataset $S1$ (a). (b) Clustering with FMS; (c) clustering with CF; (d) clustering with MCF

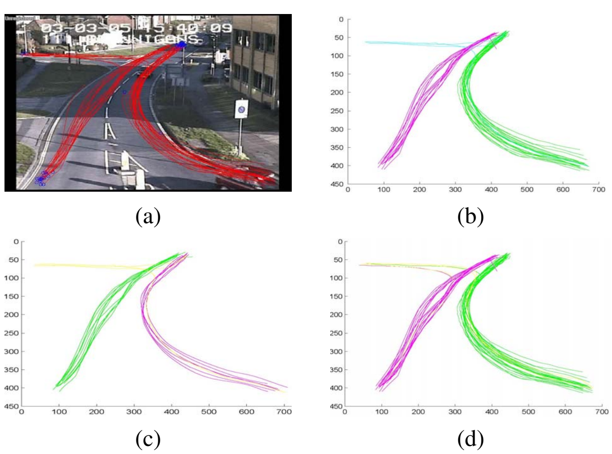

Fig. 4. Comparison of trajectory clustering results on the dataset $S2$ (a). (b) Clustering with FMS; (c) clustering with CF; (d) clustering with MCF

Fig. 3, Fig. 4, and Fig. 5 show the final clustering results for the methods under analysis. Although the three clustering techniques estimate the correct number of clusters, it is possible to notice sensible variations between the clusters discovered by the different methods. Table 1 shows a comparison of the *accuracy* obtained with the three methods (98.50% for FSM, 94.93% for CF and 97.84% for MCF). In particular, in $S1$ and $S2$, MCF performed better than FMS by 0.08% and 0.2% respectively, but, for $S3$, the performance of FMS is 1.70% better than MCF. When the number of trajectories increases and the variation among different object trajectories is low, FMS outperforms MCF. The cluster level analysis also demonstrates that , on average, MCF performed better than CF (+2.34% for S1, +3.04% for S2, and +3.08% for S3).

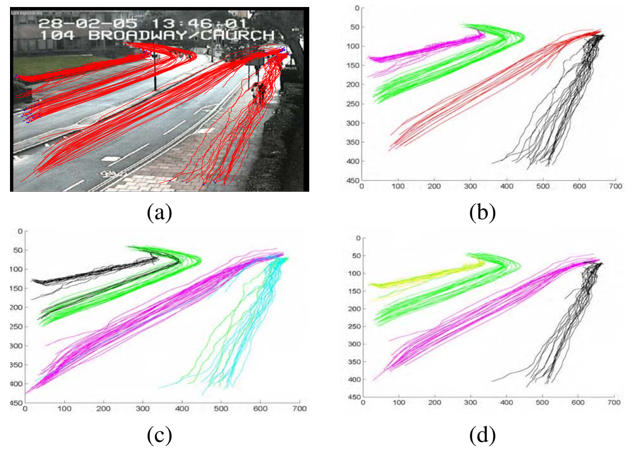

Fig. 5. Comparison of trajectory clustering results on the dataset $S3$ (a). (b) Clustering with FMS; (c) clustering with CF; (d) clustering with MCF

Finally, to test the *robustness* of the algorithms, we evaluate the clustering results when the input data are corrupted (i) by noise and (ii) by a reduction of the trajectory sampling rate (as MCF outperformed CF, we compare FMS and MCF only). Fig. 6 (left column) shows the clustering results for the $S1$ dataset after adding Gaussian noise (up to $\sigma = 10\%$ of the length of the longest trajectory) to the input data. Fig. 6 (right column) shows the clustering results after reducing the number of observations for each trajectory in the $S1$ dataset (progressively decreased sampling rate up to sub-subsampling step 5). The results for both FMS and MCF on the three dataset are visualized in Fig. 7. It is possible to notice that for FMS every 1% increase in noise causes approx. 0.45% degradation in the accuracy, whereas the degradation for *MCF* is approx. 5%. Moreover, a progressive downsampling by 1 causes 4.1% and 14.5% overall degradation for FMS and MCF, respectively. The comparison with the state of the art shows that the proposed approach is more robust to noise and to sub-sumpling of the trajectory dataset.

4. CONCLUSIONS

We proposed an unsupervised fuzzy clustering algorithm for object trajectories analysis. The algorithm is based on first applying Meanshift on distinct feature spaces, one generated from PCA and one generated by object velocities. Next, adjacent clusters in a feature space are merged and the final cluster configuration is established by finding consistent behaviors of the trajectories in all feature spaces. Moreover, final clusters with few associated trajectories are considered as outliers and removed. The algorithm was validated on real outdoor traffic scenarios from standard test sequences and compared with state-of-the-art approaches. The results demonstrated that the proposed algorithm is more robust to noise and to variations in the frequency of object observations. Our current work addresses the issue of perspective projection in trajectory clustering.

Table 1. Comparison of trajectory clustering *accuracy* for FMS, CF, and MCF on three standard datasets

	S1		S2			S3				Overall
Algorithm	C1	C2	C1	C2	C3	C1	C2	C3	C4	average
FMS	98.24	97.81	99.45	99.87	97.37	99.96	94.10	99.87	99.85	**98.50**
CF	97.31	94.23	99.41	92.27	96.57	96.87	93.68	98.65	85.43	**94.93**
MCF	98.90	97.32	99.54	99.70	98.13	93.12	98.88	99.93	95.02	**97.84**

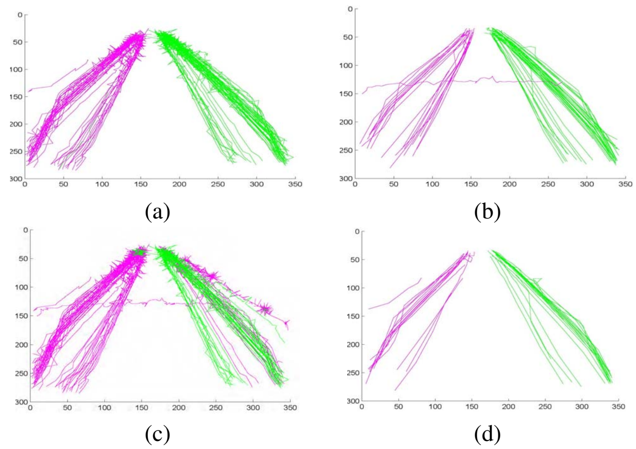

Fig. 6. Comparison of trajectory clustering *robustness* for FMS and MCF on $S1$ for noisy and sub-sampled input data. (a) FMS clustering results on noisy data ($+10\%$); (b) FMS clustering results on sub-sampled data (sub-sampling step=5); (a) MCF clustering results on noisy data ($+10\%$); (b) MCF clustering results on sub-sampled data (sub-sampling step=5)

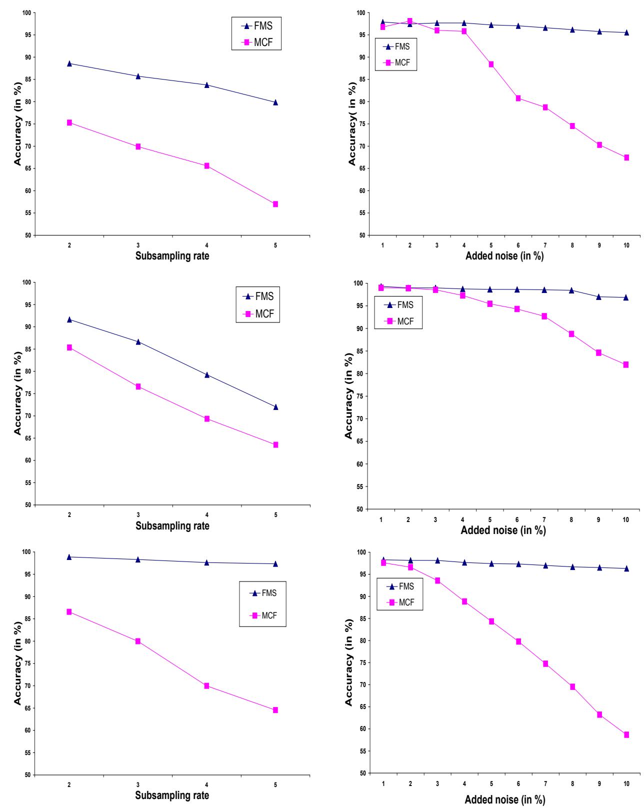

Fig. 7. Comparison of trajectory clustering *robustness* for FMS and MCF for noisy and sub-sampled input data. Left column: clustering accuracy for sub-sampled data. Right column: clustering accuracy for noisy data. First row: $S1$ dataset; second row: $S2$ dataset; third row: $S3$ dataset

5. REFERENCES

[1] F. Porikli, "Trajectory distance metric using hidden markov model based representation," *Mitsubishi Electric Research Laboratories Cambridge, TR2004-030*, 2004.

[2] M. Breitenbach and G.Z. Grudic, "Clustering through ranking on manifolds," *in Proc. of ACM intl. conf. on Machine learning*, pp. 73–80, July 2005.

[3] F.I. Bashir, A.A. Khokhur, and D. Schonfeld, "Segmented trajectory based indexing and retrieval of video data," *in Proc. of Intl. Conf. on Image Processing, ICIP*, pp. 623–626, Sept. 2003.

[4] G. Antonini and J.P. Thiran, "Counting pedestrians in video sequences using trajectory clustering," *IEEE Trans. on Circuit and Systems for Video Technology*, pp. 623–626, Aug. 2006.

[5] X. Li, W. Hu, and W. Hu, "A coarse-to-fine strategy for vehicle motion trajectory clustering," *in Proc. of Intl. Conf. on Pattern Recognition, ICPR*, pp. 591–594, Aug. 2006.

[6] J. Aleotti and S. Caselli, "Trajectory clustering and stochastic approximation for robot programming by demonstration," *in Proc. of IEEE Intl. conf. on Intelligent Robots and Systems*, pp. 1029–1034, Aug. 2005.

[7] C. Piciarelli, G. L. Foresti, and L. Snidaro, "Trajectory clustering and its applications for video surveillance," *In Proc. IEEE Intl. Conf. on Advanced Video and Signal based Surveillance*, pp. 40–45, Sept. 2005.

[8] J. Melo, A. Naftel, A. Bernardino, and J.S. Victor, "Retrieval of vehicle trajectories and estimation of lane geometry using non-stationary traffic surveillance cameras," *Advanced Concepts for Intelligent Vision Systems, ACIVS*, Aug. 2004.

[9] H. Fashandi and A.M.E. Moghaddam, "A new invariant similarity measure for trajectories," *in Proc. of IEEE Intl. Symposium on Computational Intelligence in Robotics and Automation*, pp. 631–634, June 2005.

[10] Z. Zhang, K. Huang, and T. Tan, "Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes," *in Proc. of Intl. Conf. on Pattern Recognition, ICPR*, pp. 1135–1138, Aug. 2006.

[11] D. Comaniciu and P. Meer, "Distribution free decomposition of multivariate data," *Pattern Analanlysis and Applications*, pp. 602–610, 1999.

[12] R. Kasturi, "Performance evaluation protocol for face, person and vehicle detection & tracking in video analysis and content extraction, vace-ii," Jan. 2006.