

Trajectory Abstracting with Group-Based Signal Denoising

Xiaoxiao Luo, Qing Xu^(✉), Yuejun Guo, Hao Wei, and Yimin Lv

School of Computer Science and Technology, Tianjin University, Tianjin, China
qingxu@tju.edu.cn
<http://cs.tju.edu.cn/szdw/jsfjs/xuqing/>

Abstract. Trajectory abstracting is to compendiously summarize the substance of a lot of information delivered by the trajectory data. In this paper, to cope with complex trajectory data, we propose a novel framework for abstracting trajectories from the perspective of signal processing. That is, trajectories are designated as signals, manifesting the copious information that varies with time and space, and denoising is exploited to concisely communicate the trajectory data. Resampling of trajectory data is firstly performed, based on achieving the minimum *Jensen-Shannon divergence* of the trajectories before and after being re-sampled. The resampled trajectories are matched into groups according to their similarity and, a *non-local* denoising approach based on wavelet transformation is developed to produce summaries of trajectory groups. Our new framework can not only offer multi-granularity abstractions of trajectory data, but also identify outlier trajectories. Extensive experimental studies have shown that the proposed framework achieves very potential results in trajectory summarization, in terms of both objective evaluation metrics and subjective visual effects. To the best of our knowledge, this is the first to deploy the group-based signal denoising technique in the context of summarizing the trajectory data.

Keywords: Trajectory abstracting · Multi-granularity abstractions · Signal processing · *non-local* denoising · Wavelet transformation

1 Introduction

Trajectory data is very useful for a lot of practical fields such as intelligent transportation and so on [13]. The processing of the trajectory data is fundamentally based on clustering, which is basically one of the most powerful techniques to obtain the patterns and knowledge of these data for the better abstraction and full employment of them [8]. Unfortunately, the performance of clustering may degrade when dealing with complex trajectory data. Here the meaning of complexity is at least threefold. First, different trajectories can have largely diverse numbers of sample points. Second, similarity between trajectories happening in one area of a scene can significantly differ from that in another area. Third, outliers occur together with common clusters of trajectories. An example in Fig. 4(f)

shows the difficulty of the clustering technique for treating complex trajectory data.

In this paper, in order to effectively analyze and understand complex trajectories, we propose a totally new approach, called trajectory abstracting, which is significantly different from the clustering scheme. That is, considering that trajectories are in fact signals handling the information that change with time and space, the abstraction of trajectory data is performed from the perspective of signal denoising. At first, resampling of trajectories is taken for combatting the situation that numbers of trajectory sample points are largely different. Then, variations of trajectory similarities occurring in different scene areas are tackled by iterative group-based “*non-local*” denoising. Our denoising scheme provides trajectory details and summarizations with multiple granularities, leading to better analysis and understanding of the complexity of data.

The remainder of this paper is organized as follow. The next Section covers the related work. The developed trajectory abstracting framework is described in Sect. 3. Section 4 introduces two metrics to evaluate the proposed framework for trajectory abstraction. Experimental results are presented in Sect. 5. The final Section concludes the paper.

2 Related Work

The use of clusters (with their centroids) may be the most popular way to represent the patterns of trajectory data [11]. A lot of good clustering techniques used for trajectory data have been developed [11, 13]. For example, classical algorithms include k-means [9], BIRCH [17], DBSCAN [6], OPTICS [4], and STING [16]. Among them, k-means and DBSCAN may be the mostly applied in practice due to their easy use and efficiency. However, as indicated in Sect. 1, the clustering may not benefit for the reasonable handling of the complex trajectory data. For the sake of treating complex trajectory data, data filtering is a good preprocess. For instance, Keogh et al. apply the wavelet transform to represent a single trajectory and do clustering by k-means [15]. Notice that our proposed approach is largely different from that by Keogh et al., we perform the wavelet denoising across all the trajectories in a similarity group.

Block-Matching and 3D Filtering (*BM3D*) may be the best state-of-the-art filter for noisy image/video data [5], which utilize coherence among pixel patches to do denoising very effectively. The core of *BM3D* inspires us, but we significantly extend *BM3D* to cope with the trajectory data that are greatly different from the image/video data.

3 The Framework for Trajectory Abstraction

Our proposed framework is composed of two main components, resampling and group-based *non-local* denoising. An abstraction in one granularity is obtained by an iteration of denoising. Our method iteratively outputs trajectory abstractions with multi-granularities and outliers.

3.1 Resampling

In general, complex trajectories have various numbers of sample points. Also, trajectories are usually corrupted with noise. We propose to make use of a resampling technique to smooth out noise and to make resampled trajectories have a same number of sample points, leading to a better trajectory abstraction. Each trajectory is resampled to have a equal distance interval. For the sake of obtaining the minimum bias introduced by resampling, we find an optimal number n_{opt} of sample points to preserve information of the original trajectory as much as possible. Since the trajectory shape is very important, we obtain n_{opt} by minimizing the widely used *Jensen-Shannon divergence (JSD)* between the trajectory shapes before and after being re-sampled. Actually, the trajectory shape can be characterized by the distribution of the angles at the sample points.

Suppose a trajectory $T = \{p_1, p_2, \dots, p_n\}$, is with n sample points, here $p_i = (x_i, y_i)$ is the 2-D coordinates in a ground plane. We obtain its angles $\{\theta_j\}, j = 1, 2, \dots, n - 2$ (Fig. 1), to build up a probability distribution, $\mathbf{P}^{ori} = \{\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_M\}$, using the angle histogram with M bins ($M = 10$ in this paper). For the resampled version T' , we obtain its probability distribution \mathbf{P}^{res} similarly, and then the *JSD* distance between T and T' is computed by

$$JSD(T; T') = H\left(\frac{\mathbf{P}^{ori} + \mathbf{P}^{res}}{2}\right) - \frac{H(\mathbf{P}^{ori}) + H(\mathbf{P}^{res})}{2} \tag{1}$$

where $H(\cdot)$ denotes the *Shannon* entropy. In this paper, the *JSD* value between a dataset before and after being resampled is defined as the mean of *JSD*s for the trajectories of this dataset. Apparently, the lower *JSD* indicates a higher similarity between the original and the resampled trajectory data. Thus, n_{opt} is located at the minimum in all the *JSD* values resulted from the trajectories with different numbers of resampled points. For example, Fig. 2 presents such a plot for the *Pedestrian* dataset (Sect. 5) and undoubtedly, $n_{opt} = 21$ corresponds to the minimum *JSD*.

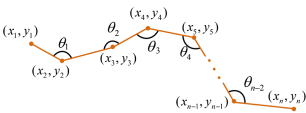


Fig. 1. A trajectory with its angles

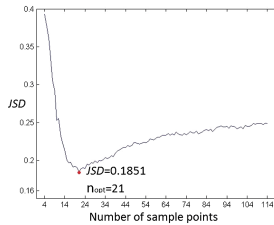


Fig. 2. Determination of n_{opt} based on the *JSD* value

3.2 non-local Denoising

With all the trajectories after being resampled, we perform a procedure called “non-local” denoising, involving three phases, *Matching*, *Thresholding* and *Combining*, in multiple iterations for obtaining the multi-granularity abstractions. In fact, the summaries in different abstracted levels give a more clear and better understanding about the trajectory data. It is worthy to point out that outliers can be detected in the process of non-local denoising. The outliers, picked out in an iteration, are not be included for the later iterative operations.

For easy understanding, we define $\mathbf{TR}_k = \{T_{k,1}, T_{k,2}, \dots, T_{k,l}\}$ ($k \geq 1$) as the k -th iterative output with l abstracted trajectories (and also as the input of the $(k + 1)$ -th iteration). Here \mathbf{TR}_0 is the product by the trajectory resampling, as the input of the 1-st iteration. The iterative execution is terminated when the output abstractions between two consecutive iterations do not change anymore.

Matching. In this phase, similar trajectories are matched to establish groups. Notice that, simultaneously, outlier trajectories may be identified by the similarity matching. In concrete, each input trajectory is considered as a “reference”, and the nearby trajectories similar to this reference are matched to form a group. Assume that there exists a reference trajectory $T_{k,r} \in \mathbf{TR}_k$, its similarity group $\mathbf{TG}_{k,r}$ is

$$\mathbf{TG}_{k,r} = \{T_{k,j} \in \mathbf{TR}_k \mid Diff(T_{k,r}, T_{k,j}) < \tau \ (j \neq r)\} \tag{2}$$

where $Diff(\cdot)$ is a distance function and the simple and widely used Euclidean distance is adopted here. τ is a threshold, selected adaptively (Sect. 3.3), to determine whether two trajectories are distant or not.

Notice that, a trajectory can be used in more than one matching-based groups. Some outliers with few similar trajectories can be distinguished in this phase. That is, a trajectory is defined as outlier if the number of trajectories in its similarity group is less than η ($\eta = 3$ is used, for simplicity).

Thresholding. The wavelet threshold technique [7] is operated for the trajectories in the whole group, rather than just for a single trajectory, to obtain a compacted and summarized representation of these trajectories.

Suppose a reference trajectory $T_{k,r}$, having g number of neighboring trajectories $T_{k,j}$ ($j = 1, 2, \dots, g; j \neq r$), is with a similarity group which is now denoted as a matrix

$$\mathbf{TG}_{k,r} = \begin{bmatrix} T_{k,1}^r \\ T_{k,2}^r \\ \dots \\ T_{k,g}^r \\ T_{k,r}^r \end{bmatrix} = \begin{bmatrix} p_1^1 & p_2^1 & p_3^1 & \dots \\ p_1^2 & p_2^2 & p_3^2 & \dots \\ \dots & \dots & \dots & \dots \\ p_1^g & p_2^g & p_3^g & \dots \\ p_1^r & p_2^r & p_3^r & \dots \end{bmatrix} = [s_1 \ s_2 \ s_3 \ \dots]. \tag{3}$$

Here $s_i = [p_i^1, p_i^2, \dots, p_i^g, p_i^r]^T$ is designated as the i -th signal of this similarity group, which is actually the collection of the i -th sample points of the different

trajectories in the group. The wavelet threshold is used for the noise filtering for s_i . At first, wavelet transform is performed on s_i to obtain the corresponding coefficients. Then the high frequency components are completely suppressed if and only if their values are smaller than a adaptively determined ϵ (Sect. 3.3). Finally, the denoised signal \tilde{s}_i is given by using inverse wavelet transform

$$\tilde{s}_i = F^{-1}(\mathcal{Y}(F(s_i))), \quad (4)$$

where \mathcal{Y} , F and F^{-1} denote the wavelet thresholding, wavelet transform and its inverse, respectively. In this paper Haar wavelet is employed and the denoised similarity group is as follows

$$\widetilde{\mathbf{TG}}_{k,r} = [\tilde{s}_1, \tilde{s}_2, \dots] = \begin{bmatrix} \widetilde{T_{k,1}^r} \\ \widetilde{T_{k,2}^r} \\ \dots \\ \widetilde{T_{k,g}^r} \\ \widetilde{T_{k,r}^r} \end{bmatrix} \quad (5)$$

Combining. Basically, for a resampled trajectory, more than one denoised versions can be obtained if it belongs to several similarity groups. In this case, we further combine these versions by averaging them to obtain a better abstraction with richer *non-local* similarity information.

3.3 Parameter Selection for Group-Based Denoising

All the important parameters used for grouping and denoising are adaptively selected, achieving the effectiveness and robustness of the proposed technique.

The distance threshold τ for the matching phase is determined based on the statistically averaged Euclidean distance between two trajectories in the dataset under consideration. Suppose there exists n trajectories in a dataset. For each trajectory T , $n - 1$ Euclidean distances between T and all the other trajectories are sorted ascendingly. The average of x ($1 \leq x < n$) smallest distances, denoted by Y , can be calculated. A plot of Y versus x is then provided (Fig. 3(a)). The Y value corresponding to the maximum of the second derivative of this plot (Fig. 3(b)) can be found. All the Y values resulted from n trajectories are averaged to obtain τ . We have observed that ϵ used for the thresholding phase is closely related with τ , $\epsilon = 2.5 \times \tau$ can do very well.

4 Evaluation

In this paper, we develop two quantitative criteria, *Integrity* (*INT*) and *Fidelity* (*FID*), to systematically and objectively evaluate the performance of our proposed framework for trajectory abstracting.

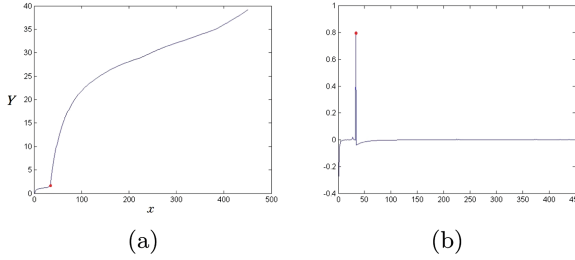


Fig. 3. (a) The plot of Y versus x of the trajectory T . (b) The second derivative of (a). The maximum point in (b) with red color corresponds to the red point in (a) (Colour figure online)

4.1 Fidelity (FID)

FID is to measure the similarity between the trajectory datasets before and after being abstracted, and this is a quality evaluation on the trajectory summarization. Given a original trajectory dataset without detected outliers $\mathbf{TR} = \{T_1, T_2, \dots, T_n\}$ and its abstract of the last iteration (or the cluster centroids) $\mathbf{TR}' = \{T'_1, T'_2, \dots, T'_m\}$, we define FID by

$$FID(\mathbf{TR}, \mathbf{TR}') = 1 - \frac{\max_i \{ \min_j Diff(T_i, T'_j) \}}{MaxDiff}, \quad (6)$$

Here $Diff()$ is the hausdorff distance, and $MaxDiff$ is the global maximum distance between \mathbf{TR} and \mathbf{TR}' . Apparently, a higher FID means the trajectory abstraction can express the dataset more accurately.

4.2 Integrality (INT)

INT measures the degree of coverage by abstracted trajectories for the trajectory dataset. Suppose $\mathbf{TR}' = \{T'_1, T'_2, \dots, T'_m\}$ is the abstracted output by the last iteration (or the cluster centroids), Its original trajectory without detected outliers is $\mathbf{TR} = \{T_1, T_2, \dots, T_n\}$. The INT between original data and the abstraction is defined as

$$INT(\mathbf{TR}, \mathbf{TR}') = \frac{1}{|\mathbf{TR}|} \sum_i Diff(T_i, T'_i) \quad (7)$$

Here $Diff()$ is the hausdorff distance. A low INT indicates a completely coverage of the original.

5 Experiments

We have done extensive tests to evaluate the performance of our proposed framework, including 7 public databases listed in Table 1. Considering that trajectory abstracting and cluster are both aimed at pattern mining, two typical and effective cluster methods widely used in practical applications, DBSCAN and k-means, are compared with our technique.

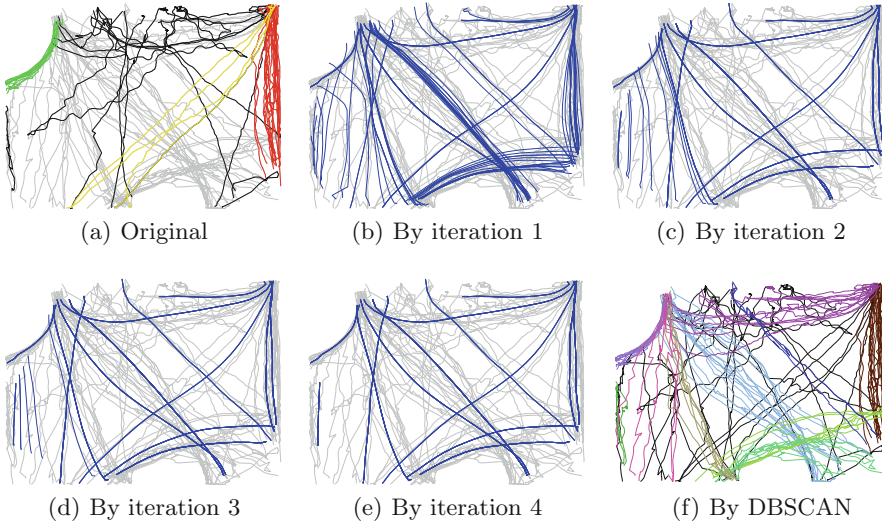


Fig. 4. Comparison of different methods on *Pedestrian* (Colour figure online)

5.1 Results Analysis of *Pedestrian* Dataset

Figure 4(a) is the original trajectories of *Pedestrian*, where twenty trajectories in black color are outliers. The numbers of sample points in this dataset are largely diverse, varying between 33 and 422. The abstracted trajectories obtained in four iterations are respectively shown in Fig. 4(b)–(e). Ten clusters and outliers given by DBSCAN are displayed in Fig. 4(f) with different colors. Overall, the proposed technique compresses and summarizes very messy data into neat abstractions with multiple granularities. Fourteen most abstracted patterns, obtained in the last iteration, in reality exactly exhibit the fourteen pedestrian paths in this scenario (Fig. 4(e)).

Complex trajectory data can be represented by the multi-granularity abstractions very well. However, DBSCAN may give out unsatisfactory outputs. For example, the trajectories in red color (Fig. 4(a)) are messy and, they have two opposite directions. Iteration 1 makes them become neat and close, and then Iteration 2 outputs two abstracted patterns presenting trajectories with two opposite directions. As a result, the products by both two iterations indicate the processing of abstracting and, more importantly, jointly present trajectory details and summarizations. In contrast, DBSCAN falsely merge these trajectories two opposite directions into a single cluster.

As for the complex trajectory data with diverse trajectory similarities in different scene areas, abstractions by the proposed technique can give a concise representation of the original data. But the clusters by DBSCAN may be mistaken in this case. For instance, four trajectories in yellow color (Fig. 4(a)) are distant, compared with the trajectories in other areas. DBSCAN wrongly iden-

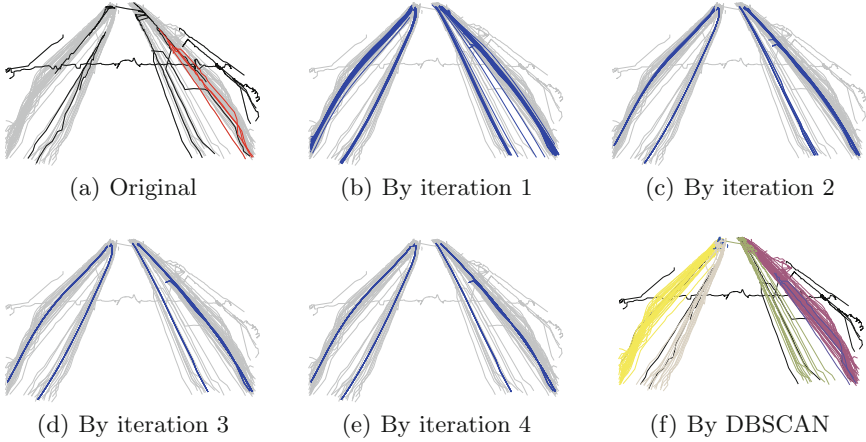


Fig. 5. Comparison of different methods on *highway* (Colour figure online)

tify these four as outliers. Notably, our approach generates a condensed pattern, depicting the data satisfactorily.

5.2 Results Analysis of *Highway* Dataset

Original trajectories of *Highway* are shown in Fig. 5(a), and eighteen black trajectories are outliers. The trajectory abstractions at multiple granularities clearly expose the four traffic lanes. The lengths of the trajectories in the leftmost lane are diverse. Most of them are longer than the four trajectories rendered in red (Fig. 5(a)). These four short trajectories are compressed into one abstracted pattern which differs from their adjacent trajectories in the same lane. DBSCAN uses a single cluster for all the trajectories belonging to this traffic lane. But here a single cluster is not enough to feature the trajectories in largely different lengths.

5.3 Objective Evaluation

The recall and precision values by several methods for anomaly detection are listed in Table 1. It is obvious that our method performs the best to detect outliers. This is due to that *non-local* denoising can emphasize outliers in the process of trajectory summarizing.

The *FID* and *INT* scores listed in Table 2 indicate that the proposed technique achieves the best, consistent with the visual results discussed above.

6 Conclusions

In this paper, for the purpose of abstracting the complex trajectory data, we have established a novel effective technique in which a *non-local* signal denoising approach is exploited to obtain the summaries of the trajectory groups.

Table 1. Evaluation on outlier detection

Database	Recall			Precision		
	<i>ours</i>	<i>DBSCAN</i>	<i>k-means</i>	<i>ours</i>	<i>DBSCAN</i>	<i>k-means</i>
<i>Pedestrian</i> [1]	0.90	0.87	0.69	1.0	0.95	0.63
<i>Highway</i> [3]	0.89	0.94	0.61	0.89	0.84	0.55
<i>Simulation</i> [14]	1.0	1.0	0.79	1.0	0.95	0.78
<i>Edinburgh</i> [1]	0.95	0.76	0.48	0.81	0.86	0.55
<i>Aircraft</i> [2]	0.88	0.89	0.67	0.95	0.81	0.60
<i>CROSS</i> [12]	0.98	0.93	0.50	0.92	0.81	0.64
<i>OMNI</i> [12]	0.96	0.84	0.55	0.79	0.85	0.66

Table 2. Objective evaluation on summarization

	<i>Pedestrian</i>		<i>Highway</i>		<i>Simulation</i>		<i>Edinburgh</i>		<i>Aircraft</i>		<i>CROSS</i>		<i>OMNI</i>	
	<i>FID</i>	<i>INT</i>	<i>FID</i>	<i>INT</i>	<i>FID</i>	<i>INT</i>	<i>FID</i>	<i>INT</i>	<i>FID</i>	<i>INT</i>	<i>FID</i>	<i>INT</i>	<i>FID</i>	<i>INT</i>
<i>ours</i>	0.59	323	0.58	147	0.79	159	0.45	871	0.88	101	0.73	87	0.63	338
<i>DBSCAN</i>	0.53	434	0.56	181	0.81	191	0.38	920	0.61	155	0.69	109	0.55	395
<i>k-means</i>	0.56	451	0.51	252	0.75	171	0.43	833	0.68	116	0.80	101	0.61	296

The widely used *JSD* is investigated to do the resampling of the trajectories with varied lengths. The group-based denoising is iterated to obtain the multi-granularity condensed and summarized trajectory representations, which in the meantime may include some outliers of the data. We have also proposed two metrics to quantitatively evaluate the new framework for trajectory abstraction. A lot of experiments have clearly shown that the proposed technique is very helpful for understanding and utilizing the complex trajectory data in practice. To our knowledge this is the first deployment of the mighty group-based signal denoising technique for trajectory abstracting.

Several improvements for the proposed trajectory summarizing will be performed in our future work. In order to have some emphasis on the local features of long trajectories, a long trajectory would be segmented and, our proposed framework would be extended to deal with the trajectory segments. Furthermore, the powerful and general *visual analytics* technique [10] would be used to develop visual interactions to improve the trajectory abstraction.

Acknowledgments. This work has been funded by Natural Science Foundation of China (61471261, 61179067, U1333110), and by grants TIN2013-47276-C6-1-R from Spanish Government and 2014-SGR-1232 from Catalan Government (Spain).

References

1. <http://homepages.inf.ed.ac.uk/rbf/forumtracking/>
2. <https://c3.nasa.gov/dashlink/resources/132/> (2011)
3. Anjum, N., Cavallaro, A.: Multifeature object trajectory clustering for video analysis. *IEEE Trans. Circ. Syst. Video Technol.* **18**(11), 1555–1564 (2008)
4. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: Optics: ordering points to identify the clustering structure. In: *ACM Sigmod Record*, vol. 28, pp. 49–60. ACM (1999)
5. Dabov, K., Foi, A., Katkovich, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. Image Proces.* **16**(8), 2080–2095 (2007)
6. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Kdd*, vol. 96, pp. 226–231 (1996)
7. Johnstone, I.M., Silverman, B.W.: Wavelet threshold estimators for data with correlated noise. *J. Royal Stat. Soc.: Ser. B (Stat. Methodol.)* **59**(2), 319–351 (1997)
8. Laxhammar, R., Falkman, G.: Online learning and sequential anomaly detection in trajectories. *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(6), 1158–1173 (2014)
9. Lloyd, S.: Least squares quantization in pcm. *IEEE Trans. Inf. Theory* **28**(2), 129–137 (1982)
10. May, R., Hanrahan, P., Keim, D.A., Shneiderman, B., Card, S.: The state of visual analytics: views on what visual analytics is and where it is going. In: *2010 IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pp. 257–259. IEEE (2010)
11. Morris, B.T., Trivedi, M.M.: A survey of vision-based trajectory learning and analysis for surveillance. *IEEE Trans. Circ. Syst. Video Technol.* **18**(8), 1114–1127 (2008)
12. Morris, B.T., Trivedi, M.M.: Trajectory learning for activity understanding: unsupervised, multilevel, and long-term adaptive approach. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(11), 2287–2301 (2011)
13. Morris, B.T., Trivedi, M.M.: Understanding vehicular traffic behavior from video: a survey of unsupervised approaches. *J. Electron. Imaging* **22**(4), 041113 (2013)
14. Piciarelli, C., Micheloni, C., Foresti, G.L.: Trajectory-based anomalous event detection. *IEEE Trans. Circ. Syst. Video Technol.* **18**(11), 1544–1554 (2008)
15. Vlachos, M., Lin, J., Keogh, E., Gunopulos, D.: A wavelet-based anytime algorithm for k-means clustering of time series. In: *Proceedings of the Workshop on Clustering High Dimensionality Data and Its Applications*, Citeseer (2003)
16. Wang, W., Yang, J., Muntz, R., et al.: Sting: a statistical information grid approach to spatial data mining. *VLDB* **97**, 186–195 (1997)
17. Zhang, T., Ramakrishnan, R., Livny, M.: Birch: an efficient data clustering method for very large databases. In: *ACM SIGMOD Record*, vol. 25, pp. 103–114. ACM (1996)