Understanding Trajectory Behavior: A Motion Pattern Approach

Mahdi M. Kalayeh, Stephen Mussman*, Alla Petrakova* and Mubarak Shah

Abstract—In today's world, a tremendous amount of data is generated and stored every second. Having tools to extract patterns from these gigantic information volumes allows data analysts to understand the data behaviors quickly and easily. Trajectories can be thought of as evolution of data in time. In this paper, we propose a motion pattern approach to mine frequent behaviors in trajectory data. Motion patterns have been shown to be very effective in extracting dominant motion behaviors in computer vision applications. We have revised the basic motion pattern idea so that it successfully solves the general task of trajectory clustering. We have evaluated the effectiveness of our proposed method in an extensive set of experiments on diverse datasets. Further, we have shown how our proposed method handles difficulties in the general task of trajectory clustering that challenge the existing state-of-the-art methods.

Index Terms—Trajectory, Pseudo-optical flow, Motion component, Reachability, Motion Pattern.

1 Introduction

Recently, there has been a great amount of research in mining frequent behaviors from trajectory data. A trajectory is an N-dimensional path that does not necessarily correspond to a physical object, but may correspond to the time evolution of a N-dimensional feature vector. This paper will propose a novel method for extracting dominant trajectory behaviors; while our method will be described for the case of 2-dimensional trajectories, it can be extended to higher dimensions for further applications.

[Several paragraphs were omitted pending publication. These paragraphs introduce the idea behind the proposed method.]

Seeking to find common behaviors in trajectory data, some methods group whole trajectories into clusters while others attempt to mine regional patterns that trajectories follow in a subsegment of their evolutions. There exist a large number of methods in the literature but due to the space limitation, only a subset of them are selected to be introduced in this section. We tried to have a balance between the most recent methods and the most popular ones.

- *Mubarak Shah and Mahdi M. Kalayeh are with Center for Research in Computer Vision at the Department of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Fl, 32816. Stephen Mussman is with Department of Mathematics and Department of Computer Science, Purdue University, West Lafayette, IN 47907. Alla Petrakova is with Department of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Fl, 32816.*
- *Authors* have contributed equally and are listed alphabetically.*

Many of the existing trajectory clustering methods approach the problem by first defining a similarity function for trajectories and then using one of the well-established clustering procedures. For instance, [2] compared clustering results obtained by using the trajectory similarity measures DTW, LCSS, and modified Hausdorff distance in combination with agglomerative and spectral clustering. Fu $et.al.$ [3] used the average distance between corresponding trajectory points as the similarity measure, which required pre-processing and resampling of trajectories. The clustering was later applied in two steps, with the first step producing clusters corresponding to the larger dominant paths, and the second one subsequently refining results obtained by the first one. Morris and Trivedi [4] surveyed performance of a wide variety of distance measures and clustering algorithms on several datasets with varying characteristics. Specifically, they compared trajectories using HU [5], PCA [6], DTW [7], LCSS [8], PF [9], and modified Hausdorff [10] distance and then obtained clusters using direct [11], divisive [12], agglomerative [8], hybrid [13], graph [14] and spectral clustering techniques [5].

Recently, Ferreira $et.al.$ [15] have proposed Vector field K-means that treats the trajectories as a whole and attempts to follow an iterative model similar in form to k-means clustering. In this approach, clusters are subsets of the trajectories. First, the trajectories are partitioned randomly into k clusters. Then, for each cluster, the trajectories in that cluster are used to form one of k vector fields. Further, clusters are updated by assigning each trajectory to the vector field that it fits best. These two steps are repeated in an iterative fashion until a convergence criteria is met. DivCluST proposed by Wu $et.al.$ [16] is a

partition based approach that clusters trajectories according to their speed in addition to their shape and position. First, it partitions the trajectories into representative segments by combining adjacent trajectory segments that have similar direction and speed. Then, a method similar to k-means is used to find cluster centers. The clusters are initialized as random subsets of the representative segments. Next, the representative segments of each cluster are averaged to form a cluster center. Finally, as in k-means, each of the representative segments is associated with a cluster center, then each cluster center is updated to be the average of the cluster's representative segments. This process will be repeated until a convergence criteria is met. Ulm $et.al.$ [17] have proposed a trajectory clustering approach that is distinctive in its online clustering ability and does not explicitly treat the trajectories as a whole, but in practice, it behaves as if it does. They defined a cluster as a vector field defined on a subset of the spatial range. When a trajectory is received, it is converted into a new vector field. If at any time two vector fields are similar, they are merged together into one. Since the entire trajectory is used to make a vector field and merging process largely conserves the shape of both fields that are being merged, the algorithm in practice behaves similarly to those approaches that treat the trajectories as a whole.

Giannotti $et.al.$ [18] proposed a very different way of treating trajectories where their density is used to form frequently visited rectangular regions. Then, each of the trajectories is represented as a sequence of visited regions with the transition times. They mined common subsequences with similar transition times that represent commonly taken paths similar in interpretation to trajectory clusters. It is worth mentioning that because of the exact phrasing of the problem, this approach gives very redundant clusters in practice. Lee $et.al.$ [19] observed that clustering trajectories as a whole fails to reveal portions of trajectories that exhibit a common behavior. To address this, they proposed TRACLUS, an algorithm that first partitions trajectories into a set of line segments and then groups similar segments into clusters using a density-based clustering algorithm for line segments (largely analogous to DBSCAN). This work has been further extended in [20], which proposed a feature generation framework, TraClass, that uses a combination of region-based and trajectory-based clusterings. Specifically, TraClass finds homogeneous regions where trajectories of a certain class are predominant, then uses a class-conscious adaptation of TRACLUS to group trajectory partitions into clusters that have high discriminative power. After generating features in this manner, TraClass maps trajectories into a feature vector, where each entry corresponds to a region-based or a trajectory-based cluster.

The remainder of this paper is organized as follows. Section 2 presents the problem definition and formulation details of our proposed method. Then, the detailed description of the experiments is given in Section 3, followed by a parameters study in Section 4. Finally, we conclude the paper in Section 5.

2 Methodology

In this section, we will introduce our algorithm for mining frequent trajectory behaviors. Each block in our pipeline model will be explained in detail while associated parameters will be discussed separately in the parameter study section.

2.1 Computing Pseudo-Optical Flows

[Section is omitted from the project report pending publication.]

2.2 Extracting Motion Components

[Section is omitted from the project report pending publication.]

2.3 Reachability

[Section is omitted from the project report pending publication.]

2.4 Forming Motion Patterns

[Section is omitted from the project report pending publication.]

3 Experimental Results

In this section, we first give a brief description of the datasets that were used for evaluation. Then, we show experimental results of our proposed trajectory clustering method on different datasets and finally compare them to the baseline methods.

3.1 Datasets

We used three different datasets to test the proposed method. These datasets are among the ones that are used by many papers in the literature and vary greatly in their properties, such as the number of trajectories, average number of points per trajectory, sampling density, spatial separation and complexity. Experimental results indicate that our proposed method is an effective solution for trajectory clustering task regardless of the dataset properties.

- **Vehicle Motion Trajectory Dataset:** This dataset contains 1500 trajectories gathered by tracking vehicles at a traffic intersection. These trajectories are annotated manually; each trajectory is assigned to one of 15 trajectory classes. The mean

number of points per trajectory is 96. This dataset is available at [21].

- **Atlantic Hurricane Dataset (HURDAT2):** This dataset is provided by the National Hurricane Service (NHS) and contains 1740 trajectories of Atlantic Hurricanes from 1851 through 2012, with trajectories containing 27 points on average. NHS also provides annotations of typical hurricane tracks for each month throughout the annual hurricane season that spans from June to November. In order to evaluate how close the motion patterns mined by our method are to the NHS annotations, we divided the Atlantic Hurricane Dataset into six subsets, one for each month. Trajectories that span more than one month were split to ensure each month includes only activity occurring within its span. This dataset is available at [22].
- **The Greek Trucks Dataset:** This dataset contains 1100 trajectories from 50 different trucks delivering concrete around Athens, Greece. As expected, the trucks follow highways giving the trajectories a distinctive appearance. The average number of points per trajectory is 86. This dataset available at [25].

3.2 Pre-Processing

For all experiments, data was first normalized so that all the trajectories were in the bounding box determined by $x \in [0, 1000]$ and $y \in [0, 1000]$. The Atlantic Hurricane Dataset went through further pre-processing: after splitting the dataset by months, each of the resulting subsets was pruned by removing trajectories consisting solely of a single coordinate pair repeated one or more times.

3.3 Intrepreting Output

Before we discuss experimental results, we first briefly explain how to read and interpret the output of the proposed algorithm. Each discovered motion pattern is displayed on top of plotted trajectories. To avoid overcrowding the figures, only a random subset of trajectories are visualized. The color of the motion pattern denotes the direction of the motion. A color wheel that appears in each figure serves as the legend for translating color into direction of motion.

Lastly, we must introduce the reasoning behind our handling of merging and diverging trajectories. An example of such a scenario is illustrated in Figure 1(a), where two clusters of trajectories partially overlap. The original motion pattern algorithm [1] was designed to see both of these clusters as one pattern. The altered version of the motion pattern algorithm we are proposing recovers three motion patterns in this case: two separate motion patterns for the distinct portions of the two clusters and a separate motion pattern for their merged portion. Therefore, it must be noted that a single trajectory may pass through several motion patterns. In other words, a motion pattern itself does not represent a homogeneous group of trajectories. Using our proposed method, trajectories shown in Figure 1(a) produce motion patterns shown in Figures 1(b), 1(c) and 1(d).

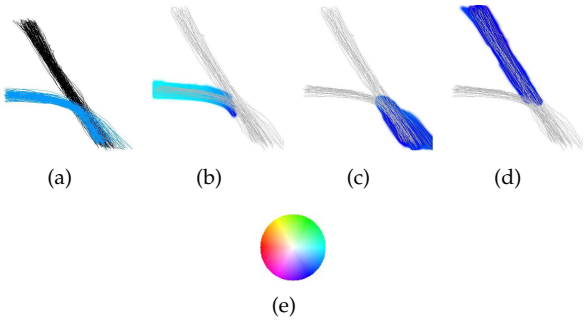


Figure 1. An example for merging in 1(a) where the proposed method has found two separate motion patterns of 1(b) and 1(d) in addition to the common part 1(c). In 1(a), colors of black and blue are used only to distinguish two classes of trajectories according to the annotations. Colorwheel is shown in 1(e).

3.4 Evaluations on Vehicle Motion Trajectory Dataset

Examining the outputs of the proposed method versus annotations shown in Figure 2(a), we see that the proposed algorithm recovers the motion patterns present in the dataset, with a few differences from the annotated version. First of all, the annotations categorize the traffic in each of the parallel lanes in Figure 2(b) into separate clusters. Because the motion in the left group of three lanes is very similar, the proposed algorithm recovers it as a single motion pattern. The same applies to the three-lane group on the right. Second, for reasons stated above, the turning trajectories shown in Figure 2(c) are segmented to differentiate their distinct portions.

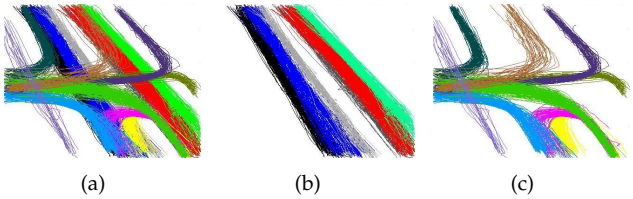


Figure 2. 15 different annotated classes of trajectories in Vehicle Motion Trajectory Dataset are shown with different colors.

Figures 3(a) and 3(b) correspond to the two directions of traffic on the main highways. Figure 3(c) corresponds to one of the left turns and is especially large since it contains the bulk of the left-side outgoing traffic. Figures 3(d), 3(e), and 3(f) correspond

to the incoming traffic from the left side. This is an example of the diverging behavior where the motion pattern from Figure 3(e) diverges into the motion pattern turning right in Figure 3(f) and the motion pattern going straight in Figure 3(d). Motion patterns in Figures 3(g) and 3(j) reflect a U-turn, where the former is a portion of the U-turn that merges with the traffic shown in Figure 3(b) and the latter is the distinct part of the U-turn. Figures 3(h) and 3(i) show two different access roads. Figure 3(k) shows a right turn and Figure 3(l) shows a rarely taken left turn.

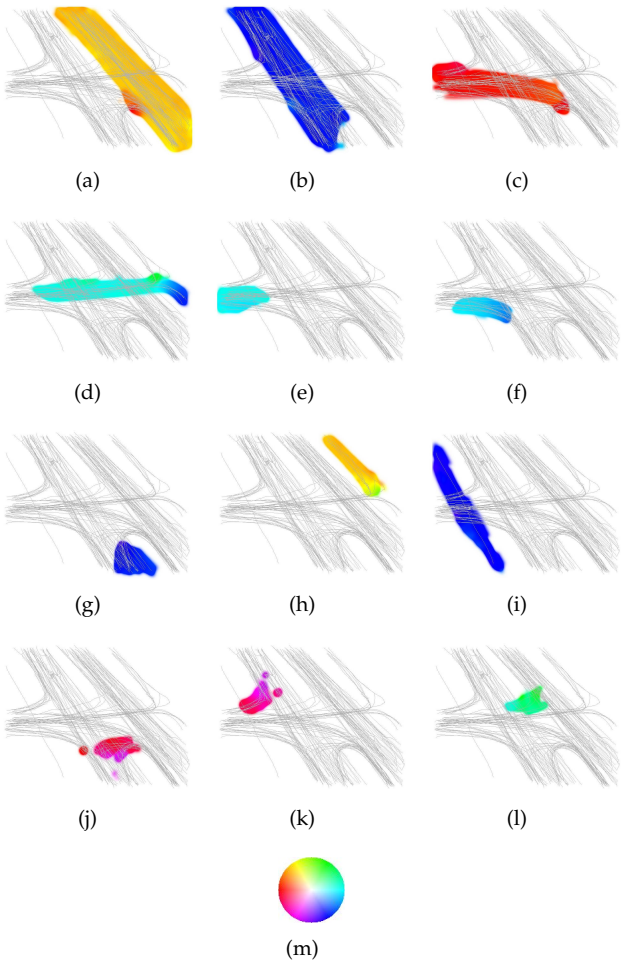


Figure 3. Extracted motion patterns from Vehicle Motion Trajectory Dataset. Colorwheel is shown in 3(m).

3.5 Evaluations on Atlantic Hurricane Dataset

We evaluated the performance of the proposed method on the Atlantic Hurricane Dataset by comparing its output with the annotations provided by the National Hurricane Service (NHS). Prevailing hurricane tracks for each month are indicated by white arrows in Figures 4(a), 4(d), 4(i), 4(m), 4(s) and 4(v). Examining the output for each month, we see that the proposed algorithm recovers motion patterns closely resembling the prevailing hurricane tracks for June in 4(b) and 4(c), for October in 4(t) and 4(u), and for November in 4(w). For July, 4(e) reflects the rightmost track, while 4(f) reflects the shared path of the leftmost tracks, and 4(g) and 4(h) serve as the distinct portions of the two leftmost paths. For August, 4(j) reflects the bottom left prevailing track, while 4(k) and 4(l) correspond to the splitting tracks on the top right. September hurricane tracks in 4(m) are split into two groups of three. The left group of three is output as motion patterns 4(o), 4(r), 4(p), while the right group of three is output as 4(n) for the arrow pointing northwest and 4(q) for the two arrows turning towards northeast. Because the density of trajectories, which corresponds to the number of hurricane tracks recorded within each given month, varies dramatically from one month to another, model parameters must be set separately for each month. As a general rule, months with fewer hurricanes, such as June, July and November, require more relaxed reachability conditions (larger double-ellipse and wedge radius), while months with high number of hurricanes, such as August, September and October, require stricter ones.

3.6 Evaluation on Greek Trucks Dataset

The proposed algorithm finds the main truck routes in the dataset. As previously noted, in the case of two routes merging, the algorithm will find three patterns: two for the routes before they merge, and one for the combined routes after they merge. The patterns in Figures 5(b) and 5(c) merge into the pattern in Figure 5(a). Similarly, the pattern in Figure 5(d) diverges into the patterns shown in Figures 5(g) and 5(h). Finally, the patterns patterns shown in Figures 5(f), 5(i), and 5(j) merged to create pattern shown in Figure 5(e).

3.7 Evaluating Baselines

Various quantitative measures for evaluating the quality of clusters exist in the literature. [26] , [27], [4] and [28] computed Correct Clustering Rate (CCR) as a measure to evaluate clustering results against the ground truth or manual annotations. [2] used an information theoretic criterion proposed by Meilă [29], Variation of Information (VI), to validate obtained clusters. VI determines the amount of information lost and gained between two different clusterings of data. Masciari compared inter-class and intra-class similarity of clusters [30], [31]. All of these and similar measures, however, were used to evaluate clusters of entire trajectories. Since this does not align with the goals of our work, an attempt to use the aforementioned metrics would not yield a meaningful comparison.

Rather than clustering entire trajectories, the proposed algorithm mines regional trends in trajectory data like [32], [33], [19], [16], [17] and [18]. [32] evaluates obtained results by visual comparison

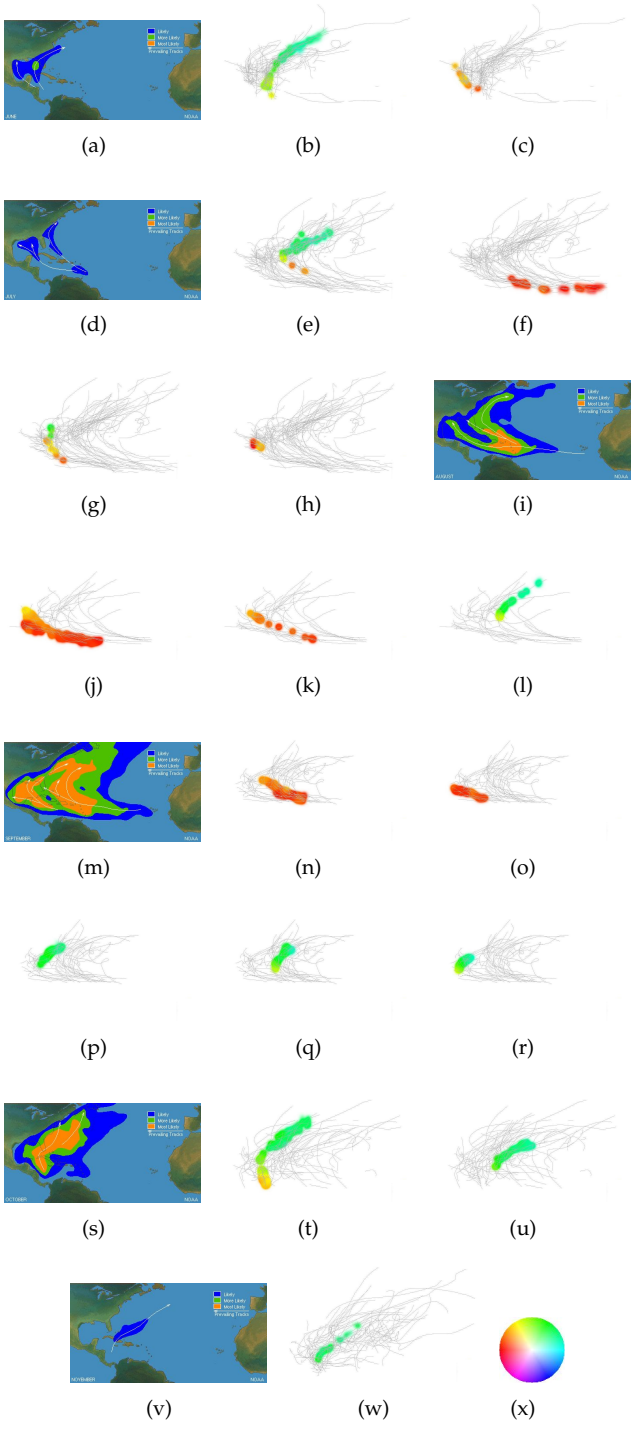


Figure 4. Extracted motion patterns from Atlantic Hurricane Dataset. Colorwheel is shown in 4(x).

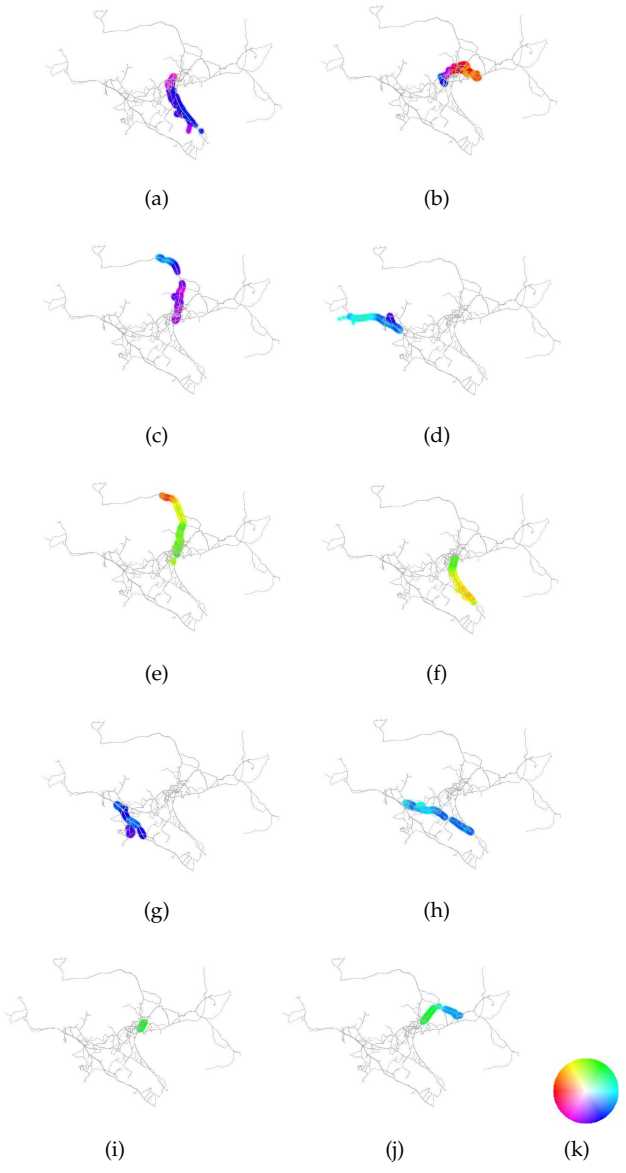
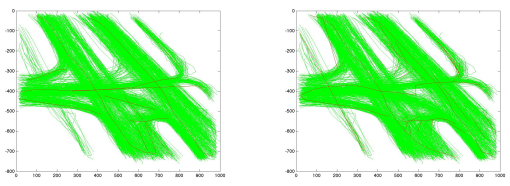


Figure 5. Extracted motion patterns from The Greek Trucks Dataset. Colorwheel is shown in 5(k).

with GSP [34] and PrefixSpan [35]. [33] offers a visual comparison of results with TRACLUS [19]. TRACLUS authors, in turn, state that there is no well-defined measure for density-based clustering methods, and suggest using a metric consisting of sum of squared error and noise penalty to, in authors' own words, "get a hint of the clustering quality". This metric cannot be easily adapted to methods other than variations of TRACLUS. Ferreira *et.al.* [15], who use the subset of the Atlantic Hurricane Dataset for their evaluations, report that their results are visually consistent with expected hurricane behavior, as does [36]. [18] provides a visual analysis and suggests a semantic interpretation of their results on the Greek Trucks Dataset. Staying close to the spirit of the works that mine regional trends, we will demonstrate the effectiveness of the proposed method by visually comparing our results with those of TRACLUS [19], DivCluST [16], [18] and [17]. Due to the space limitation in the paper, we were not able to provide all the visualizations for each of the baseline methods. Instead, we illustrated cases where they could not handle the difficulties provided by the datasets. Compared to the baseline methods, our

algorithm can successfully face those challenges.

The goal of TRACLUS [19] is to detect similar portions of trajectories, which semantically correspond to the output of our proposed method. More formally, given a set of trajectories, TRACLUS partitions trajectories, clusters them, and outputs a representative trajectory for each cluster. We used TRACLUS implementation available at [37]. It contains a program that estimates optimum parameter values. According to [19], these values are approximate and may differ slightly from the true optimal values. Therefore, we ran TRACLUS on each dataset multiple times, tuning each of the parameters. This is not unlike the approach to parameter selection that TRACLUS authors take. Lee $et.al.$ [19] suggest a heuristic approach to parameter selection, where optimal ε is selected to minimize the entropy of obtained clusters. Further, the quality of clusters is estimated using the sum of squared error and the noise penalty. Finally, [19] points out that the final selection of optimal parameters for TRACLUS is made using visual inspection and domain knowledge. One of the challenges that TRACLUS faces, as [33] also mentions, is that it is not capable of simultaneously extracting both dense and sparse trajectory clusters because its parameters are optimized globally. Modifying its parameters such that it find sparser clusters leads to redundant clusters in denser regions. Figure 6(a) shows clusters found by TRACLUS in Vehicle Motion Trajectory, when optimized ε and $MinLns$ are used. We tuned these parameters to allow TRACLUS extract clusters in both sparse and dense regions but it results in many short, local clusters and redundant ones shown in Figure 6(b). Our proposed method can handle variation in density as is illustrated in Figure 3.



(a) (b)

Figure 6. TRACLUS parameters ($\varepsilon, MinLns$) used in 6(a) and 6(b) are respectively as follows: (20,12), (20,3).

DivCluST [16] is an algorithm that seeks to find regional typical moving styles in the form of mean lines. Performance of DivCluST on the Greek Trucks Dataset is shown in Figure 7(a) where parameters are optimized according to the method described in the paper. Each arrow represents a mean line where the thickness refers to the frequency of that style and the color corresponds to the speed of that style. Warmer, reddish colors are for faster styles while cooler, bluish colors are for slower styles. This algorithm has trouble when there is large variation in the trajectory density. The Greek Trucks Dataset has this property which causes problems. In the high density regions, the mean lines are very cluttered and overlapping. Due to the kmeans-type model, there are very similar mean lines. This is because if the random initial clusters are close together in a high density region, a kmeans-type algorithm will often keep them close together. In the low density regions, there are mean lines that are not representative such as the mean lines within the yellow box in Figure 7(b). This is because in low density regions, quite different representative segments are clustered together, producing mean lines that are dissimilar from all the representative segments. Additionally, because the model is restricted to straight mean lines rather than curves, motion that would be better described as a curve is instead required to be described as one long mean line or a sequence of short mean lines. An example of this is the long cyan arrows within the purple box in Figure 7(b). Our algorithm deals much better with variation in density and has the ability to find curved motion patterns.

Giannotti $et.al.$ [18] proposed an algorithm that seeks to find aggregate motion behaviors from trajectories. These behaviors are defined as a sequence of rectangular regions. In Figure 8, some of the aggregated motion behaviors are illustrated where each is represented as a sequence of rectangular black regions. The order of the sequence is shown by the black arrows. First, it is important to note that this algorithm gives very redundant results. Many of the motion behaviors are very similar and some are even subsets of each other. This requires digging through many patterns to find the distinguishable ones. Although there are 80 generated patterns in total for the Vehicle Motion Trajectory Dataset, only 6 representative ones are shown in Figure 8. In other words, of the patterns not shown, each of them has the same shape or is a subset of the shown patterns. Requiring regions to be rectangular restricts the shape of extracted patterns. For instance, in Figure 8(a), the upper right part of the larger box covers a lane of traffic that should not be included in the pattern. Since the rectangular regions are built only based on density of trajectories without considering motion properties, they are not always suitable to represent motion behaviors. For instance, in Figures 8(b), 8(c), and 8(d), the rightmost box includes a different lane of traffic in the upper portion of the box. Finally, multiple traffic behaviors are sometimes combined into a single pattern. For instance, Figure 8(a) shows the traffic turning right and going straight combined, Figure 8(c) shows the traffic going straight and making U-turns combined, Figure 8(e) shows the access road and incoming traffic from

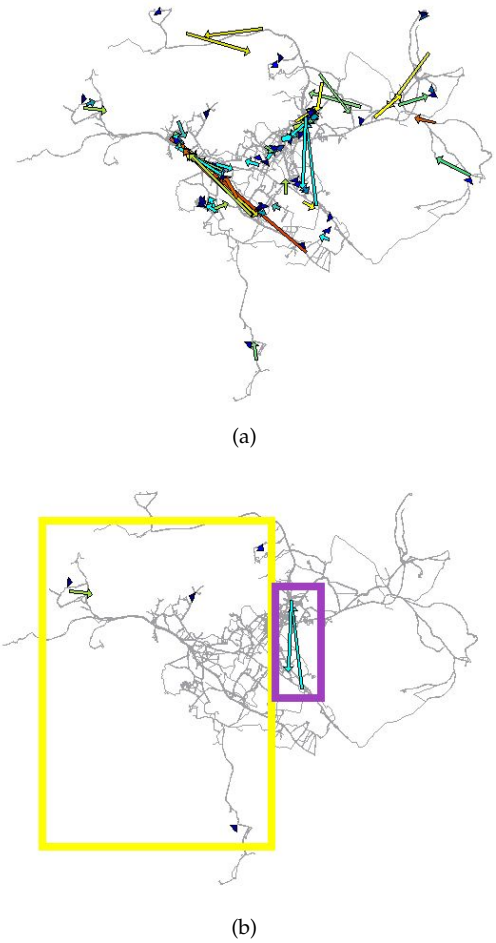


Figure 7. DivCluST on the Greek Trucks Dataset with parameters th_{len}=0.17, th_{spd}=0.50, and k=95.

the left combined and Figure 8(f) shows the U-turn and straight traffic combined. Unlike this method, the approach which we have proposed in this paper does not generate redundant or overlapping motion patterns. Our GMM based motion components are more flexible than rectangular regions used in [18] and our method does not mix multiple patterns of traffic in a motion pattern as can be seen in Figure 3.

Ulm $et.al.$ [17] proposed an algorithm that seeks to find clusters in the form of vector fields defined on a connected spatial set. While this algorithm performs well on datasets that are well structured such as Vehicle Motion Trajectory Dataset, it is not well suited for those that lack this property. This is partially due to the fact that it behaves much like the algorithms that cluster trajectories as a whole. It can be seen in Figure 9(d) that the vector fields can include directions going two opposite ways on a road. This can be avoided by changing the weights, but this results in vector fields that point perpendicular to the roads. This issue is not present in our algorithm since it can find two spatially overlapping motion patterns

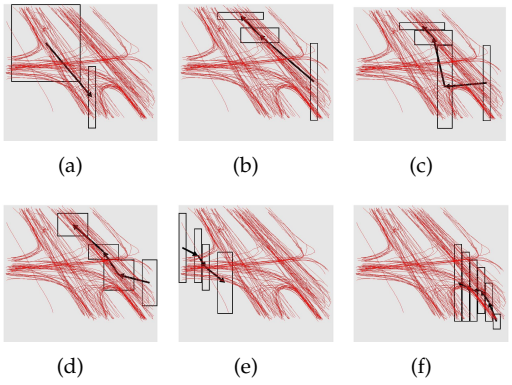


Figure 8. Parameters that are used to implement [18] on Vehicle Motion Trajectory Dataset are as: S_{min}=0.17, δ=0.17, τ=100 and ε=25.

with different directions. Secondly, the clusters are not homogeneous in movement. Some of the clusters are very sprawling while others have many small, noisy portions. This is not the effect of using too few clusters since there is a large amount of overlap among them. Instead, this is the effect of using whole trajectories to create vector fields.

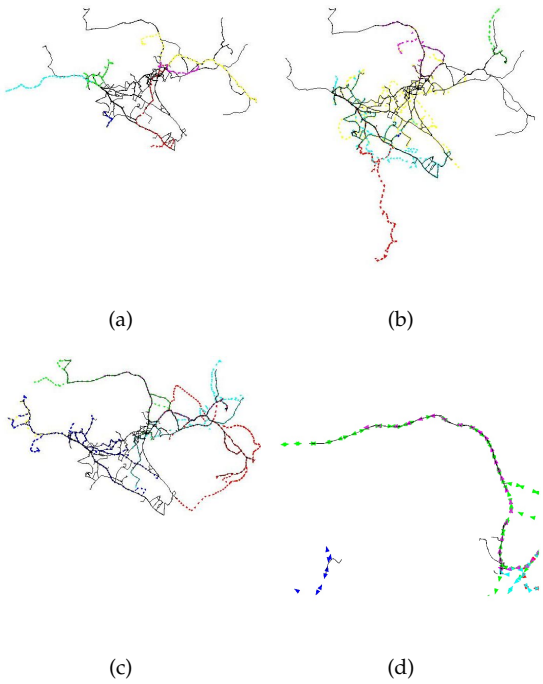


Figure 9. [17] is evaluated on the Greek Trucks Dataset with the cluster threshold of 110, the prototype threshold of 100, the position weight of 1 and the direction weight of 100. All other parameters were set to the default values.

4 Parameters Study

Section is omitted from the report pending publication.

5 Conclusion

In this paper, we looked at the general task of trajectory clustering from a novel approach inspired by the motion pattern idea [1] in computer vision. Compared to the original motion pattern work [1], we used a more effective set of conditions to find reachable motion component pairs. This allows us to extract motion patterns containing sharp turns and merge parallel, similar patterns into one. Further, we can discern merging and diverging motion patterns that [1] would see as one because of shared motion components.

We evaluated our proposed method on four different datasets. Experimental results indicate that our motion pattern approach is an effective solution for the general task of trajectory clustering regardless of the dataset properties. Extracted motion patterns closely fit the annotations, prevailing paths, or descriptions of trajectory datasets where available.

Overall, we believe that we have provided a new approach for understanding trajectory behavior. Its output is comparable to the output of those trajectory clustering methods that look for regional similarities among trajectories. It is flexible to a variety of challenges provided by different datasets. Our proposed method can provide data analysts a good starting point for understanding the hidden behavioral patterns in huge and complex trajectory datasets. We comprehensively discussed the effect of model parameters and provided selection process for them. Finally, we note that the actual optimum set of parameters will rely on domain knowledge as well as specific analytical goals.

References

[1] I. Saleemi, L. Hartung, and M. Shah, "Scene understanding by statistical modeling of motion patterns," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2010, pp. 2069–2076. 3, 8

[2] S. Atev, G. Miller, and N. P. Papanikolopoulos, "Clustering of vehicle trajectories," *Trans. Intell. Transport. Sys.*, vol. 11, no. 3, pp. 647–657, Sep. 2010. 1, 4

[3] Z. Fu, W. Hu, and T. Tan, "Similarity based vehicle trajectory clustering and anomaly detection," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 2. IEEE, 2005, pp. II–602. 1

[4] B. Morris and M. Trivedi, "Learning trajectory patterns by clustering: Experimental studies and comparative evaluation," in *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2009. 1, 4

[5] W. Hu, D. Xie, Z. Fu, W. Zeng, and S. Maybank, "Semantic-based surveillance video retrieval," *Image Processing, IEEE Transactions on*, vol. 16, no. 4, pp. 1168–1181, 2007. 1

[6] F. Bashir, A. Khokhar, and D. Schonfeld, "Object trajectory-based activity classification and recognition using hidden markov models," *Image Processing, IEEE Transactions on*, vol. 16, no. 7, pp. 1912–1919, 2007. 1

[7] E. J. Keogh and M. J. Pazzani, "Scaling up dynamic time warping for datamining applications," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2000, pp. 285–289. 1

[8] D. Buzan, S. Sclaroff, and G. Kollios, "Extraction and clustering of motion trajectories in video," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2. IEEE, 2004, pp. 521–524. 1

[9] C. Piciarelli and G. L. Foresti, "On-line trajectory clustering for anomalous events detection," *Pattern Recognition Letters*, vol. 27, no. 15, pp. 1835–1842, 2006. 1

[10] S. Atev, O. Masoud, and N. Papanikolopoulos, "Learning traffic patterns at intersections by spectral clustering of motion trajectories," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, 2006, pp. 4851–4856. 1

[11] B. Morris and M. Trivedi, "An adaptive scene description for activity analysis in surveillance video," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008, pp. 1–4. 1

[12] D. Biliotti, G. Antonini, and J. P. Thiran, "Multi-layer hierarchical clustering of pedestrian trajectories for automatic counting of people in video sequences," in *Application of Computer Vision, 2005. WACV/MOTIONS'05 Volume 1. Seventh IEEE Workshops on*, vol. 2. IEEE, 2005, pp. 50–57. 1

[13] G. Karypis, E.-H. Han, and V. Kumar, "Chameleon: Hierarchical clustering using dynamic modeling," *Computer*, vol. 32, no. 8, pp. 68–75, 1999. 1

[14] X. Li, W. Hu, and W. Hu, "A coarse-to-fine strategy for vehicle motion trajectory clustering," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 1. IEEE, 2006, pp. 591–594. 1

[15] N. Ferreira, J. T. Klosowski, C. Scheidegger, and C. Silva, "Vector field k-means: Clustering trajectories by fitting multiple vector fields," in *EuroVis*, 2013. 1, 5

[16] H. ru Wu, M.-Y. Yeh, and M.-S. Chen, "Profiling moving objects by dividing and clustering trajectories spatiotemporally," *IEEE Transactions on Knowledge and Data Engineering*, vol. 99, no. PrePrints, 2012. 1, 4, 5, 6

[17] M. Ulm and N. Brandie, "Robust online trajectory clustering without computing trajectory distances," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 2270–2273. 2, 4, 5, 7

[18] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 330–339. 2, 4, 5, 6, 7

[19] J. gil Lee and J. Han, "Trajectory clustering: A partition-and-group framework," in *In SIGMOD*, 2007, pp. 593–604. 2, 4, 5, 6

[20] J.-G. Lee, J. Han, X. Li, and H. Gonzalez, "Traclass: trajectory classification using hierarchical region-based and trajectory-based clustering," pp. 1081–1094, 2008. 2

[21] "Vehicle Motion Trajectory Dataset," http://159.226.19.33/dataset/trajectory/dataInfo.htm. 3

[22] "Atlantic Hurricane Dataset (HURDAT2)," http://www.nhc.noaa.gov/data/hurdat. 3

[23] M. R. Fuller, W. S. Seegar, and L. S. Schueck, "Routes and travel rates of migrating peregrine falcons falco peregrinus and swainson's hawks buteo swainsoni in the western hemisphere," *Journal of Avian Biology*, pp. 433–440, 1998.

[24] "Movebank Data Repository," http://www.Movebank.org.

[25] "The Greek Trucks Dataset," http://www.chorochronos.org. 3

[26] Y. Yang, Z. CUI, J. WU, G. ZHANG, and X. XIAN, "Trajectory analysis using spectral clustering and sequence pattern mining," *Journal of Computational Information Systems*, vol. 8, no. 6, pp. 2637–2645, 2012. 4

[27] Z. Zhang, K. Huang, and T. Tan, "Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 3. IEEE, pp. 1135–1138. 4

[28] E. Ricci, F. Tobia, and G. Zen, "Learning pedestrian trajectories with kernels," in *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE, 2010, pp. 149–152. 4

[29] M. Meilă, "Comparing clusteringsan information based distance," *Journal of Multivariate Analysis*, vol. 98, no. 5, pp. 873–895, 2007. 4

[30] E. Masciari, "A complete framework for clustering trajectories," in *Tools with Artificial Intelligence, 2009. ICTAI'09. 21st International Conference on*. IEEE, 2009, pp. 9–16. 4

[31] ——, "Finding homogeneous groups in trajectory streams," in *Proceedings of the Third ACM SIGSPATIAL International Workshop on GeoStreaming*. ACM, 2012, pp. 11–18. 4
[32] J. Kang and H.-S. Yong, "Mining spatio-temporal patterns in trajectory data." *JIPS*, vol. 6, no. 4, pp. 521–536, 2010. 4
[33] Y. Zhang and D. Pi, "A trajectory clustering algorithm based on symmetric neighborhood," in *Computer Science and Information Engineering, 2009 WRI World Congress on*, vol. 3. IEEE, 2009, pp. 640–645. 4, 5, 6
[34] R. Srikant and R. Agrawal, *Mining sequential patterns: Generalizations and performance improvements*. Springer, 1996. 5
[35] J. Han, J. Pei, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, "Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth," in *proceedings of the 17th international conference on data engineering*, 2001, pp. 215–224. 5
[36] B. Guan, L. Liu, and J. Chen, "Using relative distance and hausdorff distance to mine trajectory clusters," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 11, no. 1, pp. 115–122, 2013. 5
[37] "Traclus Implementation," http://dm.kaist.ac.kr/jaegil/Publications. 6