Clustering of Inertial Indoor Positioning Data

Lorenz Schauer and Martin Werner
Mobile and Distributed Systems Group
Ludwig-Maximilians Universität, Munich, Germany
lorenz.schauer@ifi.lmu.de, martin.werner@ifi.lmu.de

***Abstract*—TRACLUS is a widely-used partitioning and grouping framework for trajectories. However, suitable clustering results representing the building's topologies are hardly obtained when applying the framework to indoor trajectories with noise. In this work, this problem is demonstrated on an inertial indoor positioning data set created using a filtered dead-reckoning system based on step-counting. Using Douglas-Peucker algorithm as a different segmentation strategy to TRACLUS and a minor correction to the distance weightings of TRACLUS, we show that this framework is still applicable for this data set.**

***Keywords*-Indoor Trajectories; Segmentation; Density Clustering; TRACLUS**

I. Introduction

In the last decade, modern mobile devices with integrated sensors have generated new possibilities and challenges for indoor localization and tracking. More and more location data from mobile users is becoming available and has to be processed with the methods of data science, such as trajectory clustering algorithms, in order to understand and use this valuable personal information. This topic is of high interest for many fields in the scientific and commercial world. We see an important application domain in filtering of noise and uncertainty in indoor positioning systems without map information: the clustering structure of trajectories creates a valuable "map", which can be used to assess the probability location in particle filtering or multiple-hypothesis tracking navigation approaches.

In this work, we focus on a well-known and widely-used trajectory clustering framework and algorithm named TRACLUS [1] and show its deficits for a real-world inertial indoor positioning data set. We demonstrate, how these deficits can be solved with minimal changes of the proposed framework. Using Douglas-Peucker algorithm [2] as another segmentation strategy, and a slightly up-weighted angle distance, we receive more adequate clustering results representing the topology of the corresponding building.

The remainder of the paper is organized as follows: The next Section II shortly introduces the TRACLUS frameworks and the relevant definitions. In Section III, we first describe the used dataset, its internal structure, and the results of the original TRACLUS algorithm. Then, we show that a different segmentation leads to the expected results. Finally, Section IV concludes the paper.

II. The TRACLUS algorithm

The TRACLUS algorithm [1] is a widely-used partitioning and clustering framework performing density-based clustering on line segments, rather than grouping trajectories as a whole. This is due to the fact that capturing the distance between non-local objects is infeasible and, therefore, some locality is reconstructed by a splitting methodology. In order to get line segments, the trajectory is partitioned on segmentation points representing significant changes of the trajectory's behavior. Hence, as a first step, segmentation points have to be found. For trajectory segmentation, several approaches have been presented in literature [2]–[4]. TRACLUS performs an approximate solution based on the minimum description length (MDL) principle in order to find the optimal tradeoff between preciseness and conciseness.

In a second step, the line segments from the partitioning phase are clustered with respect to their density-connectedness. Different density-based clustering algorithms can be found in literature such as DBSCAN [5] or DENCLUE [6]. TRACLUS is similar to DBSCAN and searches density-connected sets of line segments which are marked as clusters. As in case of DBSCAN, two parameters are required: ϵ defining the threshold for the ϵ-neighborhood of line segments, and *minLns*, defining the minimum amount of lines which have to be inside an ϵ-neighborhood in order to create a cluster.

In order to compute the ϵ-neighborhood of a line segment, a distance function is needed focusing on segment characteristics. TRACLUS proposes three distances capturing different aspects of similarity: parallel distance (PD), perpendicular distance (PPD), and angle distance (AD). These are joined into a single distance measure by a weighted sum, with weight $\alpha_i = 1$ by default:

$$d = \alpha_1 PD + \alpha_2 PPD + \alpha_3 AD$$

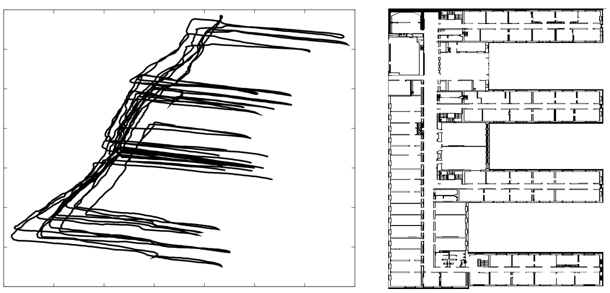
In a final step, all clusters are checked if their trajectory cardinality is greater that a predefined threshold which is *minLns* by default. Otherwise they are removed from the result set of clusters. This is done, because a cluster of segments coming from a small set of trajectories could lead to a cluster of trajectories containing less than *minLns* individual trajectories.

III. Clustering of Inertial Indoor Positioning Data

In order to extract reasonable map information from inertial sensor systems, this section proposes to use a blind clustering approach without any map information and demonstrate its feasibility on a real-life dataset.

A. The Dataset

Using a dead-reckoning system based on step counting and a digital compass, we created several traces inside our building which are depicted in Figure 1(a). For reference, Figure 1(b) depicts the schematic floor plan of the building. The dataset consists of a user walking into each of the hallways, turning there and walking back. It suffers from accumulating errors as there are no reference measurements integrated into the system, however, the used sensory and filtering led to quite accurate results as can be seen. The dataset consists of 10 trajectories for a total of 10,386 step estimates of varying length and direction.



(a) Representation of data set (b) Schematic floor plan

Figure 1. Data set created within our office building

Using this dataset, we would expect to find four clusters representing the horizontal hallways and either one or three clusters representing the vertical interconnections between them. We expected this as the trajectories in these clusters would have been similar to each other in direction and length as well as near to each other while they would not be near to trajectory parts in other hallways.

B. TRACLUS Results

For clustering, we used TRACLUS in two different fashions: First, we perform segmentation and distance weightings as proposed in the original publication.

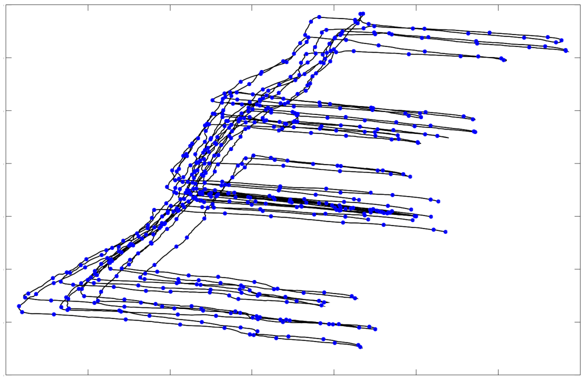
Performing this with a value of *minLns* $= 3$, we varied the parameter ϵ and were unable to produce the expected results: For large values of ϵ, of course, too few clusters were detected. For smaller values, however, some of the clusters broke up into two different ones and clusters got rejected based on that they did not meet the *minLns* threshold anymore rendering those candidates as noise. Figure 2 depicts a situation for $\epsilon = 1.8$. While some clusters start breaking wrongly into two different clusters (low part of Figure 2(b)), highly unrelated clusters are being joined such as the T-shaped cluster in the middle. This is due to problems with the segmentation and especially with short segments therein, see Figure 2(a). These short segments are taken as similar with each other by the original TRACLUS construction [1].

It is clear from this figure, that no other value of ϵ would have created the expected results: smaller values increase the splitting of relevant clusters, larger values will not lead to a breakup of the T-shaped cluster. Consequently, we have to change the algorithm in another way.

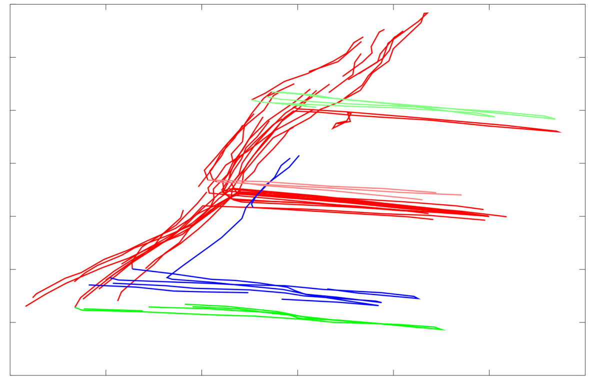
From Figure 2(a), we observe that the MDL-based segmentation of TRACLUS results in very small segments which actually do not capture any trend change in their definitions. In order to change that, we decided to use the Douglas-Peucker line simplification algorithm in order to capture a segmentation of the input trajectories which is less sensitive to small perturbations. As you can see in Figure 3(a), the Douglas-Peucker simplification rejects more points as compared to the original TRACLUS segmentation. Furthermore, it keeps those points where durable changes in orientation actually take place. As a first effect, this reduces the computational overhead. More importantly, however, the clusters capture more information: The line segments that we expect to fall into same clusters not only have similar orientation, but have similar length and some nearness of endpoints. In order to get a distance measure sensitive to the differences in angles between those segments and not overemphasizing the nearness of completely parallel segments of similar length, we had to change the weighting in the original TRACLUS distances towards angle distance. Note that executing the original TRACLUS with these modified weightings led to an even worse clustering result, as a lot of non-related segments have similar orientation. Using Douglas-Peucker segmentation, however, the result of clustering with modified weightings is depicted in Figure 3(b). This result finds all clusters as expected without the topmost cluster, which is anyways underrepresented in the dataset and correctly removed, due to its low trajectory cardinality.

In summary, we showed that the TRACLUS framework is still applicable to inertial indoor trajectories. However, the TRACLUS segmentation is too sensitive to measurement variations. We changed the segmentation as well as the distance weightings such that the intuitive notion of a cluster in the given dataset is actually reached.

Note that we did not yet show that this approach is generally advantageous for indoor trajectories. Still, we provide an example and an approach, which is tailored to the nature of inertial trajectories. Therefore, we expect that this approach will be applicable to other datasets in which measurement perturbations would lead to short segments as well as in which large segments can be extracted, which capture the actual structure of the dataset.

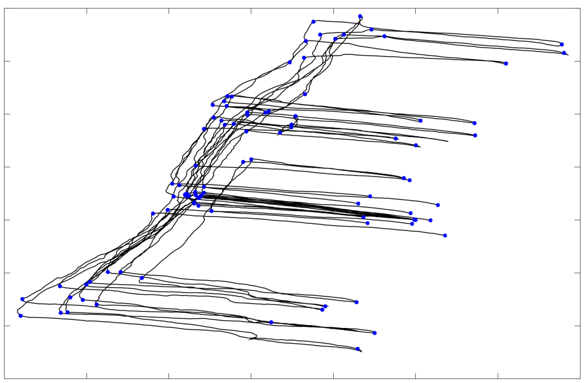


(a) TRACLUS segmentation result

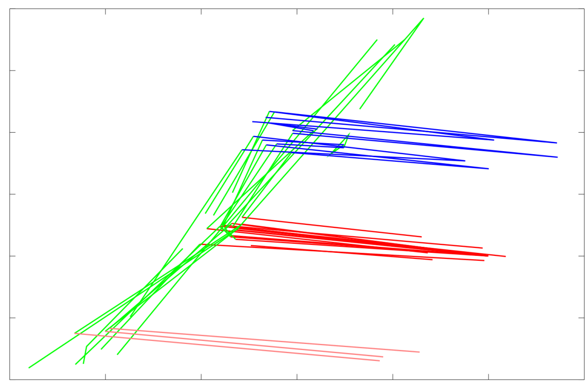


(b) TRACLUS clustering result with $\epsilon = 1.8$

Figure 2. Results of using TRACLUS in its original form on inertial tracking data.



(a) Douglas-Peucker segmentation



(b) TRACLUS using Douglas-Peucker ($\epsilon_{\text{DP}} = 2$), modified distance weight ($\alpha_3 = 2$), and $\epsilon = 6$

Figure 3. Results of using Douglas-Peucker for segmentation together with TRACLUS clustering on inertial tracking data.

IV. Conclusion and Future Work

In this work, we have addressed the problem of clustering inertial indoor positioning data. We have shown that the widely-used TRACLUS algorithm in its original form does not fit to this problem. Furthermore, we have demonstrated that the clustering results not only depend on both parameters ϵ and *minLns*, but also on the used segmentation strategy. Replacing the TRACLUS segmentation with Douglas-Peucker, the clustering results became more adequate for our purpose.

For future work, we want to enhance our investigations using different parameters and a large number of indoor data sets. We plan to use various distances for trajectories in order to compute the ϵ-neighborhood and evaluate the quality of returned clusters by using well-known quality measures for clustering as well as novel application-centered measures which capture the special situation inside buildings and other complex surroundings.

References

[1] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: a partition-and-group framework," in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. ACM, 2007, pp. 593–604.

[2] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[3] M. Potamias, K. Patroumpas, and T. Sellis, "Sampling trajectory streams with spatiotemporal criteria," in *Scientific and Statistical Database Management, 2006. 18th International Conference on*. IEEE, 2006, pp. 275–284.

[4] P. Katsikouli, R. Sarkar, and J. Gao, "Persistence based online signal and trajectory simplification for mobile devices," 2014.

[5] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, 1996, pp. 226–231.

[6] A. Hinneburg and H.-H. Gabriel, "Denclue 2.0: Fast clustering based on kernel density estimation," in *Advances in Intelligent Data Analysis VII*. Springer, 2007, pp. 70–80.