J Ambient Intell Human Comput (2018) 9:173–186
https://doi.org/10.1007/s12652-016-0426-8

ORIGINAL RESEARCH

Enhancing the quality of geometries of interest (GOIs) extracted from GPS trajectory data using spatio-temporal data aggregation and outlier detection

Seyed Morteza Mousavi[1] · Aaron Harwood[2] · Shanika Karunasekera[2] · Mojtaba Maghrebi[3]

Received: 10 September 2016 / Accepted: 10 October 2016 / Published online: 7 November 2016

Abstract One of the initial phases in the applications dealing with data processing on GPS trajectory data is to generate the time-stamped Sequence of Visited Locations (SVLs) of the mobile objects. The sequence is constructed by labeling each of the GPS observations of the trajectory using the ID of their intersecting Geometries of Interest (GOIs). In this paper, we enhance the performance of the state-of-the-art scheme for constructing the GOIs of a mobile object by proposing a data aggregation and outlier detection method. Our experimental results using geometric similarity metrics show that our improved GOI construction method outperforms the baseline methods by constructing the GOIs remarkably more geometrically similar to the real world GOIs. The geometric similarity metrics are only applicable when we have access to the geometries of the real world GOIs (ground truth). To be able to analyse the performance of the GOI extraction methods in environments which we do not have access to the ground truth, we propose two useful spatio-temporal metrics to measure the quality of GOIs based on the quality of the generated SVLs based on them. Our experimental results show that these two metrics are able to discriminate between the results of our different outlier detection methods and select the best scheme without using any external knowledge about the geometries of the real world GOIs.

Keywords Trajectory data · Geometry of interest (GOI) · Data aggregation · Outlier detection · GOI quality measurement metrics · POI

1 Introduction

In recent years, due to the development of GPS-enabled devices such as vehicles carrying navigation systems and mobile phones with GPS sensors, a very large amount of data is being collected on a daily basis. The collected data can be effectively used by data mining and knowledge discovery methods in various applications such as traffic and transportation management systems (Min and Wynter 2011), animal migration and movement monitoring (Handcock et al. 2009), location prediction (Gidófalvi and Dong 2012), transportation mode estimation (Zheng et al. 2010), tourist POI recomendation (Fenza et al. 2011), and social networks (De Maio et al. 2016).

One of the most useful information which can be derived from the GPS trajectories is the time-stamped Sequence of Visited Locations (SVLs). The time-stamped sequences represent the Points of Interest (POIs) which are visited during the trajectory period along with the arrival and departure times of the visits to each POI. The derived SVLs are widely used in location based applications dealing with trajectory data and the performance of the

✉ Seyed Morteza Mousavi
seyed.morteza.mousavi@gmail.com

Aaron Harwood
aharwood@unimelb.edu.au

Shanika Karunasekera
karus@unimelb.edu.au

Mojtaba Maghrebi
mojtabamaghrebi@um.ac.ir

[1] Data61, Department of Computing and Information Systems, The University of Melbourne, Melbourne, Australia

[2] Department of Computing and Information Systems, The University of Melbourne, Melbourne, Australia

[3] Department of Civil Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

applications are significantly dependent on the quality of the SVLs. For example, in location prediction applications the SVL is used for forecasting the next location of a mobile object and its arrival time to the predicted location (Scellato et al. 2011a; Gidófalvi and Dong 2012). Obviously, the performance of the location prediction applications rely on the quality and the accuracy of the constructed SVLs and correspondingly, the quality of the SVLs are highly dependent on the quality of the geometries of the POIs extracted from the GPS trajectories.

SVL extraction methods often use the Nearest Neighbour Queries (NNQ) to label each GPS trajectory by the ID of the visited POI at each time within the trajectory period. This approach has considerable drawbacks on the quality of the SVLs. One solution to the problem is to use the intersection geometric operator instead of NNQ in the SVL construction process. The operator needs to have access to the accurate geometry of the POIs instead of only considering the coordinate of their centroids.

The problem of extracting the geometries and the centroids of significant places which mobile object frequently visit (POIs) has been highly considered and addressed in research works dealing with GPS trajectory data (e.g. Xiao et al. 2014; Ye et al. 2009; Hariharan and Toyama 2004). Mousavi et al. (2016) have proposed a method to partition the trajectory area of a mobile object into a grid containing the geometries of the POIs of the moving object. They refer to the geometry of the POIs as the Geometries of Interest (GOIs). Based on their reported results, the geometries of the extracted GOIs are significantly more similar to the real world GOIs compared to the base line methods (Ye et al. 2009; Hariharan and Toyama 2004). Despite the fact, the quality of the extracted GOIs is required to be enhanced to resemble the real world GOIs more accurately. The main goal of this paper is to provide a method to enhance the quality of the estimated GOIs to become more geometrically similar to the real world GOIs.

Figure 1 shows the GOI extracted using the proposed method in Mousavi et al. (2016). As it is clearly evident, the GOIs (polygons depicted with blue color) have acceptable geometric similarity with the real world GOIs (depicted with red polygons). Although the performance of the GOI extraction proposed in Mousavi et al. (2016) was significantly better than its baselines, there is evidently a significant need for improvement since the geometries of the extracted GOIs are not thoroughly covering the area of the real world GOIs (which are considered as the car parks on the map). In some cases, the estimated GOIs cover large areas outside the real world GOIs.

To tackle this problem we consider the GPS trajectories of the other mobile objects (which move in the same geographical area) in the GOI extraction process. In this paper, to enhance the results of the state-of-the-art method (Mousavi et al. 2016), we propose an algorithm to aggregate the destinations extracted for all the mobile objects in our GPS trajectory dataset. The data aggregation process attempts to reconstruct the extracted destinations which have been previously extracted for one particular mobile object. The process does not extract any new destination and only reconstructs the previously extracted destinations of the particular mobile object. The data aggregation process uses the geometric similarity of the destinations of each mobile object to aggregate their GPS points in the point sets of the particular mobile object.

Our experiments show that performing the data aggregation phase leads to the GOIs of the particular mobile object to cover almost all the area of the real world GOIs (car parks on the map). However, the extracted areas are considerably larger than the real world GOIs. The main problem which causes this phenomenon is the existence of the outlier GPS points in the point set of the aggregated destinations. To tackle this problem we propose a novel spatio-temporal outlier detection method suitable for our particular problem. Our experimental results show that our proposed outlier detection method outperforms the other examined baseline outlier detection methods in maximizing the geometric similarity of the estimated GOIs and the real world GOIs.

The main goal of the GOI extraction method, data aggregation, and outlier detection method is to minimize the geometric dissimilarity between the real world GOIs and the estimated GOIs. To measure the geometric dissimilarity, in this paper, we use two geometric metrics. The results of our experimental evaluations show that our proposed data aggregation and outlier detection method outperforms the baseline method by minimizing the geometric dissimilarity between real world GOIs and the estimated GOIs.

The two proposed geometric similarity based methods can compare the performance of the GOI extraction methods in the scenarios and application which sufficient data about the real world GOIs (ground truth) are accessible. However, in various applications such as those dealing with GPS trajectories collected from animals, battlefields, the accurate geometries of the significant places are not known a priori. In this case, the geometric similarity based metrics are not applicable in experimental evaluation. To evaluate the performance of the GOI extraction methods in such scenarios, we propose two spatio-temporal metrics which rely on the quality of the extracted SVLs. Our experimental results show that the two proposed metrics are able to distinguish the best GOI extraction methods without having access to ground truth data.

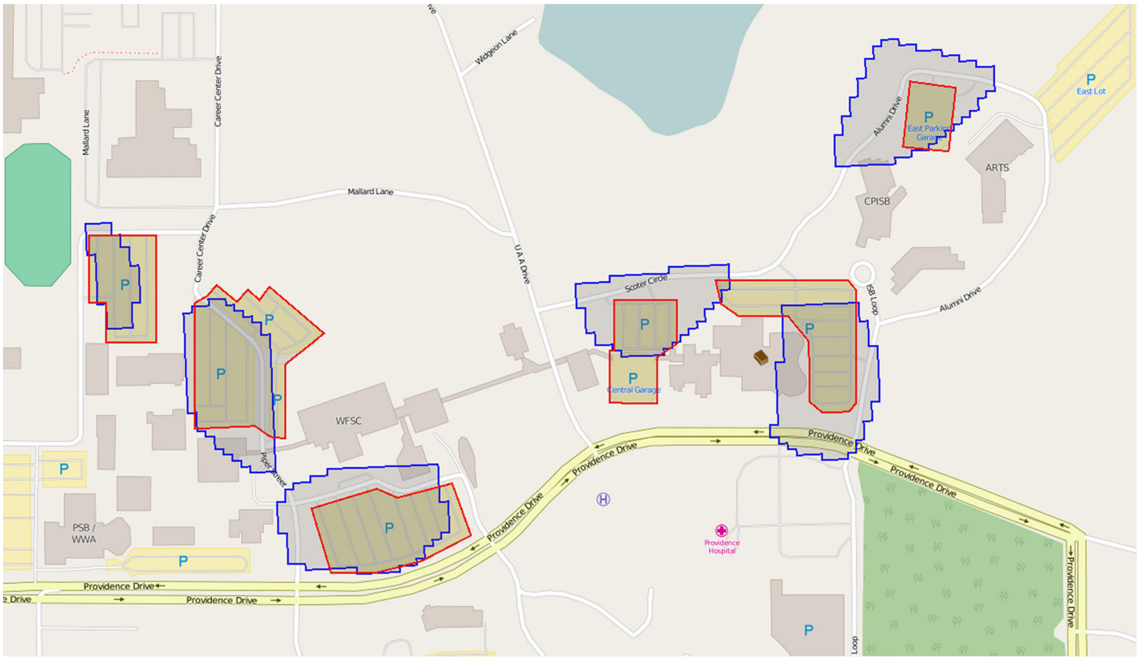

Fig. 1 Extracted GOIs of one mobile object using time-weighted geometric similarity based method (T_{min} = 60 min, D_{max} = 100 m, J_{min} = 0.10, VF_{min} = 7)

1.1 Contributions

The main contributions of this paper can be summarized as follows:

- Proposing a spatio-temporal data aggregation method to reconstruct the destinations of a particular mobile object using the GPS trajectory data from other mobile objects moving in the same geographical area.
- Proposing a novel outlier detection method and analyzing its performance compared to three famous outlier detection methods.
- Proposing two novel spatio-temporal distance metrics to analyze the quality of the GOI estimation methods without having access to the ground truth.

1.2 Paper organisation

The rest of the paper is organized as follows: In Sect. 2, we discuss the related research works. In Sect. 3, the problem is preliminarily defined. In Sect. 4, we discuss our data aggregation method. In Sect. 5, we briefly introduce the three customized outlier detection methods and discuss our proposed spatio-temporal outlier detection method. We compare the performance and the accuracy of the outlier detection methods based on the two proposed geometric metrics in Sect. 7.1. In Sect. 7.2, we discuss our two spatio-temporal SVL quality based metrics and compare the outlier detection methods based on them. Finally, in Sect. 8, the introduced method is summarized, and the achieved results and the future works are discussed.

2 Related works

In the related works aiming to partition the minimum bounding rectangle (MBR) of a mobile objects trajectories and extract the location of POIs and the geometries of the GOIs, five approaches have been taken. In this section, we briefly introduce these approaches.

The first approach is to extract the regions of interest by partitioning the MBR of the trajectory into a homogeneous grid with triangular, square, rectangular, or hexagonal polygons shape. As an example can refer to Xue et al. (2013). The major problem with this approach, which has significant drawbacks on the quality of the SVL extracted based on such partitions, is the granularity of the partitions. The coarse granularity leads to a number of POIs being covered by one partition, and the fine granularity leads to the area of one POI being divided into different cells. The labeling process in the SVL extraction method based on such grids is not straight forward.

The second approach is to consider the area which is covered by the wireless access points or covered by each cell in cellular network as the region of the POI (Song et al. 2006; Si et al. 2010). Similar to the first approach the granularity of the areas covered by the access points or the cells in cellular networks has a significant impact on the quality of the SVLs constructed based on the partitioned grid.

The third approach is to cluster the GPS points using measures such as the distance between GPS points or density connectivity in Cartesian space using clustering schemes such as KMeans (MacQueen 1967) and DBSCAN (Ester et al. 1996), without taking temporal aspects into consideration, and partition the trajectory area based on the destination geometries constructed on the grounds of that clusters (Ashbrook and Starner 2003; Zhou et al. 2004; Scellato et al. 2011b; Li et al. 2011). Considering the density and the neighbourhood system of the GPS points without considering the temporal aspects in the clusters leads to significant drawback on the accuracy of the extracted POI locations. For example, the density of the points at the conjunctions with traffic lights often have higher density while they are not often the locations of the POIs.

The fourth approach is to take the speed restrictions into consideration in finding the stop and moves (e.g. Palma et al. 2008; Bhattacharya et al. 2012). This approach assumes the clusters with the GPS track points with lower speed are more likely to be stop points. This approach also has drawbacks since the speed of the mobile objects is not always available or easily estimable. Moreover, the speed threshold in finding the stop and move points is highly dependent on the transportation mode of the mobile objects.

The fifth approach considers temporal aspects and restriction in extracting the stay regions and the destinations (Ye et al. 2009; Xiao et al. 2010). Estimation of the locations and the geometries of the significant regions are done in two phases, stay region extraction and destination extraction. They define a valid stay region (a vicinity distance) within which the mobile object has stopped or kept moving for a time span $\Delta T \geq T_{min}$, where T_{min}, is a time span threshold. The destinations which represent the POIs are extracted by merging the stay centroid points of the extracted stay regions using density-based clustering methods such as OPTICS (Ye et al. 2009). As a result, the places which the mobile objects stay for a considerable time are selected, and the other places are filtered although they might have high point densities. Similarly, the research work (Hariharan and Toyama 2004) extracts the stay regions by defining the time and vicinity distance based on the diameter of the extracted stay regions. The destinations are extracted by merging the stay regions based on the predefined maximum diameter of the destinations.

Recently Mousavi et al. (2016) introdcued a method that focuses on extracting the GOIs of a mobile object and partitioning the MBR of the trajectories based on the estimated GOIs. As opposed to the related works such as (Ye et al. 2009; Xiao et al. 2010) which focus on estimating the position of the centroid of the POIs, the main objective of their work was to extract the *geometries* of the significant places. They improve the performance and the accuracy of the stay region extraction phase of the fifth approach by incorporation the concept of time-value of the GPS points in the clustering method. Moreover, they proposed an agglomerative hierarchical clustering method to merge the stay regions and extract the geometries of the destinations on the grounds of geometric similarity metrics.

They have used the number of stays and destinations and also the geometric similarity of the estimated GOIs and the real world GOIs as the evaluation criteria. Their proposed method significantly outperforms the available benchmarks, however comparing the geometries of the estimated GOIs and the real world GOIs reveals the fact that the destination extraction methods needs to be enhanced further. To enhance the quality and geometric accuracy of the extracted GOIs, in this paper, we propose a data aggregation and outlier detection method. Moreover, instead of using a very simple geometric evaluation criterion which was used in Mousavi et al. (2016), we propose two geometric and two novel spatio-temporal evaluation criteria and analyze our methods based on them.

3 Problem definition

Before defining our research problem, we introduce two geometric dissimilarity metrics (distance metrics) which are used to evaluate the performance of GOI extraction methods as follows:

Definition 1 Given a set of real GOIs $R = \{r_1, r_2, \ldots, r_n\}$ and a set of estimated GOIs $G = \{g_1, g_2, \ldots, g_m\}$, we define the Ratio of Uncovered Real GOIs (RURG) as:

$$RURG(G, R) = \frac{\sum_i^n \sum_j^m Area(r_i) - Area(r_i \cap g_j)}{\sum_i^n Area(r_i)}. \tag{1}$$

Definition 2 Given a set of real GOIs $R = \{r_1, r_2, ..., r_n\}$ and a set of estimated GOIs $G = \{g_1, g_2, \ldots, g_m\}$, we define the Ratio of UnCovering Estimated GOIs (RUEG) as:

$$RUEG(G, R) = \frac{\sum_i^n \sum_j^m Area(g_j) - Area(r_i \cap g_j)}{\sum_i^n Area(r_i)}. \tag{2}$$

Given a GPS trajectory $\mathcal{T}$, a set of real GOIs $R = \{r_1, r_2, \ldots, r_n\}$, and a set of estimated GOIs

$G = \{g_1, g_2, \ldots, g_m\}$, our objective is to propose the best GOI estimation method, $f_o : \mathcal{T} \rightarrow G$ which minimizes the sum of the parameters *RURG*, and *RUEG*.

$$f_o = \underset{f_i \in F}{\operatorname{argmin}}[RURG(f_i(\mathcal{T}), R) + RUEG(f_i(\mathcal{T}), R)], \qquad (3)$$

where $F = \{f_1, f_2, \ldots, f_k\}$ is the set of different GOI estimation methods,

Subject to

- ✓ $\forall g_j$ and $g_k \in G :$ if $j \neq k$ then $Area(g_j \cap g_k) = 0$,
- ✓ $\forall p_t \in \mathcal{T}, \exists g_j \in G \mid p_t \cap g_j \neq \emptyset$.

The first constraint guarantees that the geometries of extracted GOIs are mutually disjoint. The second constraint ensures that all the GPS points in the trajectory intersect with one and only one estimated GOI.

4 Data aggregation

Figure 1, shows the extracted GOIs of one mobile object from the Freesim dataset (Miller 2009), using the stay and destination extraction methods proposed in Mousavi et al. (2016). As it is evident, the GOIs (depicted with blue polygons), have an acceptable geometric intersection with the geometries of the real world GOIs (depicted by red polygons). However, a considerable area of the real GOIs are not covered by the real world GOIs, and also the estimated GOIs cover a considerable area outside the corresponding real GOIs.

The major reason for this problem is the lack of sufficient data gathered from the trajectory of only one mobile object. To deal with this problem, we aggregate the trajectory data of the other mobile objects (46 mobile objects) stored in our available GPS trajectory dataset (Miller and Horowitz 2007; Miller 2009) to the extracted GOIs of the particular mobile object. The data aggregation process does not add any new GOI to our GOI grid. It only reconstructs the geometries of the GOIs.

Before the data aggregation process, firstly, we extract the destinations of the particular mobile object resulting in the reference destination grid (*RDG*) using the method proposed in Mousavi et al. (2016). Then we extract the destinations of the aggregated trajectory resulting in multi-object destination grid (*MODG*). In data aggregation process (Algorithm 1), for each destination $r_i \in RDG$, $i = 1, 2, \ldots, n$, we find the destination $d_j \in MODG$, $j = 1, 2, \ldots, m$, such that the geometric similarity between r_i and d_j is greater or equal to the a predefined similarity threshold J_{min} and merge their GPS points with the point set of reference destination r_i. We use the same geometric similarity metric used in (Mousavi et al. 2016) with parameter $J_{min} = 0.1$ to measure the geometric similarity. After the process, we compute the convex hull of all the destinations in *RDG* resulting in the aggregated destination grid (*ADG*).

We perform the partitioning method proposed in (Mousavi et al. 2016) to convert the destinations in *RDG* into the final GOI grid. Figure 2 shows the result of data aggregation in our trajectory dataset. It can be clearly seen that the aggregated GOIs fully cover the car parks. In this regards, the data aggregation method performs better than when only a single mobile object is used (depicted in Fig. 1). However, the remaining problem is the existence of the outlier points in the point sets of each of the destinations which makes the areas of their corresponding estimated GOIs much bigger than the real GOIs which leads to the Ratio of Uncovering Estimated GOIs (RUEG) to be a high. Based on our problem definition, one of the main objectives is to minimize RUEG. In Sect. 5, an outlier detection and removal method is introduced to detect and remove the outliers and prune the geometries of the aggregated GOIs.

Algorithm 1: Data Aggregation Method

```
input  : A set of destination regions of the
         reference mobile object $RDG$,
         A set of destination regions of the
aggregated trajectory $MODG$,
         Jaccard similarity threshold $J_{min}$
output: Aggregated destination grid $ADG$
Data: Destination $d$, $InterList$, $RTreeIndex$
RTreeIndex.Update($MODG$)
$i \leftarrow 0$, while $(i < |RDG|)$ do
    $interList \leftarrow$
      $RTreeIndex.FindIntersectingDestinations(r_i)$
    $j \leftarrow 0$
    while $(j < |interList|)$ do
        $JSim \leftarrow$
          $Area(r_i \bigcap interList[j])/Area(r_i \bigcup interList[j])$
        if $(JSim \geq JSim_{min})$ then
            $r_i.MergePoints(interList[j])$
        end
        $j \leftarrow j + 1$
    end
    $i \leftarrow i + 1$
end
foreach $(r_i \in RDG)$ do
    $g_i^r \leftarrow ComputeConvexHull(p_i^r)$
end
$ADG \leftarrow RDG$
return $ADG$
```

5 Outlier detection

As discussed above, although the data aggregation process makes the resulting GOIs cover the area of the real GOIs, the size of the aggregated GOIs are significantly bigger than the real GOIs. In this section, we propose an outlier detection and removal process to tackle this problem. We examine three major outlier detection methods on the aggregated destinations and analyze their performance.

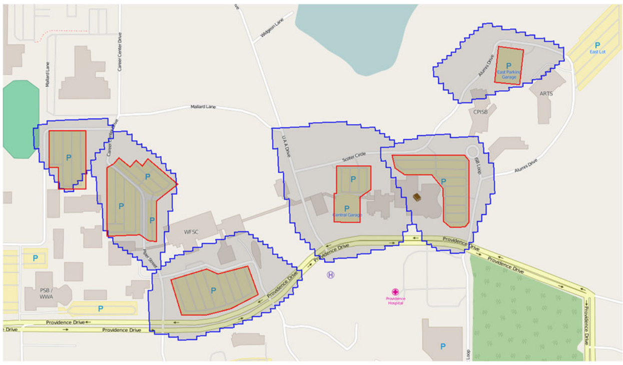

Fig. 2 Extracted aggregated GOIs of 46 mobile objects using time-weighted similarity based method (T_{min} = 60 min, D_{max} = 100 m, J_{min} = 0.10)

Moreover, we propose a novel outlier detection method customized for our particular problem by incorporating the concept of time-value (Mousavi et al. 2016) into the outlier detection process.

5.1 Speed based outlier detection

One of the most applied outlier detection methods in GPS trajectories is to filter the GPS points based on their speed. Research works such as (Bhattacharya et al. 2015) consider the GPS points which have the speed more than the speed of the pedestrian as outliers and remove them. They assume that people often stop at the GOIs or move around within their area with lower speed. Therefore, they simply remove the GPS points which have the speed greater than a speed threshold. This assumption is not very realistic since there are scenarios in which the mobile objects move with considerably higher speed around a certain destination geometry. For example, a security guard or a police officer might patrol within a GOI riding a motorbike or kinds of the vehicle which have speed more than the pedestrian speed threshold. By simply removing GPS points with the higher speeds, we lose a large amount of very useful information about the geometries of GOIs. Moreover, in various application domains such as analyzing the movement of animals, fish, or ants, we cannot simply define a valid speed threshold. Furthermore, it is likely that a point is an outlier in a destination region while its speed is zero.

As our first outlier detection method candidate, we examined the speed based outlier detection method on our aggregated destination grid (*ADG*) which are the result of our data aggregation process (Fig. 2). Figure 3a shows the outcome of the speed based outlier detection and removal process. As it is clearly evident, this method does not perform well since although the resulting GOIs cover almost all the area of the real world GOIs, the areas of the GOIs are considerably larger than real GOIs. Hence, in addition to performing speed based outlier detection method, we examine more capable methods for outlier detection which are based on outlier detection schemes discussed in statistics and machine learning.

5.2 Multivariate Mahalanobis distance based outlier detection

An outlier is an observation that lies an abnormal distance from other values in a random sample from a population. Clustering methods discussed in machine learning, as a post processing phase, attempt to detect and remove the points that are likely not a real member of the clusters (outliers). Various methods have been proposed for outlier detection (such as Zimek et al. 2012 and Gupta et al. 2014) in statistics and machine learning. Among these methods, we examine two methods proposed by Filzmoser et al. (2005) and Zhang et al. (2009).

As second outlier detection approch we use Mahalanobis distance (Filzmoser et al. 2005) to detect the multivariate outliers. The outliers are the observations which have large square Mahalanobis distance. To determine the outliers, this method uses the adjusted quantile of chi-square distribution to make the outlier detection more robust to extreme values in the population (Filzmoser et al. 2005). Figure 3b shows the results of the multivariate outlier detection method. It is evident that this method performs much better than speed based outlier detection method in minimizing the value of parameter RUEG. However, the value of parameter RURG is high since the estimated GOIs by this method leave a considerable area of the real GOIs uncovered.

5.3 LDOF based outlier detection

The third candidate method defines a Local Distance-Based Outlier Factor (LDOF), which is sensitive to outliers in a cluster. LDOF uses the relative distance from an object to its neighbors to measure how much the cluster points deviate from their neighborhood. The higher the violation degree an object has, the more likely the point is an outlier (Zhang et al. 2009).

Let $\mathcal{N}_p$ be the set of the k-nearest neighbours of point p_i in a cluster (excluding p_i). The k-nearest neighbours distance of p_i equals the average distance from p_i to all points in N_p (Zhang et al. 2009).

$$\bar{d}_{p_i} = \frac{1}{k} \sum_{p_n \in \mathcal{N}} dist(p_n, p_i).$$

Given the k-nearest neighbours set $\mathcal{N}_p$ of object p_i, the k-nearest neighbours inner distance of p_i is defined as the average distance between objects in $\mathcal{N}_p$:

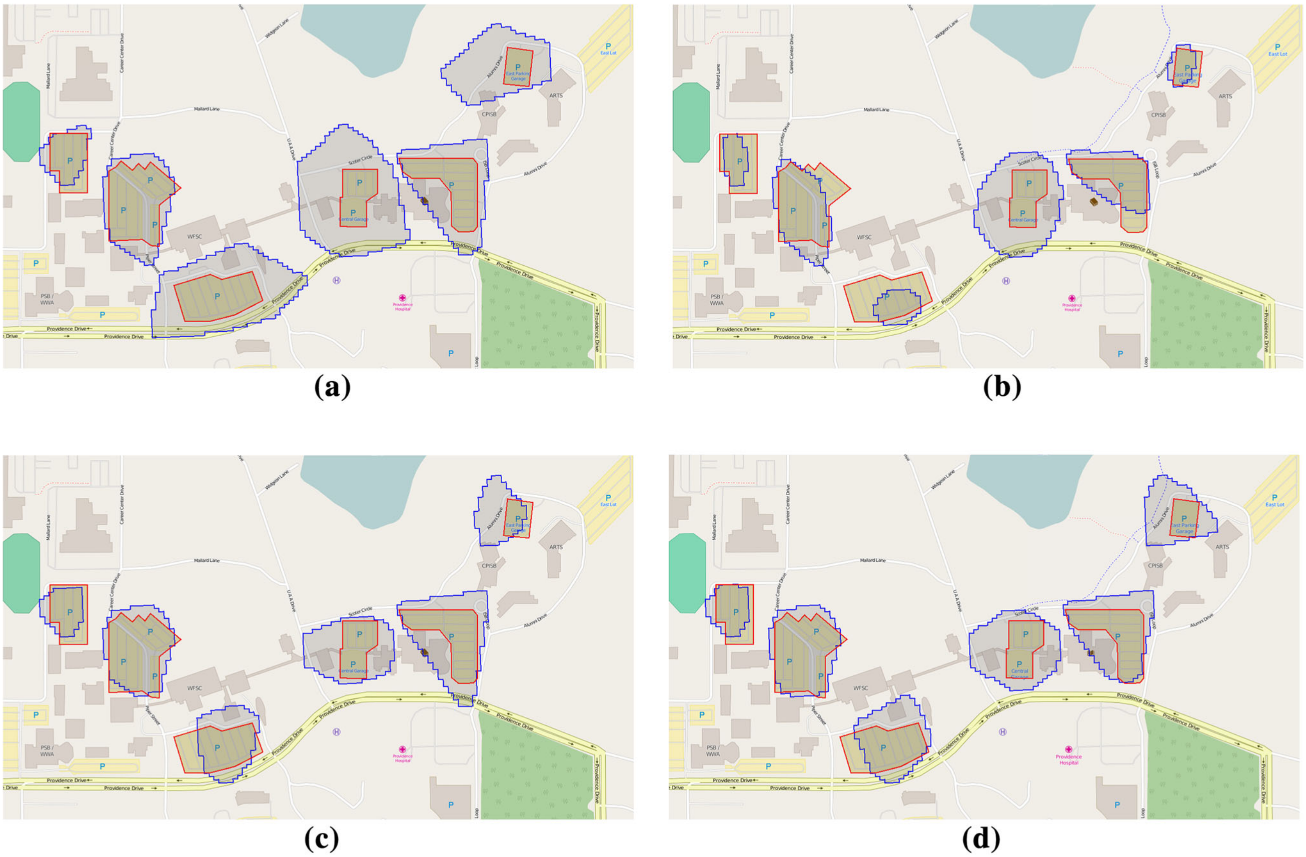

Fig. 3 Outlier detection results. **a** Speed based. **b** Mahalanobis distance based. **c** LDOF based. **d** Time-weighted LDOF based

$$\bar{D}_{p_i} = \frac{1}{k(k-1)} \sum_{p_i, p'_i \in \mathcal{N}, i \neq i'} dist(p_i, p'_i).$$

The local distance-based outlier factor (LDOF) of p_i is defined as:

$$LDOF_k(p_i) = \frac{\bar{d}_{p_i}}{\bar{D}_{p_i}}$$

If we regard the k-nearest neighbors as a neighborhood system, LDOF captures the degree to which object p_i deviates from its neighborhood system. Intuitively, this means that LDOF is the distance ratio indicating how far the object p_i lies outside its neighborhood system. LDOF $\lesssim 1$ indicates that p_i is surrounded by its neighbors. On the contrary, when LDOF $\gg 1$, p_i is outside its neighbors' data cloud. Obviously, the higher LDOF is, the farther p_i is away from its neighborhood system (Zhang et al. 2009).

Defining a lower bound for $LDOF_k(p_i)$ is not straight forward. Therefore, after computing this ratio for all $p_i \in \mathcal{S}$ we rank them based on their LDOF. Then we choose a predefined proportion of the points which have highest LDOF, consider them as the outlier and remove them from the point set. In our experiments, the proportion is set to 10%. To determine the size of the neighborhood system $\mathcal{N}_p$ of each point p_i, we consider two percent of the points which have lowest Euclidean distance to point P_i as the nearest neighbors.

This approach has a significant effect on removing outliers resulting in much cleaner data and better-shaped geometries compared to the speed based and multivariate Mahalanobis distance based outlier detection methods. Figure 3c shows the results of the LDOF based outlier detection method. As it is clearly seen, the shape and size of the resulting destination geometries are much more acceptable compared to Fig. 3a, b.

5.4 Time-weighted LDOF based outlier detection

Although the LDOF based outlier detection method outperforms the two above-discussed methods, it only considers the spatial characteristics of the points and does not consider any temporal characteristic. Based on the hypothesis that considering the time-value of points might have a positive impact on the accuracy of the outlier detection method, we alter the LDOF based outlier detection method by incorporating the concept of time-value (Mousavi et al. 2016) into the method resulting in a method called time-weighted LDOF.

Time-value of a GPS point can be considered as its degree of significant. Our hypothesis is if a GPS point in the point set of a destination has higher time value or is surrounded by the neighbors with higher time-values, the likelihood that the point is an outlier is low. The reason behind the hypothesis is that the points with higher time-values represent the positions where the mobile object has stopped for a longer time or has very low speed while the outliers normally are the points with very trivial stops.

Algorithm 2 shows the steps in computing the time-weighted LDOF of each point and removing the outliers. We incorporate the time values of each point in computing the parameters $\hat{d}_{p_i}$ (time-weighted k-nearest neighbours distance of p_i) and $\hat{D}_{p_i}$ (time-weighted k-nearest neighbours inner distance of p_i). Then we compute the time-weighted LDOF (TWLDOF) of each point and then remove a predefined percentage (10% in our experiments) of points with highest TWLDOF.

Our experimental results show that time-weighted LDOF based outlier detection method outperforms the LDOF based method. Figure 3d shows the results of performing the method on our aggregated destinations. For example comparing the shape of the GOI on the top right hand side of Fig. 3d with the same GOI in Fig. 3c, shows that assigning a weight to each point based on their time-value makes the resulting GOI cover almost all the area of the real GOI since on the right side of the destination, there were a few points with high time-values (longer stay times). LDOF based method simply considers the points as outliers since their LDOF are high. While time-weighted LDOF considers them as non-outlier points because their time-weighted LDOF are low enough. Slightly the same thing happens in the destination extracted for the second GOI from the left.

Algorithm 2: Time-Weighted LDOF Based Outlier Detection

```
input  : $PointList$: A set of GPS point in destination $\mathcal{D}$,
         The percentage of the outliers to be removed $PO$,
         The percentage of the K-nearest neighbours for each point $PK$
output: $PointList$: A set of GPS points in destination $\mathcal{D}$
Data:  $P_i^{knn}$: A set of points containing K-nearest neighbours of point $p_i$,
       $\hat{d}_{p_i}$: Average Euclidean distance to K-nearest neighbours of point $p_i$,
       $\hat{D}_{p_i}$: Average inner Euclidean distance between K-nearest neighbours of point $p_i$

foreach $(p_i \in PointList)$ do
    $P_i^{knn} \leftarrow ComputeKNN(p_i)$
    $count \leftarrow 0$
    $distance \leftarrow 0$
    foreach $(p_j \in P_i^{knn})$ do
        $distance \leftarrow distance + [tv_j^p \times tv_i^p \times EuclideanDistance(p_j, p_i)]$
        $count \leftarrow count + [tv_j^p \times tv_i^p]$
    end
    $\hat{d}_{p_i} \leftarrow {}^{distance}/_{count}$
    $count \leftarrow 0$
    $distance \leftarrow 0$
    foreach $(p_j \in P_i^{knn})$ do
        $distance \leftarrow distance + [tv_i^p \times tv_j^p \times EuclideanDistance(p_i, p_j)]$
        $count \leftarrow count + [tv_i^p \times tv_j^p]$
    end
    foreach $(p_j \in P_i^{knn})$ do
        foreach $(p_k \in P_i^{knn})$ do
            if $k \neq j$ then
                $distance \leftarrow distance + [tv_k^p \times tv_j^p \times EuclideanDistance(p_k, p_j)]$
                $count \leftarrow count + [tv_k^p \times tv_j^p]$
            end
        end
    end
    $\hat{D}_{p_i} \leftarrow {}^{distance}/_{count}$
    $p_i^{twldof} \leftarrow \hat{d}_{p_i}/\hat{D}_{p_i}$
end
$PointList \leftarrow$ RankPointsBasedOnTheirTWLDOF($PointList$)
$PointList \leftarrow$ RemoveOutliers($PointList$,$PO$)
return $PointList$
```

6 Constructing the SVL of the trajectory

To be able to evaluate the quality of the extracted GOIs, we need to generate the time-stamped sequence of visited locations of a trajectory on the grounds of the extracted GOIs.

A sequence of visited locations (SVL) of a trajectory of a mobile object is comprised of a list of visits $SVL = \{v_1, v_2, \ldots, v_n\}$. Each visit represents a visit to a cell in the GOI Grid. Each visit v_j has an arrival time (at_j^v), departure time (dt_j^v) and residence time (rt_j^v) as well as the ID of the visited cell (c_j^v). Each visit also includes a list of GPS points $pl_j^v = \{p_m, p_{m+1}, \ldots, p_n\}$ starting from the first GPS point intersected the cell (p_m) and ending to the last point intersected the cell (p_n). The arrival time of each visit indicates the time-stamp of the first GPS point in $\mathcal{T}$ intersecting the visited cell at the beginning of the visit (p_m). Departure time is defined as the time-stamp of the last GPS point intersecting the cell at the end of each visit (p_n). Algorithm 3 shows the steps of the process of constructing the SVL of a given trajectory. The algorithm receives a GPS trajectory ($\mathcal{T}$) and a GOI grid (GG) as input and returns the constructed SVL as output.

Algorithm 3: Construction of Sequence of Visited Locations (SVL)

```
input  : A GPS Trajectory $\mathcal{T}$, GOI Grid $GG$,
output: Sequence of Visited Locations $SVL$
Data:  RTReeIndex GGIndex
i ← 0
j ← 0
GGIndex ← MakeRTreeIndex($GG$)
oldCell ← GGIndex.FindIntersectingCell($p_i$)
$c_j^v \leftarrow oldCell.id$
$pl_j^v.Insert(p_i)$
$at_j^v \leftarrow t_i^p$
$i \leftarrow 1$
while $i < |\mathcal{T}|$ do
    newCell ← GGIndex.FindIntersectingCell($p_i$)
    if $newCell{=}oldCell$ then
        $pl_j^v.Insert(p_i)$
    end
    else
        $at_j^v \leftarrow pl_j^v[first].t$
        $dt_j^v \leftarrow pl_j^v[last].t$
        $rt_j^v \leftarrow dt_j^v - at_j^v$
        $SVL.Insert(v_j)$
        $j \leftarrow j + 1$
        $c_j^v \leftarrow newCell.id$
        $pl_j^v.insert(p_i)$
        $oldCell \leftarrow newCell$
    end
    $i \leftarrow i + 1$
end
return $SVL$
```

7 GOI extraction quality measurement

In this section, we introduce our geometric similarity based and spatio-temporal SVL quality based metrics for evaluation of the quality of GOI extraction methods. Then we evaluate each of the GOI extraction and outlier detection methods using the proposed metrics.

In our experiments, we use a set of GPS trajectories collected from 46 vehicles in the field to evaluate the GOI extraction methods. The dataset is collected in Anchorage, Alaska, as a part of the project FreeSim (Miller and Horowitz 2007). The trajectory used to construct the GOIs in Fig. 1 has been collected from a vehicle for the duration of about 42 months from 2010 to 2013 with varying sampling rate from one sample every 10 s to one sample every 2 min. We have cropped the trajectory to cover only the selected area depicted in Fig. 1. We also complement our database using the trajectories of other 45 mobile objects moving in the same area from the Freesim dataset in our data aggregation process.

7.1 Geometric similarity based quality measurement

We define two geometric dissimilarity metrics to measure the quality of GOI extraction. As discussed in Sect. 3, the main objectives of our spatio-temporal partitioning (comprised of stay extraction, destination extraction, data aggregation, outlier detection, and partitioning) is to minimize these three geometric distances.

The first metric measures the area of the geometries of each of the real world GOIs (ground truth) which are uncovered by the corresponding estimated GOIs. The lower intersecting area between reference geometries and the GOI indicated a lower geometric quality of the GOI extraction method. To compute the Ratio of Uncovered Real Geometries (RURG) we compute the sum of the uncovered real GOI areas (which are not intersecting with their corresponding estimated GOIs) and divide it by the sum of the areas of the real GOI geometries (Eq. 1).

RURG determines the degree of geometric quality of the GOI extraction. The lower RURG indicates a higher quality. However, considering only this geometric metric might cause a problem which is clearly evident in Figs. 2 and 3a. RURG is very low in both figures. However, the estimated GOIs (polygons depicted in blue) cover a large area outside the real world GOIs. They even cover the road segments which are obviously not related to the geometries of the car parks.

Therefore, another geometric dissimilarity which is required to be minimized is the area of the estimated GOIs which are intersecting (covering) with the corresponding real GOIs. To compute the area of the Ration of

Table 1 Geometric evaluation of the outlier detection methods

GOI extraction method	RURG	RUEG
GS based	0.435	0.745
Aggregated	0.008	2.646
Speed based	0.033	1.779
Mahalanobis distance base	0.434	1.067
LDOF based	0.334	1.208
TWLDOF based	0.103	0.845

Uncovering Estimated GOIs (RUEG), we compute the sum of the area of the estimated GOIs which are not intersecting with their corresponding real GOIs divided by the sum of the area of the real GOIs (Eq. 2). The lower RUEG in a GOI extraction method indicates better quality. For example, comparing Figs. 1 and 3a reveals that our proposed time-weighted LDOF outlier detection method has much lower RUEG compared to the speed based method.

Table 1 shows the computed values for both RURG and RUEG geometric dissimilarity metrics for each of the GOI extraction schemes. It is evident that our proposed time-weighted LDOF methods outperform the other methods in minimizing the geometric distances metrics. By looking closer at Fig. 3 can perceive that time-weighted LDOF based outlier detection method performs better than the other methods in fitting the geometries of the estimated GOIs to the real world GOIs geometries. We use a combination of the two metrics (RURG, RUEG) to find the best GOI extraction method which has the optimum performance (Eq. 3).

7.2 Spatio-temporal SVL quality based quality measurement

The main objective of the proposed spatio-temporal method is to generate the time-stamped sequence of visited locations (SVL) of the GPS trajectory by only using geometric intersection operator. The generated SVLs can be used in more advanced applications such as location prediction and other location-based applications. The quality of the generated SVLs has a significant impact on the performance of the applications, and the geometry of the extracted GOIs has a significant impact on the quality of the generated SVLs. In this section, we propose two metrics to measure the quality of the GOI extraction process based on the quality of the generated SVLs. These two metrics do not rely on any ground truth (the geometries of the real GOIs). Therefore, there are useful when we do not have access to the ground truth.

The outcome of the GOI extraction process is a grid of GOIs (Mousavi et al. 2016). A GOI of a mobile object is a place or a geometric region which the mobile object most likely stays for a considerable time inside the region. The stay duration depends on the minimum stay time (T_{min}) which was used as a parameter in the stay extraction phase of the GOI extraction process. Therefore, when a mobile object visits a GOI during the trajectory period, we expect the residence time at the particular GOI be equal or longer than the minimum stay time T_{min}. If the average residence times of the visits during the trajectory period is very short, we can conclude that it is highly likely that the area of the estimated GOIs is lower than the area of the real world GOI or the GOIs are covering some particular areas which the real world GOI do not cover.

Moreover, we expect the distance between the centroid of each visit (the centroid of the GPS points collected during each visit to a GOI) and the geometric centroid of the visited GOI to be minimal. This metric indicates how accurate the geometry of the GOI has been estimated. If the average distance between all the visit centroid and the GOI centroids during the trajectory SVL is considerably long, we can conclude that the geometries of the estimated GOIs are highly likely much bigger than the real world GOI geometries.

Following we define two metrics to measure the quality of GOI extraction based on the two above-mentioned expectations.

7.2.1 Deviation from centroid (DFC)

We define this measure as the average distance between the geometric centroid of our GOIs in our GOI-Grid and the time-weighted centroid of each visit to the GOI in the SVL. This metric examines the accuracy of the geometric centroids of our estimated GOIs and as a result the accuracy of the estimated geometry of the estimated GOI. Note that, in the process of SVL extraction; we store the GPS points that intersect each cell in the GOI-Grid during each visit to that cell. The time-weighted centroid of these GPS points represents the estimated centroid of that visit. If this centroid is less distant to the GOI centroid (more similar), we infer that the GOI extraction method was performed better. If the average dissimilarity of these two types of centroids is considerably large throughout the SVL, we can infer that the geometry of our extracted GOI corresponding to that visit was not estimated well enough.

We define metric deviation from centroid (DFC) for visit $v_i \in SVL$ is computed as follows:

$$DFC(v_i) = EucDist(Centroid(g_i^v), Centroid(ps_i^v)), \tag{4}$$

where, $centroid(g_i^v)$ finds the centroid of the GOI to which the visit v_i was happened in the SVL and $Centroid(ps_i^v)$ is the centroid of the visit v_i which is computed based on the time-weighted centroid of the GPS points in point set (ps)

of each visit. In the measurement process, we compute the DFC for every visit in the SVL.

To evaluate each of the GOI extraction methods based on the deviation from centroid metric, for each method, we have generated an SVL based on their GOI grid. Then for each visit in the SVL, we have computed the value for DFC. Figure 4 show the histogram of the value DFC throughout the trajectory period. The second column of Table 2, show the average DFC for each of the GOI extraction methods.

As it can be seen in Table 2, our proposed time-weighted LDOF based method obtained the lowest average DFC among the other available GOI extraction methods. This means that the extracted GOI of this method has the best estimated shapes. Empirical observation also confirms that the time-weighted LDOF method has the most fitted GOIs to the real GOIs. The aggregated method has the worst average DFC since the extracted GOIs has the biggest area and therefore, the average visit centroid to GOI centroid distance for each visit become longer.

Figure 4, histogram of the parameter DFC for each of the outlier detection methods. The histograms also confirm that the time-weighted centroid was more capable of minimizing the DFC of the visits. We use the data depicted in the histograms in our statistical significance analysis in Sect. 7.2.3.

7.2.2 *Residence time deviation (RTD)*

One of the conditions of a valid stay region is that the visit time must be longer than or equal to the minimum time span threshold ($\Delta t \geq T_{min}$). If the visit duration to a GOI in the SVL is considerably lower than (T_{min}), this means that the GOI geometry is likely estimated to be smaller than the real GOI or it is covering a place such as road segments which are outside the GOI. This situation often happens when the mobile object passes the GOI without stopping in the region. For example, assume a GOI corresponding to a university campus. The estimated geometry of the GOI covers an area of a street near the campus because the street is the road to and from the campus (some GPS points have been collected in the vicinity of the campus intersecting the street). It might also happen within the trajectory period that the mobile object passes the street without visiting the campus. In this case, the visit duration is often less than T_{min} (depending on the traffic conditions). This particular visit to the GOI is not an expected visit although it was recorded in the SVL because the visit duration was much shorter than the time span threshold.

We compute the residence time deviation for visit $v_i \in SVL$ as follows:

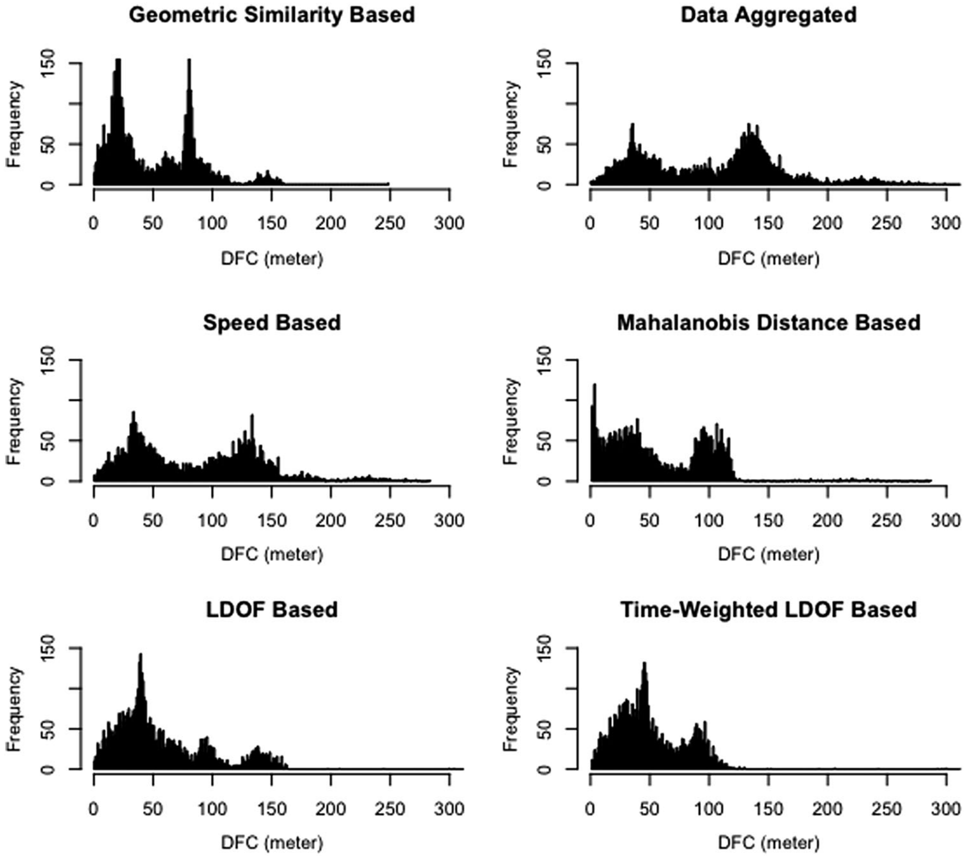

Fig. 4 The histogram of deviation from centroid (DFC)

Table 2 SVL based evaluation of the outlier detection methods

GOI extraction method	RTD (min)	DFC (m)
GS based	43.95	56
Aggregated	36.73	102
Speed based	35.33	90
Mahanalobis distance base	32.10	59
LDOF based	33.45	57
TWLDOF based	29.03	50

$$RTD(v_i) = \begin{cases} T_{min} - rs_i^v & \text{if } \mathrm{rs}_i^v < \mathrm{T}_{\min} \\ 0 & \text{if } \mathrm{rs}_i^v \geq \mathrm{T}_{\min}, \end{cases} \tag{5}$$

where rs_i^v is the residence time of the visit v_i. The value of RTD for each visit is zero if the residence time is greater or equal to T_{min} and it has a positive value in the distance $(0, T_{min})$ otherwise (Eq. 5).

To evaluate each of the GOI extraction methods, we have generated an SVL based on their GOI grid. Then for each visit in the SVL, we have computed the value for RTD. Figure 5 show the histogram of the value RTD throughout the SVL period. The third column of Table 2, show the average RTD for each of the outlier detection methods. The average RTD has been computed based on $T_{min} = 60min$. For example, the average RTD equals to 35.33 min means that on average, the residence time of the visits in the SVL is 35.33 min lower than 60 min (deviation from T_{min}). As it is seen in Table 2, the average RTD for time-weighted LDOF is considerably lower than the other methods.

Figure 5 shows the histogram of the RTD values for four SVL with about 5000 visits constructed based on GOIs extracted by each of the six methods. Each histogram shows the density of the values of the residence time deviation in the distance [0, 59] min. The density of value zero in each histogram shows the density of the visits with zero residence time deviation. Therefore, the lower value for the density of zero value shows a better result. On the other hand, the higher values around 59 min show higher deviation from T_{min} and accordingly, weaker performance. As it is evident, our proposed time-weighted LDOF outlier detection method has the highest density for zero and the lowest density around 59 and therefore, it is the most capable method for minimizing the values of RTD.

7.2.3 Statistical significance analysis

Although our experimental results show that the average value for both of the DFC and RTD metrics in time-weighted LDOF based outlier detection method are considerable lower than the other methods, we need to examine the statistical significance of the average values using statistical tests. Since depending on the data distribution and the variance of the data, the difference of averages between two data sets might be meaningful or not, techniques such as statistical analysis of variance (ANOVA) (Howell 2002) are widely used for this purpose.

ANOVA test is a parametric test. Its major assumption is the data is normally distributed. Since the distribution of our data in both DFC and RTD are not normally distributed (Kolmogorov–Smirnov test is performed), use of one-way ANOVA test (Howell 2002) is not valid. Instead, we use Kruskal–Wallis chi-squared test (Kruskal and Wallis 1952) which is the non-parametric alternative of one-way ANOVA. We use R statistical test packages (R Development Core Team 2008) to perform the tests on our data. Kruskal–Wallis chi-squared test provides a statistical test of whether or not the means of several groups are equal. In our tests, for each parameter DFC and RTD, we have six groups (Figs. 4, 5). The null hypothesis of the Kruskal–Wallis test is that the means of the groups are the same. Therefore, if the p-value of the test is calculated less that 0.05, we conclude that the difference between the average values of the groups is significant.

The experimental result of the Kruskal-Wallis chi-squared tests on our DFC and RTD data with six groups are lower than 0.05 (p-value 2.2×10^{-16}). Therefore, the null hypothesis is rejected, and the difference between means in the six groups are statistically significant. Therefore, we can conclude that the better average of DFC and RTD for our proposed time-weighted LDOF based outlier detection method are significantly different to the other five groups.

We also perform the pairwise comparisons of the mean of the six groups using pairwise Wilcoxon rank sum test (Smucker et al. 2007). The results show that the pairwise difference between the mean of DFC values of our time-weighted LDOF based method and the other five methods are significant (p-value <0.05). Similarly, the pairwise test for RTD data, shows that the mean of our Time-Weighted LDOF method is significantly different from the other five groups as well (p-value <0.05).

A quick look at the histograms of RTD (Fig. 5) and DFC (Fig. 4) in all six groups confirms the pairwise tests. The density of RTD with zero value in time-weighted LDOF which shows the number of visits with expected residence time ($\geq T_{min}$) is much higher than the other methods. On the other hand, the density of the unexpected residence time ($0 < RTD < 60$) in time-weighted LDOF base method is much lower than the other methods. These results show that our proposed outlier detection method outperforms the baseline GOI extraction and outlier detection methods. As far as the deviation from the centroid (DFC) is concerned, our proposed method outperforms the other methods because

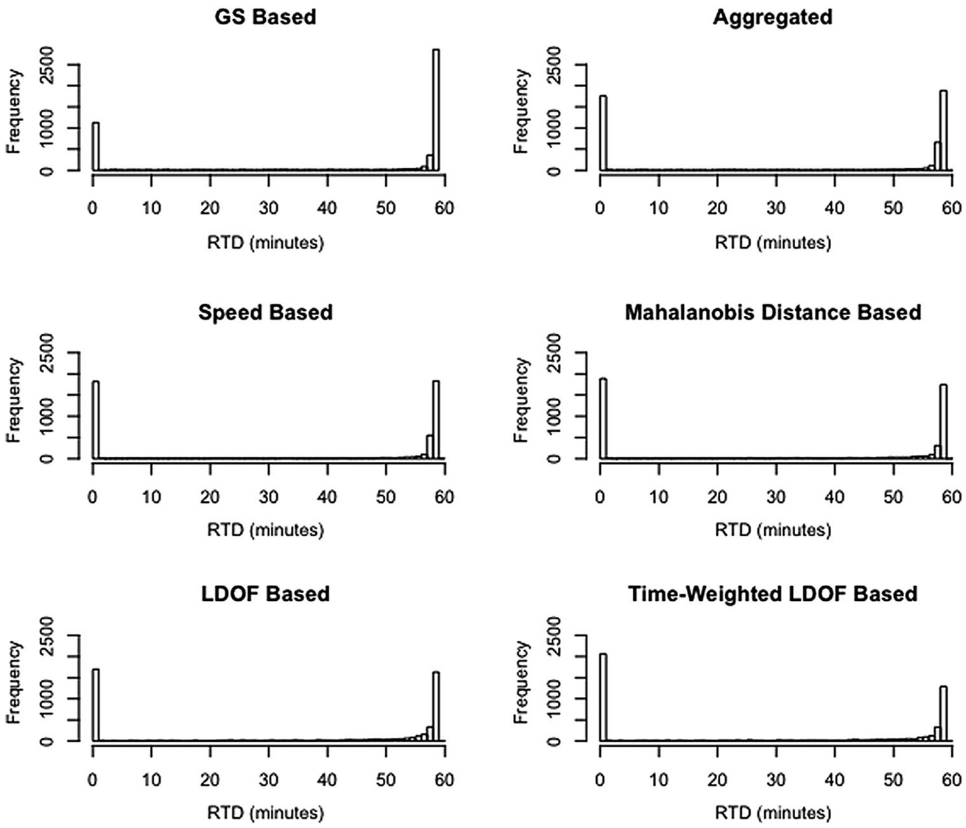

Fig. 5 The histogram of residence time deviation (RTD), $T_{min} = 60min$

the histogram of DFC (Fig. 4) in our method has the shortest tail and highest density between zero and 100 m.

8 Conclusion and future work

In this paper, we addressed the problem of data aggregation and outlier detection to improve the most recent method proposed to construct the geometries of interest in trajectory data analysis research. We proposed two geometric and two SVL quality based quality evaluation metrics. The research shows that the idea of data aggregation itself is not very capable in enhancing the geometric accuracy of GOI extraction methods. However, performing outlier detection along with considering the concept of time-value in the outlier detection improves the quality of the extracted GOIs significantly.

This research has opened up many questions in need of further investigation and can serve as a base for future studies. It would be interesting to focus on improving the performance and the accuracy of our proposed GOI extraction methods by finding the best geometric similarity metric in the data aggregation phase and finding the optimum value for the percentage of the data which should be removed from the set of points in the outlier detection phase, as a future work.

Acknowledgements We would like to acknowledge the financial support that we received from Data61 during this research project.

References

Ashbrook D, Starner T (2003) Using GPS to learn significant locations and predict movement across multiple users. Pers Ubiquitous Comput 7(5):275–286

Bhattacharya T, Kulik L, Bailey J (2012) Extracting significant places from mobile user GPS trajectories: a bearing change based approach. In: Proceedings of the 20th International Conference on Advances in Geographic Information Systems, ACM, New York, NY, USA, SIGSPATIAL '12, pp 398–401

Bhattacharya T, Kulik L, Bailey J (2015) Automatically recognizing places of interest from unreliable GPS data using spatio-temporal density estimation and line intersections. Pervasive Mob Comput 19:86–107

De Maio C, Fenza G, Loia V, Orciuoli F (2016) Unfolding social content evolution along time and semantics. Future Gener Comput Syst. doi:10.1016/j.future.2016.05.039

Ester M (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. AAAI Press, Portland, pp 226–231

Fenza G, Fischetti E, Furno D, Loia V (2011) A hybrid context aware system for tourist guidance based on collaborative filtering. In: Fuzzy Systems (FUZZ), 2011 IEEE International Conference, pp 131–138

Filzmoser P, Garrett RG, Reimann C (2005) Multivariate outlier detection in exploration geochemistry. Comput Geosci 31(5):579–587

Gidófalvi G, Dong F (2012) When and where next: individual mobility prediction. In: MobiGIS, pp 57–64

Gupta M, Gao J, Aggarwal C, Han J (2014) Outlier detection for temporal data. Synth Lect Data Mining Knowl Discov 5(1):1–129

Handcock RN, Swain DL, Bishop-Hurley GJ, Patison KP, Wark T, Valencia P, Corke P, ONeill CJ (2009) Monitoring animal behaviour and environmental interactions using wireless sensor networks, GPS collars and satellite remote sensing. Sensors 9(5):3586–3603

Hariharan R, Toyama K (2004) Project lachesis: parsing and modeling location histories. In: Egenhofer M, Freksa C, Miller H (eds) Geographic information science. Lecture notes in computer science, vol 3234. Springer, Berlin, Heidelberg, pp 106–124

Howell D (2002) Statistical methods for psychology. Duxbury/Thomson Learning, North Scituate

Kruskal WH, Wallis WA (1952) Use of ranks in one-criterion variance analysis. J Am Stat Assoc 47(260):583–621

Li Z, Han J, Ji M, Tang LA, Yu Y, Ding B, Lee JG, Kays R (2011) Movemine: mining moving object data for discovery of animal movement patterns. ACM TIST 2(4):37–57

MacQueen J (1967) Some methods for classification and analysis of multivariate observations. In: Le Cam LM, Neyman J (eds) Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, vol 1. University of California Press, Berkeley, CA, USA, pp 281–297

Mahalanobis PC (1936) On the generalized distance in statistics. Proceedings of the National Institute of Sciences (Calcutta) 2:49–55

Miller J (2009) Fastest path analysis in a vehicle-to-infrastructure intelligent transportation system architecture. In: Intelligent Vehicles Symposium, 2009 IEEE, pp 1125–1130

Miller J, Horowitz E (2007) Freesim—a free real-time freeway traffic simulator. In: Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE, pp 18–23

Min W, Wynter L (2011) Real-time road traffic prediction with spatio-temporal correlations. Transp Res Part C Emerg Technol 19(4):606–616

Mousavi SM, Harwood A, Karunasekera S, Maghrebi M (2016) Geometry of interest (GOI): spatio-temporal destination extraction and partitioning in GPS trajectory data. J Ambient Intell Human Comput. doi:10.1007/s12652-016-0400-5

Palma AT, Bogorny V, Kuijpers B, Alvares LO (2008) A clustering-based approach for discovering interesting places in trajectories. In: Proceedings of the 2008 ACM Symposium on Applied Computing, ACM, New York, NY, USA, SAC '08, pp 863–868

R Development Core Team (2008) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org **(ISBN 3-900051-07-0)**

Scellato S, Musolesi M, Mascolo C, Latora V, Campbell AT (2011a) Nextplace: a spatio-temporal prediction framework for pervasive systems., Proceedings of the 9th international conference on Pervasive computing, Pervasive ′11. Springer-Verlag, Berlin, Heidelberg

Scellato S, Musolesi M, Mascolo C, Latora V, Campbell AT (2011b) Nextplace: a spatio-temporal prediction framework for pervasive systems. In: Pervasive, pp 152–169

Si H, Wang Y, Yuan J, Shan X (2010) Mobility prediction in cellular network using hidden markov model. In: Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE, pp 1–5

Smucker MD, Allan J, Carterette B (2007) A comparison of statistical significance tests for information retrieval evaluation. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, ACM, New York, NY, USA, CIKM '07, pp 623–632

Song L, Deshpande U, Kozat U, Kotz D, Jain R (2006) Predictability of wlan mobility and its effects on bandwidth provisioning. In: INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings, pp 1–13

Xiao X, Zheng Y, Luo Q, Xie X (2010) Finding similar users using category-based location history. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, New York, NY, USA, GIS '10, pp 442–445

Xiao X, Zheng Y, Luo Q, Xie X (2014) Inferring social ties between users with human location history. J Ambient Intell Human Comput 5(1):3–19

Xue AY, Zhang R, Zheng Y, Xie X, Huang J, Xu Z (2013) Destination prediction by sub-trajectory synthesis and privacy protection against such prediction. In: IEEE International Conference on Data Engineering (ICDE 2013), IEEE

Ye Y, Zheng Y, Chen Y, Feng J, Xie X (2009) Mining individual life pattern based on location history. In: Mobile Data Management: Systems, Services and Middleware, 2009. MDM '09. Tenth International Conference, pp 1–10

Zhang K, Hutter M, Jin H (2009) A new local distance-based outlier detection approach for scattered real-world data. In: Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining. Springer-Verlag, Berlin, Heidelberg, PAKDD '09, pp 813–822

Zheng Y, Chen Y, Li Q, Xie X, Ma WY (2010) Understanding transportation modes based on GPS data for web applications. ACM Trans Web 4(1):1:1–1:36

Zhou C, Frankowski D, Ludford P, Shekhar S, Terveen L (2004) Discovering personal gazetteers: An interactive clustering approach. In: Proceedings of the 12th Annual ACM International Workshop on Geographic Information Systems, ACM, New York, NY, USA, GIS '04, pp 266–273

Zimek A, Schubert E, Kriegel HP (2012) A survey on unsupervised outlier detection in high-dimensional numerical data. Stat Anal Data Mining 5(5):363–387