Robust Time-Referenced Segmentation of Moving Object Trajectories*

Hyunjin Yoon and Cyrus Shahabi
University of Southern California
Los Angeles, CA 90089-0781
{hjy, shahabi}@usc.edu

Abstract

Trajectory segmentation is the process of partitioning a given trajectory into a small number of homogeneous segments w.r.t. some criteria. Conventional segmentation techniques only focus on the spatial features of the movement and could lead to spatially homogeneous segments but with presumably dissimilar temporal structures. Furthermore, trajectories could be over-segmented in the presence of outliers. In this paper, we propose a family of three trajectory segmentation methods that takes into account both geospatial and temporal structures of movement for the segmentation and is also robust with respect to time-referenced spatial outliers. The effectiveness of our methods is empirically demonstrated over three real-world datasets.

1. Introduction

A *trajectory* of a moving object is a series of locations sampled at discrete instances of time and defined as a sequence of pairs, $\langle(p_1,t_1),\ (p_2,t_2),\ldots,(p_n,t_n)\rangle$, where p_i is a two- or three-dimensional vector representing the geospatial position observed at a timestamp t_i ($i = 1,\ldots,n$). Various types of trajectory data tracking the movement of vehicles, animals, or human subjects have been acquired using location-aware sensors and exploited to find similar trajectories [4], discover frequent spatio-temporal patterns [3, 6, 8], and eventually obtain insights into the behavioral traits of moving objects.

Often the size of a trajectory, i.e., the number of observations n, is large. For example, the elk trajectories used in [8] contain about 1430 observations on average, and the size of bus trajectories used in [3] varies in the range from 1000 to 7000. It is therefore necessary to preprocess the trajectories to reduce the dimensionality and compress them in a compact and concise representation in order to process them efficiently in the subsequent data analysis tasks.

*This research has been funded in part by NSF grants IIS-0238560 (PECASE), IIS-0534761, IIS-0742811 and CNS-0831505 (CyberTrust), and in part from the METRANS Transportation Center, under grants from USDOT and Caltrans. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Trajectory segmentation is an attempt to partition a given trajectory into a small number of *homogeneous* segments, such that the data within each segment are similar w.r.t. some criteria and thus can be effectively described by a simple model [2]. A typical approach previously adopted for the trajectory segmentation [3, 8] takes a simple sequence of sampled locations (by dropping the timestamp component) of a trajectory as an input, which we call a *route* of a moving object to explicitly distinguish it from a trajectory. The approach first selects a subset of the sampled locations, identified as *characteristic points* (CPs), where the geometric structure (e.g., spatial closeness, co-linearity, or movement direction) of the given route changes substantially. Subsequently, only the selected CPs are retained to approximate the input trajectory as a sequence of lines, each connecting two consecutive CPs. Figure 1 illustrates a route with 9 sampled positions and its desirable segmentation into four continuous and non-overlapping segments.

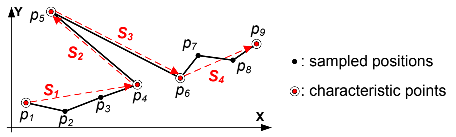


Figure 1. An example of route segmentation

Our key observation is that such segmentations discarding the time component could lead to spatially homogeneous segments but with presumably dissimilar temporal or spatio-temporal structures unless a constant sampling rate is assumed[1]. Suppose the first four locations in Figure 1 are acquired at irregular sampling rate, e.g., time-stamped at 1, 2, 3, and 13, respectively. From the timestamps together with the moving distances, it can be derived that the speed development of the moving object varies within the obtained segment S_1; it is fast at first from p_1 to p_2, similarly fast from p_2 to p_3, and then moves slowly from p_3 to p_4. Since the movement speed significantly changes at p_3, the segment S_1 should have been partitioned at p_3 to result in genuinely homogeneous segments in terms of both

[1]Irregular sampling rates are usually encountered in the real-world sensor data due to the inherent imprecisions of sensor devices, missing data, network failure or delay, disturbance signals, etc.

spatial and temporal semantics.

Another limitation of previous approaches is their vulnerability to outliers. A single extreme location value, typically depicted as a *peak* (see p_5 on the route in Figure 1), is most likely selected as a characteristic point by the conventional segmentation due to its substantial change of geometric structure. However, if p_5 is an incorrect noisy sensor value representing unrealistic movement and hence an outlier, a desired segmentation robust to noise would lead to a single segment connecting p_1 to p_9, discarding this outlier. In order to detect whether p_5 is indeed an outlier or just a surprising but a correct value, the associated temporal information must be exploited together with the spatial changes (see Section 2.2). Otherwise, trajectories will be over-segmented at all spatial outliers by erroneously considering them as CPs.

In this paper, we propose a family of three robust time-referenced trajectory segmentation algorithms that take into account both spatial and temporal structures presented in the trajectory data to identify genuine characteristics points. Intuitively, we attempt to partition a given trajectory into a small number of *spatially* and *temporally homogeneous* segments, such that each segment accurately approximates a linear movement at a constant speed. In addition, we utilize the spatio-temporal properties of movement to guide the outlier detection so as to make the segmentation robust to outliers. Our experiments on three real-world datsets indicate that our techniques outperform the conventional techniques, such the Douglas-Peucker algorithm and the one-pass approximation algorithm based on the Minimum Description Length principal as well as their simple temporal extensions, in terms of spatio-temporal homogeneity while maintaining comparable spatial homogeneity.

2. Preliminaries

In this section, we present our trajectory model and some basic definitions to be used in the rest of paper .

2.1. Trajectory Model

The movement of a moving object is typically traced by sampling its geographic locations at discrete instances of time by using location-aware sensors. A *trajectory* S is a finite sequence of sampled locations during a closed time interval $[t_1, t_n]$ and defined as a sequence of pairs $S = \langle(p_1, t_1), (p_2, t_2), \ldots, (p_n, t_n)\rangle$, where $p_i \in \Re^d$ $(d \in \{2, 3\})$ is a two- or three-dimensional location vector representing the geo-spatial position sampled at a timestamp $t_i \in \Re^+$. A *route* of the moving object is the projection of the trajectory S on the spatial space by dropping the temporal component, thus defined as the simple sequence of sampled position values $R_s = \langle p_1, p_2, \ldots, p_n\rangle$. The size of a trajectory is defined as the number of samples and denoted as $|S|$=n.

Figure 2 illustrates a trajectory as a solid *directed* polyline in the *spatio-temporal space* formed by combining the spatial plane spanned by X and Y axes (shaded in the figure) and the time dimension of Z axis. The dashed line on the (shaded) spatial plane shows the route. We assume an irregular sampling rate, that is, the time intervals between two consecutive samples, $\Delta t_i = t_{i+1} - t_i$ $(i = 1, \ldots, n-1)$, can vary even within a single trajectory. This is a reasonable assumption due to the inherent imprecision involved in the sensor data acquisition.

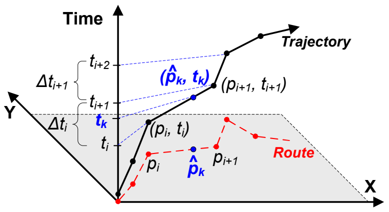


Figure 2. example of a trajectory and its route

As shown in Figure 2, during a time interval $[t_i, t_{i+1}]$, the unmeasured movement of a moving object is estimated by a linear interpolation using the observed positions p_i and p_{i+1}, assuming that the object moves straight from the location p_i to p_{i+1} at a constant speed. Therefore, at any instance of time t_k within this time interval $(i \leq k \leq i+1)$, the *estimated* position $\widehat{p}_k$ placed along the line can be calculated as follows:

$$\widehat{p}_k = \frac{t_k - t_i}{t_{i+1} - t_i} p_i + \frac{t_{i+1} - t_k}{t_{i+1} - t_i} p_{i+1}. \tag{1}$$

Definition 1. A segment s_{ij} of a trajectory S of size n $(1 \leq i < j \leq n)$ is a continuous sub-sequence of S, starting from (p_i, t_i) and ending at (p_j, t_j). A segment s_{ij} is *spatially* and *temporally homogeneous* during the time interval $[t_i, t_j]$ with respect to a pre-specified distance threshold ε if all sampled locations p_k observed at time t_k $(i \leq k \leq j)$ within this segment deviate from their estimated time-referenced positions $\widehat{p}_k$ by no more than ε.

Note that $\widehat{p}_k$ is obtained as in Equation 1 by a linear interpolation using the starting and the ending positions p_i and p_j, respectively, while assuming a linear movement with a constant speed during this time interval $[t_i, t_j]$. The distance between p_k and $\widehat{p}_k$, denoted as T-dist(p_k, $\widehat{p}_k$), is computed using the Euclidean distance, and called *time-referenced* distance in order to emphasize that the temporal structure is incorporated to measure the spatial dissimilarity. This time-reference distance can be formulated as follows;

$$\text{T-dist}(p_k, \widehat{p}_k | s_{ij}) = dist(p_k, \frac{t_k - t_i}{t_j - t_i} p_i + \frac{t_j - t_k}{t_j - t_i} p_j) \tag{2}$$

, where $i \leq k \leq j$ and $dist(\cdot)$ is the Euclidean distance.

Figure 3(a) illustrates a segment $s_{i,i+2}$, starting from (p_i,t_i) and ending at (p_{i+2},t_{i+2}). This segment is spatially and temporally homogeneous since the Euclidean distance between the actually measured location p_{i+1} and the estimated location $\widehat{p}_{i+1}$ at the time instance t_{i+1} is less than a given distance threshold ε. Figure 3(b) illustrates the projected view of the segment onto the spatial plane and shows how the time-referenced distance T-dist(p_{i+1}, $\widehat{p}_{i+1}$) is different from the perpendicular distance $d_\perp(p_{i+1}, \overrightarrow{p_i p_{i+2}})$ widely employed in the previous segmentations [3, 8].

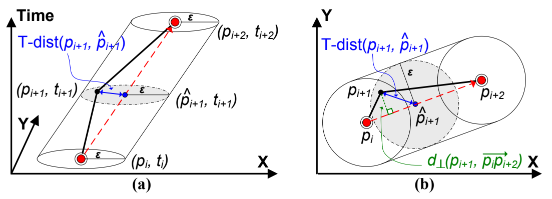

Time
T-dist(p_{i+1}, $\hat{p}_{i+1}$)
(p_{i+2}, t_{i+2})
(p_{i+1}, t_{i+1})
($\hat{p}_{i+1}$, t_{i+1})
ε
Y
(p_i, t_i)
X
(a)
Y
T-dist(p_{i+1}, $\hat{p}_{i+1}$)
ε
p_{i+1}
p_{i+2}
p_i
$\hat{p}_{i+1}$
$d_\perp(p_{i+1}, \overrightarrow{p_i p_{i+2}})$
X
(b)

Figure 3. Spatially and temporally homogeneous segment

2.2. Outlier Detection

An outlier is a single extreme value that is substantially different from the rest of data at some measures [1]. In moving object trajectories, an outlier can be identified as an extreme rate of change of location value to a unit time interval and thus visually depicted as a *peak* in the spatio-temporal space as shown in Figure 4(a). Note that the peak in the spatio-temporal space is different from the one in the spatial subspace. For example, Figure 4(a) shows a significant change of location both at p_4 and p_7, both of which are illustrated as peaks, thus outliers, in the spatial plane. However, considering that the time intervals taken to sample the locations p_4 and p_7, p_4 is not really a surprising rate of location change per unit time, while p_7 could be indeed a substantial change in a short time interval. Unfortunately, the $slow$ peak or the $fast$ peak is not distinguishable in the geo-spatial representation unless a constant sampling rate is assumed. Detecting outliers simply based on the *static* geometric structure presented in route can be thus misleading due to the missing temporal information.

In order to identify the time-referenced spatial outliers, spatio-temporal features of movement must be incorporated. This suggests us to adopt the maximum movement speed as an indicator whether a peak is an error or just a surprising but a correct value representing the real movement. The key idea is that the maximum speed of a moving object can limit the upper bound of where the moving object could traverse in a given time interval. If the sampled location is beyond this range, it is considered as an error.

Consider the segment $s_{i-1,i+1}$ of a trajectory in Figure 4(b). Suppose the moving object's maximum speed υ_m is known in advance. If the object linearly moves at maximum speed from p_{i-1}, its position at time t_i will be on the surface of a half (smaller) sphere of a radius $r_1 = \upsilon_m \times (t_i - t_{i-1})$ as shown in Figure 4(b). The points on the surface of the half sphere are the furthest positions taken at the time instance t_i. If the movement speed is less than υ_m, or it does not move, then the time-referenced location at time t_i is somewhere within the area bounded by the smaller half sphere. Similarly, in order to move to the location p_{i+1} by the time t_{i+1} during the time interval $[t_i, t_{i+1}]$, the location at the preceding time instance t_i should be at most on the surface of the (larger) half sphere of radius $r_2 = \upsilon_m \times (t_{i+1} - t_i)$ or somewhere inside it. Therefore, the time-referenced location (p_i,t_i) beyond the range represents the unrealistic movement, thus indeed an inaccurate and noisy sensor measurement.

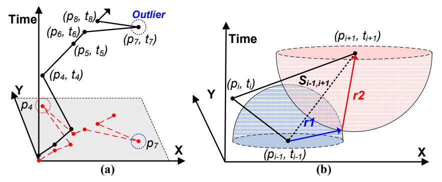


Figure 4. Outlier detection

Definition 2. A location p_i at time t_i is a time-referenced spatial outlier w.r.t. a pre-specified maximum speed υ_m if $dist(p_i - p_{i-1}) > \upsilon_m \times (t_i - t_{i-1}) \wedge dist(p_{i+1} - p_i) > \upsilon_m \times (t_i - t_{i-1})$, where $dist(\cdot)$ is the Euclidean distance.

3. Trajectory Segmentation

In this section, we formalize the problem and present three trajectory segmentation algorithms as our solution.

3.1. Problem Definition

The trajectory segmentation problem considered in this paper can be formalized as follows; given a trajectory S of size n, $S = \langle (p_1,t_1), (p_2,t_2), \ldots, (p_n,t_n) \rangle$, determine a *minimum* subset of the discrete instances of time, $CT = \{t_{c_1}, t_{c_2}, \ldots, t_{c_k}\}$ $(1 \leq c_1 < c_2 < \ldots < c_k \leq n)$, termed *characteristic timestamps* (CTs), such that each segment $s_{c_i c_{i+1}}$ $(1 \leq i \leq k-1)$ that starts from (p_{c_i}, t_{c_i}) and ends at $(p_{c_{i+1}}, t_{c_{i+1}})$ is spatially and temporally homogeneous during the corresponding time interval w.r.t. a pre-specified distance threshold ε (by Definition 1) and none of the locations sampled at the characteristic timestamps is a time-referenced spatial outlier w.r.t. a pre-specified maximum speed υ_m (by Definition 2).

The optimal solution that finds the minimum subset of the timestamps satisfying the constraints on the maximum deviation and the outliers exclusion can be obtained by using dynamic programming as in [9] or methods developed for the shortest path problem in digraph [7]. However, these are expensive solutions due to their time complexity usually ranging between $O(n^2)$ and $O(n^3)$. Instead, we adopt a greedy approach based on the *split* or *merge* heuristics, which greedily selects the local optimum at each iteration as hoping to lead to a global optimum.

3.2. Top-down Algorithm

Our top-down segmentation algorithm takes an unsegmented trajectory as an input and selects one characteristic timestamp t_i with the largest time-referenced distance between the actually observed location p_i sampled at this time t_i and its estimated time-referenced location $\widehat{p}_i$. If the observed location p_i is an outliers w.r.t. its preceding and succeeding locations and a given maximum speed, it should not be considered as the characteristic timestamp. Subsequently, it splits the sequence at t_i into two subsequences, which is recursively repeated until no more observation deviates from its estimated time-referenced position by more than the given distance threshold.

Algorithm 1 RTR-TopDown (S, ε, υ_m)

```
1:  $CT \leftarrow \emptyset$;  $n \leftarrow |S|$; // $n > 2$
2:  for $i$=2 to $n$-2 do
3:      $splitMeasure(i) \leftarrow$ RobustTDist($S_{i-1,i+1}, 1, \upsilon_m$);
4:  if max($splitMeasure$) $\leq \varepsilon$ then // No more split
5:      return $CT \leftarrow \{t_1, t_n\}$;
6:  $splitidx \leftarrow$ get the index of max($splitMeasure$);
7:      // Recursively split the subsequences
8:  $CT1 \leftarrow$ RTR-TopDown($S_{1,splitidx}$, $\varepsilon$, $\upsilon_m$);
9:  $CT2 \leftarrow$ RTR-TopDown($S_{splitidx,n}$, $\varepsilon$, $\upsilon_m$);
10: return $CT \leftarrow CT1 \bigcup CT2$; // Union the selected timestamps
```

Algorithm 1 shows the pseudocodes of our top-down algorithm. We first compute the *split* measure of each observation in a given trajectory using the procedure *RobustTDist*() in lines 2-3 based on which we choose one *splitting* characteristic timestamp. The robust time-referenced distance function *RobustTDist*($S_{i-1,i+1}$, 1, υ_m) in line 3 returns the spatial distance of the observation p_i in the segment $S_{i-1,i+1}$ to its estimated position if this observation is not an outlier w.r.t. the given maximum speed υ_m and a zero value otherwise, as shown in Algorithm 2. If the maximum value of the split measures is below the given distance threshold ε, there will be no more splits and the segmentation is stopped by returning the CTs found so far (lines 4-5). Otherwise, the input sequence is partitioned at the observation with the largest split measure value into two subsequences, each of which is recursively partitioned by the same *RTR-TopDown* procedure as in lines 8 and 9.

Algorithm 2 RobustTDist ($S_{startidx,endidx}$, i^{th}, υ_m)

```
1: $tdist \leftarrow 0$;  $cidx \leftarrow startidx + i^{th}$; // $cidx \leq endidx$
2: if dist($p_{cidx}, p_{cidx-1}$) $> \upsilon_m \times (t_{cidx} - t_{cidx-1}) \wedge$
       dist($p_{cidx+1}, p_{cidx}$) $> \upsilon_m \times (t_{cidx+1} - t_{cidx})$ then
3:     return $tdist$ ;    // $p_{cidx}$ is an outlier
4: else
5:     $\widehat{p}_{cidx} \leftarrow (t_{cidx} - t_{startidx})/(t_{endidx} - t_{startidx}) \times p_{startidx}$
           $+ (t_{endidx} - t_{cidx})/(t_{endidx} - t_{startidx}) \times p_{endidx}$;
6:     return $tdist \leftarrow$ dist($p_{cidx}, \widehat{p}_{cidx}$);
```

The *RobustTDist*($S_{startidx,endidx}$, i^{th}, υ_m) procedure in Algorithm 2 computes the time-referenced distance between the actually measured location observed at the i^{th} timestamp in the input segment and its corresponding estimated location, i.e., T-dist($p_{i^{th}}$,$\widehat{p}_{i^{th}}|S_{startidx,endidx}$) as in Equation 2, assuming a linear movement at a constant speed within the input segment if the observation $p_{i^{th}}$ at time $t_{i^{th}}$ is not an outlier (lines 5-6). If the observation is an outlier w.r.t. a given maximum speed υ_m, a zero value is returned so that its timestamp should not be selected as the characteristic timestamp in the top-down algorithm.

The time complexity of the top-down algorithm is $O(kn^2)$ after k number of splits, where n is the size of trajectory, and can be reduced to $O(kn \log n)$ if an efficient data structure such as a priority queue is used to maintain the split measure values in order.

3.3. Bottom-up Algorithm

A typical bottom-up algorithm begins with the finest possible segments of a given trajectory and greedily merge two adjacent segments with the minimum merge cost.

Algorithm 3 RTR-BottomUp (S, ε, υ_m)

```
1: $n \leftarrow |S|$;  $CT \leftarrow \{t_i | 1 \leq i \leq n\}$;
2: for $i$=2 to $n$-1 do
3:     $mgMeasure(i) \leftarrow$ RobustTDist($S_{i-1,i+1}, 1, \upsilon_m$);
4: if min($mgMeasure$) $> \varepsilon$ then // No more merge
5:     return $CT$;
6: $mgidx \leftarrow$ get the index of min($mgMeasure$);
7: $CT \leftarrow CT - \{t_{mgidx}\}$;   // Remove the selected $CT$
8: $S' \leftarrow$ concatenate($S_{1,mgidx-1}, S_{mgidx+1,n}$);
9: return RTR-BottomUp($S'$, $\varepsilon$, $\upsilon_m$);
```

As shown in Algorithm 3, we first initialize the set of characteristic timestamps (CTs) with the finest possible timestamps of a given trajectory of size n in line 1. Subsequently, we compute the *merge* measure of the observation at each CT by using the same distance function *RobustTDist*() in lines 2-3. If the minimum value of the merge measures exceeds the given distance threshold ε, the segmentation stops by returning the CTs found so far in lines 4-5. Otherwise, we remove the timestamp of the observation with the minimum value from the CT set (lines 6-7). The trajectory segment (S') without the removed observation is recursively merged until no more observation is an outlier w.r.t. the given maximum speed υ_m or deviates from its estimated location by more than the given distance threshold ε (line 8-9). Note that the robust time-referenced distance function returns a zero if an observation is an outlier so that outliers would be first removed from the CT set in the merging procedure.

The time complexity of the bottom-up algorithm is $O(kn)$ after k number of merges and can be reduced to $O(k \log n)$ if a priority queue is used to maintain the merge measure values in order.

3.4. Sliding Window Algorithm

As in other sliding window algorithms [8], our sliding window algorithm fixes the first timestamp of a given trajectory as the first characteristic timestamp and attempts to place the next CT as far as possible as long as the considering segment remains spatially and temporally homogeneous. When the spatial and temporal homogeneity of the current segment becomes unsatisfied by adding another observation, we separate the segment (without the current observation breaking the homogeneity) from the trajectory and repeat the process with the current observation until the end of the trajectory.

Algorithm 4 RTR-SlidingWindow (S, ε, v_m)

```
1: $n \leftarrow |S|$;  $CT \leftarrow \{t_1\}$;  $startidx \leftarrow 1$;  $length \leftarrow 2$;
2: while $startidx + length \leq n$ do
3:     $curridx \leftarrow startidx + length$;
4:     for $i$=1 to $length$-1 do
5:         $swMeasure(i) \leftarrow$ RobustTDist($S_{startidx,curridx}$,$i$,$v_m$);
6:     if max($swMeasure$) $> \varepsilon$ then
7:         $CT \leftarrow CT \cup \{t_{curridx-1}\}$;
8:         $startidx \leftarrow curridx - 1$;  $length \leftarrow 2$;
9:     else
10:        $length \leftarrow length + 1$;
11: return $CT \leftarrow CT \cup \{t_n\}$;;
```

As shown in Algorithm 4, we first anchor the first timestamp as the first CT in line 1, which is the starting observation of the segment under consideration. Each of the subsequent timestamps is considered in order as the potential candidate for the next CT. To this end, we compute the $slidingwindow$ measure of each observation within the current segment $S_{startidx,curridx}$ using our distance function *RobustTDist*() in lines 4-5. If any of the measures is larger than the given distance threshold ε, i.e., the segment under consideration is not spatially and temporally homogeneous, we add the immediately preceding timestamp into the CT set to separate the segment from the rest and reset the variables as necessary in line 8. Otherwise, we increase the length of the segment under consideration (line 10) and proceed with the next observation. The same procedure (lines 2-10) is repeated until the sequence ends.

The time complexity of the sliding window algorithm is $O(kn)$, where k is the number of inner-iterations required to update the slidngwindow measure of each observation in the increased segment and n is the trajectory size.

4. Performance Evaluation

In this section, we evaluate the effectiveness of our robust time-referenced trajectory segmentation algorithms on three real-world trajectory datasets, tracing the movement of buses [3], trucks [6], and hands [10], respectively. The summary information of the used datasets is shown in Table 1 and the detailed description can be found in [10].

The performance of our segmentation algorithms are compared to those of conventional segmentation techniques on the routes of moving objects; the well-known Douglas-Peucker (DP) algorithm [3,5] (DP-R) and the one-pass approximation algorithm based on the Minimum Description Length (MDL) principal [8] (MDL-R). We also apply these two methods on the trajectories (DP-T and MDL-T), considering time as another spatial dimension.

Table 1. Summary of used datasets

	Bus	Truck	Hand
Dimensions of location	2	2	3
# Trajectories Dataset size	108	273	660
Range of trajectory size	79∼1095	29∼992	65∼693
# Total observations	66,096	112,203	118,664

4.1. Selection of Maximum Speed Value

Our segmentation algorithm requires the maximum speed v_m as an input parameter to determine whether a spatio-temporal observation is indeed an outlier. We employ a heuristic based on the sorted movement speed values to determine a sensible value of v_m. We compute the movement speed between each pair of two consecutive samples from the entire trajectories in the dataset and then sort the speed values. By a visual inspection, we choose an extreme and substantially larger value than its immediate preceding one. In the bus and the truck datasets, we could observe such value around $80km/h$, corresponding to the 99^{th} percentile of the entire speed values. This implies that there exist few outliers in the vehicle datasets. Similarly, we obtained $150cm/s$ for the hand dataset, approximately corresponding to the 92^{th} percentile, which indicates that about 8% of the observations could be outliers. The same speed values are used for v_m throughout the experiments.

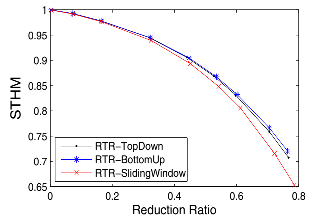
4.2. Experimental Results

In order to measure the spatio-temporal homogeneity presented in the obtained segments, we employ the *spatio-temporal homogeneity measure* ($STHM$) utilizing the statistics of the *normalized* movement speed per segment [10]. This measure attains a value closer to 1 when the movement speed varies less per segment, i.e., the segments are more spatio-temporally homogeneous.

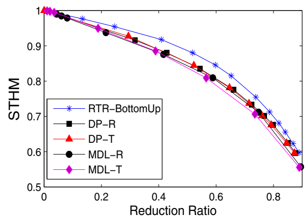
Figure 5 shows the comparative results of this spatio-temporal homogeneity measure. The X axis represent the *reduction ratio*, i.e., the ratio of the total number of non-characteristic observations discarded upon the segmentation to the total number of observations in the dataset, and the Y axis is the spatio-temporal homogeneity measure. The parameter values set for ε are specified in Figure 5(d). Each marker (e.g., denoted as $\times$, $*$, etc) along the performance curve represents the performance obtained by using the corresponding ε in the same order from the left to the right. Recall that the maximum speed parameter v_m is already determined as in Section 4.1.

In Figure 5(a), the spatio-temporal homogeneity retained by RTR-BU and RTR-TD is more or less the same on the bus dataset, which is slightly yet consistently better than RTR-SW. The same results are consistently observed in the other two datasets [10], thus omitted due to the space limitation. However, the relative difference appear to be marginal when the reduction ratio is less than about 0.4.

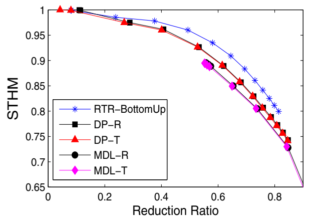
Figures 5(b) and 5(c) present the comparison of our RTR-BU to the conventional approaches in terms of the spatio-temporal homogeneity over the truck and the hand datasets, respectively. RTR-BU is selected because it consistently shows the best results among our approaches. As illustrated, the four conventional methods never beat RTR-BU for all reduction ratios and in all three datasets. This observation shows that RTR-BU identifies the characteristic observations where the spatio-temporal structure significantly changes and the partitioned segments are more spatio-temporally homogeneous. Another interesting observation is that the näive extensions of two conventional techniques simply to the trajectories, DP-T and MDL-T, only attain as similar spatio-temporal homogeneity as those of DP-R and MDL-R (on the routes), respectively. This observation ascertains that semantically, the temporal dimension is indeed different from the spatial dimension, and hence should be incorporated in its own context like we do in our time-referenced distance function. Recall that about 8% of the total observations in hand dataset are considered outliers for v_m=150cm/s, approximately corresponding to the 92th percentile, whereas the vehicle datasets have few outliers. The experiment results in Figure 5(c) show that our RTR algorithms partition trajectories into more temporally homogeneous segments in the presence of outliers.



(a) Bus dataset



(b) Truck dataset



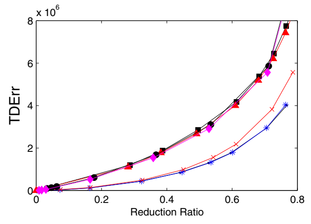
(c) Hand dataset

Dataset	Methods	Parameter Values
Bus Truck	RTR Family	Eps: 0 10 20 40 60 80 100 150 200
	DP-R/T	Eps: 0 5 10 20 40 60 80 100 150 200
	MDL-R/T	MDLc: 0 0.5 1 3 5 6.5 8 10
Hand	RTR Family	Eps: .01 .05 .1 .2 .3 .4 .5 .6 .7 .8 .9 1
	DP-R/T	Eps: .01 .05 .1 .2 .3 .4 .5 .6 .7 .8 .9
	MDL-R/T	MDLc: 0 .01 .05 .1 .5 1 2 3 4 5

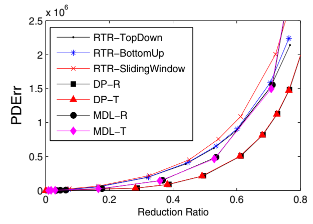
(d) Parameter values

Figure 5. Spatio-temporal homogeneity

In addition, as an attempt to approximate the true spatial discrepancy between a trajectory and a set of its segments, we aggregate the time-referenced distances and the perpendicular distances on every observation. A larger value indicates larger spatial deviation between a trajectory and its segments. Figure 6(a) presents the spatial errors measured by the time-referenced distance (TDerr) on the bus dataset. As expected, the TDerr produced by our RTR family is smaller than those of other approaches. Figure 6(b) shows the spatial error measured by the perpendicular distance (PDerr) on the bus dataset. The DP family that explicitly minimizes the perpendicular distances show the lower PDErr than other methods. Our RTR algorithms yield similar or slightly larger PDErr when the reduction ratio is no larger than 0.4. The same results are consistently observed in the other two datasets [10]. In sum, the results of two spatial discrepancy measures indicate that our RTR segmentation algorithms maintain a comparable spatial homogeneity to the conventional techniques focusing only on the spatial relationships.



(a) Time-referenced distance error on bus dataset



(b) Perpendicular distance error on bus dataset

Figure 6. Spatial error comparison

5. Conclusion

In this paper, we proposed three robust time-referenced trajectory segmentation algorithms that take into account both spatial and temporal structures presented in trajectory. The proposed segmentation methods employ our time-referenced distance function to partition a given trajectory into a small number of spatially and temporally homogeneous segments. In addition, we utilize the maximum movement speed to detect time-referenced spatial outliers. Our experiments on three real-world datsets indicate that our techniques outperform the conventional segmentation techniques in terms of spatio-temporal homogeneity and maintain comparable spatial homogeneity.

References

[1] C. C. Aggarwal and P. S. Yu. Outlier detection for high dimensional data. In *SIGMOD '01*, pages 37–46, 2001.
[2] E. Bingham, A. Gionis, N. Haiminen, H. Hiisilä, H. Mannila, and E. Terzi. Segmentation and dimensionality reduction. In *SIAM Data Mining Conf.*, pages 371–383, 2006.
[3] H. Cao, N. Mamoulis, and D. W. Cheung. Mining frequent spatio-temporal sequential patterns. In *ICDM '05*, pages 82–89, 2005.
[4] L. Chen, M. T. Özsu, and V. Oria. Robust and fast similarity search for moving object trajectories. In *SIGMOD*, pages 491–502, 2005.
[5] D. H. Douglas and T. K. Peucker. Algorithm for the reduction of the number of points required to represent a line or its caricature. *Cartographica: The Int. J. for Geographic Inf. and Geovisualization*, 10(2):112–122, 1973.
[6] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi. Trajectory pattern mining. In *KDD '07*, pages 330–339, 2007.
[7] H. Imai and M. Iri. Computational-geometric methods for polygonal approximations of a curve. *Comput. Vision Graph. Image Process.*, 36(1), 1986.
[8] J.-G. Lee, J. Han, and K.-Y. Whang. Trajectory clustering: a partition-and-group framework. In *SIGMOD*, pages 593–604, 2007.
[9] G. Papakonstantinou, P. Tsanakas, and G. Manis. Parallel approaches to piecewise linear approximation. *Signal Proc.*, 37(3):415–423, 1994.
[10] H. Yoon and C. Shahabi. Robust time-referenced segmentation of moving object trajectories. Technical report, Univ. of Southern California, 2008.