International Cartographic Association
Association Cartographique Internationale

Semantically Enriched Simplification of Trajectories

Rajesh Tamilmani [a, *], Emmanuel Stefanakis [b]

[a] *Geodesy and Geomatics Engineering, University of New Brunswick, rtamilma@unb.ca*
[b] *Department of Geomatics Engineering, University of Calgary, emmanuel.stefanakis@ucalgary.ca*

* Corresponding author

Abstract: Moving objects that are equipped with GPS devices generate huge volumes of spatio-temporal data. This spatial and temporal information is used in tracing the path travelled by the object, so called trajectory. It is often difficult to handle this massive data as it contains millions of raw data points. The number of points in a trajectory is reduced by trajectory simplification techniques. While most of the simplification algorithms use the distance offset as a criterion to eliminate the redundant points, temporal dimension in trajectories should also be considered in retaining the points which convey both the spatial and temporal characteristics of the trajectory. In addition to that the simplification process may result in losing the semantics associated with the intermediate points on the original trajectories. These intermediate points can contain attributes or characteristics depending on the application domain. For example, a trajectory of a moving vessel can contain information about distance travelled, bearing, and current speed. This paper involves implementing the Synchronized Euclidean Distance (SED) based simplification to consider the temporal dimension and building the Semantically Enriched Line simpliFication(SELF) data structure to preserve the semantic attributes associated to individual points on actual trajectories. The SED based simplification technique and the SELF data structure have been implemented in PostgreSQL 9.4 with PostGIS extension using PL/pgSQL to support dynamic lines. Extended experimental work has been carried out to better understand the impact of SED based simplification over conventional Douglas-Peucker algorithm to both synthetic and real trajectories. The efficiency of SELF structure in regard to semantic preservation has been tested at different levels of simplification.

Keywords: trajectory, line simplification, semantics, synchronous euclidean distance

1. Introduction

Over the years, the technology developments have enabled the usage of GPS devices in moving objects. These devices generate streams of points (locations) which form a path travelled by the moving object during a particular period of time. This traced path is known as trajectory. Trajectory data is commonly utilized in urban planning, fleet management systems, and other location-based service applications. With every trajectory containing enormous amount of data points, it is often required to reduce the data according to the application domain. The concept of trajectory reduction has evolved from the algorithms used in cartographic generalization for linear geometric features also known as simplification (Keates 1989). The basic idea is to retain certain points which are more significant in forming the trajectory than other points as they better convey the trajectory characteristics for a particular context. For example, the point at which a sudden speed change occurs is more important than other points in vehicle movement tracking. The conventional generalization techniques for linear features (e.g., rivers, pipelines, and roads) remove the high-density vertices based on a given criterion.

The Douglas-Peucker (DP) algorithm is a recursive approach for simplifying lines which takes the original linear geometry and a threshold distance as input. The simplified version is generated by controlling the offset while minimizing the distortion. At the end of a recursive process, only a subset of the vertices is retained to form the simplified geometry. The resultant geometry ends up in reduction in length (Douglas et al., 1973). DP simplification algorithm does not consider temporal dimension (time) associated with the vertices of the trajectories. Furthermore, as a result of simplification the semantics (e.g. speed, heading and distance travelled) at each point on the original line (trajectory) are not preserved in the simplified line. As a result, Douglas-Peucker algorithm has limited scope to be utilized in trajectory simplification. For example, in Fig.1, only the first and last points of original line are retained in the simplified line for a 50-meter threshold distance, because none of the perpendicular offset is greater than 50 meters regardless the temporal data associated with the intermediate points. Depending on the threshold distance some intermediate points can also be retained using the Douglas-Peucker algorithm.

As DP algorithm has the limitation of not being able to consider the temporal dimension of a trajectory, the notion of the Synchronous Euclidean Distance (SED) was introduced by Meratnia and de By (2004).

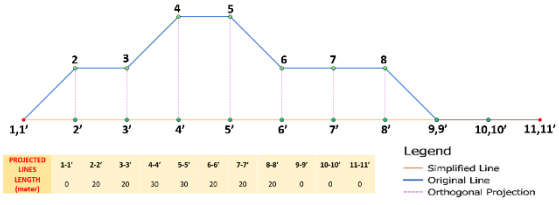


Figure 1: Comparison of original and the simplified version when DP-threshold is 50 meters

 This contribution underwent single-blind peer review based on submitted abstracts. https://doi.org/10.5194/ica-proc-2-128-2019 | © Authors 2019. CC BY 4.0 License.

This paper presents an implementation of DP with the notion of SED and combining it with SELF (Semantically Enriched Line simpliFication) data structure for dynamic linear features that preserves the semantic attributes (speed, heading and distance travelled) associated with individual locations of original trajectory. These attributes are associated with the DP-SED based simplified trajectory as an array of values corresponding to multiple locations along the simplified trajectory (Stefanakis 2015).

The overall objectives of this research work are:

1. To implement SED based trajectory simplification technique to consider spatio-temporal data in trajectory generalization
2. To implement the SELF structure to support dynamic polylines and to test with both synthetic and real-world features.
3. To compare the interpolated semantic values using SELF structure at different levels of trajectory generalization

2. Literature Review

2.1 Trajectory Generalization

Over the years the usage of GPS devices in mobility vehicles have increased exponentially and massive amount of data is being generated by these devices. The generated data is used in various public and business applications such as urban transportation planning, fleet management and traffic modelling (K. Buchin et al., 2008). The enormous volume of data makes it impossible to analyse the data manually. For example, during the trip length of 30 minutes, if the location is being recorded for every 5 seconds a total of 360 points are recorded. In a day, the dataset contains 17,280 points. It necessitates to identify the methods for reducing the complexity of the dataset. The concept of reducing a trajectory dataset is called trajectory reduction or simplification. The idea of trajectory reduction has evolved from cartographic generalization. Simplification, the common cartographic generalization technique, has been a key research area for cartographers over the years (Cromley 1991, Weibel 1997, Robinson et al. 2005, Wu et al. 2003).

The Douglas-Peucker (DP) algorithm has been revamped by many researchers since it was introduced in 1973. The problem of topological inconsistency between original and simplified geometry produced by DP algorithm was addressed by avoiding self-intersections on the simplified geometry (Wu et al., 2003). The problem of limited data storage space is addressed by experimenting the line simplification algorithms in a streaming environment (Abam 2010). Various techniques have been proposed to enforce the topological constraints while simplifying a polyline (Shahriari and Tao 2002, Tienaah et al., 2015, QiuLei et al., 2016).

Meanwhile, enriching the content of linear features has gained attention to address the problem of annotating trajectories with semantic data. (Alvares et al., 2007, Yan et al., 2011, Richter et al., 2012, Parent et al., 2013, Stefanakis 2015). The trajectory sample points have been transformed into stops and moves by adding semantic information (Alvares et al., 2007). Though the implemented model has shown significant compression of trajectories while enabling efficient query processing, the pre-processing of adding semantic information to trajectories is a time-consuming operation. The semantic enrichment platform SeMiTri, multi-tiered approach, was presented to handle heterogeneous trajectories (includes both fast and slow-moving objects). The trajectory generalization platform based on Hidden Markov Model (HMM) technique has not considered trajectories in large scale (Yan, Z., et al., 2011). Richter et al., (2012) extended concepts of network-constrained indexing in mobility object to embed human movement with the individual locations on the trajectory. The algorithm they proposed enables a user to determine the reference point and all possible movement change descriptions from that point but is limited only to urban transport network.

The problem of spatial relation violation while compressing the trajectories was ad-dressed to maintain disjoint topological relation and direction relations between the original and generalized trajectory (Stefanakis 2012). The author has extended DP algorithm to maintain the topological consistency between the trajectory and its simplified version.

2.2 Synchronous Euclidean Distance (SED)

Most of the simplification algorithms are suitable for generalizing linear geometries. The data points are retained only based on the perpendicular distance between data points and the pro-posed generalized version of it. While these algorithms can also be applied on trajectory datasets, using the perpendicular distance as a criterion becomes inappropriate as trajectories are not just linear geometries. Trajectories represent historical trace of points by associating temporal dimension with spatial data. With the above idea, the notion of the Synchronous Euclidean Distance (SED) was introduced to achieve reduction of trajectories while retaining the spatio temporal characteristics of the trajectory (Meratnia and de By 2004). The author has implemented and tested the DP-SED algorithm (extension of Douglas-Peucker algorithm with the notion of SED). The proposed algorithm retains the spatiotemporal characteristics while reducing the trajectories efficiently.

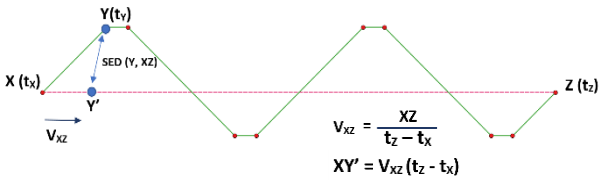


Figure 2: The Synchronous Euclidean Distance (Sed).

In Fig.2, the locations X, Y, Z represent the position of a moving vessel at the timestamps t_X, t_Y, t_Z where $t_X < t_Y < t_Z$. The spatiotemporal footprint of Y (i.e. Y') is calculated with respect to the velocity of trip V_{XZ}. The Euclidean distance YY' is known as the SED for the point Y. The perpendicular distance applied in DP algorithm is lesser or

equal to the SED (YY') as the line YY' is not perpendicular to the straight line XZ.

2.3 Semantically Enriched Line Simplification (SELF)

On the one hand, efficient generalization of trajectories can be achieved by DP-SED algorithm while retaining the spatiotemporal characteristics of the trajectory. However, the generalized version does not retain the semantics associated with the individual points on the original trajectory. SELF data structure has been introduced by Stefanakis (2015) to enrich the simplified line to convey some semantics associated with the original version. In the attempt of enriching the content of the linear geometries while reducing the number of points, SELF data structure is proposed to preserve the attributes of the original line or any semantic annotations associated with individual locations or segments of that line (Spaccapietra et al., 2008) into the generalized version (Stefanakis 2015). The author has proposed many variations of SELF and the choices can be made based on how rich the semantics attached to the simplified line are.

The *basic variant* of SELF attaches the original line length (e.g., kilometric travel distance) to the simplified line. In this variant, a line with end points 1 (start), n (end), and total length (d_n) will be represented by a simplified line defined as follows (Stefanakis 2015):

$[x_1, y_1, x_n, y_n, d_n]$ (SELF variant: basic)

An *advanced variant* for function lines will also tag the accumulated length per vertex along the line. Hence, each vertex K of the original line will orthogonally be projected on the simplified line (Fig. 1) and the footprint point K' will be assigned the accumulated length d_k from point 1 (start) to vertex K along the original line. If $d_{k'}$ is the Euclidean distance of point K′ from end point 1, the simplified line will be represented as follows (Stefanakis 2015):

$[x_1, y_1, x_n, y_n, d_n,$ ARRAY $\{(d_k', d_k); k=2, \ldots, n-1\}]$
(SELF variant: advanced-function)

In this paper, the advanced variant of SELF structure has been extended to tag trajectory semantics: speed, heading, and distance travelled. Each point on the original trajectory is projected on the generalized version based on SED. The footprint of each point will be assigned with speed, heading and distance travelled at that point.

$[x_1, y_1, x_n, y_n, d_n,$ ARRAY $\{(d_k', \text{speed, heading}, d_k); k=1, \ldots, n\}]$ (SELF variant: dynamic lines)

3. Methodology

3.1 SED Simplification

Trajectories are formed by connecting series of raw mobility data points. These individual data points include the spatiotemporal locations (latitude, longitude, time). The simplified line using Douglas-Peucker algorithm with the user defined threshold always considers perpendicular distance as a criterion to eliminate the redundant points.

The goal of the SED based simplification is to also consider temporal dimension of trajectory data while generalizing the trajectory.

The algorithm is divided into four steps:

1. Constructing the trajectory using the individual points (Trajectory Reconstruction)
2. Calculating the average velocity of the trip
3. Identify the corresponding SED point for every point on the original line
4. Removing the points by comparing the SED against the simplification threshold

3.1.1 *Constructing the trajectory using the individual points (Trajectory Reconstruction)*

Raw points ordered by the timestamp are connected sequentially to form the trajectory. These data points are in the form (time, latitude, longitude). Ten (10) points along with their corresponding attributes are given in Table 1; their individual locations are mapped in Figure 3 and the constructed trajectory is shown in Figure 4.

Table 1: Attribute table for the trajectory points in Fig.3

ID	SPEED (in KNOTS)	LOCATION	TIME	HEADING
1	0	POINT (482980 4101964)	2017-05-23 01:00:00.000	511
2	233.261	POINT (483010 4101994)	2017-05-23 01:00:01.000	400
3	233.261	POINT (483020 4101994)	2017-05-23 01:00:02.000	211
4	233.261	POINT (483070 4101944)	2017-05-23 01:00:03.000	400
5	233.261	POINT (483080 4101944)	2017-05-23 01:00:04.000	511
6	233.261	POINT (483130 4101994)	2017-05-23 01:00:05.000	400
7	233.261	POINT (483140 4101994)	2017-05-23 01:00:06.000	211
8	77.7538	POINT (483190 4101944)	2017-05-23 01:00:08.000	400
9	77.7538	POINT (483200 4101944)	2017-05-23 01:00:10.000	511
10	77.7538	POINT (483220 4101964)	2017-05-23 01:00:12.000	0

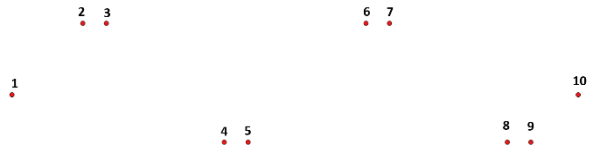


Figure 3: Individual Points on the trajectory

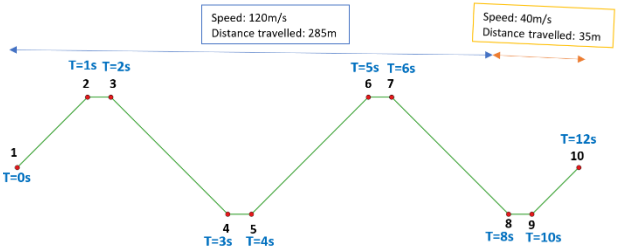


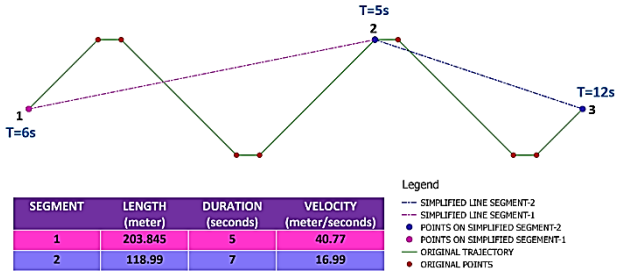
Figure 4: Formed trajectory after connecting individual points

3.1.2 *Calculating the average velocity of the trip*

Average velocity of the trip is defined as the ratio between the straight-line length of the trip and the total duration of the trip.

In case of multiple segments connecting the starting and ending points, the average velocity is calculated for each segment and retrieved as an array of numeric values (Fig. 5).

Proceedings of the International Cartographic Association, 2, 2019.
29th International Cartographic Conference (ICC 2019), 15–20 July 2019, Tokyo, Japan. This contribution underwent single-blind peer review based on submitted abstracts. https://doi.org/10.5194/ica-proc-2-128-2019 | © Authors 2019. CC BY 4.0 License.



SEGMENT	LENGTH (meter)	DURATION (seconds)	VELOCITY (meter/seconds)
1	203.845	5	40.77
2	118.99	7	16.99

Figure 5: Average velocity for multi segment line

3.1.3 *Identify the corresponding SED point for every point on the original line*

For each point on the original line the corresponding SED point is identified by calculating the distance from starting point to the SED point.

Distance to SED point = Average Velocity * Time of travel at the original point

Table 2: Distance to each SED point on straight line (Fig.6)

Point ID	1	2	3	4	5	6	7	8	9	10
SED Points	1	2’	3’	4’	5’	6’	7’	8’	9’	10
Time (Sec)	0	1	2	3	4	5	6	8	10	12
Distance to SED point (metre)	0	20	40	60	80	100	120	160	200	240

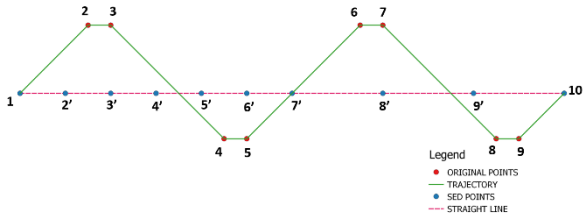


Figure 6: Individual points on the trajectory and the corresponding SED points on straight line

3.1.4 *Removing the points by comparing SED distance against simplification threshold*

SED based simplification algorithm takes the set of points ordered by timestamp and a thresh-old distance as input. The recursive algorithm divides the trajectory based on the SED against the threshold distance. Once the corresponding locations of all the points on the original trajectory are identified on the straight line connecting the first to the last point, the algorithm finds all points for which the SED is longer than threshold distance. The point with longest SED is marked to be retained for the next iteration. For the next iteration, two straight line segments are compared against original trajectory. When there are no points found with an SED longer that the threshold, the algorithm terminates. Fig. 7 demonstrates the SED based simplification algorithm in generalizing a trajectory. In the iteration 1, for each point on the original trajectory its corresponding SED projection on the straight line connecting 1 and 10 is found. Point 6 has the maximum SED and greater than the threshold (30 meters). So, point 6 is retained. For the next iteration, the intermediate simplified line contains two segments 1-6 and 6-10. Again, for each original point its corresponding SED projection point is found on the intermediate simplified version. This time point 5 has the maximum SED (>30 meters). At the end of second iteration the intermediate simplified line contains three segments 1-5, 5-6, 6-10. The iteration continues until there are no points to be retained. In this case, the algorithm terminates in the 6th iteration as then none of the points have an SED greater than the threshold (30 meters).

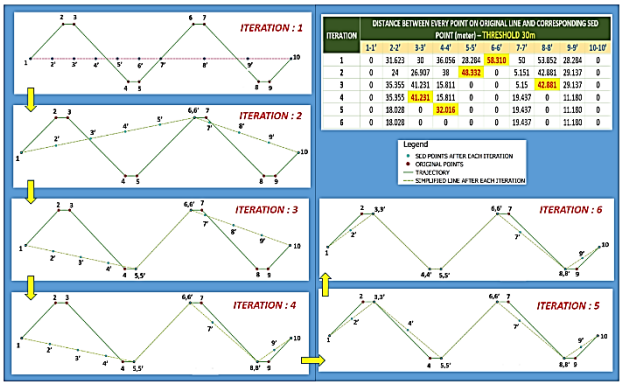


Figure 7: The SED based simplification algorithm

3.2 Building SELF structure based on SED simplification

The goal of the SELF structure is to retain the semantics at each point on the original trajectory along with the corresponding SED point on the generalized version.

The algorithm is divided into four steps.

1. Finding the SED projection for each point on the original trajectory on the simplified trajectory
2. Calculating the accumulated distance at each point on the original trajectory and its SED projection point on the generalized trajectory
3. Remove the individual points based on the change in speed and heading
4. Interpolation of the semantics on the original trajectory at any point on the generalized version

3.2.1 *Finding the SED projection for each point on the original trajectory on the simplified trajectory*

Each point on the original trajectory is projected on the generalized version based on SED.

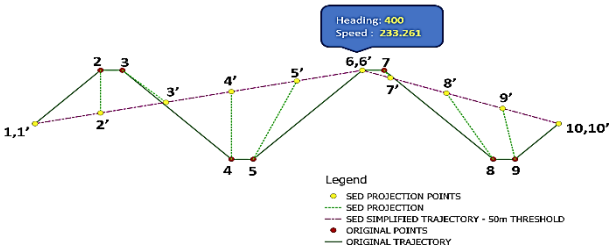


Figure 8: SED projection of each point on the simplified trajectory

3.2.2 *Calculating the accumulated distance at each point on the original trajectory and its SED projection point on the generalized trajectory*

SELF structure for dynamic lines stores the semantics associated with individual points on the original trajectory along the generalized version. As shown in Fig.8, each point on the original trajectory is projected based on the SED on the generalized version. For each point on the original trajectory, the corresponding SED point is tagged with the semantics. The entire SELF structure is represented as follows:

[**SPOINT (482980 4101964), EPOINT (483220 4101964),**
322.843, {(0.000,0,511,0.00),
(30.594,233.261,400,42.426), (61.188,233.261,211,52.426),
(91.782,233.261,400,123.137),
(122.376,233.261,511,133.137), (152.971,233.261,400,203.848),
(166.523,233.261,211,213.848), (193.628,77.7538,400,284.559),
**(220.734,77.7538,511,294.559), (247.839,77.7538,0,322.843)}]

whereas, the semantic array contains the values in the order (accumulated length on the generalized version, speed, heading, accumulated length on the original trajectory).

Table 3: Semantics at each locations of original and generalized trajectory in Fig. 8

POINT ON ORIGINAL TRAJECTORY	1	2	3	4	5	6	7	8	9	10
ACCUMULATED DISTANCE ON ORIGINAL TRAJECTORY (m)	0.0	42.43	52.43	123.14	133.14	203.85	213.85	284.60	294.60	322.84
CORRESPONDING POINT ON SIMPLIFIED TRAJECTORY	1′	2′	3′	4′	5′	6′	7′	8′	9′	10′
ACCUMULATED DISTANCE ON SIMPLIFIED TRAJECTORY (m)	0.0	30.59	61.19	91.78	122.38	152.97	166.52	193.63	220.73	247.84
HEADING	511	400	211	400	511	400	211	400	511	0
SPEED (in KNOTS)	0	233.26	233.26	233.26	233.26	233.26	233.26	77.75	77.75	77.75

3.2.3 *Semantic based Compression Levels*

SELF structure generates a large volume of data which is proportional to the number of points in the original trajectory. The number of points to be stored in SELF structure can be diminished by applying a semantic based compression: (a) Speed based compression (b) Heading based compression. These methods are described in following sub-sections.

3.2.3.1 Speed based compression

If the ratio of speed between two consecutive points on the original trajectory is less than the given threshold, then the semantics at second point is not stored.

In Fig.9 the semantic at point 7 is not stored as the ratio of speed between the points 6 and 7 is zero.

$$\frac{\text{Speed at point '6'} - \text{Speed at point '7'}}{\text{Speed at point '6'}} \times 100\ \%$$

3.2.3.2 Heading based compression

In case of heading based compression, the semantics of the points are not stored if the ratio of heading between two consecutive points is less than the threshold.

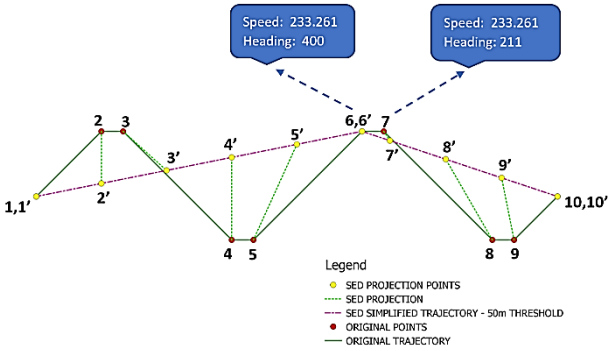


Figure 9: SED projection of each point on the simplified trajectory and the semantics

In Fig.9 the heading-based compression ratio between points 6 and 7 is 47.25 %. The semantics at point 7 will not be stored in SELF structure when the heading-based compression ratio applied as 50.0 %.

Table 4: Speed and heading based compression ratio between points (Data Source: Table 1)

POINTS	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10
SPEED BASED COMPRESSION RATIO	100.0	0.0	0.0	0.0	0.0	0.0	66.67	0.0	0.0
HEADING BASED COMPRESSION RATIO	21.72	47.25	89.57	27.75	21.72	47.25	89.57	27.75	100.0

The self structure after applying both the speed based and heading based compression thresh-olds as (10.0, 10.0) for trajectory data given in Table 1 is:

[**SPOINT (482980 4101964), EPOINT (483220 4101964),**
322.843, {(0.000,0,511,0.00),
(30.594,233.261,400,42.426), ~~(61.188,233.261,211,52.426),~~
~~(91.782,233.261,400,123.137),~~
~~(122.376,233.261,511,133.137),~~ ~~(152.971,233.261,400,203.848),~~
~~(166.523,233.261,211,213.848),~~ (193.628,77.7538,400,284.559),
~~(220.734,77.7538,511,294.559),~~ (247.839,77.7538,0,322.843)}]

3.2.4 *Interpolation of the semantics on the original trajectory at any point on the generalized version*

The SELF structure built using algorithm 6 can be used to interpolate the semantics on the original trajectory at any point on the generalized version of it. In Fig.10, the semantics at 'P' can be calculated by applying a linear interpolation on the segment defined by the projection of vertices '8' and '9' on the simplified trajectory. The algorithm 7 is used for computing the semantics at P.

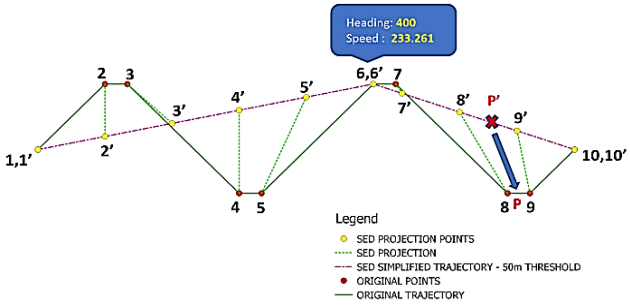


Figure 10: P' is the SED Projection of P

4. Implementation

SED based simplification algorithm and SELF data structure have been implemented in PostgreSQL 9.4 using PL/pgSQL. The spatial extension POSTGIS 2.3 has been installed in PostgreSQL 9.4. The implemented algorithm takes a set of raw mobility points and simplification threshold as input. The simplified version is then associated with the SELF data structure. The user can select any point on the simplified trajectory, to retrieve the original semantics. The experiments were performed on a sea vessel trajectory dataset obtained in Aegean Sea, Greece. The built-in functions available with PostGIS extension that were utilized for implementing SED based simplification algorithm and developing the SELF data structure. Using PL/pgSQL, the procedural language for PostgreSQL, both the SED based simplification and SELF structure algorithms were added as new (user defined) functions. Eleven new functions were implemented.

4.1 Experimental Data

To demonstrate the effectiveness of the SED based simplification and SELF structure in interpolating the semantics, experimentation is done on different trajectory datasets with different values for speed based and heading based simplification. The experiments ran over the vessel trajectories for August 2013 in the Aegean Sea as collected by the MarineTraffic Automatic Identification System (AIS) (MarineTraffic 2017). In order for the set of features to be representative for a wide range of spatiotemporal characteristics, it was decided to choose trajectories with different number of mobility data points. TR1 in Table 5 refers to the trajectory in Fig. 4

Table 5: Length and the number of points available in the selected trajectory dataset

TRAJECTORY NAME	LENGTH (in meters)	TOTAL NUMBER OF POINTS	STARTING TIME	ENDING TIME
TR1	322.842	10	"2017-05-23 01:00:00"	"2017-05-23 01:00:12"
TR2	40933.052	55	"2013-08-08 02:08:00"	"2013-08-08 06:01:00"
TR3	1032677.072	500	"2013-08-01 00:02:00"	"2013-08-02 15:17:00"
TR4	6961025.269	5030	"2013-08-01 00:04:00"	"2013-08-31 21:42:00"
TR5	17207258.378	8553	"2013-08-01 00:02:00"	"2013-08-31 23:39:00"

4.2 Experiments

4.2.1 *SED based Simplification*

The SED based simplified version and the original trajectory is shown in Fig.11 with a simplification threshold of 90.0 meters.

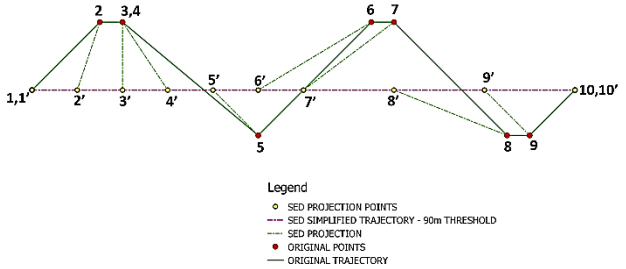


Figure 11: Original and simplified version of a trajectory

Table 6: Attribute table of the trajectory in Fig. 11

ID	SPEED (in KNOTS)	LOCATION	TIME	HEADING
1	0	POINT (482980 4101964)	2017-05-23 01:00:00.000	511
2	233.261	POINT (483010 4101994)	2017-05-23 01:00:01.000	400
3	0	POINT (483020 4101994)	2017-05-23 01:00:02.000	0
4	233.261	POINT (483020 4101994)	2017-05-23 01:00:03.000	211
5	233.261	POINT (483080 4101944)	2017-05-23 01:00:04.000	511
6	233.261	POINT (483130 4101994)	2017-05-23 01:00:05.000	400
7	233.261	POINT (483140 4101994)	2017-05-23 01:00:06.000	211
8	77.7538	POINT (483190 4101944)	2017-05-23 01:00:08.000	400
9	77.7538	POINT (483200 4101944)	2017-05-23 01:00:10.000	511
10	77.7538	POINT (483220 4101964)	2017-05-23 01:00:12.000	0

In Fig.11, the locations '3' and '4' represent the same point as the vessel has stopped at location '3' and stayed there for a minute before leaving. Even though the locations '3' and '4' are same, their SED projections are different. The cause is due to the points '3' and '4' has different timestamps.

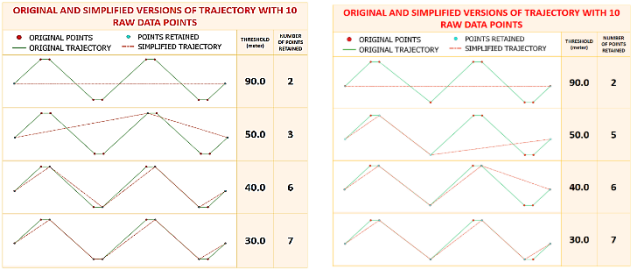


Figure 12: Comparison of original and simplified trajectory at different levels of simplification

Comparing the number of points retained by SED-DP simplification with DP simplification indicates that SED-DP simplification always retains more number of points than DP simplification (Fig. 13).

Table 7: Length and number of points in each data

TRAJECTORY NAME	THRESHOLD (meters)	NUMBER OF POINTS RETAINED		PERCENTAGE INCREASE/DECREASE IN RETAINED POINTS
		DP	SED-DP	
TR1(10 POINTS)	90	2	2	0.00
	50	2	3	33.33
	30	2	7	71.43
TR2 (55 POINTS)	5000	3	4	25.00
	1000	7	12	41.67
	500	8	29	72.41
TR3 (500 POINTS)	50000	4	7	42.86
	10000	15	19	21.05
	5000	17	27	37.04
TR4 (5030 POINTS)	500000	2	2	0.00
	200000	13	14	7.14
	300000	10	9	-11.11
TR5 (8553 POINTS)	2000000	2	4	50.00
	3000000	2	4	50.00
	5000000	2	4	50.00

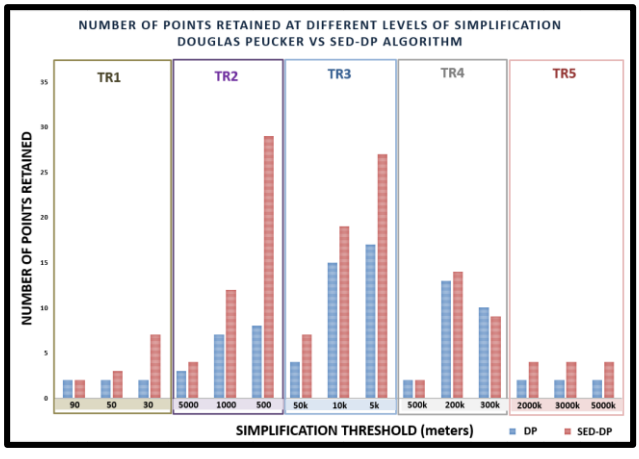


Figure 13: Comparing number of points retained

4.2.2 *SELF structure*

The SELF structure has been built on the original trajectory shown in Fig.11 with an SED based simplification threshold of 90.0 meters and both the speed and heading based compression levels set as 0. The original semantics at each point on the simplified trajectory are listed in Fig. 14

POINT ON ORIGINAL TRAJECTORY	1	2	3	4	5	6	7	8	9	10
ACCUMULATED DISTANCE ON ORIGINAL TRAJECTORY (m)	0.0	42.43	52.43	52.43	130.53	201.24	211.24	281.95	291.95	320.23
CORRESPONDING POINT ON SIMPLIFIED TRAJECTORY	1′	2′	3′	4′	5′	6′	7′	8′	9′	10′
ACCUMULATED DISTANCE ON SIMPLIFIED TRAJECTORY (m)	0.0	20.0	40.0	60.0	80.0	100.0	120.0	160.0	200.0	240.0
HEADING	511	400	211	400	511	400	211	400	511	0
SPEED (in KNOTS)	0	233.26	0	233.26	233.26	233.26	233.26	77.75	77.75	77.75

Figure 14: Semantics at each points of original trajectory and its simplified version in Fig. 11

Table 8: Interpolated semantics at each point clicked on the simplified trajectory

POINT CLICKED ON THE SIMPLIFIED TRAJECTORY	1′	2′	3′	4′	5′	6′	7′	8′	9′	10′
INTERPOLATED DISTANCE	0.0	42.43	52.43	52.43	130.53	201.24	211.24	281.95	291.95	320.23
INTERPOLATED HEADING	511	400	211	400	511	400	211	400	511	0
INTERPOLATED SPEED (in KNOTS)	0	233.26	0	233.26	233.26	233.26	233.26	77.75	77.75	77.75

In Fig. 11 the projected points 3’ and 4’ correspond to the same location (points 3 and 4) on the original trajectory as the original accumulated distance on original trajectory at the points 3 and 4 is 52.43 meters.

There are three possible outcomes when running the algorithm to interpolate the semantics on the original trajectory at any points of the simplified version from SELF structure. The type of error in interpolation is classified depending on the outcome given in Table 9.

Table 9: Error classification based on possible outcome

POSSIBLE OUTCOME	ERROR CLASSIFICATION
Interpolated Semantics > Actual Semantics	Negative
Interpolated Semantics < Actual Semantics	Positive
Interpolated Semantics = Actual Semantics	Zero

Speed based compression and heading based compression produce almost the same amount of error in the interpolated semantics. The different spatio-temporal characteristics of the datasets play major role in semantic interpolation error.

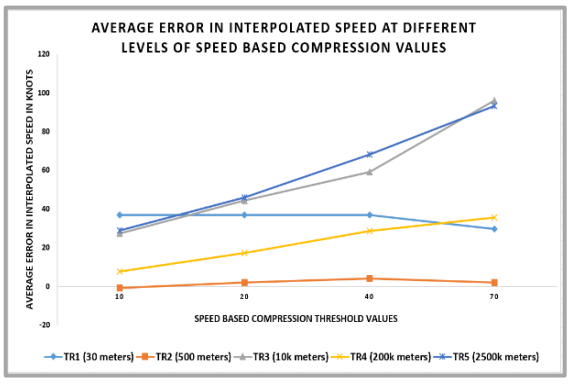


Figure 15: Error in interpolated speed

It can be seen from Fig.15-18 that there is a positive correlation between average error and the level of simplification.

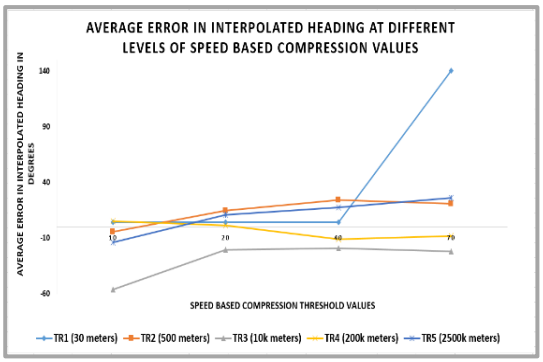


Figure 16: Error in interpolated heading

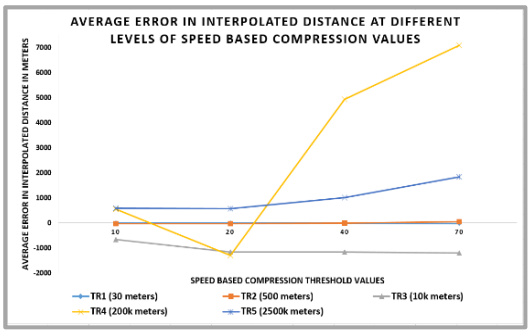


Figure 17: Error in interpolated distance

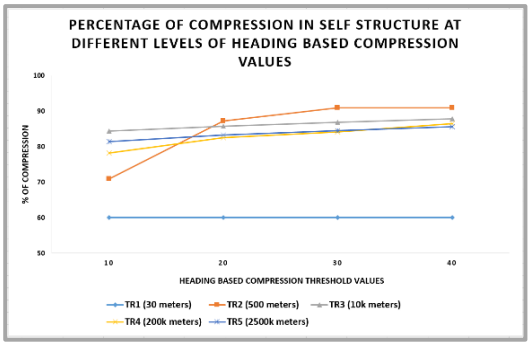


Figure 18: % of compression VS heading based compression values

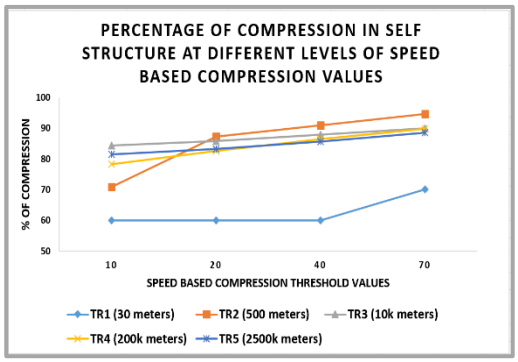


Figure 19: % of compression VS speed based compression values

The percentage error in interpolation increases when the values for compression is also increased. Noticeably, average error in distance interpolation for “TR4” suddenly increases after 20% of speed-based compression. The cause is due to an increase in compression level discards more points from SELF structure (Fig.18, 19). But this number would change due to the different spatiotemporal

Proceedings of the International Cartographic Association, 2, 2019.
29th International Cartographic Conference (ICC 2019), 15–20 July 2019, Tokyo, Japan. This contribution underwent single-blind peer review based on submitted abstracts. https://doi.org/10.5194/ica-proc-2-128-2019 | © Authors 2019. CC BY 4.0 License.

characteristics of the datasets. So, the level of compression can be decided based on the application and the required accuracy in semantics interpolation.

5. Conclusions

This paper summarizes the implementation and testing of a method for semantically enriched simplification of trajectories. The method combines the Synchronized Euclidean Distance (SED) based simplification algorithm and SELF (Semantically Enriched Line simpliFication) data structure to preserve the semantics associated with the actual trajectories. The method has been implemented in PostgreSQL 9.4 with PostGIS extension using PL/pgSQL to support dynamic lines and tested with both synthetic and real-world features.

The method applies two kinds of semantic based reduction: speed based and heading based. Both the compression techniques produce the same amount error in the interpolated semantics. However, the results of the experiments indicate that the different spatio-temporal characteristics of the datasets play a major role in the semantic interpolation error. The comparison results between SED-DP simplification and DP simplification indicate that SED-DP simplification always retains more number of points which are more significant in forming the trajectory than other points as they better convey the trajectory characteristics for a particular context.

Future work includes the development of a visualization framework to provide an enhanced user experience. This will help in facilitating the adoption of the SELF structure in various application domains with need for semantically enhanced multiscale representation of linear features. Integrating these libraries to a Graph database (such as Neo4j) so that the extended functionalities of Graph database can be utilized in trajectory data management is another future goal.

6. References

Abam, M.A., et al., (2010). Streaming algorithms for line simplification. Discrete & Computational Geometry, 43 (3), 497–515. doi:10.1007/s00454-008-9132-4

Alvares, L.O., et al., (2007). A model for enriching trajectories with semantic geographical information. In: The Proceedings of the 15th annual ACM international symposium on advances in geographic information systems, 7–9 November, Seattle, WA. Article No. 22.

Cromley, R.G., (1991). Hierarchical methods of line simplification. Cartography and Geographic Information Science, 18 (2), 125–131. doi:10.1559/152304091783805563

Douglas, D.H. and Peucker, T.K., (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. The Canadian Cartographer, 10 (2), 112–122. doi:10.3138/FM57-6770-U75U-7727

MarineTraffic, (2017). Live-ships map: AIS [online]. Available from: http://www.marinetraffic.com/ais/ [Last visited, June 27, 2017]

K. Buchin, M. Buchin, and J. Gudmundsson (2008). Detecting single file movement. In GIS '08: Proceedings of the 16thACM SIGSPATIAL international conference on Advances in geographic information systems, pages 1-10, New York, NY, USA, ACM.

Keates, J.S., (1989). Cartographic design and production. 2nd ed. Essex: Longman.

Meratnia, N. and de By, R.A., (2004). Spatiotemporal compression techniques for moving point objects. In: Proceedings of the international conference on extending database technology (EDBT). Berlin: Springer, 765–782. LNCS 2992.

Parent, C. et al., (2013). Semantic Trajectories Modeling and Analysis. ACM Computing Surveys, Vol. 45, No.4, Article 42. doi: http://dx.doi.org/10.1145/2501654.2501656

QiuLei Guo and Hassan A. Karimi, (2016). A Topology-Inferred Graph-Based Heuristic Algorithm for Map Simplification. In: Transactions in GIS, doi:10.1111/tgis.12188

Richter, K.F., Schmid, F., and Laube, P., (2012). Semantic trajectory compression: repre-senting urban movement in a nutshell. Journal of Spatial Information Science, (4), 3–30.

Robinson, Joel L. Morrison, Phillip C. Muehrcke, A. Jon Kimerling, Stephen C. Guptill, Elements of Cartography, 6th Edition ISBN: 978-0-471-55579-7

Shahriari, N., and V. Tao, (2002). Minimizing Positional Errors in Line Simplification Us-ing Adaptive Tolerance Values. In: Symposium on Geospatial Theory, Processing and Application, 4(3), 213-223.

Spaccapietra, S., et al., (2008). A conceptual view on trajectories. Data & Knowledge Engineering, 65 (1), 126–146. Retrieved from : http://www.sciencedirect.com/science/article/pii/S0169023X07002078

Stefanakis, E., (2012). Trajectory generalization under space constraints. In: The proceedings of the 7th international conference on geographic information science (GIScience 2012), 18–21 September, Columbus, OH.

Stefanakis, E., (2015). SELF: Semantically enriched Line simpliFication. In: International Journal of Geographical Information Science, Vol. 29, Iss. 10, 2015, Pages 1826-1844 doi: 10.1080/13658816.2015.1053092

Tienaah, T., Stefanakis, E., and Coleman, D., (2015).S Contextual Douglas-Peucker simplification. Geomatica Journal, 69 (3), https://doi.org/10.5623/cig2015-306

Weibel, R., (1996). A typology of constraints to line simplification. In: Proceedings of 7th international symposium on spatial data handling, 12–16 August, Delft. IGU, 533–546.

Weibel, R., (1997). Generalization of spatial data: principles and selected algorithms. In: M. Kreveld, et al., eds. Algorithmic foundations of geographic information systems. Berlin: Springer, 99–152.

Wu, et alShil – Ting and Mercedes Rocio Gonzales Marquez (2003). Proceedings of the XVI Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'03) 1530-1834/03 doi :10.1109/SIBGRA.2003.1240992

Yan, Z., et al. (2011). SeMiTri: A framework for semantic annotation of heterogeneous trajectories. In: The proceedings of the 14th international conference on Extending Database Technology (EDBT 2011). New York: ACM, 259–270.