Extensive Comparison of Trajectory Simplification Algorithms

Rence van Hunnik

Utrecht University
ICA-3820882

February 9, 2017

Abstract

In this study we compare ten simplification algorithms consisting of both line and trajectory simplification algorithms. Namely, Uniform Sampling, Douglas-Peucker, Visvalingam-Whyatt, Imai-Iri, TD-TR, SQUISH-E(μ), STTrace, OPW-TR, OPW-SP, and a newly introduced algorithm VW-TS. This newly introduced algorithm is an adaptation of the Visvalingam-Whyatt algorithm that uses Time-Space. This comparison is performed using three distinct real world datasets. Also, five error metrics are described and used to compare the simplifications' performance. These five error metrics are: Spatial Distance, Temporal Distance, Speed Deviation, Heading Deviation, and Acceleration Deviation. TD-TR, VW-TS, and SQUISH-E(μ) prove to have the lowest error across all but one error metric. On the Acceleration Deviation metric, OPW-SP gives the lowest error with a significant margin.

1 Introduction

More than ever people are capable of tracking location with small affordable GPS trackers. Nowadays it is increasingly normal to at all times know the location of your car, pet, postal deliveries, etc. It is even possible to equip your children with a GPS tracker that can make you aware of their whereabouts. Companies on the other hand have more insight into their logistics, with the goal of streamlining processes and coordinating employees beyond what they were capable of before GPS tracking. This data is often stored and processed to gain high level information in the long run.

When both latitude, longitude, and time are stored in data type Double, one sample uses $3 * 8$ $Bytes = 24$ $Bytes$ of storage. When the location of a smartphone is stored with a sample rate of every 1 second, a day of samples uses 2.07 $MegaByte$ of storage. Combine this with the knowledge that in a small country like the Netherlands there are approximately ten milion smartphones [1], and storing a years worth of all of these locations uses an approximate 7.5 $PetaByte$. While not a very real world example, a clear need for compression is indicated for companies like Google that do store enormous amounts of data about their users.

As we have seen, the storage size of the data is a good motivation for compression, but there are a few more reasons why it is important to make use of simplification algorithms to shrink the size. Data is usually collected on small portable devices and often by using wireless connections. A couple of examples can be given why it is preferable that the data that is transmitted is as small as possible in size. One could think of an application where a connection is scarce and unreliable, in this case having to upload a smaller file, would improve the chance of successful transmission. A different scenario could be where it is simply preferred to have the least amount of background data transmission as possible, like in consumer smartphones where it consumes both mobile data and battery life.

Even when storage size is not a factor and data is not collected in a decentralized manner, having to work with large data caches can prove cumbersome. Collected data is usually subject to computation, finding patterns or outliers is in itself already quite computationally intensive. Having to do it on data sets that are larger than necessary would make it even harder.

Now that we have clearly motivated why it is preferable to compress raw trajectory data, we will discuss how a compression of this data is possible. On a high level, trajectory data compression is done by reducing the number of sampled data points which will lead to a simplified trajectory. In this paper, we will use the term simplification to refer to the compression because the compression used is not lossless and produces a simplified approximation of the original trajectories.

Different algorithms have been devised around choosing which of the original data points to keep, and which to discard. Because these algorithms are lossy we need to know how much the simplified data differs from the original data, and how well one algorithm performs compared to the next. For this we use various error metrics that each describe distance (error), on a particular value, to the original sample data.

This study makes the following contributions: We evaluate different simplification algorithms on their performance on different error metrics while performing both high and moderate compression on real world data. Among the algorithms for simplifying trajectories we have modified an algorithm ourselves to perform better on the different error criteria. The results of this comparison study give indications on algorithms which might be preferred in certain situations. Overall, when no specific error metric is more important than another, TD-TR, SQUISH-E(μ) and VW-TS (the modified algorithm introduced in this paper) perform well on most metrics.

In this study we have used *C#* for implementing the algorithms, and the surrounding experiment environment. We also use three widely distributed datasets.

The remainder of this paper is organized as follows: In Section 2 we will look at error metrics that will be used in this study for comparing. In Section 3 the basics of the chosen simplification algorithms for this study are explained. Section 4 gives an overview of related work containing similar empirical studies. In Section 5 we give insight into our research method and data used in the study. Section 6 will discuss and present the results of the study. The paper concludes in Section 7 with a conclusion.

2 Error Metrics

In this section we will go into detail about the comparison of a trajectory and its simplified companion. In this study a Trajectory is represented as a polyline P that is a sequence of points $\{p_1, ..., p_n\}$, where point p_i consist of X_i, Y_i, and t_i that are respectively longitude, latitude, and sample time-stamp. The simplification of a trajectory is referred to as approximation A, and is a subset of P, the polyline of the original trajectory. Approximation A must also contain both p_1 and p_n of the original trajectory.

In our study we compare trajectories and their simplifications on five error metrics that cover all basic aspects of a trajectory. These are: Spatial Distance, Temporal Distance, Speed Deviation, Heading Deviation, and Acceleration Deviation. To calculate some of these metrics we need the following definitions.

- The length of edge $E\{p_i, p_j\}$ is defined as $\sqrt{(X_j - X_i)^2 + (Y_j - Y_i)^2}$.
- The speed at point p_i is defined as the speed on the edge $E\{p_i, p_{i+1}\}$, defined as $edgeLength(E\{p_i, p_{i+1}\})/(t_{i+1} - t_i)$.
- The speed at any time t between p_i and p_{i+1} is the same as the speed at p_i.
- The acceleration at point p_i is defined as the increase of speed on the surrounding two edges. $speed(p_{i+1}) - speed(p_i)$.
- The heading at point p_i is defined as the angle between the edge $E\{p_i, p_{i+1}\}$ and the origin.

To calculate error we need a point p_i from the original trajectory, and a point a_i from the approximation. For most error metrics we use points p_i and a_i that are synchronous in time, meaning that they have the same time-stamp value. These can be used to answer the question "*where_at*", meaning where was the moving object at a particular time. However, for the temporal distance we use points that have a similar location. This is usually referred to as the "*when_at*" question, when was the moving object at a particular location. Using these points we can calculate error on the discussed metrics as follows.

- The Spatial Distance between p_i and a_i is the length of the edge $E\{p_i, a_i\}$.
- The Temporal Distance between p_i and a_j is $abs(t - t\prime)$, where t is the time at p_i and $t\prime$ is the time at a_j.
- The Speed Deviation between p_i and a_i is $abs(speed(p_i) - speed(a_i))$.
- The Heading Deviation between p_i and a_i is $abs(heading(p_i) - heading(a_i))$.
- The Acceleration Deviation between p_i and a_i is $abs(acceleration(p_i) - acceleration(a_i))$.

Section 5 will describe how the metrics are used to give a numerical score to approximations on the metric that is used. Comparing the score of an approximation made by a certain algorithm to an approximation made by a different algorithm, will give an indication on the performance of the algorithm on the individual metrics. This can be useful information when choosing an algorithm for your application.

Algorithm	Time Complexity	Error Metrics	Online
Uniform Sampling	$O(n)$	None	Yes
Douglas-Peucker	$O(n \log n)$	Spatial distance	No
Visvalingam-Whyatt	$O(n \log n)$	Spatial distance	No
Imai and Iri	$O(n^2)$	Spatial distance	No
TD-TR	$O(n^2)$	Time distance ratio	No
SQUISH-E(μ)	$O(n \log n)$	Time distance ratio	No
STTrace	$O(n)$	Synchronous Euclidean distance, Heading, Speed	Yes
OPW-TR	$O(n^2)$	Time distance ratio	Yes
OPW-SP	$O(n^2)$	Time distance ratio, Maximum speed	Yes
VW-TS	$O(n \log n)$	Time distance ratio	No

Table 1: Simplification algorithm summary

3 Simplification Algorithms

In this study we will compare some well documented algorithms that are used both for line and trajectory simplification, and one new, adapted algorithm. These algorithms are shown in Table 1. Of these algorithms the Douglas-Peucker, Visvalingam-Whyatt, and Imai-Iri algorithms are designed for line simplification, and therefore only guarantee spatial distance. The remaining algorithms are trajectory simplification algorithms that are designed with different error criteria in mind. In the following subsections we will describe how these simplification algorithms function.

3.1 Uniform Sampling

Uniform Sampling is a basic algorithm, and does not guarantee any criteria other than reduction in points. This algorithm takes a polyline P, a sequence of points $\{p_1, ..., p_n\}$, and a user defined reduction ratio. The points are considered in order and points are uniformly added to the approximation in accordance with the reduction ratio.

3.2 Douglas-Peucker

The Douglas-Peucker algorithm [2][3] takes a polyline P a sequence of points $\{p_1, ..., p_n\}$, and a user defined allowed spatial error, $\varepsilon > 0$. The algorithm builds an approximation polyline P', initially consisting of p_1 and p_n. It continues adding the point p_i out of the original polyline that has the largest shortest-euclidean-distance to P' until that distance is smaller than ε as demonstrated in Figure 1.

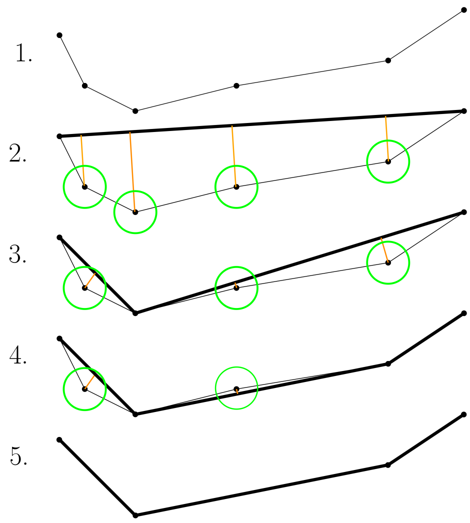

Figure 1: Illustration of how the Douglas-Peucker algorithm iteratively simplifies a line. The allowed spatial error ε is depicted with green circles.

3.3 Visvalingam-Whyatt

The Visvalingam-Whyatt algorithm [4] uses the concept of 'effective area', which is the area of the triangle formed by a point and its two neighbors. The algorithm takes a polyline P a sequence of points $\{p_1, ..., p_n\}$, and a user defined allowed spatial displacement error, $\varepsilon > 0$. For every set of three consecutive points $\{p_{i-1}, p_i, p_{i+1}\}$ a triangle is formed with its surface being the 'effective area'. Iteratively point p_i is dropped that results in the least areal displacement to form an approximation as illustrated in Figure 2. This process halts when the 'effective area' is larger than ε.

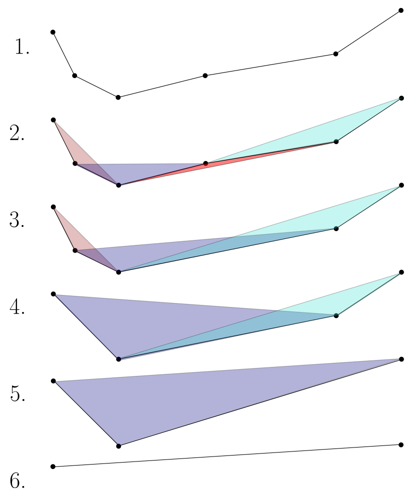

Figure 2: Illustration of how the Visvalingam-Whyatt algorithm iteratively simplifies the line.

3.4 Imai-Iri

The basis of the Imai-Iri algorithm [5] lies in the construction of an unweighted directed acyclic graph G. This graph is constructed by connecting all combinations of two points that would create an allowed shortcut. A breadth-first search is done on this graph to compute the shortest path connecting the first and last point, resulting in the approximation.

This algorithm takes a polyline P a sequence of points $\{p_1, ..., p_n\}$, and a user defined allowed spatial error, $\varepsilon > 0$. For each combination of two points (p_i and p_j) it checks if a line between them intersects all circles with radius ε that center on the points that lie between them $\{p_x > p_i, p_x < p_j\}$. When this is the case, the line $p_i p_j$ is an allowed shortcut and is added to the graph G, see Figure 3.2. After all allowed shortcuts are added to graph G, breadth-first search is done to find the shortest path through the graph from p_1 to p_n.

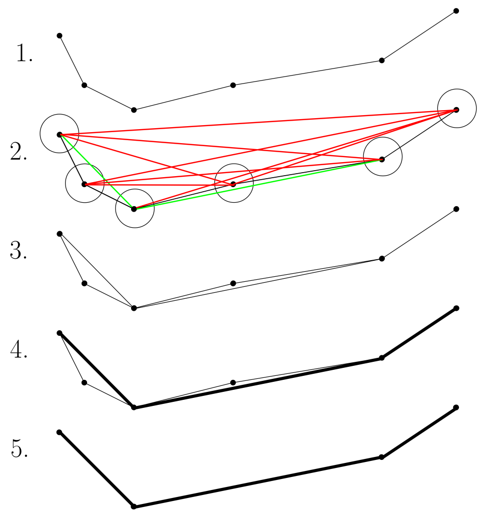

Figure 3: Illustration of how the Imai-Iri algorithm generates shortcuts. Green lines are allowed shortcuts, red lines are not allowed.

3.5 TD-TR

The TD-TR algorithm [6] functions in the same way as the Douglas-Peucker algorithm. The difference between them is that where the Douglas-Peucker algorithm uses the Euclidean distance to calculate the shortest distance between point p_i and the approximation, the TD-TR algorithm uses a time-synchronous Euclidean distance, as seen in Figure 4. This distance measure can guarantee both a maximum spatial distance as well as a maximum temporal error distance.

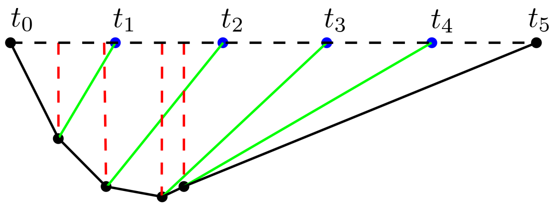

Figure 4: The distance measures used in the Douglas-Peucker and TD-TR algorithms are demonstrated here. The dashed lines show the Euclidean distance. The solid lines show the time-synchronous Euclidean distance when the original points have an equal sample rate.

3.6 SQUISH-E(μ)

The SQUISH-E(μ) algorithm [7] is a derivative of the SQUISH algorithm [8].

SQUISH-E(μ) takes a polyline P a sequence of points $\{p_1, ..., p_n\}$, and a user defined allowed priority error, $\mu > 0$. The algorithm assigns a priority score to each of the points from P and orders pointers to them in a priority queue from low to high. The priority of point p_i is the sum of two values. The first value is the time-synchronous Euclidean distance of p_i to a line between p_{i-1} and p_{i+1}. The second value is the maximum of the priorities that the neighboring points of p_i had when they were removed, if p_i has neighbors that have been removed.

The algorithm removes point p_i from P that has the lowest priority, and recalculates its neighbors' priority score accordingly. The algorithm is halted when the lowest priority score in the priority queue is higher than μ. This results in a polyline A, that is an approximation of the polyline P.

3.7 STTrace

The STTrace algorithm [9] determines which points of the original polyline are discarded from the approximation, by calculating if they are predictable within a certain error margin. This is done by constructing a 'safe area', the area where a point is predicted to be by its three original predecessors. From the original polyline, points with at least three predecessors are considered in order, and only the unpredictable points are added to the approximation.

This algorithm takes a polyline P a sequence of points $\{p_1, ..., p_n\}$, an $\alpha > 0$ and a $0 < \beta < 360$, where α and β are respectively the user defined allowed speed and heading error. A point p_i is considered predictable when it is inside of the 'safe area' which is the intersection of two prediction polygons called A_{i-2} and A_{i-1}.

These prediction polygons are annular sectors, that are constructed from two concentric circles, a heading, and an angle, see Figure 5. Annular sector A_{i-1} is centered on p_{i-1} and its two concentric circles have radii of length of edge $p_{i-2}p_{i-1}$ plus and minus $\alpha/2$. Annular sector A_{i-1} its heading is vector $\overrightarrow{p_{i-2}p_{i-1}}$ and has angle β. Annular sector A_{i-2} is constructed the same way and is also centered on p_{i-1}, but for its two concentric circles it uses radii of length of edge $p_{i-3}p_{i-2}$ plus and minus $\alpha/2$, and its heading is vector $\overrightarrow{p_{i-3}p_{i-2}}$.

When point p_i is not considered predictable, it is deemed necessary in the approximation. Vice versa, when point p_i is predictable it is omitted from the approximation.

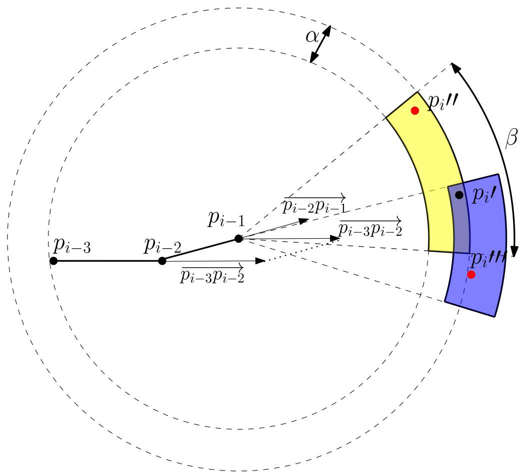

Figure 5: The annular sectors constructed by the STTrace algorithm are colored yellow (A_{i-1}) and blue (A_{i-2}). Three scenarios are drawn for point p_i. In both scenarios with p_i'' and p_i''', point p_i is only predicted by one of the two annular sectors, hence point p_i should be added to the approximation. In the scenario of p_i', point p_i is within both annular sectors and would therefore be omitted from the approximation.

3.8 OPW-TR

The OPW-TR algorithm [6] is an Opening Window (OPW) algorithm. The basic idea behind OPW algorithms is that they grow a window from a starting point called the anchor by incrementally progressing the so-called float point, and stop when some halting condition is met. The OPW-TR algorithm uses the time-synchronous Euclidean distance measure as halting condition.

This algorithm takes a polyline P a sequence of points $\{p_1, ..., p_n\}$, and a user defined allowed time-synchronous spatial error, $\varepsilon > 0$. A window is grown from the anchor p_a, that starts out as p_1. The halting condition is triggered when a point p_i between the anchor and the float p_f has a time-synchronous Euclidean distance larger than ε to edge $p_a p_f$. When this happens point p_i is added to the approximation, as demonstrated in Figure 6.

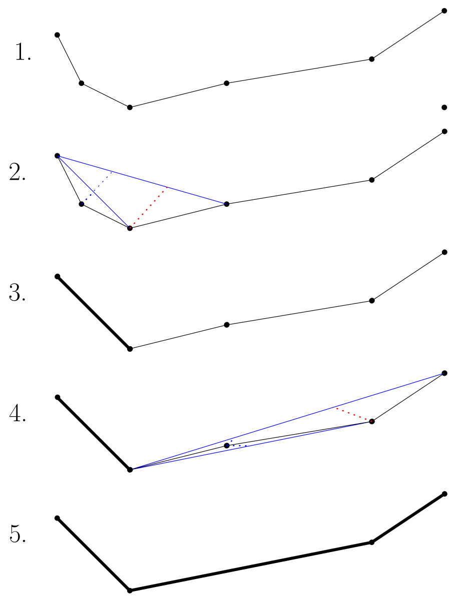

Figure 6: An approximation using the Opening Window algorithm where the red dotted lines exceed the error threshold.

3.9 OPW-SP

The OPW-SP algorithm [6] is an Opening Window algorithm just like the OPW-TR algorithm. This SP variant uses a halting condition based on the speed difference between two consecutive points.

This algorithm takes a polyline P a sequence of points $\{p_1, ..., p_n\}$, and a user defined allowed speed difference, $\varepsilon > 0$. The speed difference between point p_i and p_{i+1} is calculated in order. When this speed difference is higher than ε the halting condition is triggered, and point p_i is added to the approximation.

3.10 VW-TS

The VW-TS algorithm is a variation on the Visvalingam-Whyatt algorithm that takes the temporal component of a trajectory into account. This algorithm is not yet described in literature.

The original Visvalingam-Whyatt algorithm uses the two spatial coordinates (X, Y) to represent the polyline, and discards the time-stamps of points. This will most likely result in simplifications that have a higher error on the temporal components. VW-TS uses a third dimension, Z. This dimension represents the progression in time of the polyline, where a point p_i being farther in time than point p_j will result in the z coordinate of p_i being larger than the z coordinate of p_j.

Like the Visvalingam-Whyatt algorithm, this adaptation calculates, the 'effective area' the surface area of a triangle between three consecutive points. The original algorithm however calculates the surface area of the triangle in its two dimensions, where the adapted algorithm calculates this surface in three dimensional space.

Like the Visvalingam-Whyatt algorithm, VW-TS progressively removes point p_i that formed triangle p_{i-1}, p_i, p_{i+1} that results in the smallest 'effective area' until this area is larger than ε, a user defined spatial temporal displacement error.

This addition makes the consideration of the temporal component possible, since a variation in the temporal component now also results in a variation in triangle surface area. Thus this adaptation provides the simplification of points that are close in time as well as in space.

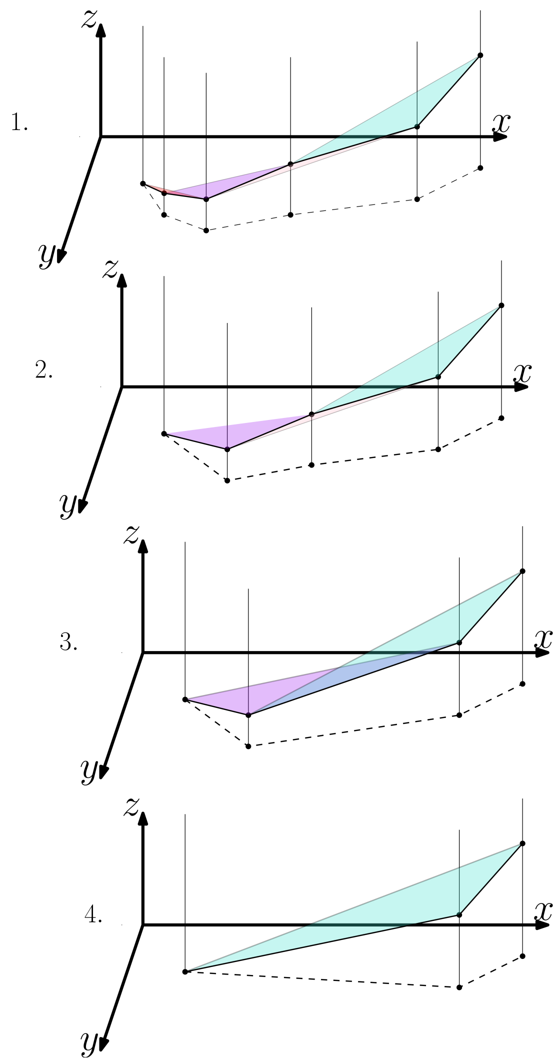

Figure 7: Illustration of how the VW-TS algorithm iteratively simplifies the line until the turquoise effective area is larger than ε.

4 Related Work

There are three studies done by Meratnia and de By (2004) [6], Muckell et al. (2010) [10], and Muckell et al. (2013) [7] that also compare algorithms in an experimental setting.

4.1 Meratnia and de By, Time-Ratio Algorithms (2004)

One of the first experimental studies for comparing trajectory compression algorithms is performed by Meratnia and de By in 2004 [6].

They introduced a distance metric that takes into account the position of a point as well as its timestamp of the original trajectory and the new trajectory. This is the same distance measure as described under the name 'time-synchronous Euclidean distance' in Section 3.5. This new distance metric introduces a class of algorithms that can be called time-ratio algorithms. They applied this to two algorithms and named them TD-TR and OPW-TR, as discussed in Section 3.5 and 3.8. TD-TR is obtained through the application of the time-ratio distance measuring technique on the Douglas-Peucker algorithm and OPW-TR is an opening window algorithm with the time-ratio distance metric applied.

Next to the time-ratio algorithm, they introduced a speed difference threshold, the spatio-temporal algorithm, indicating above which speed difference the data point will always be retained. They applied this algorithm to the top down and opening window algorithm and named the algorithms TD-SP and OPW-SP.

In their experimental study, they have done three experiments in which they tested on ten real world trajectories three experiments. They applied fifteen different spatial threshold values ranging from 30 to $100m$ and three speed difference threshold values ranging from 5 to $25m/s$.

In their first experiment, they compared Douglas-Peucker with their improved version, TD-TR. In this comparison, the TD-TR algorithm has lower errors but the compression is somewhat lower on a particular distance threshold.

In their second experiment, they analyzed the choice of breakpoint in opening window algorithms. They compared the Normal Opening Window algorithm (NOPW), removing the data point that causes the threshold violation, with the Before Opening Window algorithm (BOPW), removing the sample-point just before the sample-point causing a threshold violation. They have found that the BOPW algorithm achieves higher compression but has worse errors than the NOPW algorithm.

The last experiment concerned comparing their improved opening window algorithm, OPW-TR, with the Normal Opening Window algorithm (NOPW). Their results show that OPW-TR is superior to the NOPW algorithm. The results of the OPW-TR algorithm shows that choosing a higher threshold does not have a huge impact on the amount of error.

They also applied their second algorithm, the spatio-temporal algorithm (SP), to the top-down algorithm, resulting in TD-SP, and to the opening window algorithm, resulting in OPW-SP. The TD-SP algorithm showed high sensitivity towards speed thresholds for both error and compression. For OPW-SP, changes in speed threshold value did not show high changes in the results.

Comparing TD-TR, OPW-TR and OPW-SP, TD-TR ranks slightly better because of better compression rates. The results of their study show that algorithms with spatio-temporal characteristics outperform other algorithms.

4.2 Muckell et al., Algorithms for Compressing GPS Trajectory Data (2010)

The experimental research done by Muckell et al. (2010) [10] is a comprehensive empirical evaluation of compression algorithms such as the Douglas-Peucker algorithm, Bellman's algorithm, STTrace algorithm and the Opening Windows algorithms. They have analyzed these algorithms on how well they preserve the spatio-temporal information, on their execution time, and on error metrics.

They have used two datasets, one is obtained from a fleet of buses in Albany (Public-Transit dataset) and another was obtained from 24 volunteers at the New York Metropolitan Transportation Council (NYMTC dataset). Three groups of trajectories were created from both datasets, each group representing one travel mode and each trajectory consists of 5000 points. In total 14 trajectories were selected.

To compare the execution times, a common compression ratio of 7 was chosen. There was no significant difference in the execution time performance for any algorithm among the different travel modes. However, between the different algorithms, substantial differences in execution time were observed. STTrace is by far the slowest algorithm (40 seconds) and Uniform Sampling was the fastest (0.002 seconds). Significant differences were not apparent in algorithms that are a modification of each other. Douglas-Peucker (2.2 seconds) and TD-TR (2.5 seconds), as well as the Opening Window algorithms OPW-TR (0.4 seconds) and OPW-SP (0.5 seconds) do not differ very much in their median execution time. Finally, the Bellman's algorithm had an execution time of 3.2 seconds.

A comparison of the algorithms on the synchronized Euclidean distance (SED) error metric shows that there are significant differences between the algorithms as well as the travel modes. Again, all algorithms were compared having the same compression ratio of 7. The bus travel mode has the highest degree of error, followed by the pedestrian travel mode, and the multimodal dataset had the least amount of error. When comparing the algorithms, OPW-SP has a high SED error. Another notable result was that Douglas-Peucker performed better than the modified version TD-TR. Douglas-Peucker also outperformed the OPW-TR algorithm.

Comparing the travel modes on the speed metric shows that the bus dataset again has the highest error rate compared to the pedestrian and multimodel datasets. Comparing the algorithms shows that STTrace has the most consistent speed results and Bellman and Douglas-Peucker both have low speed errors. Comparing the opening window algorithms, OPW-SP performed slightly better than OPW-TR.

Analyzing the error in direction or heading shows that the bus travel mode has the lowest error and that Bellman's algorithm has the best performance followed by STTrace, however only for small compression ratios. When compression ratios are larger there is no significant advantage in heading accuracy between Douglas-Peucker, TD-TR, OPW-TR and OPW-SP compared to Uniform Sampling.

4.3 Muckell et al., SQUISH-E (2013)

The study done by Muckell et al. (2013) [7] is more recent and they compared a number of algorithms, namely Uniform Sampling, Douglas-Peucker, TD-TR, and Opening Window algorithms, while also introducing their own new algorithm SQUISH-E of which a version is described in this paper in Section 3.6. They have chosen these algorithms because they argued that these algorithms have unique benefits in balancing compression time and the degree of error.

The datasets chosen by Muckell et al. represent different transportation modes (bus, urban commuter, and multi-modal). They have selected 71 trajectories each containing around 20,000 points for their experiment.

They compare the algorithms in terms of compression time and error metrics. To be able to compare the algorithms on SED, average speed, and average spatial error metric, they have used a compression ratio of 10 for each algorithm. For the algorithms that did not take a target compression ratio as an input parameter, they repeatedly executed those algorithms while modifying the error bound parameter until the desired compression ratio was achieved.

In terms of compression time, their observation is that the fastest algorithm is the Uniform Sampling algorithm and the slowest algorithm is TD-TR. They also have found that the synchronized Euclidean distance (SED) metric requires more computation than spatial metric. Uniform Sampling and Opening Window were the fastest algorithms. They also acknowledge that despite their speed benefits, Uniform Sampling and Opening Window have fundamental limitations in controlling the growth or errors during compression. Furthermore, they found that their own algorithm, SQUISH-E, was faster than the other algorithms that use SED. They argue that this speed benefit is due to the use of a priority queue which enables fast and effective removal of points.

In terms of comparison of algorithms across metrics, TD-TR and SQUISH-E(μ) outperform the algorithms in case of overall accuracy. Next to that, they found that the SED error metric has the benefit of incorporating temporal data into error calculation. Algorithms that take into account the SED error metric keep spatial error at a relatively low level. According to their results, Douglas-Peucker and TD-TR have most accurate compressions in terms of the spatial error metric and the SED error metric, respectively.

5 Experimental Comparisons

In this section we will describe on which data we have experimented and give detailed insight in how our experiments are set up that lead to the results described in Section 6.

5.1 The Data

In this study three distinct datasets are used, GeoLife [11][12][13], GoTrack [14], and TaxiServiceTrajectory (TST) [15]. First the data is filtered, trajectories with irregularities like large jumps in both time and space as well as trajectories with a unsuitably small number of points are omitted from this study.

The GeoLife dataset was collected by 182 users in a period of over five years. The dataset contains 17,621 unique trajectories collected by different GPS loggers and GPS phones. 91.5 percent of the total trajectory set is sampled every $1 \sim 5$ seconds. This dataset contains data collected while traveling by foot, bike, bus, car, train, and airplane.

The GoTrack dataset was obtained via the use of the Android app called Go!Track, and contains 163 unique trajectories. The trajectories are evenly collected by car or bus, and have a sample rate of approximately 5 seconds.

The TST dataset was obtained by a taxi company in Portugal that installed telematics in their vehicles. The data was acquired over a continuous period of nine months and contains 320 unique trajectories. The data is sampled every 15 seconds.

5.2 Research Method

5.2.1 Data Preparation

All three datasets made use of the geographic coordinate system (GCS) and had time-stamps that were altogether not that friendly for calculations. So some preprocessing was done to make calculations easier and to represent the trajectories from the three databases in the same way. Because the GCS coordinates have their origin as a mapping on a spheroid (earth), this data was remapped to an orthogonal plane and refactored to be in meters. As a time-stamp system, all trajectories are remapped to number of seconds passed since the start of the trajectory, and start at zero seconds.

Experiments were done in three fold without combining datasets as a means of being able to compare results across different datasets.

5.2.2 Trajectory Subdivision

When a simplification algorithm returns an approximation A that is a simplified version of P, it is likely that A consist of less points than P. This is exactly the purpose of simplification. However, we need to score the approximation A against the original trajectory P and we would like both trajectories to have an equal number of points on which the error metrics can be calculated. Not only do we have the requirement that both A and P need an equal number of points to score an approximation. We also require the error score to represent the whole of the error between A and P, and not only error on the existing, potentially sparse, sample points.

To accurately calculate an error score that represents the whole of the error, we synchronously subdivide both A and P to gain more sample points a_i and p_i on which the error metric can be calculated. This subdivision is done into very small subsections to gain a larger number of sample points a_i p_i on which to calculate a near continuous comparison. Because we need both points a_i and p_i synchronized in time as well as synchronized in space, we have to make two separate subdivisions.

The subdivision that is used for the metrics that are related to the "*where_at*" question subdivides both A and P in sections that are 0.1 seconds. The location of the new sample points a_i and p_i are linearly interpolated along A and P respectively. Ignoring the original points form A and P, this will result in an equal number of points in A and P because both their start and finish time are the same.

For the temporal error metric we need a subdivision that can answer the "*when_at*" question, so we subdivide the original trajectory P in sections with length 0.1 meter. We cannot simply subdivide approximation A in sections with length 0.1 meter because this would result in far less sample points since the total length of A is likely to be shorter than that of P. To subdivide A in a way that creates sample points that are synchronous in space to the sample points of P we calculate $X = Length(A)/Lenght(P)$, and subdivide A in sections of length $0.1X$ meter. The time of the new sample points a_i and p_i are linear interpolated along respectively A and P. This will result in two subdivisions synchronous in space that have an equal number of sample points.

5.2.3 Simplification Error Metric Score

We have shown how we can calculate the error of a_i in relation to p_i on the five chosen metrics in Section 2. In the previous subsection we have shown how we subdivide both A and P to gain enough sample points for scoring the entire simplification accurately. In this section this information is used to calculate a numerical value that is a score $Sc(A \leftrightarrow P)$ of how well simplification A represents the original data that made up trajectory P, where a lower score means a smaller distance between a_i and p_i.

The score $Sc(A \leftrightarrow P)$ of simplification A is calculated using the root mean square (RMS). This is done by summing the root of the distance metric between a_i and p_i, where, from the subdivided polyline, i is in order from 1 to n. This sum is divided by the number of samples in the subdivision to get the mean. And finally we take the root of this mean to end up with the score of simplification A. Because we used the RMS to calculate the score, a high local error in a_i and p_i will disproportionately increase score $Sc(A \leftrightarrow P)$. This results in lower scores for approximations that have low local maxima and minima, and higher scores for approximations that have high local maxima en minima in their error distance.

5.2.4 Comparing Simplification Algorithms

Now we know how to calculate the RMS score $Sc(A \leftrightarrow P)$ of a simplification on a specific error metric. This score $Sc(A \leftrightarrow P)$ can be compared to the score of simplifications obtained by other simplification algorithms.

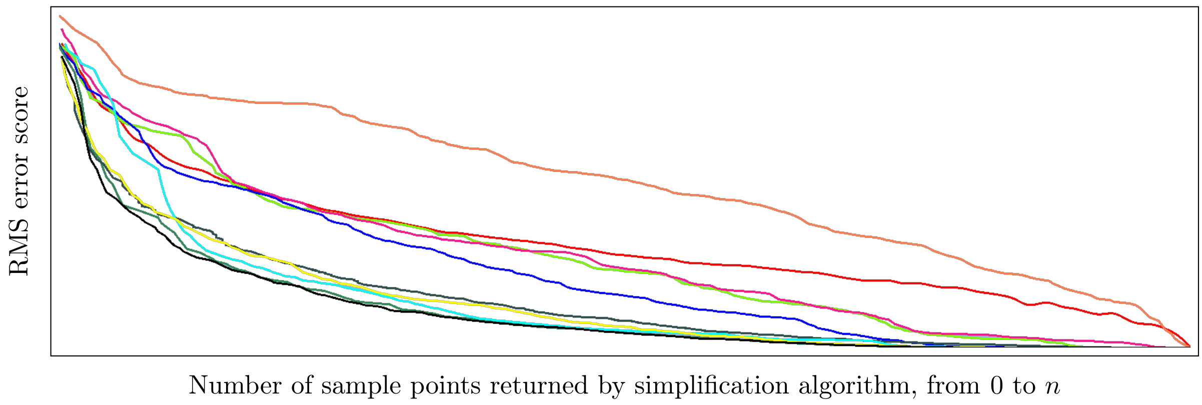

Figure 8: The RMS score ranges for all 10 simplification algorithms on the temporal error metric.

Table 2: Significance table

Heading	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	US	II	STT	OPW-TR	US	SQ	VW-TS	TD-TR
DP			DP	DP	DP	DP	DP	DP	VW-TS	DP
VW				II	STT	OPW-TR	VW	SQ	VW-TS	TD-TR
II					II	OPW-TR	II	SQ	VW-TS	TD-TR
STT						OPW-TR	STT	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	VW-TS	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									VW-TS	TD-TR
VW-TS										VW-TS
TD-TR										

However, this is only fair when both simplifications consist of an equal number of points.

In order to be able to compare simplification algorithms we would like to have multiple simplification algorithms simplify one original trajectory, and have them each produce a simplification that has a similar reduction in terms of number of sample points in A. Only then it is fair to compare their RMS score.

Each of the algorithms has either one or two parameters that is user defined, adjusting these parameters will influence the number of sample points in its simplification A. Because most of the simplification algorithms do not allow a reduction parameter as input, the particular parameters that a simplification algorithm does allow must be used to influence the reduction in number of sample-points.

For this study we have devised a method to be able to compare simplification algorithms' RMS scores over the entire range from, returning the original trajectory, to returning only the first and last sample point from P. To collect this entire range of RMS scores, we have generated simplifications with small increments in the algorithms' user defined parameter. In combination with linear interpolation this allows us to compare simplification algorithms on any given reduction percentage. This data can be graphed as can be seen in Figure 7.

6 Results

In this section, the algorithms described in Section 3 are experimentally evaluated and compared as described in Section 5.

For all three databases a mild reduction ratio of 7 is chosen, this is common practice in the literature. A second strong reduction ratio is chosen according to average trajectory length, the GeoLife dataset has a second reduction ratio of 25, GoTrack and TST have a reduction ratio of 14 since their average trajectory length does not support a reduction ratio of 25.

6.1 Significance Tables

A two-sided t-test with an $\alpha = 0.01$ is used to determine for all combinations of two algorithms, if one algorithm has a significantly better $Sc(A \leftrightarrow P)$ score than another algorithm on a metric, at a specified reduction ratio. These tests result in the data that is shown in Tables 2 and 9-14. A green cell in these tables indicates that the algorithm in that cell is significantly better

Table 3: GeoLife with reduction ratio 7

Metric \ *Algorithm*	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
Spatial	0	1.9593	-31.41	-16.48	-541.8	57.465	-204.4	70.998	**72.573**	72.557
Temporal	0	**53.887**	-17.90	26.034	-30.60	34.030	-60.04	48.293	39.364	49.084
Speed	0	16.129	17.262	11.098	-22.40	47.065	40.014	**58.247**	40.784	55.544
Heading	0	31.008	-6.675	16.252	0.9605	17.658	-13.57	28.017	**31.257**	28.618
Acceleration	0	19.582	18.492	16.950	17.861	36.407	**60.231**	47.930	27.987	43.136

than the one it was up against. A non-colored cell indicates that between the two algorithms there was no significant winner. The full set of tables can be found in the appendix (Tables 9-14).

Using these tables we can determine a ranking for each metric. The top three is shown for each dataset in Table 4. The full set of tables can be found in the appendix (Tables 5-8). There we can see that from the ten simplification algorithms tested only six appear in at least one top three. Namely, TD-TR, SQUISH-E(μ) (SQ), VW-TS, Douglas-Peucker (DP), OPW-SP, and OPW-TR. From these TD-TR, SQ, and VW-TS make up 78% of the table. Noteworthy is the appearance of the Douglas-Peucker algorithm as a line simplification algorithm among trajectory simplification algorithms. The OPW-SP algorithm clearly shows that it is specifically designed with a speed metric in mind, it performs the best on the acceleration metric across the datasets. However, it does not appear in the top three on the speed metric itself.

Among the three datasets used, similarity can be observed. In particular on the GoTrack and TST datasets, the algorithms perform almost identically. This supports the validity of the ranking in general.

Table 4: Metric top three for all datasets on reduction ratio 7

Metric \ *Dataset*		GeoLife	GoTrack	TST
Spatial	1.	VW-TS	TD-TR	TD-TR
	2.	TD-TR	VW-TS	VW-TS
	3.	SQ	SQ	SQ
Temporal	1.	DP	TD-TR	TD-TR
	2.	TD-TR	SQ	SQ
	3.	SQ	VW-TS	DP
Speed	1.	SQ	SQ	SQ
	2.	TD-TR	TD-TR	TD-TR
	3.	OPW-TR	VW-TS	OPW-TR
Heading	1.	VW-TS	DP	DP
	2.	DP	TD-TR	SQ
	3.	TD-TR	VW-TS	VW-TS
Acceleration	1.	OPW-SP	OPW-SP	OPW-SP
	2.	SQ	SQ	SQ
	3.	TD-TR	TD-TR	TD-TR

6.2 Percentage Table

As a second method of comparing the algorithms, we have calculated the percentage improvement/deterioration of an algorithm's $Sc(A \leftrightarrow P)$ compared to Uniform Sampling, using the following formula: $((USSc(A \leftrightarrow P) - XSc(A \leftrightarrow P))/USSc(A \leftrightarrow P)) * 100$ where X is the algorithm which is compared. For each dataset and reduction ratio, these percentages are calculated and stored in separate Tables 3 and 15-20. The coloring of the cells indicate whether the value is positive or negative. The highest percentage is marked in bold. The complete set of tables can be found in the appendix (Tables 15-20).

These tables can be used in future work since they are implementation independent, and only indicate performance improvement against Uniform Sampling.

These tables offer a more compact representation than the previously mentioned significance tables, while retaining most of the information needed for indicating both the algorithms that perform well, and algorithms that perform poorly. On top of this, the top three ranking gained from the significance tables are in most cases identical to a ranking on percentage improvement. The only places where the percentage tables and significance tables are in disagreement, are places where the significance tables have a non-significant improvement.

7 Conclusion

In this paper we gave an extensive comparison of trajectory data simplification algorithms, in combination with five error metrics that describe most aspects of a trajectory. We provided detailed insight into the methods used to accommodate the comparison and presented our calculations for scoring approximations. We also introduced a new adaptation of the Visvalingam-Whyatt algorithm named VW-TS.

This VW-TS algorithm is an adaptation of the Visvalingam-Whyatt algorithm that uses the time aspect as a third dimension which is usually referred to as Time-Space. The comparison described in this paper shows that this VW-TS algorithm performs well on most error metrics, and can be considered a strong trajectory simplification algorithm.

This study shows that the Imai-Iri and Visvalingam-Whyatt algorithms that are traditionally line simplification algorithms generally perform worse than Uniform Sampling when simplifying trajectory data. On the other hand, the Douglas-Peucker algorithm shows that it can still compete against trajectory simplification algorithms on some error metrics.

Among the trajectory simplification algorithms STTrace can be considered a weak algorithm since it induces error much faster than the other trajectory simplification algorithms. The TD-TR, VW-TS, and SQUISH-E(μ) algorithms prove to be very all-round, as they perform very well on all five of the error metrics. They appear as the top 3 on most error metrics. Another result is that OPW-SP outperforms all other simplification algorithms on the acceleration metric.

However, in the results it is shown that some simplification algorithms, in particular the line simplification algorithms, perform quite bad on the spatial metric. This research has the limitation that the temporal component had an unintended influence on the spatial metric scoring. Future work can improve the way that the spatial metric is calculated, to see if there is a significant performance difference on the spatial metric among trajectory and line simplification algorithms.

Next to this, future work can increase the validity of this study by performing a comparative study on different trajectory datasets to see if their findings are in line with the results of this study.

References

[1] R. Bosman and S. de Bruyckere. (2015) Dutch smartphone user 2015 q1. [Online]. Available: www.telecompaper.com/research/dutch-smartphone-user-2015-q1–1086938

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

[3] J. Hershberger and J. Snoeyink, "An o (n log n) implementation of the douglas-peucker algorithm for line simplification," in *Proceedings of the tenth annual symposium on Computational geometry*. ACM, 1994, pp. 383–384.

[4] M. Visvalingam and J. Whyatt, "Line generalisation by repeated elimination of points," *The Cartographic Journal*, vol. 30, no. 1, pp. 46–51, 1993.

[5] H. Imai and M. Iri, "Computational-geometric methods for polygonal approximations of a curve," *Computer Vision, Graphics, and Image Processing*, vol. 36, no. 1, pp. 31–41, 1986.

[6] N. Meratnia and A. Rolf, "Spatiotemporal compression techniques for moving point objects," in *Advances in Database Technology-EDBT 2004*. Springer, 2004, pp. 765–782.

[7] J.-H. Hwang, C. T. Lawson, J. Muckell, P. W. Olsen Jr, and S. Ravi, "Compression of trajectory data: a comprehensive evaluation and new approach," *GeoInformatica*, vol. 18, no. 3, pp. 435–460, 2014.

[8] J.-H. Hwang, C. T. Lawson, J. Muckell, V. Patil, F. Ping, and S. Ravi, "Squish: an online approach for gps trajectory compression," in *Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications*. ACM, 2011, p. 13.

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

[10] J.-H. Hwang, C. T. Lawson, J. Muckell, and S. Ravi, "Algorithms for compressing gps trajectory data: an empirical evaluation," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2010, pp. 402–405.

[11] W.-Y. Ma, X. Xie, L. Zhang, and Y. Zheng, "Mining interesting locations and travel sequences from gps trajectories," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 791–800.

[12] Y. Chen, Q. Li, W.-Y. Ma, X. Xie, and Y. Zheng, "Understanding mobility based on gps data," in *Proceedings of the 10th international conference on Ubiquitous computing*. ACM, 2008, pp. 312–321.

[13] W.-Y. Ma, X. Xie, and Y. Zheng, "Geolife: A collaborative social networking service among user, location and trajectory," in *IEEE Data Engineering Bulletin 33, 2*, 2010, pp. 32–40.

[14] M. O. Cruz, A. P. Guimares, and H. Macedo, "Grouping similar trajectories for carpooling purposes," in *Proceedings of Brazilian Conference on Intelligent Systems*, 2015, pp. 234–239.

[15] L. Damas, M. Ferreira, J. Gama, Jand Mendes-Moreira, and L. Moreira-Matias, "Predicting taxi-passenger demand using streaming datapredicting taxiâĂŞpassenger demand using streaming data," *IEEE Transactions on Intelligent Transportation Systems 14. 3*, pp. 1393–1402, 2013.

8 Appendix

Table 5: Metric top three for GeoLife on reduction ratio 7 and 25

Metric \ *Ratio*		1/7	1/25
Spatial	1.	VW-TS	VW-TS
	2.	TD-TR	TD-TR
	3.	SQ	SQ
Temporal	1.	DP	DP
	2.	TD-TR	TD-TR
	3.	SQ	SQ
Speed	1.	SQ	SQ
	2.	TD-TR	TD-TR
	3.	OPW-TR	OPW-TR
Heading	1.	VW-TS	VW-TS
	2.	DP	DP
	3.	TD-TR	TD-TR
Acceleration	1.	OPW-SP	OPW-SP
	2.	SQ	SQ
	3.	TD-TR	TD-TR

Table 6: Metric top three for GoTrack on reduction ratio 7 and 14

Metric \ *Ratio*		1/7	1/14
Spatial	1.	TD-TR	TD-TR
	2.	VW-TS	SQ
	3.	SQ	VW-TS
Temporal	1.	TD-TR	TD-TR
	2.	SQ	SQ
	3.	VW-TS	VW-TS
Speed	1.	SQ	SQ
	2.	TD-TR	TD-TR
	3.	VW-TS	OPW-TR
Heading	1.	DP	DP
	2.	TD-TR	VW-TS
	3.	VW-TS	SQ
Acceleration	1.	OPW-SP	OPW-SP
	2.	SQ	SQ
	3.	TD-TR	TD-TR

Table 7: Metric top three for TST on reduction ratio 7 and 14

Metric \ *Ratio*		1/7	1/14
Spatial	1.	TD-TR	TD-TR
	2.	VW-TS	VW-TS
	3.	SQ	SQ
Temporal	1.	TD-TR	SQ
	2.	SQ	TD-TR
	3.	DP	OPW-TR
Speed	1.	SQ	SQ
	2.	TD-TR	TD-TR
	3.	OPW-TR	OPW-TR
Heading	1.	DP	DP
	2.	SQ	SQ
	3.	VW-TS	OPW-TR
Acceleration	1.	OPW-SP	OPW-SP
	2.	SQ	SQ
	3.	TD-TR	TD-TR

Table 8: Metric top three for all datasets and reduction ratios

Metric \ *Dataset*		GeoLife		GoTrack		TST	
		1/7	1/25	1/7	1/14	1/7	1/14
Spatial	1.	VW-ST	VW-TS	TD-TR	TD-TR	TD-TR	TD-TR
	2.	TD-TR	TD-TR	VW-TS	SQ	VW-TS	VW-TS
	3.	SQ	SQ	SQ	VW-TS	SQ	SQ
Temporal	1.	DP	DP	TD-TR	TD-TR	TD-TR	SQ
	2.	TD-TR	TD-TR	SQ	SQ	SQ	TD-TR
	3.	SQ	SQ	VW-TS	VW-TS	DP	OPW-TR
Speed	1.	SQ	SQ	SQ	SQ	SQ	SQ
	2.	TD-TR	TD-TR	TD-TR	TD-TR	TD-TR	TD-TR
	3.	OPW-TR	OPW-TR	VW-TS	OPW-TR	OPW-TR	OPW-TR
Heading	1.	VW-TS	VW-TS	DP	DP	DP	DP
	2.	DP	DP	TD-TR	VW-TS	SQ	SQ
	3.	TD-TR	TD-TR	VW-TS	SQ	VW-TS	OPW-TR
Acceleration	1.	OPW-SP	OPW-SP	OPW-SP	OPW-SP	OPW-SP	OPW-SP
	2.	SQ	SQ	SQ	SQ	SQ	SQ
	3.	TD-TR	TD-TR	TD-TR	TD-TR	TD-TR	TD-TR

Table 9: GeoLife significance table with reduction ratio 7

Spatial	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	VW	II	US	OPW-TR	US	SQ	VW-TS	TD-TR
DP			DP	DP	DP	OPW-TR	DP	SQ	VW-TS	TD-TR
VW				II	VW	OPW-TR	VW	SQ	VW-TS	TD-TR
II					II	OPW-TR	II	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	VW-TS	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									VW-TS	TD-TR
VW-TS										VW-TS
TD-TR										

Temporal	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	US	II	US	OPW-TR	US	SQ	VW-TS	TD-TR
DP			DP	DP	DP	DP	DP	DP	DP	DP
VW				II	VW	OPW-TR	VW	SQ	VW-TS	TD-TR
II					II	OPW-TR	II	SQ	VW-TS	TD-TR
STT						OPW-TR	STT	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	OPW-TR	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									SQ	TD-TR
VW-TS										TD-TR
TD-TR										

Speed	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	VW	II	US	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
DP			VW	DP	DP	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
VW				VW	VW	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
II					II	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	OPW-TR	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									SQ	SQ
VW-TS										TD-TR
TD-TR										

Heading	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	US	II	STT	OPW-TR	US	SQ	VW-TS	TD-TR
DP			DP	DP	DP	DP	DP	DP	VW-TS	DP
VW				II	STT	OPW-TR	VW	SQ	VW-TS	TD-TR
II					II	OPW-TR	II	SQ	VW-TS	TD-TR
STT						OPW-TR	STT	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	VW-TS	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									VW-TS	TD-TR
VW-TS										VW-TS
TD-TR										

Acceleration	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
DP			VW	DP	DP	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
VW				VW	VW	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
II					STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-SP	SQ	OPW-TR	TD-TR
OPW-SP								OPW-SP	OPW-SP	OPW-SP
SQ									SQ	SQ
VW-TS										TD-TR
TD-TR										

Table 10: GeoLife significance table with reduction ratio 25

Spatial	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		US	US	US	US	OPW-TR	US	SQ	VW-TS	TD-TR
DP			DP	DP	DP	OPW-TR	DP	SQ	VW-TS	TD-TR
VW				II	VW	OPW-TR	VW	SQ	VW-TS	TD-TR
II					II	OPW-TR	II	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	VW-TS	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									VW-TS	TD-TR
VW-TS										VW-TS
TD-TR										

Temporal	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	US	II	US	OPW-TR	US	SQ	VW-TS	TD-TR
DP			DP	DP	DP	DP	DP	DP	DP	DP
VW				II	VW	OPW-TR	VW	SQ	VW-TS	TD-TR
II					II	OPW-TR	II	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	VW-TS	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									SQ	TD-TR
VW-TS										TD-TR
TD-TR										

Speed	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	VW	II	US	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
DP			VW	DP	DP	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
VW				VW	VW	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
II					II	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	OPW-TR	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									SQ	SQ
VW-TS										TD-TR
TD-TR										

Heading	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	US	II	US	OPW-TR	US	SQ	VW-TS	TD-TR
DP			DP	DP	DP	DP	DP	DP	VW-TS	DP
VW				II	VW	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
II					II	OPW-TR	II	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	VW-TS	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									VW-TS	TD-TR
VW-TS										VW-TS
TD-TR										

Acceleration	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
DP			VW	DP	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
VW				VW	VW	OPW-TR	OPW-SP	SQ	VW	TD-TR
II					STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-SP	SQ	OPW-TR	TD-TR
OPW-SP								OPW-SP	OPW-SP	OPW-SP
SQ									SQ	SQ
VW-TS										TD-TR
TD-TR										

Table 11: GoTrack significance table with reduction ratio 7

Spatial	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		US	US	US	US	OPW-TR	US	SQ	VW-TS	TD-TR
DP			VW	DP	DP	OPW-TR	DP	SQ	VW-TS	TD-TR
VW				VW	VW	OPW-TR	VW	SQ	VW-TS	TD-TR
II					II	OPW-TR	II	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	VW-TS	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									VW-TS	TD-TR
VW-TS										TD-TR
TD-TR										

Temporal	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	US	US	US	OPW-TR	US	SQ	VW-TS	TD-TR
DP			DP	DP	DP	OPW-TR	DP	SQ	VW-TS	TD-TR
VW				II	VW	OPW-TR	VW	SQ	VW-TS	TD-TR
II					II	OPW-TR	II	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	VW-TS	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									SQ	TD-TR
VW-TS										TD-TR
TD-TR										

Speed	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	VW	US	US	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
DP			VW	DP	DP	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
VW				VW	VW	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
II					II	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	VW-TS	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									SQ	SQ
VW-TS										TD-TR
TD-TR										

Heading	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	US	II	US	OPW-TR	US	SQ	VW-TS	TD-TR
DP			DP	DP	DP	DP	DP	DP	DP	DP
VW				II	VW	OPW-TR	VW	SQ	VW-TS	TD-TR
II					II	II	II	SQ	VW-TS	TD-TR
STT						OPW-TR	STT	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	VW-TS	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									VW-TS	TD-TR
VW-TS										TD-TR
TD-TR										

Acceleration	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
DP			VW	DP	DP	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
VW				VW	VW	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
II					II	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-SP	SQ	VW-TS	TD-TR
OPW-SP								OPW-SP	OPW-SP	OPW-SP
SQ									SQ	SQ
VW-TS										TD-TR
TD-TR										

Table 12: GoTrack significance table with reduction ratio 14

Spatial	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		US	US	US	US	OPW-TR	US	SQ	VW-TS	TD-TR
DP			DP	DP	DP	OPW-TR	DP	SQ	VW-TS	TD-TR
VW				VW	VW	OPW-TR	VW	SQ	VW-TS	TD-TR
II					II	OPW-TR	II	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	VW-TS	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									SQ	TD-TR
VW-TS										TD-TR
TD-TR										

Temporal	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	US	US	US	OPW-TR	US	SQ	VW-TS	TD-TR
DP			DP	DP	DP	OPW-TR	DP	SQ	VW-TS	TD-TR
VW				II	VW	OPW-TR	VW	SQ	VW-TS	TD-TR
II					II	OPW-TR	II	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	VW-TS	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									SQ	TD-TR
VW-TS										TD-TR
TD-TR										

Speed	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	VW	US	US	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
DP			DP	DP	DP	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
VW				VW	VW	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
II					II	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	OPW-TR	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									SQ	SQ
VW-TS										TD-TR
TD-TR										

Heading	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	VW	II	US	OPW-TR	US	SQ	VW-TS	TD-TR
DP			DP	DP	DP	DP	DP	DP	DP	DP
VW				II	VW	OPW-TR	VW	SQ	VW-TS	TD-TR
II					II	OPW-TR	II	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	VW-TS	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									VW-TS	SQ
VW-TS										VW-TS
TD-TR										

Acceleration	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	VW	US	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
DP			DP	DP	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
VW				VW	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
II					STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-SP	SQ	VW-TS	TD-TR
OPW-SP								OPW-SP	OPW-SP	OPW-SP
SQ									SQ	SQ
VW-TS										TD-TR
TD-TR										

Table 13: TST significance table with reduction ratio 7

Spatial	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		US	US	US	US	OPW-TR	US	SQ	VW-TS	TD-TR
DP			DP	DP	DP	OPW-TR	DP	SQ	VW-TS	TD-TR
VW				II	VW	OPW-TR	VW	SQ	VW-TS	TD-TR
II					II	OPW-TR	II	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	VW-TS	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									VW-TS	TD-TR
VW-TS										TD-TR
TD-TR										

Temporal	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	VW	II	US	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
DP			DP	DP	DP	DP	DP	SQ	DP	TD-TR
VW				II	VW	OPW-TR	VW	SQ	VW-TS	TD-TR
II					II	OPW-TR	II	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	OPW-TR	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									SQ	TD-TR
VW-TS										TD-TR
TD-TR										

Speed	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	VW	II	US	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
DP			DP	DP	DP	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
VW				II	VW	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
II					II	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	OPW-TR	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									SQ	SQ
VW-TS										TD-TR
TD-TR										

Heading	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	US	US	US	US	US	SQ	VW-TS	US
DP			DP	DP	DP	DP	DP	DP	DP	DP
VW				II	VW	OPW-TR	VW	SQ	VW-TS	TD-TR
II					II	OPW-TR	II	SQ	VW-TS	TD-TR
STT						OPW-TR	STT	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	VW-TS	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									SQ	SQ
VW-TS										VW-TS
TD-TR										

Acceleration	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
DP			DP	DP	DP	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
VW				II	VW	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
II					II	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-SP	SQ	OPW-TR	TD-TR
OPW-SP								OPW-SP	OPW-SP	OPW-SP
SQ									SQ	SQ
VW-TS										TD-TR
TD-TR										

Table 14: TST significance table with reduction ratio 14

Spatial	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	US	US	US	OPW-TR	US	SQ	VW-TS	TD-TR
DP			DP	DP	DP	OPW-TR	DP	SQ	VW-TS	TD-TR
VW				II	VW	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
II					II	OPW-TR	II	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	VW-TS	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									VW-TS	TD-TR
VW-TS										TD-TR
TD-TR										

Temporal	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	US	II	US	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
DP			DP	DP	DP	OPW-TR	DP	SQ	DP	TD-TR
VW				II	VW	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
II					II	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	OPW-TR	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									SQ	SQ
VW-TS										TD-TR
TD-TR										

Speed	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	VW	II	US	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
DP			DP	DP	DP	OPW-TR	DP	SQ	VW-TS	TD-TR
VW				II	VW	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
II					II	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	OPW-TR	TD-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									SQ	SQ
VW-TS										TD-TR
TD-TR										

Heading	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	US	US	US	OPW-TR	US	SQ	VW-TS	TD-TR
DP			DP	DP	DP	DP	DP	DP	DP	DP
VW				VW	VW	OPW-TR	VW	SQ	VW-TS	TD-TR
II					II	OPW-TR	II	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-TR	SQ	OPW-TR	OPW-TR
OPW-SP								SQ	VW-TS	TD-TR
SQ									SQ	SQ
VW-TS										TD-TR
TD-TR										

Acceleration	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
US		DP	VW	US	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
DP			DP	DP	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
VW				VW	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
II					STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
STT						OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
OPW-TR							OPW-SP	SQ	VW-TS	TD-TR
OPW-SP								OPW-SP	OPW-SP	OPW-SP
SQ									SQ	SQ
VW-TS										TD-TR
TD-TR										

Table 15: GeoLife with reduction ratio 7

Algorithm / *Metric*	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
Spatial	0	1.9593	-31.41	-16.48	-541.8	57.465	-204.4	70.998	**72.573**	72.557
Temporal	0	**53.887**	-17.90	26.034	-30.60	34.030	-60.04	48.293	39.364	49.084
Speed	0	16.129	17.262	11.098	-22.40	47.065	40.014	**58.247**	40.784	55.544
Heading	0	31.008	-6.675	16.252	0.9605	17.658	-13.57	28.017	**31.257**	28.618
Acceleration	0	19.582	18.492	16.950	17.861	36.407	**60.231**	47.930	27.987	43.136

Table 16: GeoLife with reduction ratio 25

Algorithm / *Metric*	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
Spatial	0	-26.70	-102.0	-40.42	-1766.	40.709	-308.1	48.432	**56.521**	53.692
Temporal	0	26.036	-38.53	-7.189	-127.7	13.726	-61.74	22.310	24.431	**26.430**
Speed	0	5.9416	3.2165	1.7214	-33.71	26.636	10.075	**35.618**	21.496	33.509
Heading	0	26.964	-5.246	7.0779	-12.67	8.5790	-10.82	15.095	**30.374**	19.278
Acceleration	0	4.9976	5.8879	4.1051	6.3741	9.5933	**39.650**	17.398	7.0229	12.969

Table 17: GoTrack with reduction ratio 7

Algorithm / *Metric*	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
Spatial	0	-60.19	-46.75	-72.68	-386.9	35.684	-81.11	50.188	51.035	**53.621**
Temporal	0	-1.565	-29.10	-18.27	-115.7	11.139	-60.96	27.814	27.805	**33.994**
Speed	0	-0.564	2.2388	-4.319	-31.14	29.172	22.206	**41.971**	33.223	40.149
Heading	0	**10.076**	0.5027	3.1870	-2.533	2.3245	-9.905	4.8445	4.7945	5.8999
Acceleration	0	5.3361	5.6543	3.7473	3.8255	15.527	**34.104**	25.091	17.808	21.484

Table 18: GoTrack with reduction ratio 14

Algorithm / *Metric*	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
Spatial	0	-18.05	-43.96	-43.88	-260.6	27.212	-103.5	32.987	33.561	**36.699**
Temporal	0	-0.867	-45.33	-28.02	-118.3	4.3191	-70.91	10.512	9.5965	**13.603**
Speed	0	4.8354	1.6495	-0.587	-19.78	20.678	6.2334	**28.846**	20.713	27.165
Heading	0	**9.6724**	1.4338	2.0847	-6.475	2.9394	-5.745	5.0527	6.2439	4.9210
Acceleration	0	0.3046	-0.432	-0.342	1.7660	4.3516	**16.568**	10.249	6.0539	7.4770

Table 19: TST with reduction ratio 7

Algorithm / *Metric*	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
Spatial	0	-8.151	-37.12	-20.00	-396.7	45.271	-66.86	52.999	54.096	**56.022**
Temporal	0	11.565	-12.17	6.2397	-74.50	28.487	-16.74	**30.272**	24.376	27.836
Speed	0	18.200	14.852	15.541	-36.05	36.036	31.158	46.087	**34.807**	43.007
Heading	0	**6.2634**	-2.418	-2.632	-6.343	0.8893	-7.905	2.6692	1.5896	1.5366
Acceleration	0	8.2236	8.0072	7.0783	2.4677	19.250	**38.847**	29.353	17.111	25.409

Table 20: TST with reduction ratio 14

Algorithm / *Metric*	US	DP	VW	II	STT	OPW-TR	OPW-SP	SQ	VW-TS	TD-TR
Spatial	0	-0.608	-70.82	-19.20	-639.2	32.332	-82.03	35.326	38.872	**42.245**
Temporal	0	-4.688	-37.84	-22.12	-196.2	4.0973	-27.60	**4.7402**	1.7239	3.9220
Speed	0	15.856	4.9811	9.4851	-40.15	25.190	11.175	**32.842**	24.598	30.607
Heading	0	**9.6659**	-0.736	-1.556	-12.06	3.5082	-4.481	5.0556	2.2584	3.6490
Acceleration	0	1.1062	1.0995	-0.174	2.1710	6.1258	**20.549**	13.368	6.4580	9.9283