Dear Dr. Shuai Ma,

I have received 3 reviews, included below along with a message from the handling Associate Editor, of your paper entitled "Error Bounded Line Simplification Algorithms for Trajectory Compression: An Experimental Evaluation," which you submitted for publication in the ACM Transactions on Database Systems.

These reviews, by recognized experts in the field, have obviously been prepared with care. Based on the reviews, the recommendation by handling Associate Editor Dr. Seeger, and my own assessment, I find that the paper needs to undergo a successful major revision to be acceptable for publication in TODS. As Dr. Seeger wrote to me, the reviews are quite consistent: two of them ask for a major revision, one for a minor revision. All of them agree that the survey paper is well written. Reviewer 1 rises most concerns regarding the variety of methods considered (the paper does not consider techniques where new points are used for compression), not considering map matching as a preprocessing step, error metrics, impact on applications, and the small data sets used in the experiments. The authors should carefully consider these points.

Assuming that you proceed to prepare a revision, I ask that you include comments, indicating how you have addressed each of the points raised in the reviews.

To revise your manuscript, log into https://mc.manuscriptcentral.com/tods and enter your Author Center, where you will find your manuscript title listed under "Manuscripts with Decisions." Under "Actions," click on "Create a Revision." Your manuscript number has been appended to denote a revision.

You will be unable to make your revisions on the originally submitted version of the manuscript. Instead, revise your manuscript using a word processing program and save it on your computer. Please also highlight the changes to your manuscript within the document by using colored text.

Once the revised manuscript is prepared, you can upload it and submit it through your Author Center.

When submitting your revised manuscript, you will be able to respond to the comments made by the reviewer(s) in the space provided. You can use this space to document any changes you make to the original manuscript. In order to expedite the processing of the revised manuscript, please be as specific as possible in your response to the reviewer(s).

IMPORTANT: Your original files are available to you when you upload your revised manuscript. Please delete any redundant files before completing the submission.

Please ensure that I receive the revision within 3 months, that is, by November 14, 2020. To submit your revision, please go to https://mc.manuscriptcentral.com/tods. If I do not hear from you within this period, I will assume that you do not wish to revise the paper.

Thank you for submitting your paper to ACM Transactions on Database Systems.

Sincerely,

Chris Jermaine

Editor-in-Chief, ACM TODS

----

Reviewer(s)' Comments to Author:

**Referee: 1**

Recommendation: Needs Major Revision

Comments:

C1. What is "min-# problem" mentioned in Section 1 (second paragraph)?

C2. It is stated in the paper that "The idea of piece-wise line simplification (LS) comes from computational geometry, whose target is to approximate a fine piece-wise linear curve with a coarse one (whose corresponding data points are a subset of the original one), such that the maximum distance of the former to the latter is bounded by a user specified threshold." This is actually misleading. Line simplification actually could be performed by using points other than original ones to represent the original line. The authors did mention those algorithms which consider points outside the original trajectories in the appendix. However, to make sure the correctness of the statements, it's better to mention that in the beginning of the paper. In addition, it is necessary to explain why this paper only considers the line simplification algorithms that consider the original points only.

C3. For trajectories that capture the moving objects' movement along the road network, the points are normally mapped to road networks. In other words, it is not necessary to use piece-wise line representation. It looks like the authors have not considered this case, which is actually one of the most common cases. Can the authors please explain why map-matching step is not considered even for trajectories that capture the movements in road networks?

C4. In Page 6 (line 19), it is stated "all points in the simplified trajectory belong to the original trajectory"? Why? If the error bound is the requirement, why we are not allowed to introduce new points, if it helps to improve the compression rate without violating the error bound defined? This is related to my comment C1.

C5. The three distance metrics used in this paper are mainly for line simplification. However, if we apply line simplification to compress trajectories, some of the metrics are no longer that useful. For example, perpendicular distance metric (PED) does not consider temporal dimension, which means the error bounds based on PED consider spatial distances only. However, temporal dimension is very important to trajectories. I think it is necessary to revisit the metrics and highlight their limitations. Particularly, the error-bounds provided by these metrics could be misleading as the metrics do not fully reflect the loss of information.

C6. The reason that we decide to store the trajectories (or the compressed version of raw trajectories to save storage space) is mainly to support some applications if required later, which might not require 100% accuracy. For example, in order to find out the witness of a fatal car accident, police might want to locate all the drivers who pass by a specific location within a given time window by querying the trajectories. I am not very sure whether the error bounds based on different distance metrics are still valid for specific types of queries. This survey does not consider the applications to be supported by the compressed trajectories at all. However, personally I think it is very important. The authors did include one set of experiments towards the end of the paper. However, I think it is very important to explain the criteria of trajectory compression in the beginning of the paper, to offer the audience a complete image. In this paper, what are the key considerations when we compress trajectories? Is the compression ratio the only requirement? Is it necessary to support certain applications with error bounds?

C7. The definition of Error Bounded Algorithms is not precise as the error bounds are not presented.

C8. Fig.2 in Page 7 gives an example of reachability graph. However, that graph is incomplete. For example, there is no link between P\_0 and P\_5 but they are reachable. This is very misleading.

C9. The datasets used in the experimental study were rather small. With 20M+ points, there is no need to compress them. Much larger datasets will be more realistic for the topic of trajectory compression.

C10. The average errors introduced in Page 20 are derived based on ONLY points contained in a line segment of a piece-wise line representation. Why? Why not consider all the points in the original trajectories?

C11. In Page 21 (lines 18-19), it is stated that "We then choose 10 trajectories from each dataset, and vary the size |T| of a trajectory from 1,000 points to 10,000 points while fixing the error bound ϵ = 40 metres or ϵ = 45 degrees". It is not clear how points are selected. For example, a raw trajectory contains 100 points (p1, p2, ..., p100). If we want to only consider 10 points, do you consider a sub-trajectory of 10 points (say p1, p2, ..., p10) or a trajectory of much lower sampling rate (say p1, p11, p21, ..., p91). Please state it clearly!

C11. In Page 21 (lines 44 - 46), it is stated that "The dataset collected by cars (e.g., UCar) also has better compression ratios than the datasets partially collected by individuals (e.g., Geolife and Mopsi), as cares typically move more regularly than individuals." However, this is different from our everyday observation. Because of the traffic light, cars are not expected to move so regularly. Second, it is different from the observations we could make from Figures 15, 16, 17, 18, 19, and 20. Based on the results reported in those six figures, compression ratios under Geolife and Mopsi are lower than that under UCar, which means the dataset collected by cars has poorer compression ratios. Third, it shall be 'cars' but not 'cares'.

C12. The impact of error bounds on compression ratios is straightforward. The bigger the error bounds, the better the compression ratio. However, the size of trajectory on the compression ratio is not that straightforward. Can the authors please comment on the observations we could make from Figures 18 -20?

C13. In Page 23 (lines 24 - 31), it is stated that "given the same error bound ϵ, the compression ratios of algorithms using PED are obviously better than using SED". This is actually misleading. Although the observation is that algorithms using PED could achieve higher compression ratios, the statement ignores the fact the PED does not consider the temporal dimension at all while SED does. The comparison is not fair at all! If the higher compression ratio is achieved by ignoring the error in the temporal dimension, it has to be stated clearly.

C14. In Page 27 (lines 44 - 45), it is stated that "When using DAD, the running time from the smallest to the largest is one-pass algorithms Intersect and Interval, batch algorithms TP and DP, and online algorithm OPW." However, algorithms actually perform slightly different at various datasets. It's not accurate to state that online algorithm OPW is always the worst, as it actually performs much better than TP and DP in the dataset Geolife (as shown in Figure 32(2)).

C15. When evaluating the running time of different algorithms under various error bounds, different algorithms demonstrate different trends. The authors might want to explain why algorithms change the trends in certain ways (e.g., why error bounds do not have any impact on the running time of SIPED, why TP, OPW, BQS incur longer running time as error bounds increase, why ......).

C16. Figures 33 - 38 report the average errors of where\_at queries based on trajectories compressed using different line simplification algorithms. The title of 'Evaluation of spatio-temporal queries' is misleading as only one type of queries is considered. In addition, it is necessary to consider when\_at query as it is a critical building block for many applications too.

C17. Based on the results reported in Figures 33 -38, PED and DAD are actually not proper distance metrics that should be considered when we apply line simplification to compress the trajectories and meanwhile want to preserve the utility of the compressed trajectories. Especially, Figure 21 (avg PED errors under different error bounds) and Figure 33 (avg PED query errors under different error bounds) have different trends (Figure 23 and Figure 35 have different trends too), which further demonstrates that PED and DAD are actually misleading. An algorithm that can achieve good performance in terms of PED/DAD might not be able to guarantee its performance in real applications.

C18. This paper focuses on applying line simplification for trajectory compression. Given the fact that there are different ways to compress trajectories (e.g., [1], [2]), it might be very helpful to give a brief introduction on different trajectory compression techniques, which could help audience to understand the pros/cons of applying line simplification to compress trajectories.

[1] CiNCT: Compression and Retrieval for Massive Vehicular Trajectories via Relative Movement Labeling. Satoshi Koide, Yukihiro Tadokoro, Chuan Xiao, Yoshiharu Ishikawa. ICDE'2018.

[2] COMPRESS: A Comprehensive Framework of Trajectory Compression in Road Networks. Yunheng Han, Weiwei Sun, Baihua Zheng. TODS 2017.

Additional Questions:

Relevance to Databases: High

Significance of Contribution: Marginal

Readability and Organization: High

Fusion of Theory and Practice: Marginal

Length (Relative to the useful contents of the paper): Just Right

Please help ACM create a more efficient time-to-publication process: Using your best judgment, what amount of copy editing do you think this paper needs?: Heavy

Most ACM journal papers are researcher-oriented. Is this paper of potential interest to developers and engineers?: Maybe

**Referee: 2**

Recommendation: Needs Major Revision

Comments:

This paper presents an empirical study on trajectory simplification. The authors paid great efforts to differentiate this paper from a previous experimental study published in PVLDB [41], even though both papers share the same objective to recommend suitable trajectory compression solutions for practitioners. Overall, this paper is very well written and it is enjoyable to read the paper. The existing algorithms are well summarized and explained. Multiple working examples are provided to facilitate understanding of the algorithms. The experimental findings are also clearly presented.

Detailed comments are listed in the following:

D1: In page 2, the authors mention that “important aspects of trajectory simplification (compression ratios, running time and aging friendliness) are not systematically studied” for [41]. Not sure which “important aspects” the authors refer to, as I found running time and the tradeoff between compression ratios and errors have been reported in [41].

D2: In page 3: epsilon\_1 and epsilon\_2 appeared without explanation.

D3: In page 4: It’s better to move the comparison with [41] to the motivation part, in order to assure the necessity of a new empirical study.

D4: The authors mention that one of the main contributions is to re-implement the methods with Java. So compared with the running time results reported in [41], any new or different findings are revealed?

D5: In Table 1, what are the criteria for the selection of “representative” algorithms? The authors may need to explicitly explain this in this paper.

D6: Obviously, data aging is a very important contribution in this paper, because it was not examined in [41]. However, the situation becomes very awkward when most of the existing algorithms are not data aging friendly. Among all the methods in Table 1, only DP (with SED and PED as the distance measure) is data aging friendly. If a user considers data aging is important, then he/she has no choice but to use DP. The authors may need to provide more convincing arguments for the necessity of examining data aging.

D7: In this paper, only three types of distance measure PED, SED and DAD are presented and compared. The other distance measures are briefly touched in the Appendix. Question here is that method like Dots is recommended for online compression in [41] as it shows better trade-off in terms of compression ratio and error. So when recommending users with suitable compression algorithms, would the guidance be biased if the authors exclude certain existing methods?

D8: In page 23: the authors mentioned that “in practice, SED has obviously better compression ratios than DAD..”. This is confusing to me because the distance unit for SED is meter and unit for DAD is degree. How can they be compared? It seems like the effect of epsilon=100m in SED is similar to epsilon=60 degree in DAD?

D9. Besides clarifying the differences with [41] in the introduction, it’s better to summarize the differences of experimental findings at the end of the paper, e.g., from the experimental results, what are the new insights not covered in [41] and what are the different/conflicting findings?

Additional Questions:

Relevance to Databases: High

Significance of Contribution: Marginal

Readability and Organization: High

Fusion of Theory and Practice: Adequate

Length (Relative to the useful contents of the paper): Just Right

Please help ACM create a more efficient time-to-publication process: Using your best judgment, what amount of copy editing do you think this paper needs?: Light

Most ACM journal papers are researcher-oriented. Is this paper of potential interest to developers and engineers?: Yes

**Referee: 3**

Recommendation: Needs Minor Revision

Comments:

The article presents an experimental comparison of different lossy compression algorithms for trajectories. The authors provide a taxonomy based on the algorithmic categories and distance metric. Experimental evaluations compare several aspects - compression ratio, average error, running time, aging friendliness and query friendliness - and a discussion is provided regarding the impact of each "context dimension" from the taxonomy.

The work extends the most recent similar work in [41] in several aspects. Firstly, it extends the "universe" and also considers the issue of aging. Secondly, a more robust verstion of the implementation is provided (i.e., all approaches are implemented in Java).

Overall, the paper is not too hard to read and follow, and it does a relatively decent job in terms of positioning the contributions.

However, there are certain details to which the authors should pay a bit more attention, as the may be mis-interpreted by the readership. A list of observation follows:

Sec. 4.1

The min-# optimal version from [4] has different complexities depending on whether the polyline is open or closed (as well as convex vs. concave). The authors have not properly brought this in context.

Also, in multiple places, the authors are bringing the notion of the "3D" - however, they do not mention the reference [r1] which was the first one to extend the algorithm in [4] to 3D.

Sec. 4.2

One aspect that is context-dependent (and often implicitly ignored) in online algorithms, is the communication overhead (which, in turn, may imply energy consumption in sensor networks); along with the complementary aspect of "freshness" of the compressed data in the respective sink. Please see [r2] below.

Sec. 4.3

Actually [13] does provide a kind of a taxonomy, and it would be nice to compare it with the present article. Also, [37] has a claim/proof that if the online version uses $\varepsilon$ as an error-bound in the LDR, then the final outcome is an equivalent to a compression of $2 \cdot \varepsilon$ applied to the entire historic trajectory (if it were available). This, in a sense, is a form of a justification for the use of $\varepsilon$/2 in works like CISED ([14]).

Other comments:

In several places, the authors are making statements that are not properly phrased, and some that can be taken out of context. Examples:

1. The authors should try to define terms before their first use. Example: on p.1 in the Introduction, the "min-#" is brought up, without properly explaing its meaning.

2. p.2: "...no need of extra knowledge and suitable for freely moving objects [28]...". A statement of this sort does not do justice to [28].

Namely, [28] consideres what happens with compression if the trajectories are constrained to move along road networks - but, it also is based on the motivation that in such settings there may be a different perspective - i.e., the overall-compression of the DB.

3. p.33: the authors make a claim to the effect of: "These methods lack the capability of compressing..." when describing the min-$\varepsilon$ variant. The statement needs to be clarified, or put in the more appropriate contexts: the min-$\varepsilon$ is a complementary one to min-# and, as such, it never makes any other claim that it will yield the minimum error, given a fixed "budget" of m points. However, that does not mean that this variant of the problem, and the algorithmic solution, doesn't have its own merits.

References:

[r1] Gill Barequet, Danny Z. Chen, Ovidiu Daescu, Michael T. Goodrich, and Jack Snoeyink. Efficiently approximating polygonal paths in three and higher dimensions. Algorithmica, 33(2):150–167, 2002.

[r2] Oliviu Ghica, Goce Trajcevski, Ouri Wolfson, Ugo Buy, Peter Scheuermann, Fan Zhou, Dennis Vaccaro: Trajectory Data Reduction in Wireless Sensor Networks. IJNGC 1(1) (2010)

Additional Questions:

Relevance to Databases: Marginal

Significance of Contribution: High

Readability and Organization: High

Fusion of Theory and Practice: Adequate

Length (Relative to the useful contents of the paper): Just Right

Please help ACM create a more efficient time-to-publication process: Using your best judgment, what amount of copy editing do you think this paper needs?: Moderate

Most ACM journal papers are researcher-oriented. Is this paper of potential interest to developers and engineers?: Yes

----