|  |  |
| --- | --- |
| **Subject:** | VLDB: Your manuscript entitled One-Pass Trajectory Simplification Using the Synchronous Euclidean Distance |
| **Date:** | 31 Jul 2018 11:19:14 -0400 |
| **From:** | Renée Miller [<em@editorialmanager.com>](mailto:em@editorialmanager.com) |
| **Reply-To:** | Renée Miller [<miller@cs.toronto.edu>](mailto:miller@cs.toronto.edu) |
| **To:** | Shuai Ma [<shuai.ma@gmail.com>](mailto:shuai.ma@gmail.com) |

Ref.: Ms. No. VLDB-D-18-00003

One-Pass Trajectory Simplification Using the Synchronous Euclidean Distance

The VLDB Journal

Dear Mr. Ma,

Reviewers have now commented on your paper. Some revisions have been suggested. If you are prepared to make the revisions, I would be pleased to consider publication.

The reviewers' comments can be found at the end of this email or can be accessed by following the provided link.

Your username is: shuai.ma

If you forgot your password, you can click the 'Send Login Details' link on the EM Login page at <https://vldb.editorialmanager.com/>

When revising your work, please submit a list of changes or a rebuttal against each point which is being raised when you submit the revised manuscript.

Your revision is due by 01 Sep 2018.

Please make sure to submit your editable source files (i.e. Word, TeX).

To submit a revision, go to the journal's Editorial Manager site and log in as an Author. You will see a menu item called 'Submissions Needing Revision'. You will find your submission record there.

Yours sincerely

Renée Miller

Editor-in-Chief

The VLDB Journal

## Reviewers' comments:

The reviewers uniformly agree that the paper presents a technically solid solution to an important problem. The experimental evaluation is quite thorough; and a good discussion of related work. However, the paper suffers from over-formalization that loses the intuition of the solution. In particular, the crucial sections 3 and 4 are too dense and fail to convey to readers the key technical ideas. The paper also contains numerous typos and grammatical errors. The revision should address these shortcomings. Consider providing an overview/outline of the approach first, rather than laying out a sequence of propositions. Moving some of the proofs to an appendix might be an option to improve readability.

## Reviewer #1

**Summary:**

This paper proposes a trajectory compression (simplification) algorithm that takes into account both the spatial and temporal attributes of the points. The proposed algorithm runs in one pass over the trajectory and provides a simplified version using the Synchronous Euclidean Distance. The key idea is to extend the existing sector intersection based algorithm from two dimensions (spatial) to three dimensions (spatio-temporal). The paper provides the necessary proofs that the proposed algorithm will indeed produce the desired results. The paper also provides an interesting improved regular convex polygon intersection algorithm that can speed up the polygon intersection query. Experiments on real data show that the proposed algorithms are fast and provide a good compression ratio when compared to baselines. The paper, in general, is well-written and supported with helpful figures and examples to make it easy to follow. There are some changes that will make the paper ready for

publication.

**Strong points**

S1. The problem is important and well-motivated.

S2. The paper provides both an analysis to the proposed algorithm and some implementation details.

S3. Experiments show that the proposed algorithm is both fast and efficient.

S4. The paper provides a nice related work section and covers the necessary background to understand the proposed method.

**Weak points**

W1. Some parts in the paper can be improved for readability.

W2. The experimental section needs some clarification on some points.

W3. The paper has some minor grammatical mistakes.

**Detailed comments**

D1. On page 2, the first column is a little bit confusing. Going back and forth between SED and PED might confuse the readers. The second paragraph discusses optimal PED-based algorithms that run in O(n3) and O(n2). Then it says that the O(n2) algorithm is not applicable to SED distance. The third paragraph then describes some sub-optimal algorithms for SED-based simplification. The fourth paragraph talks again about PED-based algorithms and mentions some linear-time algorithms but it's clear whether these algorithms are optimal or approximate. It is not clear either why do we need to reiterate over PED-based algorithms if they are not applicable to the addresses problem.

D2. Page 2 second column, it was a bit confusing what you mean by effective and efficient. I understood later that effective refers to the compression ratio and efficient refers to the running time. It would help to highlight this earlier or use clearer terms, e.g., faster instead of efficient.

D3. The last few paragraphs in Section 1 mention too many numbers that the reader cannot digest at that early point (all the comparison numbers to existing algorithms with different datasets.) I suggest mentioning only two or three numbers, e.g., the average speed up or the average compression ratio.

D4. In Section 2.1, it is not clear whether the Directed line segment (L), its length, and angle are in the 2D or 3D space. I assume it is in the 2D space because the angle (theta) is measured in the 2D space (x,y) but you would better highlight this part.

D5. In Section 3.1, the definition of spatio-temporal cones (C) mentions "w.r.t. a Point Ps" is this the starting point Ps or any arbitrary point? If it is always the starting point Ps, then you might better rephrase it to "w.r.t. the starting point Ps."

D6. In Section 3.4, it could be better if you mention the key idea of the proposed algorithm to help the readers understand it. For example, you can mention that the key observation is that every segment in the two polygons being intersected has to originate from one of the m edges of the regular polygon. Then you can mention how this helps skipping multiple segments at a time.

D7. In the experiments section, Figures 12 and 16 are confusing as they compare the compression ratio and error of the proposed algorithm with and without the improved polygon intersection algorithm. It looks like the proposed improvement only affects the running time but produces the same output which makes it unclear how it might affect the quality of the results. It looks like the numbers in the two figures are exactly the same with both intersection algorithms (and they had to be the same). I suggest making two lines only CISED-S and CISED-W similar to figures 13-15 and 17-19.

D8. For readability of the results, I suggest using consistent point types in all the figures. Currently, some point types are reused with different algorithms which can be confusing, e.g., CISED-S-C in Figure 12 is the same as CISED-W in Figure 13 and CISED-W-C in Figure 12 is the same as SQUISH-E in Figure 13.

D9. In Figure 14, it is not clear how the proposed CISED-W algorithm produced a compression ratio that is better than the optimal algorithm. You need to comment on this or correct it.

D10. The paper has some minor language issues that can be easily corrected with a proofread.

- "an one" => "a one"

- "to fast the ..." => "to speed up the ..."

- "they forms a ..." => "they form a ..."

- "in the executing of ..." => "in the execution of ..."

## Reviewer #2:

**Summary:**

The paper presents two lossy, single pass algorithms with bounded error for summarizing spatio-temporal trajectories using connected line segments to reduce the amount of data points

that will need to be stored in devices with limited resources.

This would also reduce the communication bandwidth required to send the data.

The proposed algorithms use the synchronous Euclidean distance as a similarity metric for spatio-temporal trajectory compression which would preserve the temporal information.

The algorithms are based on discarding the data points as long as the distance between the discarded points and the representative line segment is less than a user given threshold.

Maintaining the representative line segment is one of the major challenges in solving such a problem.

**Strong Points:**

1. The problem is important as there are many applications on mobile devices that need to store trajectory data

2. The related work seems to cover most of the existing work and nicely discusses the limitations of existing methods

3. Simple solution which improves the state-of-the-art

Three Weak Points:

1. The contribution of the paper seems to be limited; the key idea is to approximate circles intersection with polygons intersection to determine when to start a new line segment in the summarized trajectory

2. The paper is over formalized with many propositions, some of which may be incorrect

3. There are too many typos and grammatical errors.

**Detailed Comments:**

Overall the paper is hard to read and follow. The many typos and grammatical errors didn't help.

Sections 3 and 4, which constitute the core of the paper, are written as a series of propositions, some of which are not needed or are not easy to grasp.

The authors could easily remove some and also add some intuition to others.

In terms of the setting of the problem, the authors did not discuss the device range error which varies from device to device and may affect the algorithms.

In addition, it seems that the authors assume that the devices move in uniform speed which may not be realistic.

Now looking at some of the propositions.

The proof of proposition looks incorrect, take as an example a moving device along the y-axis only over the time axis. In this case, p's+i.x - Ps.x = 0 while p's+i.y - Ps.y is not zero.

Except if I am missing something, projecting a cone over a plane is not necessarily a circle.

Looking at Proposition 4, if you take only two polygons with m edges, their intersection could have more than m edges. Or are the authors doing the intersection differently.

Propositions 6 and 7 are not really needed as they follow up from the way your algorithms function.

Theorem 8 is not really needed.

Referring to Exp-1.1, I wouldn't say that CISED-S is comparable to DPSED, a ~10% difference is not small.

In almost all the experiments, the behavior of the different algorithms across all datasets is quite similar; the authors should have commented on this.

In the summary about the experiments, the authors should have discussed, based on compression ratios, average errors, and running time, which of the proposed algorithms should be used in practice under which circumstances.

In particular, looking at the average errors, how acceptable are these for the applications that will use the proposed compression algorithms. Being fast with high compression is good, but if the quality is not that good then no one will use these algorithms.

\* Few examples of the typos and the grammatical errors:

a. In th abstract "algorithms have are been"

b. Abstract "comparable with and 19.6%" --> The last sentence is a very hard to read.

c. Introduction (first paragraph) "devices have been using their sensors" --> devices use their sensors

d. Page 2 -- Column 1 -- Last paragraph "to design an one-pass LS algorithm"

e. Page 4 -- Column 1 -- "is no greater than"

f. Page 9 -- Last paragraph -- "Algorithm FastRPolyInter. The presented" --> it is the first time to talk about the algorithm so how it has been presented.

g. Many single sentence paragraph throughout the paper.

## Reviewer #3

**Summary:**

This paper proposes one-pass algorithms that compress trajectory data using the Synchronous Euclidean Distance (SED) metric. Authors show that their proposed algorithms, called CISED-S and CISED-W, are comparable to or even better than two state-of-the-art algorithms that use the SED metric, namely, DPSED (the most effective one in terms of compression ratio) and SQUISH-E (the most efficient one in terms of running time): CISED-S and CISED-W can achieve better compression ratio in a much shorter time.

The paper is technically solid, and experimental evaluation is complete. However, this paper focuses on a very specific technical problem and may not be interesting for readers that do not have background. I do not see a good way that can change this.

A couple of detailed comments are in order:

1. From the example in Fig. 1, it is really difficult to tell the difference between PED and SED. It seems that SED will use more line segments due to the given constraint enforced by epsilon. However, why is the compression using SED better than that using PED? If PED is good enough, why do we care about SED? Essentially, the motivation of leaning towards SED is never clear from a practical perspective. I think it all depends on the tolerance at the application level. So I would like the authors to evaluate some real applications on top of the compressed trajectory data, to demonstrate the benefits of using SED. At least, the authors should give some convincing arguments on that.

2. The authors should give more intuitive explanation of the semantics of "synchronized points." Maybe this has been documented in the literature, but the authors should give sufficient explanation to make this paper self-contained.

3. Examples 1, 2, 3, 4, and 5 are not illustrative: They basically explained what happened in the corresponding figures but did not explain why. So it is not helpful if readers try to understand the algorithms by reading these examples. More details should be given. For instance, in Example 1, you can give specific coordinates to certain points so that readers can verify how the SED points are computed.

4. Following my first comment, I would suggest at least comparing with one algorithm based on PED (e.g., your previous work [15]). Also, I am not convinced by just using compression ratio as the single metric for measuring effectiveness. As I mentioned, I recommend using the compressed data to evaluate some real applications.