An Optimized Trajectory Simplification by applying Quantum Minimum Searching Algorithm

*Note: Sub-titles are not captured in Xplore and should not be used

1st Ratthamontree Burimas
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Teerayut Horanont
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—As data grows every single moment nowadays, compression becomes vital to keep data as compact while maintaining its error low. Trajectory data, a locational data of moving object, is logged from GPS devices at high frequency rate unavoidably. the trajectory data are to be compressed too and a good way to do the job is trajectory simplification. Trajectory simplification simply compresses trajectory data by removing some points out. By doing so, a simplified data is obviously distorted from the orginal one. The distortion can be measure by local integral square synchronous Euclidean distance (LSSED). the LSSED collects distance of the removed points to the corresponding points that the simplified data can generate. In short, it says how much the simplified data shape different from the original one.**

The trajectory simplification become one of the hard combinatorial problem and is being challenged by many algorithms [7] but those can only achieve approximate answer. Because to achieve the exact answer by demonstrating all simplified solutions and compare them, the complexity is as high as $O(2^N)$ since trying to remove / retain all points in length N.

Today, one of state of the art in computation field is quantum computing. There many quantum algorithms that can do classical jobs i.e., searching, factorizing and etc. They can also find the same solutions while exploioting quantum specialities which makes them comsume lower complexity than classical algorithms for the same job. This paper proposes one way of trajectory simplification that can achieve the exact answer with complexity lower that the classics by bringing quantum algorithms to be a mechanism inside. It also work by demonstrating all simplified solutions and compare like the classics except the job is run by Quantum Maximum or Minimum Searching Algorithm (QUMMSA) [1]. Because the QUMMSA works in the same way by finding the location of the minimum value in an unsorted database, we replace database mapping part with the computation of LSSED. And the result shows that this integration allow us to be able find the exact answer as the classics but comsume complexity of $O(\frac{\pi}{2}\sqrt{\frac{N}{M}})$ as the quantum algorithm does

***Index Terms*—trajectory data, trajectory simplification, quantum algorithm, local integral square synchronous Euclidean distance, Quantum Maximum or Minimum Searching Algorithm**

Identify applicable funding agency here. If none, delete this.

I. INTRODUCTION

Data is being kept frequently everyday in every field of technologies. Location is the data that is usually collected for many purposes so, trajectory data becomes one of the essential data that is in use in most of fields.

Trajectory data is a sequence of point dataset which mostly contain 3 attributes those are lattitude, longitude and timestamp. Each point in trajectory data is usually produced from GPS devices. The frequency of that process is depended on their criteria and capacity of the devices. Nowadays, those GPS devices are capable to log each point in every 1 to 5 seconds. Unaviodably, the data can easily consume too much capacity unnecessarily. So, the compression is needed and one way to achieve is trajectory simplification.

Trajectory simplification is one of the way to compress trajectory data. As the trajectory data is a sequence of points, the trajectory simplification is the process of removing some points which can affect a distortion to the data as less as possible. The process of removing some points out from the trajectory data will create some error. local integral square synchronous Euclidean distance (LSSED) is a decent error criteria to be used for measuring a quality of the simplification.

The simplification in this paper is divided into 2 types. Fixed-# is a simplification that can be determined an amount of points to be retained and returns the simplified trajectory data that contains the desired amount of points with lowest LSSED and Fixed-LSSED is a simplification that can be determined a threshold of LSSED to be produced and returns the simplified trajectory data that contains lowest number of points with LSSED that does not exceed the desired threshold.

Both types can be considered as a combinatorial problem in mathamatics. The computation time of finding the finest solution is $O(2^N)$ as compute all possible solutions and find the one with the most satisfactory to the requirement. That is called “exact solution”. To consume lower computation time with the result close to the finest one, we go for “approximate solution”

This paper proposes algorithms for approximate solution of both Fixed-# and Fixed-LSSED simplifications. By exploiting Quantum Minimum Searching Algorithm (QMSA) [1] that is able to find the best solution by looking for the solution that can minimize an error metric. This allows us to find the finest solution while comsuming lower computation time by bringing specialities of quantum computing.

II. BACKGROUND

A. Trajectory Data

Trajectory Data is a locational data of a single object based on sequence of time. Normally, trajectory data contains trajectory points and each point has 3 attributes which are latitude, longitude and timestamp.

$$p = [p_1 ... p_n] \tag{1}$$

where, p is a trajectory data contains trajectory points p_i

$$p_i = (\text{latitude}, \text{longitude}, \text{timestamp}) \tag{2}$$

Fig. 1 shows a trajectory data contains 10 trajectory points $p_1 = (1,3,1)$ $p_2 = (2,3,2)$ $p_3 = (3,3,3)$ $p_4 = (4,2,4)$ $p_5 = (5,1,5)$ $p_6 = (5,1,6)$ $p_7 = (5,1,7)$ $p_8 = (5,1,8)$ $p_9 = (5,1,9)$ $p_10 = (5,1,10)$.

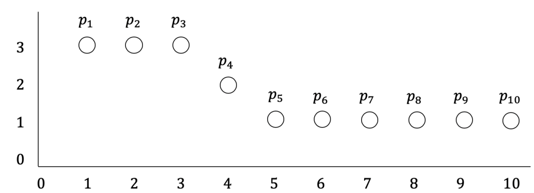

Fig. 1. Example of a trajectory data.

B. Trajectory Simplification

Trajectory simplification is a commonly used method for compressing trajectory data. By discarding some trajectory point, the distortion is produced unavoidably but also be kept as low as possible in each criteria. There are many algorithms proposed already in classical computing i.e., Top-down Time-ratio algorithm [7] that is a decent one. The algorithm is processed repetitively by selecting the furthest one point from the a single line from first and last point until reach specific threshold.

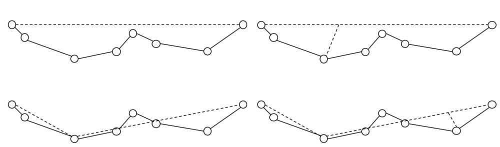

Fig. 2. Example of performing TDTR.

C. Error Metric

As mentioned in Section 2B that distortion is unavoidable when performing simplification. The error measure ment in trajectory is represented with local integral square synchronous Euclidean distance (LSSED) that can be calcualted by synchronous Euclidean distance (SED).

1) Synchronous Euclidean Distance: Synchronous Euclidean distance is a measurement of the distortion discarded point created. The SED is computed at discarded point only and is computable if there are points with previous timestamp and following timestamp existing. Those 2 points are to be computed to approximately find the point in between with the timestamp matched to the discarded point. The distance the found point and the discarded point is called SED.

Fig. 3 shows that p_i is discarded so, we do not know exactly where is the location at timestamp t_i. But we can find the point with the corresponding timestamp by the previous point p_s and following point p_e. Then we found p_i' that is believed to be where p_i is but is not. The distance between them represented as frequent-dashed line is SED. The coordinate of the point on the retained line with timestamp corresponding to the discarded point can be computed by,

$$p_i' \text{ is located at } (lat_i', lon_i') \tag{3}$$

,

$$p_s \text{ is located at } (lat_s, lon_s) \tag{4}$$

and

$$p_e \text{ is located at } (lat_e, lon_e) \tag{5}$$

where

$$lat_i' = lat_s + \frac{t_i - t_s}{t_e - t_s} \cdot (lat_e - lat_s) \tag{6}$$

$$lon_i' = lon_s + \frac{t_i - t_s}{t_e - t_s} \cdot (lon_e - lon_s) \tag{7}$$

and SED is computed by distance from p_i to p_i',

$$SED(p_i, p_j') = \sqrt{(lat_i - lat_i')^2 + (lon_i - lon_i')^2} \tag{8}$$

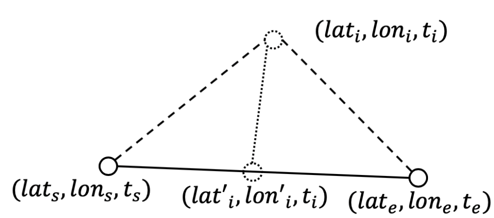

Fig. 3. Example of SED calculation.

2) LSSED: Local integral square synchronous Euclidean distance is a summation of all SED along the trajectory data T to the trajectory data T'.

$$LSSED(T,T') = \sum_{i}^{j} SED(p_i, p'_i) \tag{9}$$

Note that if p_i is a retained point as it is in both T and T', $SED(p_i, p'_i)$ equals to 0.

Fig. 4 shows that the dashed-line points are discarded so, SED can be computed from each of them. The solid line drawn through the retained point is used to find points with timestamp corresponding to those discard point. By summing up all SED in frequent-dashed line, it returns LSSED value. Apparently, our goal is to find T' that result the most satisfying LSSED with 2 criterias of Fixed-# and Fixed-LSSED.

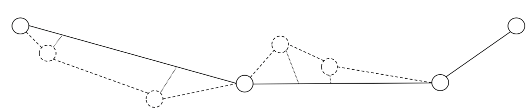

Fig. 4. Example of a LSSED.

D. LSSEDSTATE function

LSSEDPAIR function is a special function proposed in this paper to be used in simplifications. LSSEDSTATE is a summation of SED like LSSED but LSSEDSTATE takes binary state as an input instead of simplified trajectory T'.

$$LSSEDSTATE(T,y) = \sum_{i=1}^{N} SED(p_i, p'_i) \tag{10}$$

where y is a binary state with length N. $y_i = 0$ represents trajectory point at index i is discarded and $y_i = 0$ represents trajectory point at index i is retained. p'_i is computed by the closest previous retained point p_s and the closest following retained point p_e Eq. 3. $SED(p_i, p'_i)$ is only computable where $y_i = 0$ which means p_i is a discarded point. So $SED(p_i, p'_i)$ is equal to 0 if $y_i = 1$ since p_i and p'_i are the same point.

E. Quantum Minimum Searching Algorithm

Quantum Minimum Searching Algorithm (QMSA), originally called Quantum Maximum or Minimum Searching Algorithm in [1], is an algorithm for finding minimum or maximum value in an unsorted database. In this paper, we are only focusing on finding the minimum part as we defined our goal is to minimize LSSED value.

Classically, finding minimum value in an unsorted database with length N unavoidably costs complexity of $O(N)$ because we need to explore all data before deciding which one is the minimum amongst them. But QMSA, by exploiting quantum computing specialities, can do the same thing with complexity of approximatedly $O(\frac{\pi}{2}\sqrt{\frac{N}{M}})$ where M is the number of the minimum value.

Originally, The QMSA is used to find minimum value in an unsorted database by processing a set of qubits. Let x be the set of qubits which are bits with quantum state that can be computed in quatum manners. The x represented index of value in database. Apparently, the length of the database is needed to be a power of two. The QMSA adjust x to be very close to the exact answer then, measure quantum state of x. An output of measuring qubits x is a normal binary bits with the same length. Let z be a set of those bits. z is believed to be the index of the minimum in the database.

In short, The QMSA can find index of value that is a minimum amongst others under a certain condition i.e., mapping in a database . In this paper, the QMSA is applied to find solution of trajectory simplification that can returns the most satisfying result under both Fixed-# (**??**) and Fixed-LSSED (**??**) condition.

Recall that x is a set of binaries and the QMSA is to find x that can results the smallest $f(x)$ amongst others.

The QMSA performs by followings:
1.Randomly select d_0 with some x.
2.Declare d_1 with ∞.
3.Select variable c with an integer.
4.Mark all x that less than or equal to d_0 to be 1 with oracle function and store to a mapping set F
5.Map all $F(x)$s to quantum initial state z.
6.Apply Grover-Long algorithm on the initial state and store output to z'.
7.Measure those z' and reassign to d_1.
8.If d_1 is greater than d_0, and go back to Step 5.
9.If d_1 is smaller than d_0, and go back to Step 4 loop with c iteration.
11.If d_1 is equal to d_0, return d_0 as a considerably most satisfying x.
12.Reassign d_0 with d_1, and go back to Step 4

F. Grover-Long algorithm

Grover-Long algorithm [2] is a very well-known algorithm in a quantum computing field that takes quantum state as input and processed to bend the one that is marked to have the highest measurable possibility. And return a the processed quantum state as an output.

There are a lot of detail about the algorithm and most of them are described in quantum aspect. In short, after looking at a mapping of marked or unmarked for each state, this algorithm takes quantum initial states which has the same measurable possibility and adjust possibility of the unmarked state to be measured at 0% and the marked state to be measured at 100%.

III. Proposed Method

There are 2 main requirements in all types of compression which are comprssion ratio and error rate. Apparently, In our case, compression rate is ..... and error metric is LSSED of the whole simplified trajectory.

A. State the problem

Taking an input as a trajectory data T that contains n trajectory points.

$$T = [t_1...t_n] \tag{11}$$

Each point t_i has 3 attributes which are latitude, longituge and timestamp. The output of the process will be T' which contains m points where $m \leq n$.

$$T' = [t'_1...t'_m] \tag{12}$$

The T' is considered as one with desired amount of retained points resulting lowest LSSED as possible in Fixed-# or having LSSED lower than desired threhold with lowest amount of points as possible in Fixed-LSSED.

B. Formulate the problem

Consider trajectory data T with size n. Trajectory points in T at index 1 and n are obviously unremovable because SED as they are removed is not computable. So, those 2 are fixed as retained and only points in the middle from index 2 to $n-1$ will be further decided to be retained or discarded so, we mark points at index 1 and n as set and let x be a serie of n digit binaries to represent selection of each point from index 1 to n, where bit 1 means that point is retained and bit 0 means that point is discarded. Let x_i represents selection of point in T at index i.

$$x = x_1x_2...x_n \tag{13}$$

if $x_i = 1$, point in T at index i is retained. if $x_i = 0$, point in T at index i is discarded. Note that $x_1 = 1$ and $x_n = 1$. T' can be restated as

$$T' = [t'_1...t'_m] \tag{14}$$

where,

$$t'_1 = t_1, t'_m = t_n \tag{15}$$

and

$$\text{for k = 2 to n-1 : if } x_k = 1, t_k => t'_{\text{from 2 to m-1}} \tag{16}$$

The T''s LSSED can be computed by :

$$T'_{LSSED} = LSSED(T, T') \tag{17}$$

where $LSSED$ is a function measuring error trajectory data T' to a trajectory data T by summing up SED created from the points T' discarded.

C. Solve the problem

Let T_1 contains n trajectory points. That causes x to contain n binaries.

First, we cast x:

$$x = x_1x_2...x_n \tag{18}$$

,

$$x_1, x_n = 1 \tag{19}$$

and

$$x_2...x_{n-2} \in \{0, 1\} \tag{20}$$

Which means that if $x_i = 1$ (Let it called set x_i) , point in T_1 at index i is retained. if $x_i = 0$ (Let it called unset x_i), point in T_1 at index i is discarded. Note that $x_1 = 1$ and $x_n = 1$.

D. Solve the problem: Quantum Part

The QSMA [1] play a crucial role from this point. Since we know that $x_1, x_n = 1$ so, we can economize cost of 2 qubits by taking only $x_2...x_{n-1}$ to be processed.

Let z be a set of qubits representing $x_2...x_{n-1}$

$$z = z_1z_2...z_{n-2} \tag{21}$$

where z_i is for x_{i+1}

At this understanding, we can talk about our 2 goals according to 2 simplifications this paper proposes and the QSMA is used to find the most satisfying x' for each.

1) Fixed-#: Let w be a desired number of retained points. Recall that 2 x_is are fixed-set bit so, $w \geq 2$. The QSMA is to find x' that can result lowest LSSED where number of set x' is w.

The QSMA performs to achieve Fixed-# by followings:
1.Randomly select d_0 with some x state that satisfies w set bits i.e., $d_{0,2}...d_{0,2+(w-2)-1}$ to be set bit and $d_{0,2+(w-2)}...d_{0,n-1}$ is unset, $d_{0,1}$ and $d_{0,n}$ are fixed-set so we leave 2 set bits to them.
2.Declare d_1 with ∞.
3.Select variable c with some positive integer.
4.Design an oracle function that takes d_0 as input and returns a mapping set of all xs to a marking binary. Mark with 1 if that $LSSEDSTATE(T, x)$ is less than $LSSEDSTATE(T, d_0)$ and satisfy w set bits, Unmark with 0 otherwise.
Since 2 bits of x are fixed. The length of the output will be 2^{N-2} for all possible $N-2$ binary combination.
5.Perform the function in Step 4 and store it to F.
6.Initialize quantum state with $N-2$ qubits which can represent value up to 2^{N-2}. Since there are 2 fix-set bits, we can economize 2 qubits.
7.Map all $F(x)$s to the quantum initial states. The 2 economized qubits create no effect because the F also map with 2 fixed-set bit.
8.Apply Grover-Long algorithm on the quantum state. This returns one z that is believed to be one of the state marked by the oracle funtion F and can result lowest LSSED amongst others z.
9.Measure quantum state z will result an unfinished $x'_2...x'_{n-1}$.
10.Construct that $x'_2...x'_{n-1}$ to the fix-set x_1,x_n and store to d_1 .
11.If $LSSEDSTATE(T, d_1)$ is greater than $LSSEDSTATE(T, d_0)$, it means this d_1 is a mistake and we go back to Step 5.
12.If $LSSEDSTATE(T, d_1)$ is less than $LSSEDSTATE(T, d_0)$, it means this d_1 is a good progress and we assign d_1 to d_0 and go back to Step 4 for c iteration.
13.If $LSSEDSTATE(T, d_1)$ is equal to $LSSEDSTATE(T, d_0)$, it means this d_1 is considerably the

one that has minimum LSSED. We break the loop and return d_0 as an output.

The pseudocode of the algorithm is shown in Algor. 1.

Algorithm 1: Fixed-#

Result: x'

Let w be a desired number of retained points;
Let T be a trajectoty data with length N;
Let d_0 be an empty binary bits with length N;
Set the first and last bits of d_0 with 1;
Set $d_{0,2}...d_{0,2+(w-2)-1}$ bits with 1;
Set $d_{0,2+(w-2)}...d_{0,n-1}$ bits with 0;
Let c be some positive integer;
Let $d_1 = \infty$;
for $i = 1 \to c$ **do**
  Perform an oracle to mark all solutions that have w set bits and result LSSED less than d_0 does, then store to array F;
  while $LSSEDSTATE(T, d_1) > LSSEDSTATE(T, d_0)$ *or* $d_1 = \infty$ **do**
    Initialize quantum state z with length $N - 2$ for $d_{0,2}...d_{0,n-1}$;
    Map F to that quantum initial state;
    Apply Grover-Long algoritm on the quantum state;
    Measue the quantum state z and store to $x'_2...x'_{n-1}$;
    Construct x' with set bit at index 1 and n and store to d_1;
  end
  if $LSSEDSTATE(T, d_1) < LSSEDSTATE(T, d_0)$ **then**
    Reset i to 1;
  end
  else if $LSSEDSTATE(T, d_1) = LSSEDSTATE(T, d_0)$ **then**
    Return d_0;
  Set d_0 with d_1;
end
Return d_0;

2) Fixed-LSSED: Let b be a LSSED threshold. The QSMA is to find x that can result LSSED not exceeding b where number of set x_i is minimized.

The QSMA performs to achieve Fixed-LSSED by followings:
1.Randomly select d_0 with some x state that satisfies b set bits i.e., all $d_{0,i}$ to be set bits which will make $LSSEDSTATE(T, d_0) = 0$ which surely satisfies b.
2.Declare d_1 with 0.
3.Select variable c with some positive integer.
4.Design an oracle function that takes d_0 as input and returns a mapping set of all xs to a marking binary. Mark with 1 if that $LSSEDSTATE(T, x)$ is less than b and that x contains set bit less than or equal to d_0, Unmark with 0 otherwise. Since 2 bits of x are fixed. The length of the output will be 2^{N-2} for all possible $N - 2$ binary combination.
5.Perform the function in Step 4 and store it to F.
6.Initialize quantum state with $N - 2$ qubits which can represent value up to 2^{N-2}. Since there are 2 fix-set bits, we can economize 2 qubits.
7.Map all $F(x)$s to the quantum initial states. The 2 economized qubits create no effect because the F also map with 2 fixed-set bit.
8.Apply Grover-Long algorithm on the quantum state. This returns one z that is believed to be one of the state marked by the oracle funtion F and contain set bit less than or equal to d_0
9.Measure quantum state z will result an unfinished $x'_2...x'_{n-1}$.
10.Construct that $x'_2...x'_{n-1}$ to the fix-set x_1,x_n and store to d_1 .
11.If d_1 contains set bit more than d_0 does, it means this d_1 is a mistake and we go back to Step 5.
12.If d_1 contains set bit less than d_0 does, it means this d_1 is a good progress and we assign d_1 to d_0 and go back to Step 4 for c iteration.
13.If d_1 contains set bits that equals to d_0 does, it means this d_1 is considerably the one that has minimum set bit with LSSED not exceeding b. We break the loop and return d_0 as an output.

The pseudocode of the algorithm is shown in Algor. 2.

E. Obtain the answer

After performing those algorithm according to 2 simplications, we obtain d_0 that is considerably the finest solution x' that can satisfy Fixed-# or Fixed-LSSED the most amongst other x's.

$$x' = x_1 x'_2...x'_{n-1} x_n \tag{22}$$

F. Evaluation

An evalution is achieved with quantum simulator provided by Huawei's HiQ Simulator written in Python 3.5. Code is available on $github$. An environment under the evaluation is Ubuntu debian 18.04 Operating system, 16-core CPU, 32-GB of ram computer.

As parameters stated in the previous section, we demostrate trajectory simplification with 2 criterias for each simplification.

a) Fixed-# Evaluation: Let $w = 5$ and T be a trajectory data of length 10. QMSA is to find solution with minimum LSSED and keeping 5 points as retained out of 10 points. So, $x = [x_1...x_{10}]$ where $x_1 = 1$, $x_{10} = 1$ and $x_2...x_9$ is optimized by QMSA. Let $c = 5$. The result is shown in [Table. I] where "#iteration" means a number of iterations, "#satisfying condition" means a number of time that x' contains w retained points and "#finest solution" means a number of time that x' contains w retained points and result $LSSEDSTATE(T, x')$ lowest amongst others possibilities.

Algorithm 2: Fixed-LSSED

Result: x'
Let b be an LSSED threshold;
Let T be a trajectoty data with length N;
Let d_0 be an empty binary bits with length N;
Set the all bits of d_0 with 1;
Let c be some positive integer;
Let $d_1 = 0$;
for $i = 1 \to c$ **do**
  Perform an oracle to mark all solutions that have LSSED not exceeding b and contain set bits less than or equal to d_0 does, then store to array F;
  while *(set bits of d_1 > set bits of d_0) or $d_1 = 0$* **do**
    Initialize quantum state z with length $N-2$ for $d_{0,2}...d_{0,n-1}$;
    Map F to that quantum initial state;
    Apply Grover-Long algoritm on the quantum state;
    Measue the quantum state z and store to $x'_2...x'_{n-1}$;
    Construct x' with set bit at index 1 and n and store to d_1;
  end
  if *set bits of d_1 < set bits of d_0* **then**
    Reset i to 1;
  end
  else if *set bits of d_1 = set bits of d_0* **then**
    Return d_0;
  Set d_0 with d_1;
end
Return d_0;

TABLE I
FIXED-# EVALUATION

Trajectory Data	#iteration	#satisfying condition	#finest solution
Geolife	100	100	78
Taxi	100	100	80
Truck	100	100	82
Walk	100	100	68

b) Fixed-LSSED Evaluation: Let $b = 0.00125$ and T be a trajectory data of length 10. QMSA is to find solution with LSSED not exceeding the threshold b while minimizing a number of retained points as lowest. So, $x = [x_1...x_{10}]$ where $x_1 = 1$, $x_{10} = 1$ and $x_2...x_9$ is optimized by QMSA. Let $c = 5$. The result is shown in [Table. II] where "#iteration" means a number of iterations, "#satisfying condition" means a number of time that x' results $LSSEDSTATE(T, x')$ not exceeding b and "#finest solution" means a number of time that x' result $LSSEDSTATE(T, x')$ not exceeding b and contains lowest retained points amongst others possibilities.

TABLE II
FIXED-LSSED EVALUATION

Trajectory Data	#iteration	#satisfying condition	#finest solution
Geolife	100	80	80
Taxi	100	83	83
Truck	100	85	85
Walk	100	83	83

CONCLUSION AND FUTURE DEVELOPMENT

This paper proposes 2 algorithms for 2 simplification types Fixed-# and Fixed-LSSED. Classically, to achievement the best satisfying result for both types, the complexity of $O(2^{N-2})$ is needed where N is a number of trajectory point. By exploiting quantum mechanics from the QSMA, we can do the same job by comsuming only the complexity of $O(\frac{\pi}{2}\sqrt{\frac{2^{N-2}}{M}})$ where M is a number of finest solution.

Due to the current lack of quantum property, we can not provide a high number of qubits to represent solutions for trajectory data with high number of points. When qubits are more available, this can simply apply to a bigger trajectory data. The 100-iteration evaluation result shows that the solutions satisfy conditions almost 100% and is the one that is possibly the best around 80%.

Since the lack of qubits does not affect the algorithms but shorten the length to represent the solutions only, this can apply for a bigger trajectory data if the solutions can enough map to the answer i.e., to fix size of desired retained points and map only those satisfying answer to smaller quantum qubit length. This can utilize more power of the algorithms. Apparently, The QSMA is not usable only for searching anymore, but can also be for other kinds of combinatory if it is adjusted appropriately. we may suggest to try exploring other possibilities by exploiting the QSMA or our algorithms.

ACKNOWLEDGMENT

The preferred spelling of the word "acknowledgment" in America is without an "e" after the "g". Avoid the stilted expression "one of us (R. B. G.) thanks ...". Instead, try "R. B. G. thanks...". Put sponsor acknowledgments in the unnumbered footnote on the first page.

REFERENCES

[1] Y. Chen, S. Wei, X. Gao, C. Wang, J. Wu and H. Guo, "An Optimized Quantum Maximum or Minimum Searching Algorithm and its Circuits," arXiv:1908.07943 [quant-ph] 21 Aug 2019.
[2] G. L. Long, "Grover algorithm with zero theoretical failure rate," arXiv:quant-ph/0106071, 13 Jun 2001.
[3] A. Ahuja and S. Kapoor, "A Quantum Algorithm for finding the Maximum," arXiv:quant-ph/9911082, 18 Nov 1999.
[4] C. Dürr and P. Høyer, "A quantum algorithm for finding the minimum," arXiv:quant-ph/9911082, 18 Nov 1999.
[5] M. Chen, M. Xu and P. Fränti, "A Fast Multiresolution Polygonal Approximation Algorithm for GPS Trajectory Simplification," IEEE Transactions on image processing, col. 21, No. 5, May 2012.
[6] D. Zhang, M. Ding, D. Yang, Y. Liu, J. Fan, H. T. Shen, "Trajectory Simplification: An Experimental Study and Quality Analysis," Proceedings of the VLDB Endowment, vol. 11, No. 9, Rio de Janeiro, Brazil, August 2018.
[7] N. Meratnia and R. A. de By, "Spatiotemporal compression techniques for moving point objects," In EDBT, pages 765–782, 2004.

template text is removed from your conference paper prior to submission to the conference. Failure to remove the template text from your paper may result in your paper not being published.