Future Generation Computer Systems 68 (2017) 150–162

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

An efficient online direction-preserving compression approach for trajectory streaming data

Ze Deng [a,b], Wei Han [a,b], Lizhe Wang [a,b,*], Rajiv Ranjan [c], Albert Y. Zomaya [d], Wei Jie [e]

[a] School of Computer Science, China University of Geosciences, Wuhan, 430074, China
[b] Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences, Wuhan 430074, China
[c] Newcastle University, UK
[d] The Sydney University, Australia
[e] The West London University, UK

HIGHLIGHTS

- An online compression method for trajectory streaming data is developed to preserve error-bound direction information.
- An advanced online DPTS algorithm with a BQS structure is proposed which can significantly reduce the compression time.
- A parallel method of the online DPTS+ on a GPU platform is implemented, which further improved the time efficiency.

ARTICLE INFO

Article history:
Received 29 July 2016
Received in revised form
12 September 2016
Accepted 29 September 2016
Available online 4 October 2016

Keywords:
Streaming data
Compression computing
GPU
Big data
Parallel processing

ABSTRACT

Online trajectory compression is an important method of efficiently managing massive volumes of trajectory streaming data. Current online trajectory methods generally do not preserve direction information and lack high computing performance for the fast compression. Aiming to solve these problems, this paper first proposed an online direction-preserving simplification method for trajectory streaming data, online DPTS by modifying an offline direction-preserving trajectory simplification (DPTS) method. We further proposed an optimized version of online DPTS called online DPTS$^+$ by employing a data structure called bound quadrant system (BQS) to reduce the compression time of online DPTS. To provide a more efficient solution to reduce compression time, this paper explored the feasibility of using contemporary general-purpose computing on a graphics processing unit (GPU). The GPU-aided approach paralleled the major computing part of online DPTS$^+$ that is the SP-theo algorithm. The results show that by maintaining a comparable compression error and compression rate, (1) the online DPTS outperform offline DPTS with up to 21% compression time, (2) the compression time of online DPTS$^+$ algorithm is 3.95 times faster than that of online DPTS, and (3) the GPU-aided method can significantly reduce the time for graph construction and for finding the shortest path with a speedup of 31.4 and 7.88 (on average), respectively. The current approach provides a new tool for fast online trajectory streaming data compression.

1. Introduction

Recent advances in sensing, networking, smart grid [1], smart home [2], and location acquisition technologies have led to a huge volume of trajectory streaming data (e.g., Global Positioning System (GPS) trajectories). There are three main challenges when there is such a huge volume of data: (1) storing the sheer volume of trajectory data may overwhelm available storage space, (2) the cost of transmitting a large amount of trajectory data over cellular or satellite networks can be expensive, and the large size of trajectory data makes it very difficult to discover useful patterns. Trajectory compression technologies can provide a solution for these challenges [3].

Trajectory data compression approaches are generally divided into two categories: offline or online compression [4]. The offline methods (e.g., Douglas–Peucker [5] and TD–TR [6]) discard some locations with negligible errors from an original trajectory,

* Corresponding author at: School of Computer Science, China University of Geosciences, Wuhan, 430074, China.
E-mail address: lizhe.wang@gmail.com (L. Wang).

http://dx.doi.org/10.1016/j.future.2016.09.019
0167-739X/

which is already obtained before the compression process [7]. However, in many applications the trajectory data of the moving objects arrive in a stream. These applications include real-time trajectory tracking [8] and long-term location tracking [9]. Therefore, some online compression approaches have been proposed to deal with this case. The basic idea is to use segment heuristic for trajectory and remove some points of the recently received trajectories. Representative methods include the opening window algorithm [10], dead reckoning [11], and SQUISH-E(λ) [12]. However, existing online compression approaches have some drawbacks. First, according to [13], nearly all trajectory compression approaches are position-preserving trajectory simplification (PPTS) methods. These methods lose direction information so that many applications based on location-based service (LBS) cannot be broadly supported. Although two direction-preserving trajectory simplification (DPTS) methods [13,14] have been proposed to solve this issue, these two methods are designed for offline compression. To the best of our knowledge, no online DPTS method has been proposed to date. Second, current online approaches have individual drawbacks in terms of either time costs, compression ratio, and error boundaries. For example, the opening window algorithm suffers from $O(n^2)$ time complexity [12] and dead reckoning suffers from a low compression ratio. Meanwhile, the high efficiency of compression is a key requirement for current online trajectory compression methods because the volume and density of streaming data have been rapidly growing. Therefore, there is a need for an efficient direction-preserving compression approach for trajectory streaming data.

To address these research challenges, we propose an online direction-preserving trajectory compression method for trajectory streaming data that modifies an offline DPTS method [15], after which a data structure called a bounded quadrant system (BQS) [9,16] is used to optimize our compression method. Furthermore, a modern GPU platform is used to improve the performance of our trajectory compression method.

The main contributions of this study are as follows:

1. We have developed an online trajectory compression method for trajectory streaming data called online DPTS that preserves error-bound direction information and has a high compression ratio.
2. We have introduced an advanced online DPTS algorithm called online DPTS$^+$ with a BQS structure in [9,16] can significantly reduce the compression time of online DPTS.
3. We designed a parallel method of the online DPTS$^+$ on a GPU platform, which further improved the time efficiency of trajectory streaming data compression. The GPU-aided method accelerate the SP-theo algorithm in the online DPTS$^+$, with two well-designed GPU parallel schemes.
4. We performed extensive experiments to evaluate the proposed methods using real trajectory datasets.

To the best of our knowledge, the proposed compression method is the first online trajectory compression method that takes both direction preserving and parallel processing into consideration.

The remainder of this paper is organized as follows: Section 2 discusses work relating to online trajectory compression. Section 3 introduces our online trajectory compression approaches (i.e., online DPTS and online DPTS$^+$). Section 4 describes the GPU-aided compression approach on a modern GPU platform. Section 5 presents the experiments and performance evaluation results of the proposed approaches. Section 6 concludes with a summary and a plan for future work.

2. Related work

A number of successful attempts have been made regarding online trajectory compression. The most salient works are described.

Opening window (OPW) is a kind of traditional online trajectory compression algorithm. Such algorithms, including NOWA and BOPW [10], slide a window over the points on the original trajectory to approximate each trajectory using the number of points in the window so that the resulting spatial error is smaller than a bound. This process is repeated until the last point of the original trajectory is processed. The worst-case time complexity of OPW is $O(n^2)$. Opening window time ratio (OPW-TR) [10] extended OPW using a synchronized Euclidean distance (SED) error instead of spatial error.

Some fast online trajectory compression algorithms have been proposed to overcome the high time overheads of OPW and OPW-TR. These include uniform sampling [17] and dead reckoning [11]. The uniform sampling method carefully selects a few points to store and discards the remaining points at every given time interval or distance interval. Dead reckoning stores the location of the first point and the velocity at this point. It then skips every subsequent point whose location can be estimated from the information about the first point within the given SED value until it finds one point whose location cannot be estimated. The location of the point and the speed at the point are stored and used to estimate the locations of following points. This process is repeated until the input trajectory ends. The computational complexity of this kind of method is $O(n)$. However, the major drawback of this kind of method is the lower compression rates compared with OPW and OPW-TR. Therefore, a few online trajectory compression methods that can ensure both a high compression ratio and low computing overheads have been presented. For example, given parameters λ and μ, SQUISH-E can ensure a compression ratio of λ while preventing the SED errors that are not beyond μ. However, this algorithm does not preserve direction information.

Significantly different from the existing online trajectory compression methods, this paper focuses on the emerging challenges of (1) the direction-preserved online trajectory compression with error boundary and high compression rate and (2) enabling a high-performance solution to maintain the computational performance of the proposed method for trajectory streaming data. The proposed compression method is the first online trajectory compression method that takes direction preserving and parallel processing into consideration.

3. Online DPTS: online direction-preserved trajectory simplification

In this section, we formulate the problem, present the details of the proposed compression algorithm, and describe the algorithm optimization.

3.1. Problem formulation

In our setting, a central server continuously collects the location points of moving objects over time. Thus, such points relating to a moving object O form a trajectory stream. Noted that, the issue of streaming inconsistency, which means the order of location points in the original stream (in input) is different from the output, may happen because the server needs to receive location information of multiple moving objects concurrently. However, in this paper, we assume that the consistency of streaming has been achieved. In the future, we will consider the issue of streaming inconsistency and attempt to employ some methods such as in [18] or [19] to reorder the input streaming data before compressing trajectories.