**GPS Trajectory Compression and Recovery based on Compressive sensing**

**Abstract:** With the extensive use of location based devices, trajectories of various kind of moving objects can be collected. As time going on, the amount of trajectory data increases exponentially, which brings a series of problems in storage, transmission and analysis. In this paper , we propose a novel trajectory compression and recovery algorithm based on

compressing sensing(CS),which takes full account of movement pattern and structure features in trajectories.The CS\_based method proposed in the paper could not only achieve a fairly high compression rate and recovery accuracy,but also reduce the error generated by GPS measurement. Moreover ,the Particle filtering(PF) technology is applied for optimizing the system performance.Comprehensive experiments on real dataset show that: the CS-based compression algorithm achieves high compression and recovery performance compared to current trajectory compression algorithms.

**Keywords:** GPS trajectory, compression and recovery,compressing sensing,Particle filtering,movement features

1. **Introduction**

In recent years, with the rapid growth of GPS-equipped mobile devices, sensor network and wireless communication technologies, various kinds of moving objects can be traced all over the world. The popularity of these devices and technologies has leading to an exponential growth in the amount of trajectory data as time going on. For instance, there are 5000 taxis in a city and we track the trajectory of each taxi by sampling its position once every 5 seconds, so we will overwhelm 2 GB of storage capacity to store a single day trajectory data. These data are the foundation for us to analyze activities and patterns for moving objects. However, the enormous volume of data has brought several problems [1]. First, it is quite expensive and time-consuming to transmit these large amounts of data. Second, it is computationally expensive operations to query and extract useful patterns from these large amounts of trajectory data. Third, GPS trajectories are often with much redundant and trivial data that waste storage and cause increased disk I/O time. These issues can be addressed by reducing the size of trajectory data. Therefore, the aim of data compression technique is to decrease the occupied memory space and improves the transmission, storage and processing by reducing data volume without obviously losing information, or by reorganizing data with certain strategies to reduce the redundancy and memory cost. For moving objects trajectories, it is essential to preserve as much features, including position, direction, corner and velocity as possible while reducing redundant sampling points.

Currently, a number of trajectory compression algorithms have been studied. In many researches, the main idea of line simplification is widely used to reduce the number of trajectory points by introducing a bounded error, which loses some information after compression [2, 3]. This kind of line simplification is mainly derived from the well-known Douglas-Peucker (DP) algorithm [4], which makes use of the divide-and-conquer approach to keep the most important points of a polyline. In order to take both spatial and temporal dimension into account, Meratnia et al. [3] replace the perpendicular Euclidean distance with Synchronous Euclidean Distance (SED) in DP algorithm, with which, compressed data is confirmed be superiority than the former ones. Besides DP algorithm, there are also various trajectory compression algorithm exists in the literature. Each offers a different trade off among compression time, compression ratio, and accuracy. Uniform sampling, which is fast and can archive the specified compression ratio by sampling trajectory at fixed time interval, but introduce large spatial and SED errors. To-Down Time Ratio (TD-TR) algorithm [3], is a variant of DP algorithm with SED instead of spatial error. It’s running time is *O*(*n2*). Opening Window (OW) algorithm [5] is an online approximate line simplification algorithm by introducing a slide window. OW algorithm runs with the window anchored at the first point, and gradually checks the forthcoming points until the spatial error is greater than the given threshold. The spatial error is the distance of the point to the line segment between the first point and the last point in the window. Then it executes iteratively until the last point of trajectory is included. The running time of OW algorithm is *O*(*n2*). Opening Window Time Ratio (OW-TR) algorithm [3] is an extension to OW algorithm which takes temporal data into account and uses SED to represent the error. Like OW algorithm, the worst running time of OW-TR is *O*(*n2*). Dead Reckoning (DR) algorithm [6] is an efficient compression algorithm that considers not only spatial dimension but also velocity information. DR algorithm firstly marks the start point *p0* as the key point, and stores *p0* and its velocity in the compressed representation. Then the next point *pi* is estimated whether it’s location within the SED threshold from *p0*. If true then continue the next point of *pi*, else *pi* is marked as the key point and stored to the compressed representation with its velocity. The DR algorithm will execute iteratively to the end of trajectory. The computation complexity of DR algorithm is *O*(*n*).

All these algorithms take the spatial and temporal information as the basis to reduce points in trajectory, and do not take trajectory movement patterns and internal features into consideration. Due to the trivialness and redundancy of trajectory data, almost all trajectory compress algorithms are lossy. When we query trajectories from database or discover the hidden knowledge from trajectories, we hope the compressed trajectory can represent their original ones well. However, if the movement pattern and internal features are neglect, applications, such as trajectory clustering [7, 8], outlier detection [9] and activity discovery [10] may be not so accuracy as we expected. Therefore, in this paper, we present a novel trajectory compression algorithm based on corner and velocity, with which, trajectory movement patterns and internal features can be retained when compressing trajectories.

In literature [7], Lee pointed out the important attributes for trajectory clustering, and in our previous work [8], we gave the formal definitions on trajectory structure. In general, trajectory structure feature can be derived mainly by the corner and velocity at sampling points. Therefore, in this paper, we take the corner and velocity as our main goal to conquer in trajectory compression.

In order to solve the problem mentioned above, a twothree-phase compression algorithm is proposed in this paper, which is called as Trajectory Simplification Algorithm based on Structure Features (SF). Firstly, SF algorithm compresses the original trajectory based on moving direction of the object; then, it simplifies the original trajectory according to the internal fluctuations in trajectory; finally, it compresses the original trajectory by the trajectory velocity.

（对本文研究内容进行简单描述）

Therefore, we should emphasize trajectory compression should satisfy the following goals [1, 2]. Firstly, compression method should identify which parts are redundant, and which are noise or outliers. Secondly, compression method should make sure how to organize the compressed data after filtering noise and removing redundant data. Finally, compression method should make sure as much as important information be obtained.

To summarize, the main contributions of this paper are as follows:

1) This paper firstly compresses trajectories based on moving direction of objects, which can better keep the outline geometrical characters of trajectories.

The velocity corner of moving objects is introduced to compress trajectory data, through which, the movement characteristics and the internal characteristic information in trajectories will be kept in detail.

2) Then, the algorithm proposed in this paper simplifies trajectories according to internal fluctuation in trajectories, which will better keep the movement pattern and structure features in trajectories.

3) Finally, this algorithm compresses trajectories by trajectory velocity, which can better keep the movement pattern in trajectories.

Based on the velocity value of moving objects, this paper proposes a method to smooth the compressed trajectory.

34) To verify the performance of VTC, we carried carry out a comprehensive comparison with other algorithms such as DP and TD-SP.

The rest of this paper is organized as follows. Section 2 introduces the related work. Section 3 describes our compression motivation and related definitions. In section 4, the VTC method including VC and VVcompression algorithm SF is introduced in detail. An evaluation of VTC SF and other algorithms is provided in section 5. Finally, Section 6 draws conclusions and points out some possible research opportunities.

**2. Related works**

**3. Motivation and related definitions**