Local Temporal Compression for (Globally) Evolving Spatial Surfaces

Xu Teng, Prabin Giri, Matthew Dwyer, Jidong Sun, and Goce Trajcevski(✉)

Department of Electrical and Computer Engineering, Iowa State University,
Ames, IA 50014, USA
{xuteng,pgiri,dwyer,jidongs,gocet25}@iastate.edu

Abstract. The advances in the Internet of Things (IoT) paradigm have enabled generation of large volumes of data from multiple domains, capturing the evolution of various physical and social phenomena of interest. One of the consequences of such enormous data generation is that it needs to be stored, processed and queried – along with having the answers presented in an intuitive manner. A number of techniques have been proposed to alleviate the impact of the sheer volume of the data on the storage and processing overheads, along with bandwidth consumption – and, among them, the most dominant is compression. In this paper, we consider a setting in which multiple geographically dispersed data sources are generating data streams – however, the values from the discrete locations are used to construct a representation of continuous (time-evolving) surface. We have used different compression techniques to reduce the size of the raw measurements in each location, and we analyzed the impact of the compression on the quality of approximating the evolution of the shapes corresponding to a particular phenomenon. Specifically, we use the data from discrete locations to construct a TIN (triangulated irregular networks), which evolves over time as the measurements in each locations change. To analyze the global impact of the different compression techniques that are applied locally, we used different surface distance functions between raw-data TINs and compressed data TINs. We provide detailed discussions based on our experimental observations regarding the corresponding *(compression method, distance function)* pairs.

Keywords: Location-aware time series · Triangulated Irregular Network (TIN) · Surface distance · Time series data

1 Introduction and Motivation

The inter-connectivity and collaboration of multiple heterogeneous smart objects enabled by the Internet of Things (IoT) [41] have spurred a plethora of novel

X. Teng—Research supported by NSF grant III 1823267.
P. Giri—Research supported by NSF grant CNS 182367.
J. Sun—Research supported by NSF-REU grant 018522
G. Trajcevski—Research supported by NSF grants III-1823279 and CNS-1823267.

S. Madria et al. (Eds.): BDA 2019, LNCS 11932, pp. 324–340, 2019.
https://doi.org/10.1007/978-3-030-37188-3_19

applications – from smart homes [28], through personalized health care [9] and intelligent transportation system [32], to smart cities [42] and precision agriculture [11]. The multitude of sensors in the devices that define individual smart objects—be it personal (e.g., smart phone and other wearable devices with GPS features) [44] or public (e.g., roadside sensors and traffic cameras) [20] – enable the generation of unprecedented volumes of data which, in turn, provides opportunities for performing variety of analytics, prediction and recommendation tasks integrating variety of sources and contexts [10].

Most, if not all, of the data values are associated with a time-stamp indicating the instant of time at which are particular value was detected. This, in turn, allows for perceiving the data as a time series [40], or even casting it as multidimensional time series [33].

Part of the motivation for this work stems from the traditional and ever-present problem when dealing with Big Data: the Volume. The most common approach to enable storage savings; faster execution time; and saving the bandwidth consumption is to rely on some form of data compression [31]. This topic has been well studied for time-series data [22] and spatio-temporal data [4,37] and many techniques have been proposed in the literature. However, there is another part of the motivation for this work – namely, in many practical applications (e.g., the ones that depend on participatory sensing [17]) – it is often the case that:

- The data is obtained from discrete sources (e.g., measuring carbon footprint or measuring precipitation at given locations/stations [29,43]).
- However, the data is used to "generate" a spatial surface that can be used to represent a continuous distribution of the phenomena of interest over the entire domain (e.g., geo-space).

Hence, the problem that we studied in this work can be succinctly stated as: *How is a spatial shape representing a continuous phenomenon based on values from discrete locations, affected by compressing the corresponding time series with the individually sensed values at each location.*

Towards that, we conducted a series of experiments that were aiming at:

1. Using different compression techniques for time series in order to generate a more compact representation.
2. Using different distance functions to asses the difference between the surfaces obtained from the raw (i.e., uncompressed) time series.
3. Compare the impact of a particular compression technique on a particular distance function.

To our knowledge, this is the first work to systematically address the impact that the compression of location-based time series has on the surface obtained from the discrete set of values from the corresponding locations.

1.1 Organization of the Paper

The rest of this paper is organized as follows: Sect. 2 provided the necessary background and describes the main settings of the problem. Section 3 presents the details of the methodologies that we used Sect. 4 gives a detailed presentation of our experimental observations, along with a discussion of the results, and comparison of advantages and disadvantages of particular involved approach. We conclude the paper in Sect. 5.

2 Preliminaries

We now provide the background for the two main (and complementary) aspects of this work, related to time series comparission and spatial surfaces representation.

2.1 Compressing Time Series Data

A *time series* typically corresponds to a sequence of values $\{t_1, t_2, \ldots, t_n\}$ where each t_i can be perceived as the i-th measurement of a (value of a) particular phenomenon. Often times, the values are assumed to be taken at equi-distant time instants and a time series database is a collection $\{T_1, T_2, \ldots, T_k\}$ where each T_j is a time series – $T_j = \{t_{j1}, t_{j2}, \ldots, t_{jn}\}$.

Time series have attracted a lot of research interest in the past 2–3 decades due to their relevance for a plethora of application domains: from economy and business (stock market, trends detection, economic forecasting), through scientific databases (observations and simulations) databases, to medicine (EEG, gene expressions analysis), environmental data (air quality, hydrology), etc. [15,19]. As a consequence, a large body of works have emerged, targeting problems broadly related to querying and mining (i.e., clustering, classification, motif-discovery) of such data [13,23,26]. One typical feature of the time series databases is that they are very large and, as such, any kind of retrieval may suffer intolerable delays for practical use. Towards that, one would prefer to use the traditional *filter* + *refine* approach, where the filtering stage uses some kind of an index to prune as much data as possible, without introducing false negatives. However, individual time series also tend to be large – thus, attempting to index them as points in n-dimensional space creates problems in the sense of "dimensionality curse". Hence, one of the first data reduction objectives in time series was to reduce the dimensionality and then use spatial access methods to index the data in the transformed space [30]. The list of desirable properties of an indexing scheme introduced in [15] are:

- It should be much faster than sequential scanning.
- The method should require little space overhead.
- The method should be able to handle queries of various lengths.
- The method should allow insertions and deletions without requiring the index to be rebuilt (from the scratch).
- It should be correct, i.e. there should be no false dismissals.

The list was augmented by two more desiderata in [24]:

- It should be possible to build the index in "reasonable time".
- The index should be able to handle different distance measures, where appropriate.

The notion of data reduction for the purpose of indexing in the context of querying (e.g., similarity search) and mining time series data falls into the category of the, so called, *representation methods*. Essentially, a representation method attempts to reduce the dimensionality of the data, while not loosing the essential characteristics of a given "shape" that it represents – and there are two basic kinds:

- Data Adaptive – where a common representation is chosen for all items in the database, in a manner that minimizes the global reconstruction error.
- Non-Data Adaptive – which exploit local properties of the data, and construct an approximate representation accordingly.

As it turned out, an important property of any representation is the one of being able to have a lower-bound when conducting the search, which would ensure the absence of false negatives/dismissals induced by the pruning [15]).

However, there is another notion brought about in the time series literature which has influenced works in clustering, mining and compressing trajectories' data – namely, the *similarity measure* (equivalently, *distance measure*). Similarity measures aim at formalizing the intuition behind assessing how (diss)similar are two series. More formally, for two time series T_1 and T_2, a similarity function $Dist$ calculates the distance between the two time series, denoted by $Dist(T_1, T_2)$, and the desirable properties that $Dist(T_1, T_2)$ should include (cf. [13]) are:

- Provide a recognition of perceptually similar objects, even though they are not mathematically identical.
- Be consistent with human intuition.
- Emphasize the most salient features on both local and global scales.
- Be universal in the sense that it allows to identify or distinguish arbitrary objects, that is, no restrictions on time series are assumed.
- Abstract from distortions and be invariant to a set of transformations.

While some of the desiderata above may be favored over the others for a particular application domain, another categorization of similarity measures, based more on the way that they treat the matching points of the two series (cf. [40]) can be specified as:

- *Lock-step measures* – the distance measures that compare the $i-$th point of one time series to the $i-$th point of another, such as the Euclidean distance and the other Lp norms.
- *Elastic measures* – ones allowing a comparison of one-to-many points (e.g., DTW) and one-to-many/one-to-none points (e.g., LCSS).

To cater to the desirable features of the distance measures, one feature was to enable time warping in the similarity computation. A well known example – DTW (Dynamic Time Warping) distance [40] is used to allow a time series to be "stretched" or "compressed" to provide a better match with another time series (i.e., a "one-to-many" mapping of the data points is allowed for as long as each data point from one series is matched to a data points from another).

In addition to the lock-step and elastic measures, other distance functions have been introduced, motivated by a particular application context. Thus, for example, a group of measures has been developed based on the *edit distance* for strings – e.g., LCSS (*longest common subsequence*) [39] which introduced a *threshold parameter* ε specifying that two points from two time series are considered to match if their distance is less than ε. Other examples include ERP distance [7] which combines the features of DTW and EDR, by introducing the concept of a *constant reference point* for computing the distance between gaps of two time series; SpADe [8], which is a pattern-based similarity measure for time series; etc.

2.2 Triangulated Irregular Networks

The main rationalé behind using TIN is two-fold:

1. They are the most popular method for building a surface from a set of irregularly spaced points [27].
2. They enable focusing on small details in highly variable input feature [14]

TIN is a representation of choice whenever a surface can be constructed from a collection of non overlapping surfaces having triangular facets [27]. In addition, TINs are capable of preserving multiple resolutions [3].

As data structure, they consist of set of vertices (x_i, y_i, z_i) that originate in 2D triangles (the (x_i, y_i) projection), and have a vertical component z_i corresponding to a value measured at (x_i, y_i). An illustration of the 2D collection of triangles and the corresponding TIN is provided in Figs. 1 and 2, respectively.

One can also perceive TINs as a special case of Digital Elevation Model which have the surface of the triangular mesh – i.e., a set of T triangles for the finite set of points S (cf. [16]), that satisfies the following three conditions:

- S are the set of vertices of T.
- Interior angles of any two triangles cannot intersect.
- If boundaries of triangle are intersected, then it should be common edge or vertex.

There are several algorithmic solutions for constructing TIN surfaces[1] and the most popular one is based on Delanuay triangulation. A distinct property of Delaunay triangulation of a given set of planar points S is that for any triplet of points $s_i, s_j, s_k (\in S)$ that are selected to form a triangle, there will

[1] The very first implementation dating back to 1973s, due to W. Randolph Franklin.

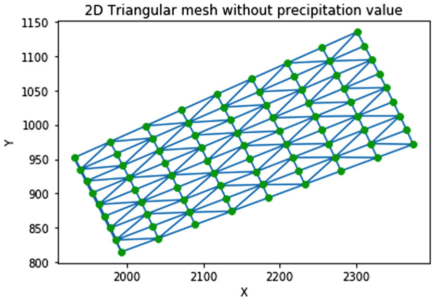

Fig. 1. 2D representation surface in TIN

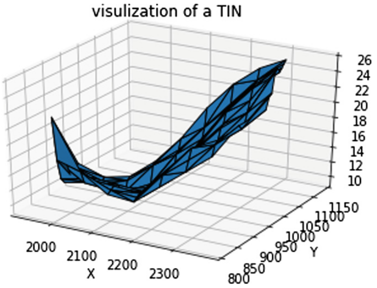

Fig. 2. 3D TIN with precipitation value

not exist a point $s_m \in S$ that will be in the interior of the circumscribed circle of $\Delta(s_i, s_j, s_k)$ [14]. Equivalently, the Delaunay triangulation maximizes the minimum angle of all the angles of triangles. An illustration of Delaunay triangluation for constructing TIN is shown in Fig. 3 (cf. [27]).

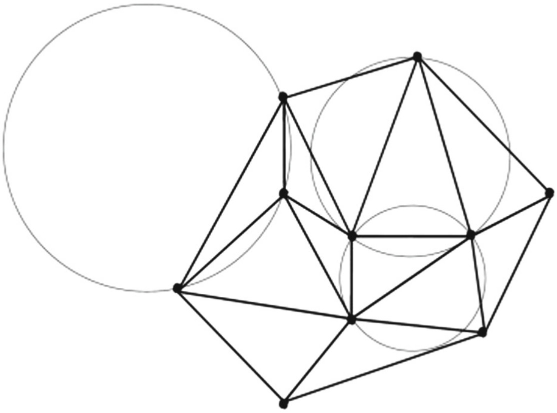

Fig. 3. Delaunay triangulation for constructing TIN

If S is the number of vertices (in our application domain, corresponding to locations of the weather stations) and b is the number of vertices on the boundary of the convex hull of all the points in S, the maximum number of triangles obtained by Delaunay triangulation would be:

$$Number\ of\ Triangles = 2 \times S - b - 2 \tag{1}$$

3 Methodology of Comparative Study

We now present the methodology used for the comparative analysis, and discuss in detail the specific approaches that we used for compression and evaluating the impact of compressing location-based time series on the global TIN.

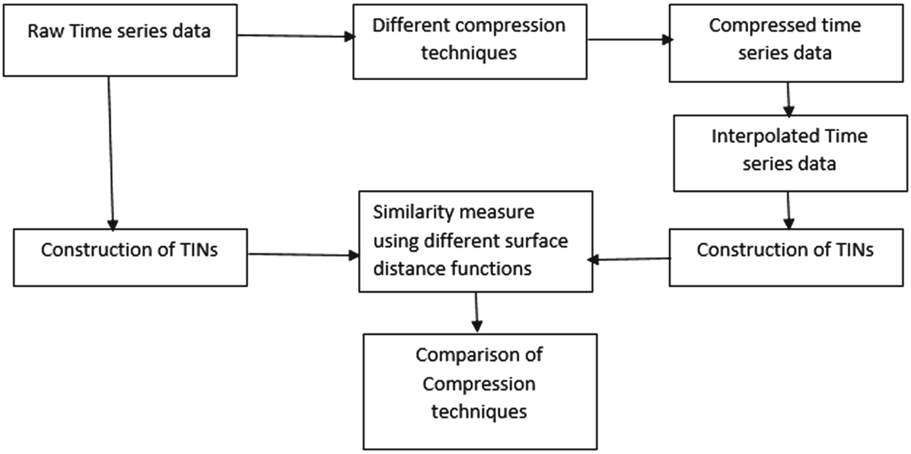
Raw Time series data
Different compression techniques
Compressed time series data
Interpolated Time series data
Construction of TINs
Similarity measure using different surface distance functions
Construction of TINs
Comparison of Compression techniques

Fig. 4. Flowchart of comparative evaluations

The workflow used is illustrated in Fig. 4. We first collected the data set of locations and measurements in each location from around the globe and categorized it into 50 different clusters based on their latitude and longitude values. The raw data was used to generate Delaunay triangulation and subsequently the TIN – i.e., a collection (or, "time series") of TINs for the time instants of each measurement in each location.

Next, we applied a compression to each of the time series containing the measurement values for the respective location and proceeded with constructing a new collection of TINs. However, due to the compression, certain points in the original time series may not be present. For such points we are using linear interpolation to generate the z-value (i.e., a measurement).

Finally, we compared the TINs at each time instant before the compression and TINs at each time instant after the compression.

3.1 Compression Techniques

In this work we used five different compression approaches which belong to two broad categories of time series compression. They consist of two dimensionality reduction techniques (Piecewise Aggregate Approximation (PAA) and Discrete Fourier Transform (DFT)), and three more native-space compression methods. The details follow

Piecewise Aggregate Approximation: The main idea of Piecewise Aggregate Approximation (PAA) [21] is to divide the original time series into N equal-size frames, where N is the desired dimensionality, and use the average value of all the data in each frame to represent each window. Mathematically, the formula of using PAA over n-dimensional time series to compress it into N dimensionality is shown in Eq. 2:

$$\bar{t_i} = \frac{N}{n} \sum_{j=\frac{n}{N}(i-1)+1}^{\frac{n}{N}i} t_j, i = 1, 2, ..., N \tag{2}$$

There is always a trade-off between compression ratio and information preservation – one extreme case would be selecting $N = n$ and the compressed representation would be identical to the original time series.

Discrete Fourier Transform: Discrete Fourier Transform (DFT) [2] is a widely-used method to find the spectrum of the finite domain signal. In theory, any n-length time series can be transformed into the frequency domain with equal number of sine and cosine waves associated with corresponding amplitudes, which enables us to inverse the transform and reconstruct the original time series. In such sense, to represent the original time series into N dimensionality, only keeping the waves with the largest N amplitudes would be a possible solution without losing too much significant information.

(Adapted) Douglas-Peucker Algorithm: Douglas-Peucker (DP) [12] algorithm is well-known for its capability of reducing the number of points while keeping the outline shape of original data. Given tolerance threshold ε, the steps of DP algorithm are as follows:

1. Construct a line segment by connecting the initiator (first point initially) and terminus (last point initially)
2. Find the "anchor" having the largest distance from the line segment and use it as the new terminus of its left part and new initiator of its right part
3. "Anchor" divides the line segment at step.1 into 2 parts and then repeat step.1 and step.2 until the largest distance in any line segment is less than ε

DP algorithm is a classical polylines compression approach. [36] adapts it into time series compression technique by considering vertical distance in stead of perpendicular. Formally, the vertical distance between a point t_k and line segment (t_i, t_j), $i < k < j$, is calculated by $|t^{'}_k - t_k|$, where $t^{'}_k$ is the intersection of the line segment and the line passing through t_k and perpendicular to time-axis.

Visvalingam-Whyatt Algorithm: "Effective area" is the key concept behind Visvalingam-Whyatt (VW) [38] algorithm, which represents the area of the triangle constructed by a point with its two neighbors. Given a sequence of time series and a error tolerance ε, the algorithm would iteratively drop the middle point of the triangle with the smallest "effective area" and updating the triangles related to that removed point until the "effective area" of any triangle is larger than ε.

(Adapted) Optimal Algorithm: Optimal (OPT) [5] algorithm considers both direction of every time series points; forward and backward. For a time series point t_i,$t_{i+1}, t_{i+2}, ..., t_n$ can be forward points and $t_{i-1}, t_{i-2}, ..., t_1$ is the backward. Here, ith pass of the algorithm draws the circle centered in every forward and backward points with the radius of ϵ. When a new point t_k in forward is being touched, such that k is $i < k \leq n$. Let t_i generates U_k and L_k as the upper and lower spectrum which defines the wedge that is related to point t_k and apex at t_i, while passing through the top and bottom of formed circle centered at point t_k. Highest and lower boundary will be maintained until the intersection of wedges is not empty and if the intersection is empty, denote the point t_k which makes the intersection empty. And then store t_i and t_{k-1} in result and repeat the steps from event point t_{k-1} to forwards and do similar for the backward part.

3.2 Distance Function

We now discuss the distance functions that we used to assess the $|TIN_{raw} - TIN_{compressed}|$, for each of the compression methods.

Hausdorff Distance: Haudsorff distance is a min-max distance measure which defines the property of similarity between two surfaces based on the positions. Hence, it is widely used as a measure of the degree of resemblance between two objects [18].

Mathematically [34], for given two set of finite points A and B, such that $A = a_1, a_2....a_n$ and $B = b_1, b_2....b_n$, Hausdorff's distance is defined as

$$H(A, B) = \max(h(A, B), h(B, A)), \tag{3}$$

where

$$h(A, B) = \max_{a \in A}(\min_{b \in B}(d(a, b))) \tag{4}$$

and

$$h(B, A) = \max_{b \in B}(\min_{a \in A}(d(b, a))) \tag{5}$$

We are going to compare the interpolated surface with the raw/original surface by finding the Hausdorff distance between them.

Volume Based Distance: The second distance function that we used is based on comparing the volumes of the TINs obtained from the original/raw time series and the TINs constructed for each time instant after compression. Volume similarity measure has been widely used as one of the standard techniques to measure the similarity between segments [35].

Each triangle of the TINs are considered as truncated triangular prism having unequal heights at a particular time. Each height is the precipitation value of the corresponding vertex in the base (i.e., coordinates of the weather stations). Figure 5 illustrates a truncated prism as a component of the TIN.

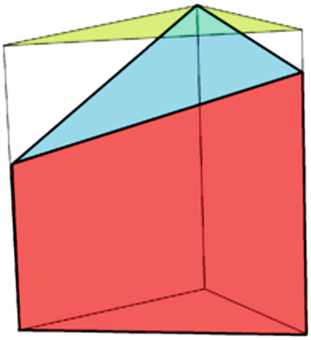

Fig. 5. Truncated triangular prism [25]

Mathematically, for a given four vertices of tetrahedron a, b, c, d, the volume is defined as:

For a given prism with a base consisting of a triplet of vertices a, b, c, and a height C, the volume is:

$$V = \frac{1}{2}|(\boldsymbol{ab} \times \boldsymbol{ac})| \cdot C \tag{6}$$

In our settings, C corresponds to the average value of the precipitation recorded in the three weather stations (i.e., $C = (height(a) + height(b) + height(c))/3$), the locations of which constitute the vertices of the triangle. Volume based distance function will show how strongly the original volume differs from interpolated volume after using compression techniques.

Angular Distance: Angular Distance is a metric which corresponds to an inverse of Cosine Similarity. Cosine similarity is used in 3D surfaces to measure the similarity between the perpendicular vectors of two corresponding triangles For a given triangle with vertices a, b, c having coordinates $(x_a, y_a), (x_b, y_b), (x_c, y_c)$, and each with corresponding height (i.e., measurement value) of z_a, z_b and z_c, the normal vector is calculated as:

$$N = (b - a) \times (c - a) \tag{7}$$

The cosine similarity for the normal vector N_1 and N_2 of two triangles is defined as [6]:

$$CS(N_1, N_2) = \frac{N_1 \cdot N_2}{||N_1|| \cdot ||N_2||} \tag{8}$$

Based on this, their Angular distance can be measured by cos-inverse of cosine similarity [6].

$$AngularDistance(N_1, N_2) = \frac{cos^{-1}(CS(N_1, N_2))}{\pi}. \tag{9}$$

4 Experimental Observations

We now present the details of our experimental observations, based on the methodology described in Sect. 3.

We note that, for reproducibility, both the source code of all the compression methods and distance functions used in the experiments, along with the datasets (before and after compression), are publicly available at https://github.com/XTRunner/Compression_Spatial_Surface.

4.1 Dataset Description

For the study, we took precipitation measurements of different weather stations across the globe [1]. Due to the geographic dispersion – i.e., having subsets of spatially co-located input points that were significantly far from other subsets of such points, we grouped the input location data into fifty clusters. The number of weather stations with precipitation measurements range from 40 to 81 across the clusters. Each location contains a time series corresponding to 50 years of monthly precipitation recordings. During the construction of the Delaunay triangulation, we converted the (*latitude, longitude*) values of the weather stations' locations into (x, y) (i.e., Cartesian) ones using ECEF (Earth Centered, Earth Fixed) methodology.

4.2 Setting of Parameters

We used multiple values for the parameters to ensure reliability and validity of our observations.

For a given dataset D represented by β_D bits, let $\mathcal{C}(D)$ denote its compressed version obtained by applying a particular compression function $\mathcal{C}$. Assume that the size of $\mathcal{C}(D)$ is $\beta_{\mathcal{C}(D)}$ bits. Then the compression ratio of $\mathcal{C}$ on D is calculated as $\mathcal{R}_{\mathcal{C}}(D) = \frac{\beta_D}{\beta_{\mathcal{C}(D)}}$.

In the experiments, the compression ratios for both DFT and PAA were set to $[10, 20/3, 5, 4, 10/3, 2]$. The rationalé is that: (a) if the compression ratio is too low, data will not be compressed much; (b) if the compression ratio is very high, it might lead to an increased loss of fine details (i.e., information).

The native-domain compression techniques (DP, VW and OPT) were implemented in such a way that they could meet a particular error tolerance. Error tolerance values were defined in a manner that ensures they are most comparable to PAA and DFT in term of compression ratio. The values for the error tolerances used in DP, VW and OPT were $[15, 25, 35, 50, 65, 80]$.

4.3 Observations

We firstly present a high-level observation regarding the impact of the compression. Namely, Fig. 6 shows the TIN corresponding to a particular cluster obtained from the raw data at a randomly chosen time instant. For comparison, in Fig. 7 we show the same cluster and at the same time instant – however, the values of the time series corresponding to the locations (i.e., vertices of Delaunay triangulation) correspond to the ones after interpolation has been applied to the compressed ones, obtained using DFT compression (z-axis indicates the precipitation values).

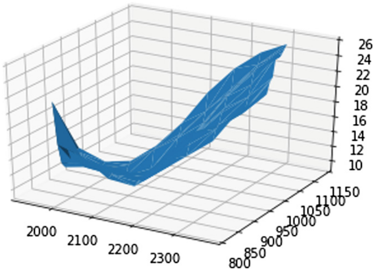

Fig. 6. 3D raw data representation of fifteenth cluster

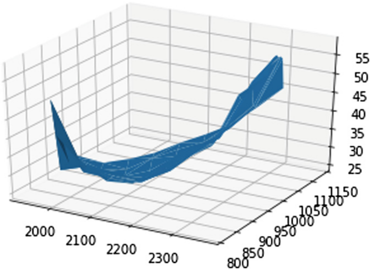

Fig. 7. 3D interpolated data representation of fifteenth cluster after DFT

For each distance measure, we firstly compute the maximum and mean value of the difference between original TINs and compressed ones in each cluster and across all the time instants. To present the results in a more general way, the average of the maximum and mean value among all the geo-clusters are calculated.

Figure 8 illustrates the effectiveness of different compression techniques in terms of Hausdorff distance measurement. Note that the x-axis represents the $1/\mathcal{R}_{\mathcal{C}}(D)$ and the y-axis represents the logarithm (with base 10) value of Hausdorff distance between the original TINs and compressed TINs. From the left side of Fig. 8, which shows the average of all the maximum Hausdorff distance in each cluster, we observe that PAA and DFT algorithms are outperformed by the other three native-domain techniques, especially when the compression ratio is

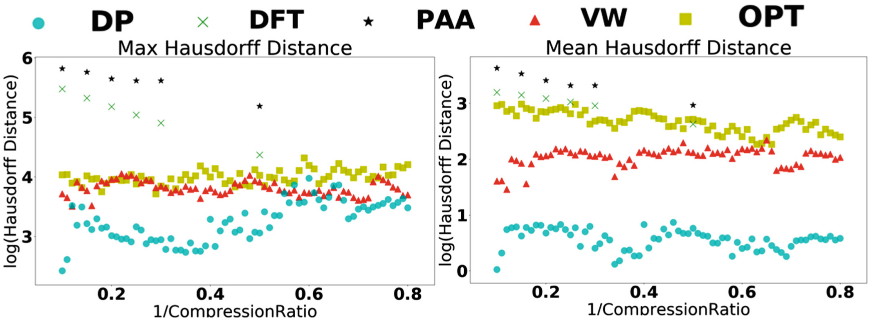

Fig. 8. Results of Hausdorff distance

relatively high. The performances of VW and OPT algorithms are very close to each other. As can be observed, the DP algorithms obtains the best performance when the compression ratio is higher than 2. The right side illustrates another different picture. By using the average of the Hausdorff distance in each cluster as measurement, DP algorithm always achieves the greatest accuracy. Different as the left side, VW algorithm has a distinguishable edge than OPT algorithm. But still, PAA and DFT has the worst performances.

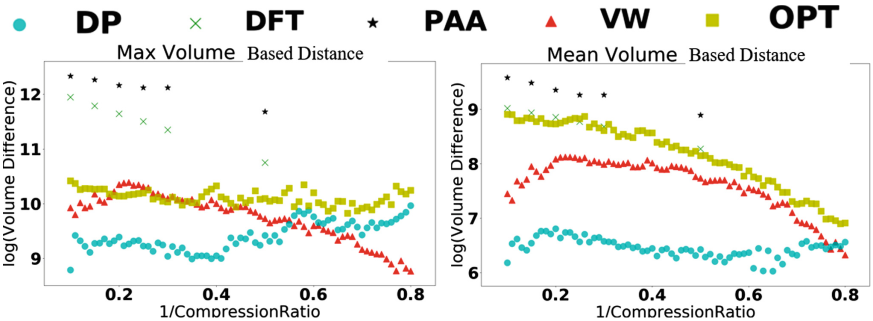

Fig. 9. Results of Volume based distance

In addition to the Hausdorff distance, we also introduce the volume based distance in Sect. 3 to evaluate the performances of different compression techniques. Compared with Figs. 8, 9 presents a different story. From the left figure, we can observe that DP algorithm is not always the best one anymore. When the compression ratio is high, DP algorithm is still able to guarantee the closest result. However, we note that the VW algorithm outperforms the DP algorithm after $1/\mathcal{R}_{\mathcal{C}}(D)$ is higher than 0.6. Thus, when the required compression ratio is higher than 2, DP algorithm should still be the first choice. But if the accuracy has a higher priority than the compression ratio, VW algorithm can be considered as a compression method of choice. Moreover, we observe that on the right side, the trend is similar to the left, i.e., the performances of VW algorithm is getting closer to DP algorithm and eventually exceeds DP algorithms for compression ratios smaller than 1.25.

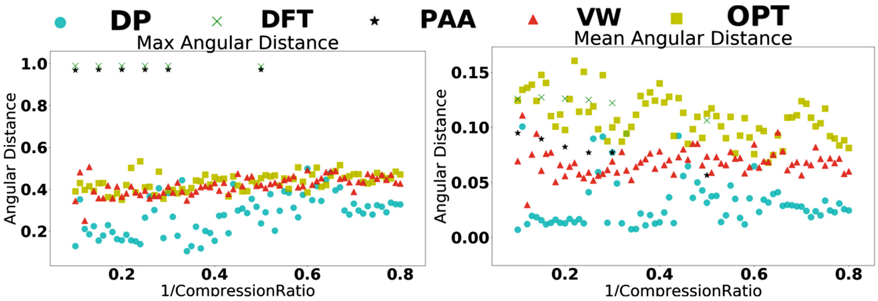

Fig. 10. Results of Angular distance

The third measurement is angular distance of the norm vectors of original and compressed TINs. As shown in the left side of Fig. 10, DP, VW and OPT algorithms have relatively similar results compared with the other two distance functions, especially when the compression ratio is between 1.5 and 2.5. Besides, PAA and DFT algorithms still have the worst performance, while the right side illustrates totally different scenarios. When the compression ratio is higher than 2.5, both PAA and DFT get much better results, which are comparable with VW algorithm and even better than OPT algorithm. And for those three native-domain compression techniques, although the performances are very close for some values of the compression ratio.

The last observation that we report is a single instance of the experiments – with a sole purpose to provide an intuitive illustration for the errors induced by the compression. Specifically, for values of the compression ratio in [0.2, 0.5], we picked the absolute worst-case scenario in terms of the Hausdorff and Volume-based distance between TINs obtained from the raw data and the TINs after the compression has been applied. The results are shown in Fig. 11.

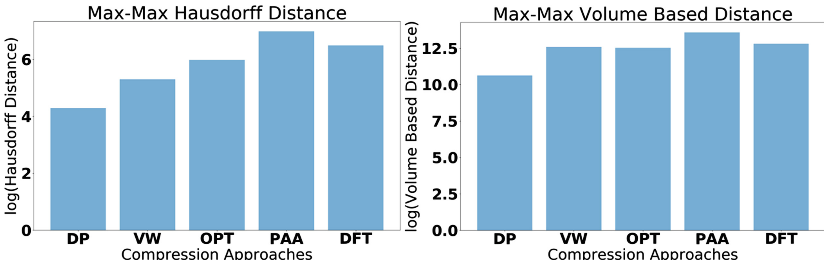

Fig. 11. Worst-case scenarios

5 Concluding Remarks and Future Work

We provided a detailed experimental comparison of the impact that compressing time series data can have when interpreted in a broader context. Specifically, we considered the settings in which each time series data is associated with a discrete location *and* the instantaneous values in each location were used to generate a TIN-based representation of the continuous surface representing a particular phenomenon of interest. When compressing the original/raw data, some of the original measurement values will not be present in the compressed version – and those are obtained by interpolation. However, using the interpolated data in the corresponding locations to generate the TIN, may yield a surface which differs from the one constructed from the raw time series. In this work, we investigated the impact that a particular compression method may have on the "distortion" of the surface, with respect to a particular distance function.

We used five different compression approaches (Discrete Fourier Transform, Piece-wise Aggregrate Approximation, Douglas-Peucker Algorithm, Visvalingam-Whyat Algorithm and Adaptive optimal Algorithm) and two different distance functions (Hausdorff and Volume-based). From among all the combinations of pairs (*compression method*, *distance function*) we made observations regarding the similarity/difference between the original TIN surfaces and post-compression ones. Our observations indicate, for example, that when it comes to volume-based distances, Douglas Peucker yielded the highest similarity and PAA showed the least similarity. Similarly, when Hausdorff distance was used to calculate the (dis)similarity, we also observed that DP was performing better than the rest. OPT and VW were showing more similarity to raw TINs inc comparison to PAA and DFT, PAA being worst.

As part of our future work, we are expanding the types of compression methods and distance functions used, for the purpose of a more complete classification of the impacts. Another one of our goals is to include applying spatial compression and combining it with the time series data compression, and evaluating the impacts on more heterogeneous datasets (e.g., time series of traffic data). Lastly, we would like to investigate the impact that the compression has on the quality of prediction in time series.

References

1. GPCC: Global Precipitation Climatology Centre. https://climatedataguide.ucar.edu/climate-data/gpcc-global-precipitation-climatology-centre
2. Agrawal, R., Faloutsos, C., Swami, A.: Efficient similarity search in sequence databases. In: Lomet, D.B. (ed.) FODO 1993. LNCS, vol. 730, pp. 69–84. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-57301-1_5
3. Bertilsson, E., Goswami, P.: Dynamic creation of multi-resolution triangulated irregular network. In: Proceedings of SIGRAD (2016)
4. Cao, H., Wolfson, O., Trajcevski, G.: Spatio-temporal data reduction with deterministic error bounds. VLDB J. **15**(3), 211–228 (2006)
5. Chan, W.S., Chin, F.: Approximation of polygonal curves with minimum number of line segments. Int. J. Comput. Geom. Appl. **6**, 59–77 (1992)

6. Chanwimalueang, T., Mandic, D.: Cosine similarity entropy: self-correlation-based complexity analysis of dynamical systems. Entropy **19**, 652 (2017). https://doi.org/10.3390/e19120652
7. Chen, L., Ng, R.T.: On the marriage of lp-norms and edit distance. In: Proceedings of the Thirtieth International Conference on Very Large Data Bases (VLDB), Toronto, Canada, 31 August– 3 September 2004, pp. 792–803 (2004)
8. Chen, Y., Nascimento, M.A., Ooi, B.C., Tung, A.K.H.: SpADe: on shape-based pattern detection in streaming time series. In: IEEE International Conference on Data Engineering (ICDE) (2007)
9. Cheng, X., Fang, L., Yang, L., Cui, S.: Mobile big data: the fuel for data-driven wireless. IEEE Internet Things J. **4**(5), 1489–1516 (2017)
10. Chudzicki, C., Pritchard, D.E., Chen, Z.: Geosoca: exploiting geographical, social and categorical correlations for point-of-interest recommendations. In: Proceedings of the International Conference On Research and Development in Information Retrieval (SIGIR), pp. 443–452. ACM (2015)
11. Deepika, G., Rajapirian, P.: Wireless sensor network in precision agriculture: a survey. In: 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS) (2016)
12. Douglas, D.H., Peucker, T.K.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica: Int. J. Geograph. Inf. Geovisualization **10**, 112–122 (1973)
13. Esling, P., Agon, C.: Time-series data mining. ACM Comput. Surv. **45**, 1 (2012)
14. ESRI: Arcgis desktop help 9.2 - about TIN surfaces (2019)
15. Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast subsequence matching in time-series databases. In: SIGMOD Conference, pp. 419–429 (1994)
16. Floriani, L.D., Magillo, P.: Triangulated irregular network. In: Liu, L., Özsu, M.T. (eds.) Encyclopedia of Database Systems, pp. 3178–3179. Springer, Boston (2009). https://doi.org/10.1007/978-0-387-39940-9_437
17. Gao, H., et al.: A survey of incentive mechanisms for participatory sensing. IEEE Commun. Surv. Tutorials **17**(2), 918–943 (2015)
18. Guo, B., Lam, K.M., Lin, K.H., Siu, W.C.: Human face recognition based on spatially weighted hausdorff distance. Pattern Recogn. Lett. **24**(1), 499–507 (2003)
19. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn. Morgan Kaufmann, Burlington (2012)
20. Jang, J., Kim, H., Cho, H.: Smart roadside server for driver assistance and safety warning: framework and applications. In: Proceedings of the International Conference on Ubiquitous Information Technologies and Applications (2010)
21. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Dimensionality reduction for fast similarity search in large time series databases. Knowl. Inf. Syst. **3**, 263–286 (2001)
22. Keogh, E., Lonardi, S., Ratanamahatana, C.A., Wei, L., Lee, S.H., Handley, J.: Compression-based data mining of sequential data. Data Min. Knowl. Discov. **14**(1), 99–129 (2007)
23. Keogh, E.J.: A decade of progress in indexing and mining large time series databases. In: VLDB (2006)
24. Keogh, E.J., Chakrabarti, K., Mehrotra, S., Pazzani, M.J.: Locally adaptive dimensionality reduction for indexing large time series databases. In: SIGMOD Conference, pp. 151–162 (2001)
25. Kern, W.F., Bland, J.R.: Solid Mensuration. Wiley/Chapman & Hall, Limited, New York/London (1934)

26. Kotsakos, D., Trajcevski, G., Gunopulos, D., Aggarwal, C.C.: Time-series data clustering. In: Data Clustering: Algorithms and Applications, pp. 357–380 (2013)
27. Liang, S.: Geometric processing and positioning techniques. In: Liang, S., Li, X., Wang, J. (eds.) Advanced Remote Sensing, pp. 33–74. Academic Press, Boston (2012). Chapter 2
28. Maselli, G., Piva, M., Stankovic, J.A.: Adaptive communication for battery-free devices in smart homes. IEEE Internet Things J. **6**, 6977–6988 (2019)
29. Mekis, E., Hogg, W.D.: Rehabilitation and analysis of Canadian daily precipitation time series. Atmos. Ocean **37**(1), 53–85 (2010)
30. Rafiei, D., Mendelzon, A.O.: Similarity-based queries for time series data. In: Proceedings ACM SIGMOD International Conference on Management of Data, SIGMOD 1997, Tucson, Arizona, USA, 13–15 May 1997, pp. 13–25 (1997)
31. ur Rehman, M.H., Liew, C.S., Abbas, A., Jayaraman, P.P., Wah, T.Y., Khan, S.U.: Big data reduction methods: a survey. Data Sci. Eng. **1**(4), 265–284 (2016)
32. Shi, D., et al.: Deep Q-network based route scheduling for TNC vehicles with passengers' location differential privacy. IEEE Internet Things J. **6**, 7681–7692 (2019)
33. Shokoohi-Yekta, M., Wang, J., Keogh, E.J.: On the non-trivial generalization of dynamic time warping to the multi-dimensional case. In: Proceedings of the 2015 SIAM International Conference on Data Mining, pp. 289–297 (2015)
34. Sim, K., Nia, M., Tso, C., Kho, T.: Chapter 34 - brain ventricle detection using hausdorff distance. In: Tran, Q.N., Arabnia, H.R. (eds.) Emerging Trends in Applications and Infrastructures for Computational Biology, Bioinformatics, and Systems Biology. Emerging Trends in Computer Science and Applied Computing, pp. 523–531. Morgan Kaufmann, Boston (2016)
35. Taha, A.A., Hanbury, A.: Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. BMC Med. Imaging **15**, 29–29 (2015)
36. Teng, X., Züfle, A., Trajcevski, G., Klabjan, D.: Location-awareness in time series compression. In: Benczúr, A., Thalheim, B., Horváth, T. (eds.) ADBIS 2018. LNCS, vol. 11019, pp. 82–95. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98398-1_6
37. Trajcevski, G.: Compression of spatio-temporal data. In: IEEE 17th International Conference on Mobile Data Management, MDM 2016, 2016 - Workshops, Porto, Portugal, 13–16 June, pp. 4–7 (2016)
38. Visvalingam, M., Whyatt, J.D.: Line generalisation by repeated elimination of points. Cartographic J. **30**, 46–51 (1993)
39. Vlachos, M., Kollios, G., Gunopulos, D.: Elastic translation invariant matching of trajectories. Mach. Learn. **58**(2–3), 301–334 (2005)
40. Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P., Keogh, E.J.: Experimental comparison of representation methods and distance measures for time series data. Data Min. Knowl. Discov. **26**(2), 275–309 (2013)
41. Whitmore, A., Agarwal, A., Xu, L.D.: The Internet of Things: a survey of topics and trends. Inf. Syst. Front. **17**(2), 261–274 (2015)
42. Yao, H., Gao, P., Wang, J., Zhang, P., Jiang, C., Han, Z.: Capsule network assisted IoT traffic classification mechanism for smart cities. IEEE Internet Things J. **6**, 7515–7525 (2019)
43. Yi, W.Y., Lo, K.M., Mak, T., Leung, K.S., Leung, Y., Meng, M.L.: A survey of wireless sensor network based air pollution monitoring systems. Sensors **15**, 31392–31427 (2015)
44. Zhuang, C., Yuan, N.J., Song, R., Xie, X., Ma, Q.: Understanding people lifestyles: construction of urban movement knowledge graph from GPS trajectory. In: International Joint Conference on Artificial Intelligence (IJCAI), pp. 3616–3623 (2017)