CET-LATS: Compressing Evolution of TINs from Location Aware Time Series (Demo Paper)

Prabin Giri
Iowa State University
Ames, Iowa

Hooman Hashemi
Iowa State University
Ames, Iowa

Evan Gossling
Iowa State University
Ames, Iowa

Jason T Guo
Iowa State University
Ames, Iowa

Koshal P Shah
Iowa State University
Ames, Iowa

Goce Trajcevski
Iowa State University
Ames, Iowa

ABSTRACT

In this paper, we present the CET-LATS (Compressing Evolution of TINs from Location Aware Time Series) system, which enables testing the impacts of various compression approaches on evolving Triangulated Irregular Networks (TINs). Specifically, we consider the settings in which values measured in distinct locations *and* at different time instants, are represented as time series of the corresponding measurements, generating a sequence of TINs. Different compression techniques applied to location-specific time series may have different impacts on the representation of the global evolution of TINs – depending on the distance functions used to evaluate the distortion. CET-LATS users can view and analyze compression vs. (im)precision trade-offs over multiple compression methods and distance functions, and decide which method works best for their application. We also provide an option to investigate the impact of the choice of a compression method on the quality of prediction. Our prototype is a web-based system using Flask, a lightweight Python framework, relying on Apache Spark for data management and JSON files to communicate with the front-end, enabling extensibility in terms of adding new data sources as well as compression techniques, distance functions and prediction methods.

CCS CONCEPTS

• **Information systems** → **Spatial-temporal systems**; **Web applications**.

KEYWORDS

Data Compression, Triangular irregular networks, Surface distance measures, Time series prediction

ACM Reference Format:
Prabin Giri, Hooman Hashemi, Evan Gossling, Jason T Guo, Koshal P Shah, and Goce Trajcevski. 2020. CET-LATS: Compressing Evolution of TINs from Location Aware Time Series (Demo Paper). In *28th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '20), November 3–6, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397536.3422352

SIGSPATIAL '20, November 3–6, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-8019-5/20/11.
https://doi.org/10.1145/3397536.3422352

1 INTRODUCTION AND MOTIVATION

The advances in sensing technologies and the proliferation of location-aware IoT devices have enabled generation of large volumes of spatial data, augmented with semantic contexts such as type of data, measurement frequencies, etc. One of the major concerns when dealing with big data is its sheer size, affecting the efficiency of scientific explorations [8], which became a part of the motivation for this work. The typical way of dealing with large datasets is to apply compression techniques, yielding benefits not only in terms of storage, but also in the efficiency of algorithms processing time as well as transmission [13].

Among the most popular data types in geo-spatial applications are TINs (Triangulated Irregular Networks), which approximate a continuous surface representing the distribution of a phenomenon of interest by using triangular facets. The 2D projections of vertices of the triangles typically correspond to locations in which measurements are taken, and the height of each vertex in the TIN corresponds to a value taken at a particular location.

In this paper we focus on specific settings in which measurements in particular locations are taken continuously over time (i.e., with some sampling frequency). Thus, the collection of measurements at a given location can be perceived as a time series. Much work has been done in terms of investigating different representation methods and distance functions for evaluating similarity of time series [14], and the impact of compression in such settings [6]. The trade-offs between different compression techniques and distance functions in terms of similarity of location-bound time series have been investigated in [12].

In this paper, we focus on investigating how different compression methods applied to location-bound time series affect the accuracy of the TIN-based representation of the evolving shapes representing the distribution of the (coverage of the) certain phenomena over a geographic region [11]. To this end, we developed the CET-LATS (Compressing Evolution of TINs from Location Aware Time Series) system which we demonstrate here. CET-LATS enables the users to test the impact of different compression methods on different shape-distance functions, thereby providing a tool for domain experts to compare and select a particular methodology that best serves their application needs. In addition to visualization capability, it also gives the option to compare the impact of compression methods on the quality of prediction.

In the rest of this paper, we overview the preliminary background in Sec. 2. Next, we discuss the overall architecture of CET-LATS in Sec. 3. The details of the steps of the demo experience for the users are presented in Sec. 4. We conclude the paper and outline directions for future work in Sec. 5.

2 PRELIMINARIES

Before describing the details of CET-LATS, we now present the necessary background.

A *time series* is sequence of values $\{v_1, v_2, \ldots, v_n\}$ where each v_i can be perceived as the measurement of a (value of a) particular phenomenon of interest at time $t = t_i$, for a given base station. A group of $\{T_1, T_2, \ldots, T_k\}$ where each T_j is a time series – $T_j = \{v_{j1}, v_{j2}, \ldots, v_{jn}\}$ is a time series database. The values in each time series sequence, in most cases, are recorded in periodically, at regular intervals [11]. Specifically, each T_i is bound to a unique location $L(T_i)$.

TIN is a collection of triangular facets used to represent (approximation of) surfaces [7], possibly in different resolutions. In our settings, we have a collection of TINs, each obtained by the values read at locations $\{L(T1), \ldots, L(T_k)\}$ for a particular time-instant.

Compression Methods and Surface Distances. We implemented different compression techniques to reduce the size of raw measurements in each location and analyzed the effects Multiple compression error tolerance/ratios and the out-turn of the compression was studied Linear interpolation to impute the data that is missing in particular time points (i.e., was eliminated during compression). The specific compression techniques that we implemented belong to two broad categories: (1) Dimensionality reduction techniques: Discrete Fourier Transform (DFT), Piecewise aggregate Approximation (PAA); and (2) Native space compression methods: Visvalingam-Whyatt Algorithm (VW), (Adapted) Optimal Algorithm (OP), (Adapted) Douglas-Peucker Algorithm (DP). The reason behind choosing these compression methods was based on their popularity and flexibility [11].

Analyzing the consequences of compression is equally important; therefore, distance metrics were chosen that can highlight the difference between raw TINS and compressed TINs in a descriptive way by comparing their surfaces. The distance functions that we implemented are Volume-based distance, Hausdorff distance, and Angular difference.

Hausdorff distance is a widely used min-max distance methods for assessing similarity between two surfaces based on their positions in terms of resemblance between two objects [5].

Volume based distance – this metric compares the volume of raw TINS with the volumes obtained using interpolated TINS from the compressed time series. It is also widely used as a technique to measure the similarity between segments based on volume preservation property [9]. In our case, each triangle in TIN is considered as a truncated prism, as shown in Figure 1. The volume of both raw truncated prism and interpolated prism constructed by interpolated data points are calculated and compared.

Angular distance – The inverse of the cosine similarity, is used as another distance metric in CET-LATS [1]. The broad use of cosine similarity in measuring 3d surface similarity between the perpendicular vectors of two corresponding triangles makes it easier to pick Angular distance to compare the TIN surfaces in CET-LATS.

Prediction Presently, we implemented Prophet, Autoregression (AR), and Autoregressive Integrated Moving Average (ARIMA).

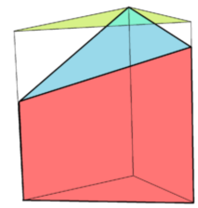

Figure 1: Truncated triangular prism

3 SYSTEM ARCHITECTURE

The Basic Architecture of CET-LATS is depicted in Figure 2, showing the three main components: *backend*, *frontend*, and the communicators – RESTful services, and the data layer (Apache Spark) communicating via SQL queries.

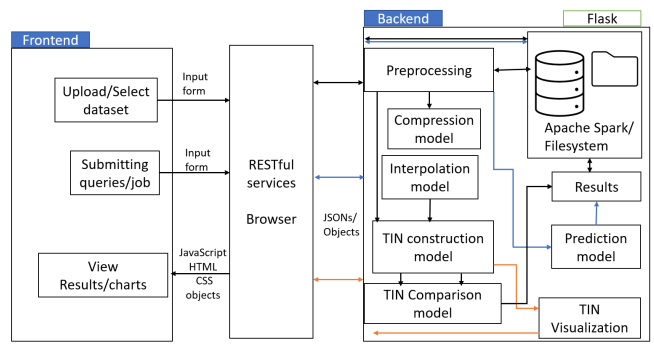

Figure 2: Architecture of the Application

Backend: Flask. The reason for choosing this python framework is because of its the most common web handling tasks such as mapping URLs, Template rendering, session management, flexible HTTP request and response handling, as well as easy to use and flexible application management [2].

Backend: Apache spark. AS a general-purpose cluster computing environment that supports SQL queries, it much faster than Hadoop and disk in terms of accessing the data. In addition, it provides a high-level API that supports different programming languages such as Java, Python, R, etc.

Frontend. This part uses HTML, CSS, and JavaScript to interact with the end-users. HTML provides the basic framework of the sites, which is then and JavaScript is is used to control and handle the presentations and behavior of different elements in web applications. CET-LATS frontend utilizes one of the libraries of Javascript, D3.js, to generate an interactive graph in the web browser. D3.js brings the data back to life with the help of SVG, HTML and CSS. The returned queries from the backend as JSON are then reconstructed into interactive visualizations in the Webpage with the help of D3 charts. The reason behind choosing D3 is its capability to customize the mapping values into graphics such as display, color [3].

We note that Flask has its own built-in template feature, Jinga, for web application, which is utilized in this system. Alongside

JSON we have used Jinga template to handle some of the backend tasks.

Information Parsing/Passing. Transmission of structured data between the server and web application needs a format of communication. A number of such formats exist and among them, two widely used messaging passing formats are XML and JSON (JavaScript Object Notation); we picked JSON because

4 DEMONSTRATION STEPS

CET-LATS features and internals are demonstrated using the dataset that incorporates daily temperature of the whole world in 2019 [4]. The final format of the data is as follows: Latitude (of the station), Longitude (of the station), and the rest of the columns are the measured temperature values for a corresponding date.

We note that the dataset, the code used for the implementation CET-LATS, as well as the code used for the different compression methods and distance function, is publicly available at: *https://github.com/Prabingiri/CET-LATS*

In the sequel, we present in detail the steps of the demo scenarios, illustrating the functionalities of the CET-LATS available for the users.

STEP 1: Selecting a region of interest. CET-LATS gives the flexibility for the end user to focus on a particular spatial range. The user can select the range of latitudes and longitudes values, creating a subset of the dataset available in the system. This feature allows the users to create a scenario for a more focused spatial investigation.

The User Interface (UI) of CET-LATS is shown in Figure 4, which also depicts the selection of the spatial range of interest.

Step 2: Selecting a compression method and distance function. CET-LATS offers the users to specify one of the presently available compression methods. Subsequently, for the chosen compression method, the user is given an option to specify the *compression ratio* (i.e., the desired size of the compressed version of the data), or the *error tolerance*. We note that a specific error tolerance may yield different output compression ratios on different input data.

Upon completion of this request, the backend systems checks the dataset in its Apache module and proceeds with the basic activities – e.g., construction of TINS and compression of the time series data in the qualified location (cf. Step 1). Next, the interpolation is applied for the values in time instants in which the original data was eliminated during the compression.

When the user selects a particular distance function to examine the impact of the chosen compression method and parameters, CET-LATS proceeds with generating the data and preparing the graphs for visual comparison. At present, the output for visually comparing the result, provide the values of min and max – in terms of which time instants generate the smallest and the largest discrepancy between the TIN from the original data and the TIN from the compressed data (for the chosen distance function). Users will also see the mean discrepancy, averaged over all the time instants.

An illustration of a possible output for the user from this step is shown in Figure 3.

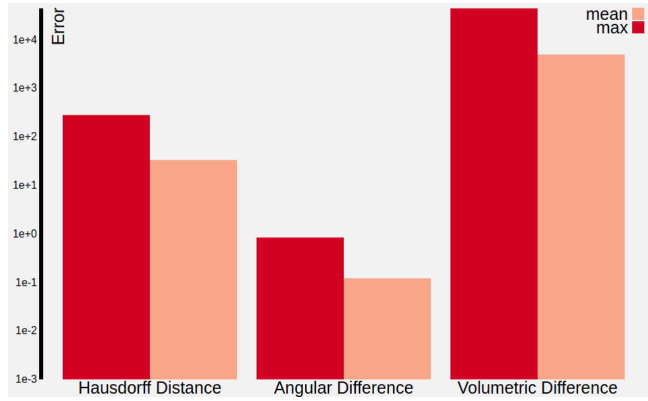

Figure 3: Visualization of the errors with different surface distance for PAA with compression ratio 0.5

Step 3: Grouped comparisons. This additional feature of the system allows users to conduct another kind of comparative analysis for the multiple compression techniques and distance functions at once. More specifically, the user can select more than one compression method *and* select multiple values for the parameters (i.e., error tolerance and compression ratio). Upon receiving the entire collection, the backend system will execute all the comparisons and returns the compared results grouped around the distant functions. Those are subsequently sent to the frontend as JSON format, which is manipulated by frontend part to enable the user to view the multiple comparisons. The lower part of Figure 4 illustrates this functionality – i.e., the user can check the boxes of the desired compression methods, after which the system will ask for specification of parameters and the selection of distance functions.

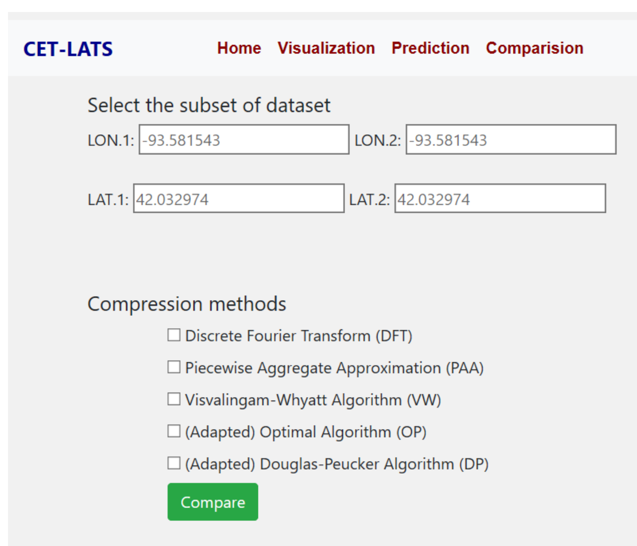

Figure 4: UI for the comparison

Step 4: Visualizing the TIN instances. Another feature of CET-LATS is that it gives the opportunity to visualize the TIN – both with the original and the compressed data – at a particular time instant. This option expects that the user will provide the value of interest for the temporal domain, and select the spatial subset of interest. An example of the output – i.e., a comparative visualization of TINS constructed before and after compression of the selected dataset is depicted in Figure 5.

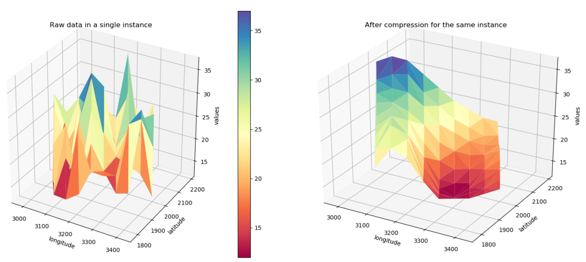

Figure 5: 3D plot of TINs before and after compression

Step 5: Compression impact on predictions.

Another feature of the proposed application is the prediction tab. AT present, we have AR, ARIMA and Prophet [10] available in CET-LATS. In the prediction part for Prophet, for both the original and the compressed datasets, in the current version we used the daily temperature of the first eleven months (January-November) of 2019 as the training data and the last month (December) as the test data. User can get the prediction values of a station by providing the latitude and longitude of the station's location. The predicted values of one of the measuring stations (latitude: 78.250, longitude: 22.817) are shown in Figure 6.

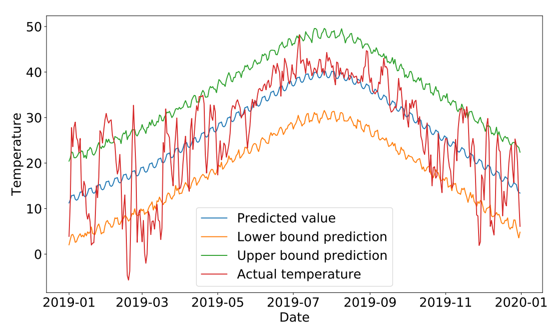

Figure 6: Predicted value versus real data of one of the measuring stations. The lower and upper bound of the prediction are shown in the figure.

We close this section with a note that the users also have the option to upload their own datasets. The only constraint is that presently the format of each record is expected to be: {lon, lat, $val_{t1}, val_{t2}, \ldots, val_{tn}$}

5 CONCLUSION AND FUTURE WORK

We presented CET-LATS, a prototype system that allows the users to explore the impact of different time series compression methods on (the evolution of) TINs. Specifically, we considered the settings in which each time series is associated with a spatial location, and at each time instant, a TIN is constructed using the coordinates of the locations and the values from the corresponding time series.

At present, there are five compression methods readily available, and three different distance metrics. For each paired combination, the user can provide compression parameters, and examine the discrepancy between the original and compressed datasets in terms of a particular distance function. The system also provides visualization options for comparing TINs at a particular time instant, as well as comparison of the compression on three prediction methods.

Part of the future work incorporates extending the functionality of CET-LATS to include compression in the spatial dimension (i.e., eliminating some of the locations, when the density of points is high enough in a given region), as well as incorporating multivariate time series data.

ACKNOWLEDGEMENT: The work was partially supported by the NSF grant CNS 1646107.

REFERENCES

[1] Theerasak Chanwimalueang and Danilo Mandic. 2017. Cosine Similarity Entropy: Self-Correlation-Based Complexity Analysis of Dynamical Systems. *Entropy* 19 (11 2017), 652. https://doi.org/10.3390/e19120652

[2] M. Copperwaite and C. Leifer. 2015. *Learning Flask Framework*. Packt Publishing. https://books.google.com/books?id=HPGoCwAAQBAJ

[3] Fan Bao and Jia Chen. 2014. Visual framework for big data in d3.js. In *2014 IEEE Workshop on Electronics, Computer and Applications*. 47–50.

[4] GSOD. [n.d.]. *Global Surface Summary of the Day - GSOD*. https://catalog.data.gov/dataset/global-surface-summary-of-the-day-gsod.

[5] Baofeng Guo, Kin-Man Lam, Kwan-Ho Lin, and Wan-Chi Siu. 2003. Human face recognition based on spatially weighted Hausdorff distance. *Pattern Recognition Letters* 24, 1 (2003), 499 – 507.

[6] Eamonn J. Keogh, Stefano Lonardi, Chotirat Ann Ratanamahatana, Li Wei, Sang-Hee Lee, and John Handley. 2007. Compression-based data mining of sequential data. *Data Min. Knowl. Discov.* 14, 1 (2007).

[7] Shunlin Liang. 2012. Chapter 2 - Geometric Processing and Positioning Techniques. In *Advanced Remote Sensing*, Shunlin Liang, Xiaowen Li, and Jindi Wang (Eds.). Academic Press, Boston, 33 – 74.

[8] Ivan Merelli, Horacio Pérez-Sánchez, Sandra Gesing, and Daniele D'Agostino. 2014. Managing, Analysing, and Integrating Big Data in Medical Bioinformatics: Open Problems and Future Perspectives. *BioMed Research International* 2014 (01 Sep 2014), 134023. https://doi.org/10.1155/2014/134023

[9] Abdel Aziz Taha and Allan Hanbury. 2015. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC medical imaging* 15 (12 Aug 2015), 29–29.

[10] Sean J Taylor and Benjamin Letham. 2018. Forecasting at scale. *The American Statistician* 72, 1 (2018), 37–45.

[11] Xu Teng, Prabin Giri, Matthew Dwyer, Jidong Sun, and Goce Trajcevski. 2019. Local Temporal Compression for (Globally) Evolving Spatial Surfaces. In *Big Data Analytics - 7th International Conference, BDA.*

[12] Xu Teng, Andreas Züfle, Goce Trajcevski, and Diego Klabjan. 2018. Location-Awareness in Time Series Compression. In *Advances in Databases and Information Systems ADBIS*, András Benczúr, Bernhard Thalheim, and Tomás Horváth (Eds.).

[13] Muhammad Habib ur Rehman, Chee Sun Liew, Assad Abbas, Prem Prakash Jayaraman, Teh Ying Wah, and Samee U. Khan. 2016. Big Data Reduction Methods: A Survey. *Data Science and Engineering* 1, 4 (2016).

[14] Xiaoyue Wang, Abdullah Mueen, Hui Ding, Goce Trajcevski, Peter Scheuermann, and Eamonn J. Keogh. 2013. Experimental comparison of representation methods and distance measures for time series data. *Data Min. Knowl. Discov.* 26, 2 (2013).