engineering proceedings

MDPI

Proceeding Paper

Succinct Data Structures in the Realm of GIS †

Nieves R. Brisaboa , Pablo Gutiérrez-Asorey , Miguel R. Luaces and Tirso V. Rodeiro *

Database Lab. Elviña, CITIC, Universidade da Coruña, 15071 A Coruña, Spain; nieves.brisaboa@udc.es (N.R.B.); pablo.gutierrez@udc.es (P.G.-A.); miguel.luaces@udc.es (M.R.L.)

* Correspondence: tirso.varela.rodeiro@udc.es

† Presented at the 4th XoveTIC Conference, A Coruña, Spain, 7–8 October 2021.

Abstract: Geographic Information Systems (GIS) have spread all over our technological environment in the last decade. The inclusion of GPS technologies in everyday portable devices along with the creation of massive shareable geographical data banks has boosted the rise of geoinformatics. Despite the technological maturity of this field, there are still relevant research challenges concerning efficient information storage and representation. One of the most powerful techniques to tackle these issues is designing new Succinct Data Structures (SDS). These structures are defined by three main characteristics: they use a compact representation of the data, they have self-index properties and, as a consequence, they do not need decompression to process the enclosed information. Thus, SDS are not only capable of storing geographical data using as little space as possible, but they can also solve queries efficiently without any previous decompression. This work introduces how SDS can be successfully applied in the GIS context through several novel approaches and practical use cases.

Keywords: data structures; geoinformatics; geographic information systems

Citation: Brisaboa, N.R.; Asorey, P.G.; Luaces, M.R.; Rodeiro, T.V. Succinct Data Structures in the Realm of GIS. *Eng. Proc.* **2021**, *7*, 29. https://doi.org/10.3390/engproc2021007029

Academic Editors: Joaquim de Moura, Marco A. González, Javier Pereira and Manuel G. Penedo

Published: 14 October 2021

1. Introduction

Since the dawn of mankind, we have been trying to capture the geography that surrounds us in an attempt to establish order in nature. From the very beginning of our species, we were forced to memorize important locations in order to survive; little by little we improved our cartographic skills until the current degree of sophistication. Currently, Geographic Information Systems (GIS) are still one of the most important tools in our society. Almost every gadget tracks its position or uses some kind of map. The rise of modern GIS can be understood via two main reasons:

- The improvements to Global Positioning Systems (GPS) technologies that enable all kinds of devices to geolocate any object with the highest precision.
- The popularization of massive geographic data banks.

On the one hand, advances in GPS techniques have allowed accurate sensors to become cheaper and disseminated all over the technological ecosystem. On the other hand, it was also necessary to design and implement shareable geographic data banks in order to represent within them the information of interest gathered by GPS (e.g., points, trajectories, etc.).

These new features translated directly to a constant production of large amounts of data that can be exploited in order to optimize a wide range of tasks and ease our daily life (e.g., package tracking, food delivery, etc.). Accordingly, an interest in programs able to handle trajectories and geographical information systems has grown in recent years, giving birth to all kinds of algorithms and systems that are able to locate mobile objects in real time or recover any past trajectory, attending to some user-defined criteria.

2. Succinct Data Structures

One of the most effective techniques to fight against the exponential growth of big data scenarios is compression. A vast amount of compression algorithms have been proposed in almost every context of computer science, trying to reduce as much as possible the space used. However, the aim of this field is not just to store information by reducing the space used but also to avoid interactions with secondary memory as much as possible. Usually, the information used in computers is too large to fit in primary memory, so it is necessary to load it by parts, one part at a time; this loading process is so slow that it would be preferable to perform more operations (e.g., decompression) in the main memory than to load more data from the disk.

The most common approach to achieving compression is based on *repetitiveness*. Usually, compression algorithms search for repeated chunks of data and store them as few times as possible. Sometimes, repetitiveness is not found in an individual bank of data by itself, but it does appear when compared with others. For example, DNA strands are not particularly repetitive themselves; however, different samples of the same species' genome are so similar to one another that they can be represented by their similarities and differences with respect to reference DNA.

In recent years, a new branch in the compression field has attracted a lot of attention: Succinct Data Structures (SDS). These structures tackle the space usage issue, but, in opposition to traditional approaches, they are autoindexes. Thus, the ultimate goal of SDS is to store information in a compact way while still being able to use the compacted data, i.e., perform queries without any decompression process, or at least a minimal decompression.

3. Trips over Public Transport Networks

Inhabitants of large cities increasingly choose public transport (bus, train, etc.) as their first option to move around the city. Common public transport systems should provide users with basic information about the available offerings (at least timetables, lines and stops). One of the main challenges of these systems is matching the available offerings with the historical passengers' demand. For this purpose, they need to gather information regarding how users move along the network. With the increasing use of passenger tracking technology on public transport networks (e.g., smart cards), it is now becoming possible to assemble (or accurately estimate) the actual trips a given user made along a network.

We introduced in [1] a new flexible representation based on efficient indexes [2] that support the analysis of the historical demand in real transport networks. A naive approach to represent the trips of each passenger would be to store the sequence of the traversed stops, e.g., $< S1, S5, S8, S9 >$. However, as all passengers of a bus are traversing the same stops at the same times, we just need to store that a user boards or leaves a vehicle following the journey j of line l at a given stop s, as a triple (s, l, j). Since we want to represent a user trip as a sequence of such stages, and it holds that the final stop of a stage and the starting stop of the next stage are the same (or close in walking distance), it is not necessary to explicitly represent the final stop of each stage, except for the final stop.

This solution requires less than half the space compared to a plain (not indexed) representation, while also being capable of directly solving queries about users' movement patterns such as *how many users start their trips at stop X and end at stop Y* or any of its combinations (e.g., filtering by line, only using initial stops, etc.).

4. Free Trajectories of Ships Furrowing the Sea

Millions of vessels sail across the oceans almost without movement constraints, i.e., describing free trajectories. Once again, this is a big data scenario with geographic information involved. The easiest way to apply compression techniques is to find a repetitive sequence in those arbitrary movements. Our work [3] focuses on speed and direction. The aim is to exploit the fact that in many applications, trajectories tend to be similar to others, wholly or piecewise. Thus, instead of storing all the real geographical positions measured by the GPS devices as traditional solutions, our work stores *snapshots* of the positions of all the objects at regular time intervals and the sequence of relative movements between snapshots (*logs*). As the trajectory of a vessel is not likely to perform sharp turns, the direction and the speed of the boat will remain constant for a while, translating into a sequence of repeated directions. One efficient way to capture these logs is spiral encoding, that is, using a grid with its center cell representing the vessel position in a particular snapshot and the surrounding cells representing reachable areas, each one with a different identifier (e.g., the log of a vessel traveling at slow speed to the east during two time instants and then turning to the southeast would be represented as $< 1, 1, 2 >$). Accordingly, the snapshots are saved in efficient spatial indexes, while well-known compression techniques [4] can be applied over the logs.

Our proposal can efficiently solve a wide range of queries (time-slice, time-interval and k-nearest neighbor queries) while reducing the raw data to 4–7% of its original size (two orders of magnitude less space than traditional spatio-temporal indexes).

5. Indoor Trajectories in a Nursing Home

Health care facilities are the perfect environment to track repetitive trajectories. As all the movement happens inside a building, the map where all the trajectories take place turns out to be a graph, each node/cell of the graph being a particular room [5]. Usually, the residents do not have much path diversity, so they need to roam through the same rooms throughout the day, generating massive repetitive trajectories. For example, it is likely that all residents perform the sequence *bedroom–bathroom–dining room* every single day.

Thus, in our last work [6], we presented a new approach to deal with these repetitive trajectories based on enhanced compression techniques. We designed and implemented a whole system capable of tracking the daily movements of residents and caregivers. The actual positioning data are gathered through Bluetooth Low-Energy (BLE) strategically placed all over the nursing home. Essentially, we use the sequence of room identifiers as the trajectory of a particular user. Then, our structure stores this information in a compact way, avoiding repetitiveness without resorting to random access.

Our solution reaches compression ratios of 1.44% with high-repetitive datasets, surpassing even well-known solutions such as *gzip* or *7zip* (both without random access capabilities). Furthermore, our proposal can perform random subtrajectory retrieval (30 position window) in roughly 25 µs.

6. Conclusions and Future Work

Despite the maturity level of the current Geographic Information Systems, there is still room for improvements, specifically regarding data storage and data management. In this paper we have introduced several success stories in different contexts where Succinct Data Structures and Geographic Information Systems work together to achieve higher efficiency.

For future work, we will pay attention not only to storage capabilities and retrieval performance but also to transmission times, in an attempt to send a smaller number of points of a geometry through aggregation.

Funding: This work is partially funded by the CITIC research center funded by Xunta/FEDER-UE 2014-2020 Program, ED431G 2019/01. MICINN(PGE/ERDF) [EXTRA-Compact: PID2020-114635RB-I00].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

References

1. Brisaboa, N.R.; Fariña, A.; Galaktionov, D.; Rodeiro, T.V.; Rodríguez, M.A. New Structures to Solve Aggregated Queries for Trips over Public Transportation Networks. In Proceedings of the String Processing and Information Retrieval 25th International Symposium, (SPIRE), Lima, Perú, 9–11 October 2018; pp. 88–101.
2. Sadakane, K. New text indexing functionalities of the compressed suffix arrays. *J. Algorithms* **2003**, *48*, 294–313. [CrossRef]
3. Brisaboa, N.R.; Gómez-Brandón, A.; Navarro, G.; Paramá, J.R. GraCT: A Grammar-based Compressed Index for Trajectory Data. *Inf. Sci.* **2019**, *483*, 106–135. [CrossRef]
4. Larsson, N.J.; Moffat, A. Offline dictionary-based compression. In Proceedings of the Data Compression Conference, (DCC99), Snowbird, Utah, 29–31 March 1999; pp. 296–305.

5. IndoorGML OGC. Available online: www.indoorgml.net (accessed on 26 July 2021).
6. Fariña, A.; Gutiérrez-Asorey, P.; Ladra, S.; Penabad, M.R.; Rodeiro, T.V. A Compact Representation of Indoor Trajectories. *IEEE Pervasive Comput.* under review.