Dynamic Aggregation of Geo-Objects for the Interactive Exploration of Research Data

ANNIKA BONERATH[1], BENJAMIN NIEDERMANN[1] & JAN-HENRIK HAUNERT[1]

Abstract: In the context of research data management, the interactive exploration of data is a useful tool to make data findable and reusable. For exploring spatio- and temporal data we developed in a previously published work a data structure that provides the user with simple visualizations of the data for time window queries. We considered the data to be points in space each associated with a time stamp and we visualized it using α-shapes, which generalize convex hulls. In general, time windowed data structures support time window queries for geometric shapes or more general problems from computational geometry such as counting and intersection problems. With this paper, we contribute a review of our previously published method in the context of time windowed data structures, which is a relatively new concept of computational geometry. In particular, we highlight the relevance of our work for a growing domain of research.

1 Introduction

In times of big and heterogeneous research data, the data management and the exploration of available data becomes more and more important. Especially in the scientific context, projects often deal with these large amounts of spatio- and temporal data that need to be findable and reusable. For example, consider a project that deals with a database of meteorological data; see Figure 1. Each object in the data set is a storm event that is represented as a point in space and time. For scientific tasks often the researcher is not interested in all the data but only in a subset that is limited to a time window. In order to find the right time window, the user might want to interactively explore the data, by retrieving simplified visualizations of the data in time windows; see Figure 2.

In this work, we visualize the queried data by sketching its outline. This outline provides the user with the possibility of roughly assessing the spatial distribution of the data. Since for most data sets, simple representations as the convex hull are not adequate, a wide range of more sophisticated polygonal representations exists; some of these are based on Delaunay-triangulations (DE BERG et al. 2011; DUCKHAM 2008; EDELSBRUNNER et al. 1983) while others use spatial grids to define the representation (ATTALI 1997; JONES et al. 2008; PURVES et al. 2005).

In this paper, we use α-shapes (EDELSBRUNNER et al. 1983; EDELSBRUNNER 2010) for representing point sets, which are a generalization of convex hulls and strongly related to Delaunay triangulations. Among others, this technique finds its application in pattern recognition (VAUHKONEN et al. 2010) and micro-biology (LIANG et al. 1998).

[1] Rheinische Friedrich-Wilhelms-Universität Bonn, Institut für Geodäsie und Geoinformation, Meckenheimer Allee 172, D-53115 Bonn, E-Mail: [bonerath, niedermann, haunert]@igg.uni-bonn.de

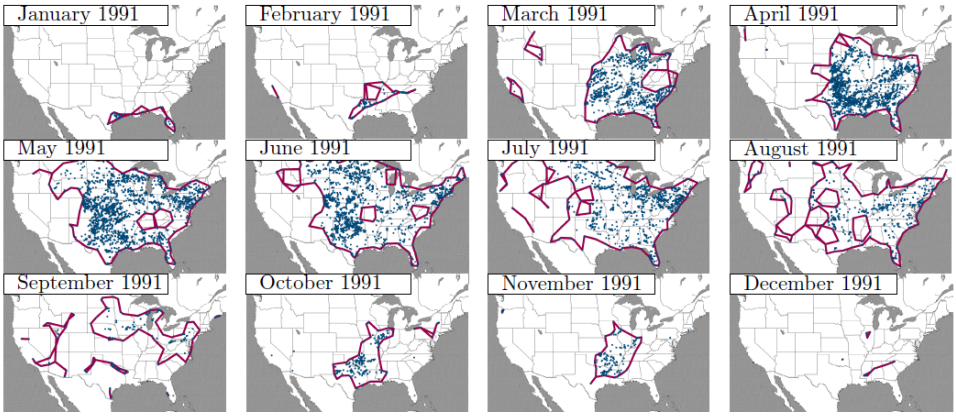

Fig. 1: Storm events for the months of 1991 represented by α-shapes (lilac). The actual point set (blue) is drawn for illustration. Data retrieved from Data.gov. Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under ODbL.

In the following, we give the definition of α-shapes. Let $P \subseteq \mathbb{R}^2$ be a set of n points in the plane and let $\alpha > 0$. Further, let $pq \in P \times P$ be a directed edge with $|q - p| \leq \alpha$. We define the *edge domain* of pq as the open disk D_{pq} with radius $\alpha/2$ whose center lies to the right of pq and whose boundary contains the points p and q. Let $S_\alpha(P) \subseteq P \times P$ be the set of all edges that are shorter than α and do not contain any point of P in their edge domain. We call $S_\alpha(P)$ the *α-shape* of P; see Figure 3(a). Based on the Delaunay triangulation, the α-shape of n points can be computed in $O(n \log n)$ time (EDELSBRUNNER et al. 1983).

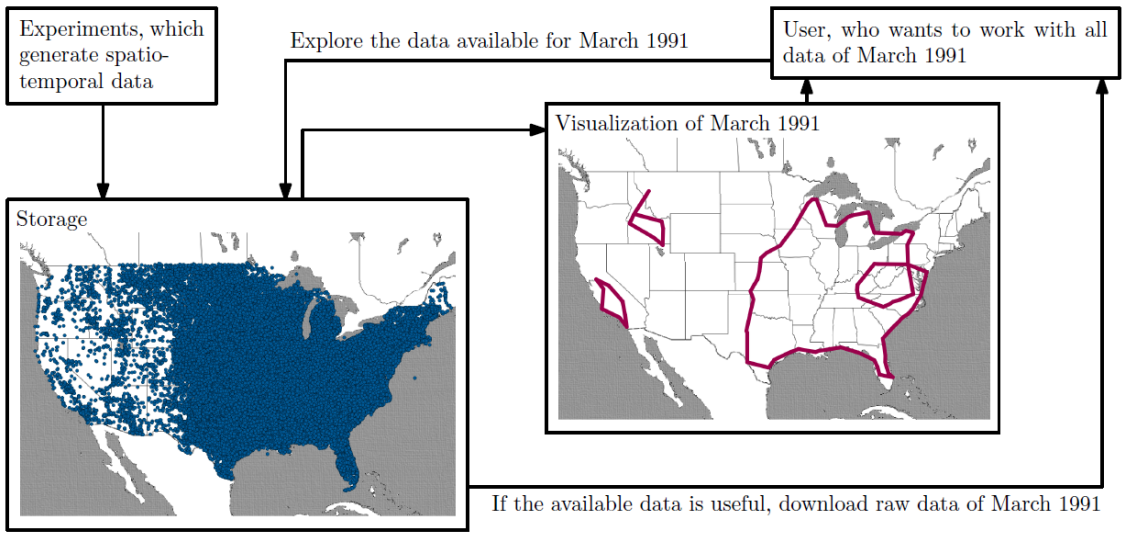

Fig. 2: Scenario for the case that the user queries a simplified visualization for all storm events in March 1991.

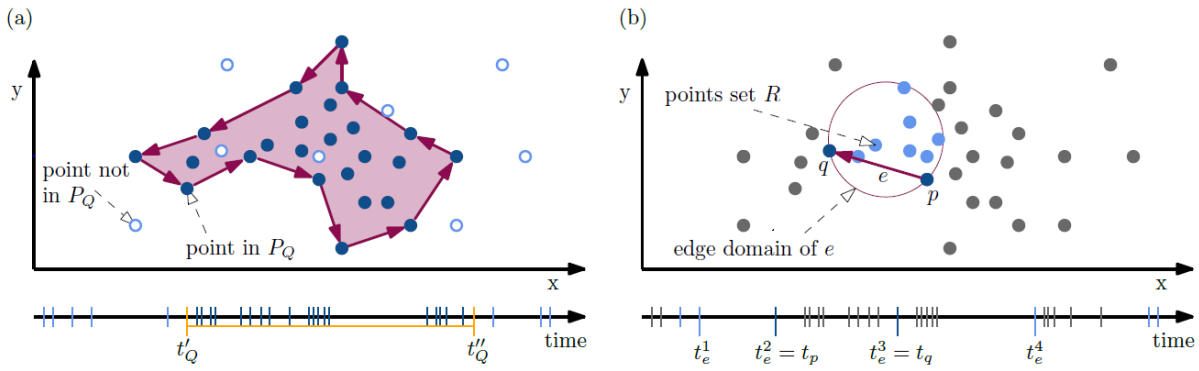

Fig. 3: (a) α-shape (lilac arcs) of a queried point set P_Q (filled blue discs) and (b) an edge with its edge domain and time attributes.

In order to apply the definition of α-*shapes* to our problem setting, we first introduce two more concepts. For a point p we denote its *time stamp* by $t_p \in \mathbb{R}$. Further, a *time window query* Q is a query for a temporal range $[t'_Q, t''_Q]$ and $P_Q = \{p \in P | t_p \in Q\}$ is the subset of P that is contained in Q. As described in the running example, the point set P is queried frequently for the α-shape $S_\alpha(P_Q)$ of some time window query Q. A straight-forward approach for a query Q first queries the set P obtaining P_Q and then computes the α-shape $S_\alpha(P_Q)$. Utilizing a balanced binary search-tree for finding P_Q and additionally computing the α-shape, we obtain $O(\log n + |P_Q| \log |P_Q|)$ running time. For our use-case, we aim at a better running time per query. In particular, we propopse to use a data structure for filtering search (CHAZELLE 1986) that allows us to answer a query in $O(\log n + k)$ where k is the size of the returned α-shape. In a pre-processing phase, we compute a data structure that aggregates the α-shapes of all possible queries; we call it the α-*structure* of P. We use this data structure in the query phase to obtain the α-shapes of the incoming queries. Note that data structures, like the α-structure that preprocess temporal data so that specific queries for a time window can be handled efficiently are called *time windowed data structures*. In Section 2, we provide an overview about existing time windowed data structures.

We have published two works concerning the α-structure, one as a preprint at the workshop EuroCG'19 and an extended conference version at ACM SIGSPATIAL'19 (BONERATH et al. 2019a, BONERATH et al. 2019b). For the conference version we extended our approach to also provide schematized α-shapes. In particular, we can provide octilinear α-shapes where each edge has a direction that stems from a predefined set of eight directions, namely horizontal, vertical, and diagonal directions. Further, we provided a method, where the α parameter of the queried α-shape does not need to be fixed before preprocessing the α-structure.

2 Related Work

In general, a time windowed data structure answers queries concerning specific properties of the input data set for time windows. This concept was first considered by BANNISTER et al. (2013). They considered social network data as their input. On this data a *relational event graph* is built, where each vertex represents an entity of the social network and each edge represents a

communication between two entities. Each edge is associated with a timestamp. In their work they provide data structures that handle counting problems for both general graph properties such as the number of connected components, and properties for social network analysis. Further improvements, extensions and results on time windowed data structures for relational event graphs are presented by CHANCHARY & MAHESHWARI (2019) and CHANCHARY et al. (2017).
Time windowed data structures are also considered for sets of geometric objects where each geometric object is associated with a point in time. For example, closely related to our work, BANNISTER et al. (2014) discussed the problem of reporting the convex hull, the skyline, and other basic problems from computational geometry for time window queries. Especially, closely related to our work is the problem of reporting the convex hulls for points with time stamps since the α-shape is a generalization of the convex hull. As a basic concept for their data structure, they build a decomposition tree in which each node contains the convex hull of its descendants. Their data structure reports the convex hull in $O(h \log^2 w)$ time where h is the number of edges of the convex hull and w is the number of points contained in the queried time window. In contrast, the running time of our method depends not directly on the number of points in the queried time window but only on the size of the α-shape for those points
RUDI (2018) discusses the design of a time windowed data structure for answering hotspot queries on trajectories. The hotspot of a trajectory is the square of fixed side length that contains the longest contiguous part of the trajectory. As input data each trajectory point is associated with a time stamp. Answering a time window query takes $O(\log^2 n)$ time. The algorithm provides only an approximation. The idea of the data structure is to enrich each vertex with additional attributes with which the result can be computed. This idea is similar to our approach, since we also enrich the edges of the α-structure with additional attributes that we use in the query.
Further work on time windowed data structures for sets of geometric objects with time stamps concerns decision problems such as the problem if the convex hull area is larger than some threshold (BOKAL et al. 2015, CHAN & PRATT 2016), intersection decision problems and counting problems of extreme points (CHANCHARY et al. 2018), and geometric properties like reporting the closest pair (CHAN & PRATT 2015).

3 On α-Structures

In the following, we define the α-structure of P.We say that an edge $pq \in P$ is *active* for a temporal query Q if the α-shape $S_\alpha(P_Q)$ contains pq. We observe that an edge pq can be active for an infinite set of temporal queries, but it can only be active for $O(n^2)$ different subsets of P. To characterize this set, we introduce the following notation; see Figure 3(b).
Let $e = pq \in P \times P$ with $t_p < t_q$ and let $R \subseteq P$ be the set of points contained in the edge domain of pq. Further, let t_r with $r \in R$ be the largest time stamp that is smaller than t_p; if r does not exist, we set $t_r = -\infty$. Similarly, let t_s with $s \in R$ be the smallest time stamp that is greater than t_q; if s does not exist, we set $t_s = \infty$. We call $t_e^1 = t_r$, $t_e^2 = t_p$, $t_e^3 = t_q$, $t_e^4 = t_s$ the *time attributes* of pq. Using the time attributes we can formulate the *conditions of an active edge* for query Q

(1) the distance between p and q is smaller than α,
(2) $\forall r \in R : t_r \notin [t_p, t_q]$, and
(3) $t'_Q \in [t_e^1, t_e^2]$ and $t''_Q \in [t_e^3, t_e^4]$.

We show that the conditions of an active edge are necessary and sufficient for an edge e to be active for Q. Assume that e is active for Q. This is equivalent to the following three conditions; (i) $p, q \in P_Q$, which is equivalent to $t_p, t_q \in Q$, (ii) e is shorter than α (equivalent to Condition (1)) and (iii) no point $r \in P_Q$ is contained in R, which is equivalent to $\forall r \in R : t_r \notin Q$. Applying the definition of the time attributes t_e^1, t_e^2, t_e^3, t_e^4, Condition (i) and (iii) are equivalent to Condition (2) and (3).
The *α-structure* $S_\alpha(P) \subseteq P \times P$ of P is the set of all active edges over all possible temporal queries. We show that Condition (1) and (2) are necessary and sufficient for an edge pq to be contained in $S_\alpha(P)$.

Lemma 2.1. *The edge* $e = pq \in P \times P$ *is contained in* $S_\alpha(P)$ *if and only if*

(1) the distance between p *and* q *is smaller than* α*, and*

(2) $\forall r \in R : t_r \notin [t_p, t_q]$.

Proof of Lemma 2.1. Let $e \in S_\alpha(P)$, and let $Q = [t'_Q, t''_Q]$ be a query for which $e \in S_\alpha(P_Q)$. Then e fulfills the conditions of an active edge for query Q and therefore the conditions of Lemma 2.1. Conversely, let e be shorter than α and all points $r \in R$ be temporally not in $[t_p, t_q]$. Then the α-shape of the query Q with $t'_Q = t_p$ and $t''_Q = t_q$ contains the edge e. □
For our use-case of a database, the memory consumption of our approach is decisive for being deployed in practice. We first observe that $O(n^2)$ is an upper bound for the size of an α-structure. The following theorem shows that this is also a lower bound in the worst case.

Theorem 2.2. *For a set* P *of* n *points the* α*-structure has size* $\Omega(n^2)$ *in the worst case.*

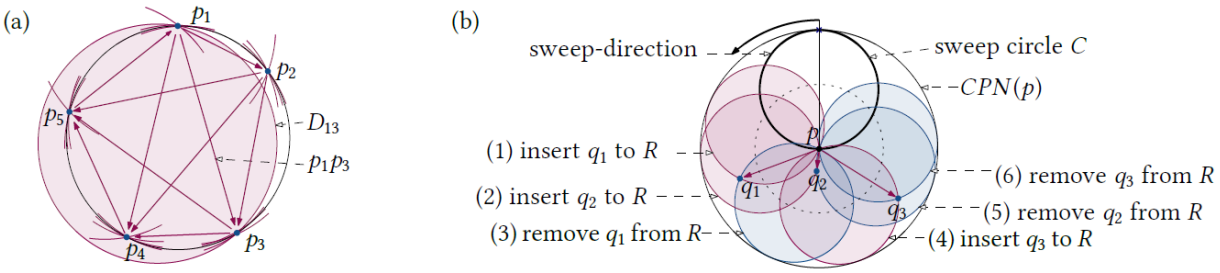

Fig. 4: (a) worst-case example for the size of the α-structure as described in Theorem 2.2 and (b) the rotational sweep CPN-check for a point p and CPN $T_q = \{q_1, q_2, q_3\}$.

Proof of Theorem 2.2. Let $P = \{p_1, p_2, \dots, p_n\}$ be a point set with time stamps $t_1 < t_2 < \cdots < t_n$ such that the points lie on a circle C of radius $r < \alpha/2$ ordered clockwise according to their time stamps; see Figure 4(a). Let $p_i, p_j \in P$ be two points with $i < j$. We show that p_ip_j is contained in the α-structure $S_\alpha(P)$ by proving the two conditions of Lemma 2.1. Due to $r < \alpha/2$ the points p_ip_j have distance smaller than α. Hence, Condition (1) of Lemma 2.1 is satisfied. For the second condition let R_{ij} be the set of points contained in edge domain D_{ij} of p_ip_j.

We observe that D_{ij} and C intersect in p_i and p_j. Since the radius of C is smaller than the radius of D_{ij} the boundary of D_{ij} partitions C into two parts. One part is contained in D_{ij} and the other lies outside of D_{ij}. Since the points $p_1, p_2, \dots, p_n$ appear in clockwise order on C, and since the center D_{ij} lies to the right of $p_i p_j$ by definition, we obtain $R_{ij} = \{p_1, \dots, p_{i-1}, p_{j+1}, \dots, p_n\}$. Consequently, Condition (2) is satisfied.
Hence, the database may exceed a size that is applicable in practice. However, the example is rather unlikely to occur in practice. Generally, the size of the database is bounded by $O(nm)$ where m is the largest number of points in a distance α to a point in P. Assuming that the point density is bounded by a constant and α is fixed, m is also constant. If, on the other hand, the density increases, it becomes more likely that an edge gets destroyed. In our experiments, we observe a linear relation between the number of points and α-structure size; see Section 4.

4 Constructing and Querying α-Structures

We introduce an algorithm that constructs the α-structure of a point set P in $O\big(n(\log n + \bar{m} \log \bar{m})\big)$ time, where $\bar{m}$ is the maximum number of points in a square with width2α. Further, we describe how to query this data structure.
The construction algorithm consists of two parts; see Algorithm 1. Each part is applied for each point $p \in P$. We call the first part *CPN-Search.* In this part we compute the set of all points $T_p \subseteq P$ that fulfill Condition (1) of Lemma 2.1, i.e., all points that lie in a circle with center at p and radius α. We call this circle the *circle of potential neighbors (CPN)* of p. We implement the CPN-Search with a sweep line approach to find T_p in $O(\log n + \bar{m})$ time (PENG 2014).

Algorithm 1. Computation of the α-structure

Input: point set P, parameter α
Output: α-structure $S_\alpha(P)$
foreach $p \in P$ **do**
    CPN-Search: Find all points $T_p \subseteq P$ in the CPN of p
    CPN-Check: Check for each edge pq with $q \in T_p$ whether it fulfils Condition (2) of Lemma 2.1, possibly compute the time attributes and add to $S_\alpha(P)$

The second part of the algorithm, which we call *CPN-Check*, checks for each point $q \in T_p$ whether the edge pq fulfills Condition (2) of Lemma 2.1. If this is the case, it computes the time attributes of pq. We use a rotational sweep for the CPN-Check. More precisely, we use a circle C of radius $\alpha/2$ which sweeps around p such that the center of C moves along the circle with center p and radius $\alpha/2$; see Figure 4(b). We call C the *sweep circle* of p. Let $\hat{R}$ be the points contained in C; we represent $\hat{R}$ using a binary search tree ordered by the time stamps of the points. The sweep circle C stops its rotation whenever its boundary intersects with a point $q \in T_p$. Two kind of events are possible; either the point q enters C, or it leaves C. Whenever a point q enters C, the sweep circle equals the edge domain of pq. Utilizing the properties of the binary search tree $\hat{R}$, Condition (2) of Lemma 2.1 can be checked in $O(\log \bar{m})$ time. If this is the case, the time attributes of pq can be computed using the temporal order of $\hat{R}$ in $O(\log \bar{m})$ time. This rotational sweep can be done in $O(\bar{m} \log \bar{m})$ time.

Theorem 2.3. *Computing the* α*-structure of a set of* n *points takes* $O\left(n(\log n + \bar{m}\log\bar{m})\right)$ *time.*

For the query phase we represent each edge e with t_e^1, t_e^2, t_e^3, t_e^4 of the α-structure by a rectangle $[t_e^1, t_e^2] \times [t_e^3, t_e^4]$. A query $[t'_Q, t''_Q]$ corresponds to finding all rectangles containing (t'_Q, t''_Q). For these rectangles we propopse to use a data structure for filtering search (CHAZELLE 1986) that allows us to answer a query in $O(\log n + k)$, where k is the size of the returned α-shape.

5 Experimental Evaluation

We analyze the performance of α-structures using on the one hand a data set of storm events in the United States in the years 1991-2000 obtained from Data.gov; see Figure 1 for the year 1991 and on the other hand a set of synthetic generated data with uniform distribution in time and space with density of 0.001 points per m^2. We performed the experiments on a 4-core Intel Core i7-7700T CPU with 16 GiB RAM using an implementation in Java.
The experiments indicate for both data sets that the memory consumption is linear in n; see Figure 5(a). The construction time for a point set of size $n = 70000$ varies between several seconds and hours depending on the value of α. For smaller α we obtain a shorter running time than for larger α. We assume this to be acceptable, since it is a pre-processing step.

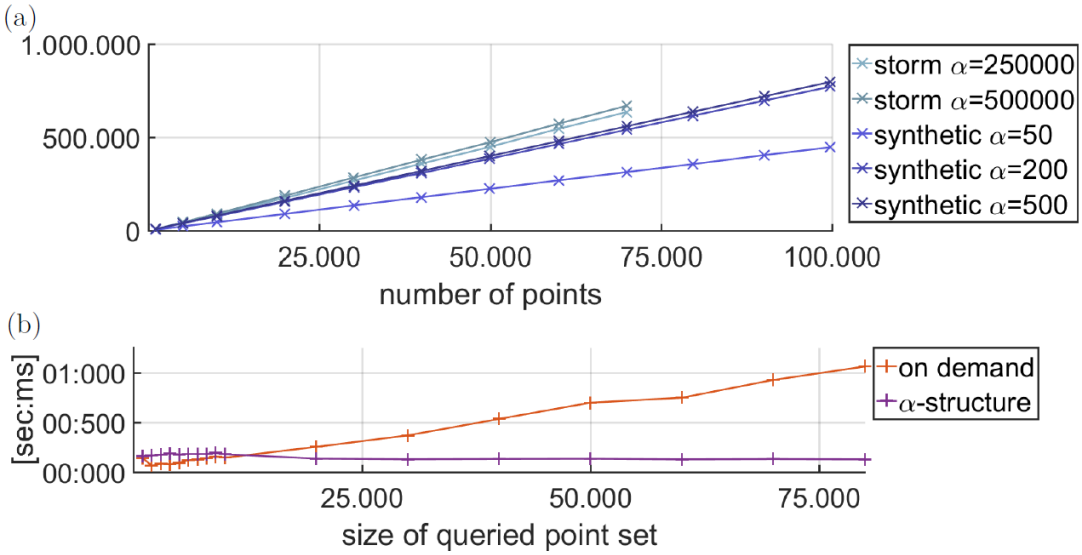

Fig. 5: (a) memory consumption of an α-structure (parameter α in meter) and (b) query phase time of the α-structure compared to an on demand approach for the synthetic dataset and $\alpha = 200$.

Figure 5(b) illustrates the query time. For this experiment, we queried the data structure for different window sizes. The x-axis shows the size of the queried point set P_Q, the y-axis the query time. The experiments show that the query time using the α-structure is nearly constant around 250[ms] with respect to the size of P_Q; see Figure 5(b). In contrast, the results for an implemented on demand approach indicate a dependency to the subset size. For this data set the query times for the α-structure are smaller than for the on demand approach for all subsets of size 20.000 or larger.

6 Conclusion and Outlook

Overall, we presented the design and construction of a data structure that provides the edges of α-shapes for time window queries on point sets. Using this data structure the query time depends not on the number of points in the time window, but on the size of the α-shape. The experiments indicate that with respect to preprocessing time and memory consumption this data structure is applicable in practice. Further, the experiments show that the α-structure performs better than the on demand approach with respect to the query time.
For future work we plan to extend our approach of α-structures such that we can handle also other geometric objects than points, e.g. lines and polygons. Further, we want to consider other aggregated representations of geographic objects for time windowed data structures. Even more general, we are interested in the concept of time windowed data structures for interactive visualization such as labeling problems or cartograms.

References

ATTALI, D., 1997: r-regular shape reconstruction from unorganized points. Proceedings of the Thirteenth Annual Symposium on Computational Geometry, Nice, France, June 4-6, 248-253.

BANNISTER, M.J., DEVANNY, W.E., GOODRICH, M.T., SIMONS, J.A. & TROTT, L., 2014: Windows into geometric events: Data structures for time-windowed querying of temporal point sets. Proceedings of the 26th Canadian Conference on Computational Geometry, CCCG 2014, Halifax, Nova Scotia, Canada, 11-19.

BANNISTER, M.J., DUBOIS, C., EPPSTEIN, D. & SMYTH, P., 2013: Windows into relational events: Data structures for contiguous subsequences of edges. Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013, 856-864.

BOKAL, D., CABELLO, S. & EPPSTEIN, D., 2015: Finding all maximal subsequences with hereditary properties. Proceedings of the 31st International Symposium on Computational Geometry, SoCG 2015, June 22-25, 2015, Eindhoven, Netherlands, 240-254.

BONERATH, A., HAUNERT, J.-H. & NIEDERMANN, B., 2019: Computing alpha-shapes for temporal range queries on point sets. Proceedings of the 35th European Workshop on Computational Geometry, EuroCG 2019, March 18-20, Utrecht, Netherlands,

BONERATH, A., NIEDERMANN, B. & HAUNERT, J.-H., 2019: Retrieving alpha-shapes and schematic polygonal approximations for sets of points within queried temporal ranges. Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL 2019, Chicago, IL, USA, November 5-8, 249-258.

BERG, M., MEULEMANS, W. & SPECKMANN, B., 2011: Delineating imprecise regions via shortest-path graphs. Proceedings of the 19th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, SIGSPATIAL 2011, November 1-4, Chicago, IL, USA, 271-280.

CHAN, T.M. & PRATT, S., 2015: Time-windowed closest pair. Proceedings of the 27th Canadian Conference on Computational Geometry, CCCG 2015 , Kingston, Ontario, Canada, August 10-12, 141-144.

CHAN, T.M. & PRATT, S., 2016: Two approaches to building time-windowed geometric data structures. Proceedings of the 32nd International Symposium on Computational Geometry, SoCG 2016, June 14-18, Boston, MA, USA, 28:1-28:15.

CHANCHARY, F. & MAHESHWARI., A., 2019: Time windowed data structures for graphs. Journal of Graph Algorithms and Applications, **23**(2), 191-226.

CHANCHARY, F., MAHESHWARI., A. & SMID, M., 2017: Querying Relational Event Graphs Using Colored Range Searching Data Structures. Discrete Applied Mathematics, CALDAM 2017, Lecture Notes in Computer Science, **10156**, Springer, 83-95.

CHANCHARY, F., MAHESHWARI., A. & SMID, M., 2018: Window Queries for Problems on Intersecting Objects and Maximal Points*. Algorithms and Discrete Applied Mathematics, CALDAM 2018. Lecture Notes in Computer Science, **10743**, Springer, 199-213.

CHAZELLE, B., 1986: Filtering Search: A New Approach to Query-Answering. SIAM Journal on Computing, **15** (3), 703-724.

DUCKHAM, M., KULIK, L., WORBOYS, M. & GALTON, A., 2008: Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. Pattern recognition, **41**(10), 3224-3236.

EDELESBRUNNER, H., 1992: Weighted alpha shapes. Technical Report UIUCDCS-R-92-1760, University of Illinois at Urbana-Champaign, Department of Computer Science.

EDELESBRUNNER, H., 2010: Alpha shapes - a survey. Tessellations in the Sciences, **27**, 1-25.

EDELESBRUNNER, H., KIRKPATRICK, D., & SEIDEL, R., 1983: On the shape of a set of points in the plane. IEEE Transactions on Information Theory, **29**(4), 551-559.

JONES, C., PURVES, R., CLOUGH, P. & JOHO, H. 2008: Modelling vague places with knowledge from the web. International Journal of Geographical Information Science, **22**(10), 1045-1065.

LIANG, J., EDELSBRUNNER, H., FU, P., SUDHAKAR, P. & SUBRAMANIAM, S., 1998: Analytical shape computation of macromolecules: I. molecular area and volume through alpha shape. Proteins: Structure, Function, and Bioinformatics, **33**(1), 1-17.

PENG, D. & WOLFF, A., 2014: Watch Your Data Structures! Proceedings of the GIS Research UK 22nd annual conference, GISRUK 2014. Glasgow, Scotland.

PURVES, R., CLOUGH, P. & JOHO, H., 2005: Identifying imprecise regions for geographic information retrieval using the web. Proceedings of the 13th Annual GIS Research UK Conference, Glasgow, Scotland, 313-318.

RUDI, A.G., 2017: Answering Time-Windowed Contiguous Hotspot Queries. Proceedings of the 1st Iranian Conference on Computational Geometry, ICCG, February 27, Teheran, Iran.

VAUHKONEN, J., KORPELA, I., MALTAMO, M. & TOKOLA, T., 2010: Imputation of single-tree attributes using airborne laser scanning-based height, intensity, and alpha shape metrics. Remote Sensing of Environment, **114**(6), 1263-1276.