

A Survey on Data Mining Methods for Clustering Complex Spatiotemporal Data

Piotr S. Maciąg^(✉)

Institute of Computer Science, Warsaw University of Technology,
Nowowiejska 15/19, 00-665 Warsaw, Poland
pmaciag@ii.pw.edu.pl
<http://www.ii.pw.edu.pl/ii.eng>

Abstract. This publication presents a survey on the clustering algorithms proposed for spatiotemporal data. We begin our study with definitions of spatiotemporal datatypes. Next we provide a categorization of spatiotemporal datatypes with the special emphasis on the spatial representation and diversity in temporal aspect. We conduct our deliberation focusing mainly on the complex spatiotemporal objects. In particular, we review algorithms for two problems already proposed in literature: clustering complex spatiotemporal objects as polygons or geographical areas and measuring distances between complex spatial objects. In addition to description of the problems mentioned above, we also attempt to provide a comprehensive references review and provide a general look on the different problems related to the clustering spatiotemporal data.

Keywords: Data mining · Clustering spatiotemporal data · Clustering algorithms

1 Introduction

Exploration of spatiotemporal data is a key aspect in many areas of management, design and business. Rapid increase of collected spatiotemporal data is associated with an intensive development of wireless sensor networks, improving sensors design techniques and increasing transmission capacity in mobile networks. Spatiotemporal data may be related to the following areas of applications: collections of events generated by sensors deployed over certain geographical regions, information about trajectories of vehicles, animals or groups of people or evolutions of phenomenons in both spatial and temporal aspects. Analysis of changes in climate and weather is the field which may generate huge amounts of spatiotemporal data described not only by the sets of points, but also by the complex objects like polygons. The problem of discovering frequent patterns in spatiotemporal data is related to several applications tasks like: analysis of traffic in cities [20], movements prediction of celestial bodies in astronomy [6] or crime analysis [41].

In addition, standard methods used in canonical data mining problems like *apriori* based algorithms [1], efficient clustering [18], periodicity detection [39],

fast validation and data interpolation [38] need to be integrated into non-trivial, sophisticated approaches which can deal with data uncertainty, shifts in spatial or temporal dimensions and non-invariant scaling problems. In the publication, we attempt to provide a survey on the most recent methods developed for the spatiotemporal data clustering.

In particular, we propose a review of clustering methods for both spatiotemporal points and polygons (geographical areas). Previous reviews on that matter consider only the categorization of spatial objects [34, 48]. Additionally, we review distance measures proposed for complex spatial or geographical objects. To the best of our knowledge this is the first survey on the mentioned measures, gathering their properties and computing algorithms from multiple resources and recently proposed publications. In opposition to our paper, the survey proposed in [47] considers the frequent patterns discovery methods rather than clustering event-based spatiotemporal data. An attempt to provide a review of patterns discovery methods for trajectory-based spatiotemporal data has been proposed in [39], which do not consider at all clustering methods for complex geographical objects and focus mainly on moving objects.

The layout of the paper is as follows: in Sect. 2 we review a categorization of spatiotemporal datatypes in the view of their adaptations to clustering algorithms. Section 3 summarizes results in the area of clustering complex spatiotemporal objects as polygons and areas. Section 4 recalls the most important distance measures for complex spatial objects. Section 5 provides a survey on the recently proposed clustering algorithms for moving objects and trajectories. Conclusions to the survey are given in Sect. 6.

2 Spatiotemporal Datatypes

Spatiotemporal datatypes are dependent on the real-world applications. Based on literature [34, 39, 48], we can distinguish two types of spatiotemporal data: event-based (also known as location-based), collected from stationary deployed sensors and trajectory-based (also referred as ID-based [39]) used to describe movements of objects. For the event-based data case, each event may be associated with a property p which value is denoted by the function $f(x, y, t, p)$ where (x, y) is a location (usually expressed in terms of longitude and latitude), t is a time stamp during which the event has been collected. Considering the more complex spatiotemporal objects (as polygons or areas), the location of an object may be denoted by the set of its coordinates. In the case of trajectory-based spatiotemporal data, for a given set of n objects o_1, o_2, \dots, o_n , a trajectory of an object o_i is represented by a sequence of geographical points $(x_1, y_1, t_1), (x_2, y_2, t_2), \dots, (x_n, y_n, t_n)$, where (x_j, y_j) is a location at the time stamp t_j . The above distinction between event-based and trajectory-based spatiotemporal data has been mainly introduced in [39].

The categorization presented above can be in addition broaden with the specification of different spatial datatypes (points, lines and polygons) and their extensions to time domain: database may contain only the last snapshot of actual

positions of observed objects or the whole history of an evolution of a spatiotemporal phenomenon. In some other cases only a stream of spatiotemporal data may be available.

The categorization of spatiotemporal datatypes is given in [34]. The authors of [34] distinguish the following categories: **Fixed location** denoting datasets containing occurrences of events of predefined types on geographical areas. Algorithms for clustering (and partially patterns discovery) for that datatype have been proposed in [9, 23, 41]. If the spatial dimension is extended to polygons or geographical areas, then one may refer to the clustering algorithms presented in [30, 50]. On the other hand, category defined as **Dynamic location** refers to trajectory-based spatiotemporal data. Datatypes denoted by **Updated snapshot** and either by **Dynamic location** or **Fixed location** labels refer to spatiotemporal data streams. The former describes moving objects reporting only the current location or position. The latter denotes streams of events occurrences (with each event described as above). Algorithms for spatiotemporal data streams clustering and classification have been recently proposed in [7, 31, 32].

The forms denoted by **Time Series** (according to categorization presented in [34]) are particularly useful because provide means for adaptation of classical algorithms and similarity measures used in time series analysis. For example, [14] presents a new similarity measure (*Edit Distance on Real sequence, EDR*) developed for the comparison of trajectories of objects. On the other hand, the time series representations of event-based spatiotemporal data are still unknown and will be developed in the future years.

In Table 1 we provide references contributing to the spatiotemporal datatypes definitions and data mining techniques proposed for them.

First propositions of spatial and spatiotemporal clustering using statistical approach have been given in [24, 36]. [36] gives a clustering method using spatial scan statistics (the approach has been improved in [27]), whereas [24] proposes an extension taking into account spatial shifts in the nature of evolving phenomenon. Due to this, proposed algorithm is able to detect clusters of spatiotemporal data which dynamically change their position and shape.

Clustering complex spatial and spatiotemporal objects is gaining attention of researchers nowadays [29, 30, 50]. The idea is to discover neighboring areas or geographical regions characterized by the same (or similar) value of non-spatiotemporal attribute (f.e. pollution). [26] adapts *Fuzzy C-Means* algorithm to spatiotemporal data. [25] raise the problem of anomaly detection in spatial time series using spatiotemporal clustering.

3 Clustering Spatiotemporal Events and Complex Geographical Objects

Proposed algorithms often operate on the more complex spatial objects f.e. polygons or lines. Classical density-based clustering algorithm - DBSCAN has been proposed in [18]. Many variations of the well-known density clustering algorithms like DBSCAN, OPTICS, NN were adapted to operate on spatiotemporal data.

Table 1. Summary of the publications on spatiotemporal data types and data mining techniques proposed for them.

No	First author	Spatiotemporal datatype	Data mining method
1	Erwig [17]	Datatypes definitions	
2	Wang [50]	Polygons (geographical areas)	Clustering
3	Birant [9]	Points (events)	Clustering
4	Joshi [30]	Polygons (geographical areas)	Clustering
5	Wang [49]	Points (events)	Clustering
6	Estvill-Castro [19]	Points (events)	Clustering
7	Wang [51]	Polygons	Clustering
8	Zhang [53]	Polygons (geographical areas)	Clustering
9	Damiani [15]	Points\trajectories	Clustering
10	Izakian [26, 27]	Points (time series)	Clustering
11	Izakian [25]	Points (time series)	Clustering\anomaly detection
12	Kulldorff [36]	Discrete events	Clustering\statistical approach
13	Iyengar [24]	Discrete events	Clustering
14	Schubert [46]	Discrete events	Anomaly detection
15	Mohan [41]	Discrete events	Patterns analysis
16	Shekhar [47]	Discrete events	Patterns analysis\anomaly detection
17	Nanni [43]	Trajectories	Clustering
18	Li [39]	Trajectories\discrete events	Clustering\patterns analysis
19	Palma [45]	Trajectories	Clustering
20	Gudmundsson [21]	Moving objects	Clustering
21	Jeung [28]	Moving objects	Clustering
22	Li [40]	Moving objects	Clustering

In addition, some non-standard grouping algorithms have been proposed: f.e. Spatio-Temporal Polygonal Clustering (STPC). Clustering algorithms are categorized into five main domains: *Partitioning*, *Hierarchical*, *Grid-based*, *Model-based* and *Density-based* [22].

In *Partitioning* methods, clusters are computed according to the mean value in a cluster (K-means) or based on the selection of an object which is nearest to the cluster's center (K-Medoid). The name of the category: *Partitioning methods* is inspired by the fact, that each object in a dataset is assigned to one and only one cluster (there are no objects classified as a noise). Objects partitioning is performed according to the predefined optimization criterion - for a given

number of clusters we would like to find assignments which minimizes the sum of distances between objects and centers of clusters (their mean values or the most central elements) to which they belong. *Hierarchical* clustering methods can be divided into two approaches: *ascending* and *descending*. In the ascending approach, each data object is initially assigned to its own cluster. Then, the algorithm gradually merges clusters until a predefined number of groups will be reached. On the other hand, descending approach divides one cluster (into which all data objects are initially assigned) until a predefined number of clusters will be reached.

Grid-based methods proceed with dividing data space into cells (a grid). Then, in the clustering phase, some cells are merged based on a predefined condition. For example: two dense, neighboring cells are merged into one (a cell is dense if it contains a number of objects greater than the predefined threshold). STING is an example of grid-based clustering algorithm [50]. *Model-based* methods try to fit clusters according to the predefined model of the data (like probability distribution). An example of a model-based method is the Expectation-Maximization algorithm [42]. *Density-based* methods try to find clusters according to distributions of density in a dataset. Due to this, appropriate density threshold should be specified by user: f.e. an estimated number of objects in a predefined neighborhood of an object [18]. Due to their simplicity, density-based clustering algorithms are widely used in data mining. Attempts to improve their efficiency and reduce needs of expert knowledge during parameter specification have been made in [4, 19].

3.1 Algorithms for Clustering Complex Spatiotemporal Objects

In this section, we proceed with description of clustering algorithms for complex spatiotemporal objects.

ST-GRID (SpatioTemporal-GRID) is a clustering method based on the partitioning spatiotemporal space into two separate grids: for spatial and temporal dimensions. In [49], the authors propose to compute the precision of a grid, based on the so-called *k-dist* graph which is constructed by random sampling of a dataset, calculating distance from each sample to its *k*-nearest neighbor and sorting calculated distances in decreasing order. The presence of clusters will be indicated by the easily noticeable threshold in the sorted distances. Calculated thresholds may be used as grid resolutions. As in the typical grid clustering algorithms, dense neighboring cells are merged to create spatiotemporal clusters. The above procedure has been originally developed only for spatiotemporal points.

ST-DBSCAN (SpatioTemporal-DBSCAN) is the algorithm developed on the conceptions of the well-known density clustering algorithm - DBSCAN (Density Based Clustering Algorithm with Noise) [18]. ST-DBSCAN has been introduced in [49] and then rearranged in [9]. ST-DBSCAN modifies DBSCAN to detect clusters according to their non-spatial, spatial and temporal dimensions. Before we describe ST-DBSCAN algorithm, we took a quick glance on the pure DBSCAN algorithm. One of the most important properties of DBSCAN is the

ability to detect clusters with an arbitrary shape: circular, ellipsoidal, linear or even more complicated. However, need to specify density thresholds may result that the algorithm will do not detect proper but sparse clusters. That problem has been addressed in the another density based clustering algorithm - OPTICS [4]. ST-DBSCAN considers cluster densities according to both spatial and temporal thresholds (assuming that for many applications they are very different). Additionally, ST-DBSCAN is able to include or exclude an object from the cluster on the basis of its non-spatiotemporal attributes: f.e. if the represented temperature is very different from the cluster's average temperature.

ST-SNN and ST-SEP-SNN Algorithms are two variations of the well known Shared Nearest Neighbor (SNN) density-based clustering algorithm. It is a noteworthy fact, that both algorithms *ST-SNN and ST-SEP-SNN* have been originally presented (in [50]) for clustering sets of polygons rather than points. Similarity between two objects according to the SNN algorithm is defined as a number of nearest neighbors shared by these two objects.

- A list of spatiotemporal neighbors of any polygon p is denoted by $NN(p) = k\text{-SPN-List}(p) \cap k\text{-TN-List}(p)$ where $k\text{-SPN-List}(p)$ and $k\text{-TN-List}(p)$ are lists of k neighbors of a polygon p in respectively spatial and temporal dimensions.
- Similarity between a pair of polygons p and q is the number of nearest spatiotemporal neighbors that they share: $similarity(p, q) = |NN(p) \cap NN(q)|$.
- The density of a polygon p is defined as the number of polygons that share at least Eps neighbors with p - $density(p) = |\{q \in \mathcal{D} | similarity(p, q) \geq Eps\}|$.
- A core polygon is a polygon where $CoreP(\mathcal{D}) = \{p \in \mathcal{D} | density(p) \geq MinPs\}$ where $MinPs$ is a user specified threshold.

The above conceptions determine clustering spatiotemporal polygons according to the *ST-SEP-SNN* algorithm. After marking each polygon either as a core or non-core, the algorithm proceed with clusters creation by processing each polygon in the dataset. During processing step, if an unprocessed core polygon p has been encountered, a new cluster is created and all polygons in the $NN(p)$ list are assigned to the new cluster (the same is recursively applied to the unprocessed core polygons encountered in the $NN(p)$ list).

ST-SNN is an algorithm that proceeds similarly to the *ST-SEP-SNN* algorithm presented above, with exception that the list of nearest neighbors $NN(p)$ of a polygon p is created using slightly different method. Rather than separately compute and then intersect lists of k -nearest spatial and temporal neighbors, *ST-SNN* combines spatial and temporal dimensions into one measure and computes only one list of the k -nearest neighbors.

STPC [30] is another density-based clustering algorithm developed for spatiotemporal polygons or areas. Again, the algorithm has been developed on the basis of the conceptions of the DBSCAN algorithm. Referring to the above mentioned *ST-SEP-SNN* algorithm, *STPC* computes lists of spatial and temporal neighbors on the basis of predefined distances (rather than k -nearest neighbors). The union of both lists contain spatiotemporal neighborhood of a polygon. If the neighborhood is appropriately dense, then the polygon is marked as a core polygon and the algorithm proceeds similarly to the above.

4 Distance Measures for Complex Spatiotemporal Objects

It is noteworthy to recall spatial distance measures used for polygons or other complex geographical objects. Table 2 provides a comparison of developed distance measures for complex spatiotemporal objects: polygons and trajectories. In the case of polygons, m and n denote the numbers of their vertices and in the case of trajectories their constituting sequences of locations.

Table 2. A comparison of distance measures for complex spatiotemporal data types.

No.	Distance	Spatiotemporal data type	Computational complexity
1	Hausdorff distance	Polygons and trajectories	$\mathcal{O}(m+n)$ [5] - only for polygons $\mathcal{O}(m*n)$ [2] - both for polygons and trajectories
2	Fréchet distance	Polygons and trajectories	$\mathcal{O}((m+n)\log(m+n))$ [10] - for convex polygons ^a $\mathcal{O}((m*n)\log(m*n))$ [3] - for trajectories
3	Exact separation distance	Polygons	$\mathcal{O}(\log(m) + \log(n))$ [44]
4	Minimum vertices approximation	Polygons	$\mathcal{O}(m*n)$ [44]
5	Centroid distance	Polygons	$\mathcal{O}(m+n)$ [44]
6	Simplified hausdorff distance	Polygons	$\mathcal{O}(m+n)$ [29]
7	Discrete Fréchet distance	Trajectories	$\mathcal{O}(m*n)$ [16]
8	Edit distance on real sequence	Trajectories	$\mathcal{O}(m*n)$ [14]
9	Dynamic time warping	Trajectories	$\mathcal{O}(m*n)$ [52]
10	Edit distance with real penalty	Trajectories	$\mathcal{O}(m*n)$ [13]

^aFor non-convex but simple polygons one may refer also to [10] where the algorithm with a non-trivial complexity is given.

Figure 1 presents a comparison between **Minimum Vertices Approximation**, **Exact Separation Distance** and **Centroid Distance**. Also, in the figure we show the Hausdorff distance for two polygons. The simplified Hausdorff distance is computed using the same formula as shown in Fig. 1, but only between vertices of polygons. Formula 1 presents a method for computing the Hausdorff distance for two polygons.

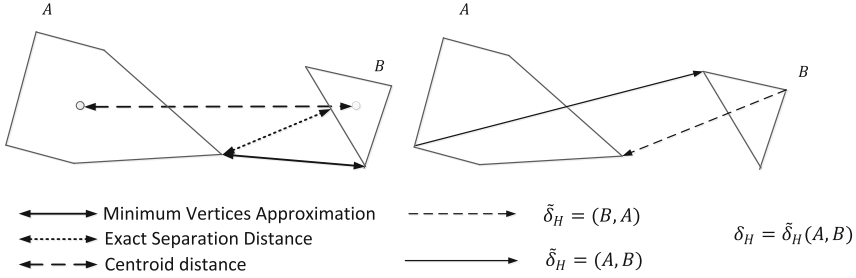


Fig. 1. Examples of distance measures for spatial polygons.

$$\delta_H = \max(\tilde{\delta}_H(A, B), \tilde{\delta}_H(B, A)) \tag{1}$$

$$\tilde{\delta}_H(A, B) = \max_{a \in A} \min_{b \in B} \| (a, b) \| .$$

A few distance measures presented in Table 2 may be used both for polygons (geographical areas) and trajectories. If particular distance measure preserves triangle inequality property, then it is possible to reduce the time of computations performed during clustering: f.e. neighborhood search [35]. Also, the methods combining the above measures with spatial and metrical indexes have been proposed in [11, 33].

5 Other Clustering Problems in the Area of Spatiotemporal Data

In this section, we provide a look on the other clustering problems related to spatiotemporal data. In particular, we attempt to provide a general overview of the most important methods proposed for clustering trajectory-based data.

Finding groups of similar moving objects - let assume that for a set of objects o_1, o_2, \dots, o_n a database stores the trajectory of a movement of each object. Additionally, let assume that each trajectory is represented in the form of a sequence of points, each associated with a timestamp. The problem of discovering *flocks* in the dataset is described as the problem of finding those sets of objects which for a predefined time interval stay within a disk which radius length is a parameter specified by an expert. A time interval is expressed as the sequence of consecutive timestamps. The problem of finding *flocks* of objects have been introduced in [21] and also developed in [8]. The above problem has been extended to finding *convoy*s [28] and *swarms* of moving objects [39]. A convoy is created from a flock by relaxing containment within a disk constraint i.e. rather than looking for the fixed disks of objects, the algorithm searches for dense regions using a clustering algorithm.

Clustering trajectories - the problem has been well studied in literature. Among the most popular algorithms for clustering spatiotemporal trajectories and their similar segments are: Trajectory-OPTICS [43], TRACLUS [37] or DENTRAC [12]. The important property of these algorithms is the ability

to cluster segments of trajectories rather than whole trajectories. The property is motivated by the fact that objects usually move together only in small segments of their trajectories. For example: TRACCLUS proceeds with three phases: in the first, a trajectory represented in the form of a sequence of points is simplified. The number of points in the sequence is reduced and the resulting parts of each trajectory are replaced with line segments. The replacement should preserve trends and angles representing turns in movements. Then, in the second phase, clustering of similar line segments is performed. In the last step, for each discovered group of similar line segments a representative trajectory is computed. Due to the complex nature of trajectories, an appropriate similarity metric should be selected. The proposed distance measure between two trajectories contains the following components: perpendicular, parallel and angle. The components are computed as follows: the perpendicular components is computed as a Lehmer mean of the distances between ending points of one segment projected into another. The parallel component is computed as a minimum from the distances between endings of one line segment projected into another. The angle component of the distance measure is defined as a product of the length of one of line segments and sinus of an angle between line segments.

6 Conclusions

In the publication, we provide a descriptive review of recently proposed algorithms for clustering complex spatiotemporal objects. In particular, we conduct a survey on the algorithms for clustering complex spatial objects: polygons or dynamically changing areas. Among the reviewed algorithms are ST-GRID, ST-DBSCAN, ST-SNN and STPC. Additionally, we provide references and a brief summary of the distance measures proposed for complex spatial objects (i.e. the Hausdorff distance, simplified Hausdorff distance and the other recently proposed heuristics). We also attempt to provide a look on the other methods proposed for clustering spatiotemporal objects i.e. trajectory-based data. The categorization of spatiotemporal datatypes presented at the beginning of the paper provides a starting point for considering new research fields in the area of spatiotemporal data mining. In particular, the most promising directions are: developing algorithms for spatiotemporal data streams and adaptation of knowledge discovery methods proposed in time series analysis to spatiotemporal data.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB 1994, pp. 487–499. Morgan Kaufmann Publishers Inc., San Francisco (1994)
2. Alt, H., Behrends, B., Blömer, J.: Approximate matching of polygonal shapes. *Ann. Math. Artif. Intell.* **13**(3), 251–265 (1995)
3. Alt, H., Godau, M.: Computing the Fréchet distance between two polygonal curves. *Int. J. Comput. Geom. Appl.* **05**(01n02), 75–91 (1995)

4. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: OPTICS: ordering points to identify the clustering structure. In: Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data, SIGMOD 1999, pp. 49–60. ACM, New York (1999)
5. Atallah, M.J.: A linear time algorithm for the hausdorff distance between convex polygons. *Inf. Process. Lett.* **17**(4), 207–209 (1983)
6. Aydin, B., Angryk, R.: Spatiotemporal frequent pattern mining on solar data: current algorithms and future directions. In: 2015 IEEE International Conference on Data Mining Workshop (ICDMW), pp. 575–581, November 2015
7. Bazan, J.G.: Hierarchical classifiers for complex spatio-temporal concepts. In: Peters, J.F., Skowron, A., Rybiński, H. (eds.) *Transactions on Rough Sets IX*. LNCS, vol. 5390, pp. 474–750. Springer, Heidelberg (2008). doi:[10.1007/978-3-540-89876-4_26](https://doi.org/10.1007/978-3-540-89876-4_26)
8. Benkert, M., Gudmundsson, J., Hübner, F., Wollé, T.: Reporting flock patterns. *Comput. Geom.* **41**(3), 111–125 (2008)
9. Birant, D., Kut, A.: ST-DBSCAN: an algorithm for clustering spatial-temporal data. *Data Knowl. Eng.* **60**(1), 208–221 (2007). *Intelligent Data Mining*
10. Buchin, K., Buchin, M., Wenk, C.: Computing the fréchet distance between simple polygons. *Comput. Geom.* **41**(1–2), 2–20 (2008). special Issue on the 22nd European Workshop on Computational Geometry (EuroCG)22nd European Workshop on Computational Geometry
11. Chan, K.P., Fu, A.W.C.: Efficient time series matching by wavelets. In: Proceedings 15th International Conference on Data Engineering (Cat. No. 99CB36337), pp. 126–133, March 1999
12. Chen, C.-S., Eick, C.F., Rizk, N.J.: Mining spatial trajectories using non-parametric density functions. In: Perner, P. (ed.) *MLDM 2011*. LNCS (LNAI), vol. 6871, pp. 496–510. Springer, Heidelberg (2011). doi:[10.1007/978-3-642-23199-5_37](https://doi.org/10.1007/978-3-642-23199-5_37)
13. Chen, L., Ng, R.: On the marriage of Lp-norms and edit distance. In: Proceedings of the Thirtieth International Conference on Very Large Data Bases, VLDB 2004, vol. 30, pp. 792–803. VLDB Endowment (2004)
14. Chen, L., Özsu, M.T., Oria, V.: Robust and fast similarity search for moving object trajectories. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, SIGMOD 2005, pp. 491–502. ACM, New York (2005)
15. Damiani, M.L., Issa, H., Fotino, G., Heurich, M., Cagnacci, F.: Introducing ‘presence’ and ‘stationarity index’ to study partial migration patterns: an application of a spatio-temporal clustering technique. *Int. J. Geogr. Inf. Sci.* **30**(5), 907–928 (2016)
16. Eiter, T., Mannila, H.: Computing discrete fréchet distance. Technical report, Vienna University of Technology (1994)
17. Erwig, M., Güting, R.H., Schneider, M., Vazirgiannis, M.: Spatio-temporal data types: an approach to modeling and querying moving objects in databases. *GeoInformatica* **3**(3), 269–296 (1999)
18. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Second International Conference on Knowledge Discovery and Data Mining, pp. 226–231. AAAI Press (1996)
19. Estivill-Castro, V., Lee, I.: Autoclust: automatic clustering via boundary extraction for mining massive point-data sets. In: Proceedings of the 5th International Conference on Geocomputation, pp. 23–25 (2000)
20. Gora, P., Rüb, I.: Traffic models for self-driving connected cars. *Transp. Res. Procedia* **14**, 2207–2216 (2016). Transport Research Arena (TRA 2016)

21. Gudmundsson, J., van Kreveld, M.: Computing longest duration flocks in trajectory data. In: Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems, GIS 2006, pp. 35–42. ACM, New York (2006)
22. Han, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers Inc., San Francisco (2005)
23. Huang, Y., Zhang, L., Zhang, P.: A framework for mining sequential patterns from spatio-temporal event data sets. *IEEE Trans. Knowl. Data Eng.* **20**(4), 433–448 (2008)
24. Iyengar, V.S.: On detecting space-time clusters. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 587–592. ACM (2004)
25. Izakian, H., Pedrycz, W.: Anomaly detection and characterization in spatial time series data: a cluster-centric approach. *IEEE Trans. Fuzzy Syst.* **22**(6), 1612–1624 (2014)
26. Izakian, H., Pedrycz, W., Jamal, I.: Clustering spatiotemporal data: an augmented fuzzy c-means. *IEEE Trans. Fuzzy Syst.* **21**(5), 855–868 (2013)
27. Izakian, H., Pedrycz, W.: A new PSO-optimized geometry of spatial and spatio-temporal scan statistics for disease outbreak detection. *Swarm Evol. Comput.* **4**, 1–11 (2012)
28. Jeung, H., Yiu, M.L., Zhou, X., Jensen, C.S., Shen, H.T.: Discovery of convoys in trajectory databases. *Proc. VLDB Endow.* **1**(1), 1068–1080 (2008)
29. Joshi, D., Samal, A., Soh, L.K.: A dissimilarity function for clustering geospatial polygons. In: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS 2009, pp. 384–387. ACM, New York (2009)
30. Joshi, D., Samal, A., Soh, L.K.: Spatio-temporal polygonal clustering with space and time as first-class citizens. *Geoinformatica* **17**(2), 387–412 (2013)
31. Kasabov, N., Capecci, E.: Spiking neural network methodology for modelling, classification and understanding of EEG spatio-temporal data measuring cognitive processes. *Inf. Sci.* **294**, 565–575 (2015). Innovative Applications of Artificial Neural Networks in Engineering
32. Kasabov, N., Scott, N.M., Tu, E., Marks, S., Sengupta, N., Capecci, E., Othman, M., Dobarjeh, M.G., Murli, N., Hartono, R., Espinosa-Ramos, J.I., Zhou, L., Alvi, F.B., Wang, G., Taylor, D., Feigin, V., Gulyaev, S., Mahmoud, M., Hou, Z.G., Yang, J.: Evolving spatio-temporal data machines based on the neucube neuro-morphic framework: design methodology and selected applications. *Neural Netw.* **78**, 1–14 (2016). special Issue on “Neural Network Learning in Big Data”
33. Keogh, E., Ratanamahatana, C.A.: Exact indexing of dynamic time warping. *Knowl. Inf. Syst.* **7**(3), 358–386 (2005)
34. Kisilevich, S., Mansmann, F., Nanni, M., Rinzivillo, S.: Spatio-temporal clustering. In: Maimon, O., Rokach, L. (eds.) *Data Mining and Knowledge Discovery Handbook*, pp. 855–874. Springer, Boston (2010)
35. Kryszkiewicz, M., Lasek, P.: TI-DBSCAN: clustering with DBSCAN by means of the triangle inequality. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) *RSCTC 2010. LNCS (LNAI)*, vol. 6086, pp. 60–69. Springer, Heidelberg (2010). doi:[10.1007/978-3-642-13529-3_8](https://doi.org/10.1007/978-3-642-13529-3_8)
36. Kulldorff, M.: A spatial scan statistic. *Commun. Stat. Theory Methods* **26**(6), 1481–1496 (1997)

37. Lee, J.G., Han, J., Whang, K.Y.: Trajectory clustering: a partition-and-group framework. In: Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, SIGMOD 2007, pp. 593–604. ACM, New York (2007)
38. Li, L., Revesz, P.: A comparison of spatio-temporal interpolation methods. In: Egenhofer, M.J., Mark, D.M. (eds.) GIScience 2002. LNCS, vol. 2478, pp. 145–160. Springer, Heidelberg (2002). doi:[10.1007/3-540-45799-2-11](https://doi.org/10.1007/3-540-45799-2-11)
39. Li, Z.: Spatiotemporal pattern mining: algorithms and applications. In: Aggarwal, C.C., Han, J. (eds.) Frequent Pattern Mining, pp. 283–306. Springer, Cham (2014). doi:[10.1007/978-3-319-07821-2-12](https://doi.org/10.1007/978-3-319-07821-2-12)
40. Li, Z., Ding, B., Han, J., Kays, R.: Swarm: mining relaxed temporal moving object clusters. Proc. VLDB Endow. **3**(1–2), 723–734 (2010)
41. Mohan, P., Shekhar, S., Shine, J.A., Rogers, J.P.: Cascading spatio-temporal pattern discovery. IEEE Trans. Knowl. Data Eng. **24**(11), 1977–1992 (2012)
42. Moon, T.K.: The expectation-maximization algorithm. IEEE Sig. Process. Mag. **13**(6), 47–60 (1996)
43. Nanni, M., Pedreschi, D.: Time-focused clustering of trajectories of moving objects. J. Intell. Inf. Syst. **27**(3), 267–289 (2006)
44. Ng, R.T., Han, J.: CLARANS: a method for clustering objects for spatial data mining. IEEE Trans. Knowl. Data Eng. **14**(5), 1003–1016 (2002)
45. Palma, A.T., Bogorny, V., Kuijpers, B., Alvares, L.O.: A clustering-based approach for discovering interesting places in trajectories. In: Proceedings of the 2008 ACM Symposium on Applied Computing, SAC 2008, pp. 863–868. ACM, New York (2008)
46. Schubert, E., Zimek, A., Kriegel, H.P.: Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. Data Min. Knowl. Disc. **28**(1), 190–237 (2014)
47. Shekhar, S., Evans, M.R., Kang, J.M., Mohan, P.: Identifying patterns in spatial information: a survey of methods. Wiley Interdisc. Rev.: Data Mining Knowl. Discov. **1**(3), 193–214 (2011)
48. Tork, H.F.: Spatio-temporal clustering methods classification. In: Doctoral Symposium on Informatics Engineering, vol. 1, no. 1, pp. 199–209. FEUP (2012)
49. Wang, M., Wang, A., Li, A.: Mining spatial-temporal clusters from geo-databases. In: Li, X., Zaïane, O.R., Li, Z. (eds.) ADMA 2006. LNCS (LNAI), vol. 4093, pp. 263–270. Springer, Heidelberg (2006). doi:[10.1007/11811305_29](https://doi.org/10.1007/11811305_29)
50. Wang, S., Cai, T., Eick, C.F.: New spatiotemporal clustering algorithms and their applications to ozone pollution. In: Proceedings of the 2013 IEEE 13th International Conference on Data Mining Workshops, ICDMW 2013, pp. 1061–1068. IEEE Computer Society, Washington, DC (2013)
51. Wang, W., Du, S., Guo, Z., Luo, L.: Polygonal clustering analysis using multilevel graph-partition. Trans. GIS **19**(5), 716–736 (2015)
52. Yi, B.K., Jagadish, H.V., Faloutsos, C.: Efficient retrieval of similar time sequences under time warping. In: Proceedings of the Fourteenth International Conference on Data Engineering, ICDE 1998, pp. 201–208. IEEE Computer Society, Washington, DC (1998)
53. Zhang, Y., Eick, C.F.: Novel clustering and analysis techniques for mining spatio-temporal data. In: Proceedings of the 1st ACM SIGSPATIAL PhD Workshop, SIGSPATIAL PhD 2014, pp. 2:1–2:5. ACM, New York (2014)