Noname manuscript No.
(will be inserted by the editor)

Operations to Support Temporal Coverage Aggregates over Moving Regions.

Mark McKenney · Roger Frye · Zachary Benchly · Logan Maughan

Received: date / Accepted: date

Abstract A temporal coverage operation computes the duration that a moving object covers a spatial area. We extend this notion into temporal coverage aggregates, in which the spatial area covered for a maximum or minimum amount of time by a moving region, or set of moving regions, is discovered. We define the max temporal aggregate coverage operation and the min temporal aggregate coverage operation. We provide an algorithm to compute these operations, and show that it is correct. Finally, the algorithm is implemented in the open source, Pyspatiotemporalgeom library to verify the algorithm under a variety of test cases.

Keywords Spatiotemporal Data; Moving Regions; Data Models; Aggregate Operations

1 Introduction

The growth of sensing platforms coupled with advances in cyber-infrastructure allowing the integration, storage, and availability of data has resulted in a rapid growth of spatiotemporal data that fits the paradigms of streaming data and stream analysis. Spatiotemporal data with extent, ie., data items represented as regions that change shape and position over time, provide significant opportunities in terms of i) analysis of the spatiotemporal attribute data associated with the data items, and ii) in terms of the information naturally encoded in the spatial and temporal behavior of the geometric representations of the data items themselves. Such data items, often referred to as *moving regions*, also present unique challenges: i) aggregate operations over moving regions are not fully defined and ii) algorithms to compute such operations tend to be

M. McKenney, R. Frye, Z. Benchly, L. Maughan
Department of Computer Science
Southern Illinois University Edwardsville
E-mail: {marmcke,rfrye,zbenchl,lmaugha}@siue.edu

complex since notions of a region's topological properties (ie., its interior, exterior and boundary) tend to have more distinct semantic differences in terms of data interpretation than do moving points or moving lines. Thus, the full algebra of analysis operations, especially at the implementation level, remains incomplete.

In this paper, we consider the notion of purely temporal aggregate operations over moving regions. In other words, this paper is concerned with identifying aggregate operations that are based purely on the spatiotemporal components of moving regions, and not on the attributes associated with space-time geometries. For example, given a set of moving regions, such that each moving region depicts the extent of hurricane force winds for a hurricane as it changes shape and position over time, an interesting operation is to discover the spatial areas covered by all hurricanes for the longest amount of time. Such areas represent places that have been impacted directly by hurricanes for the most amount of time, as opposed to places affected by the greatest number of hurricanes; for example, a single slow moving hurricane could inflict more damage on an area than several fast moving hurricanes. This type of query is agnostic of the attribute values associated with the moving regions, considering only spatiotemporal properties. We denote the operation to find the area covered for the longest by a set of moving regions the *max temporal coverage aggregate* (maxTCA) for a set of moving regions. Clearly the maxTCA has value in certain data sets, as does its converse, the *min temporal coverage aggregate* (minTCA), but such aggregates are not developed in moving region models. Furthermore, such aggregates have compelling use in stream processing in which, for example, the max time aggregate over the last hour of a data stream can be continually maintained.

We approach the problem of the maxTCA and minTCA from a general perspective and define it based solely on spatiotemporal regions; thus, the algorithms are not tied to a specific application domain. The contributions of this paper include 1) formally defining the concept of temporal coverage aggregates for moving regions, 2) identifying specific aggregate operations, 3) providing an algorithmic framework for computing these operations, and 4) developing an open source, reference implementation of the algorithm. The source code of the reference implementation is available at [11,12]. Furthermore, we discuss implementation methods for the algorithm that result in an expected linearithmic worst case time bound using appropriate data structures. A quadratic time variant of the algorithm is significantly easier to implement.

The paper is structured as follows. Related literature is presented in Section 2. Section 3 introduces the concept of temporal coverage aggregates and builds the formalism to define them. An algorithm implementing the temporal coverage aggregate operations is provided in Section 4. Section 5 discusses an implementation of the algorithm and provides examples. Problems that arise when extending the algorithm to use moving regions, as opposed to just interval regions, are discussed in Section 6. Finally, in Section 7, we draw some conclusions.

2 Related Work and Data Model

We group related work into two categories: work dealing with data types that we use to as the data model for this paper, and work dealing with aggregate operations, particularly existing approaches to spatial aggregates.

2.1 Data Types

In this paper, we consider temporal coverage aggregates over *complex regions*. Complex regions, as defined in [16], are areal data types representing a region containing one or more components, denoted *faces*, such that each face may contain zero or more *holes*. For example, the country of Italy can be represented as a single complex region in which the mainland and islands are each individual faces, with the face representing the mainland containing a single hole representing the Vatican City (which does not belong to Italy).

In addition to complex regions, the work in this paper uses complex points and complex lines. A complex point is a collection of one or more coordinates; thus, a complex point can contain multiple individual points. A complex line is a line that can have multiple disconnected features, and those features can branch and split [16].

Spatiotemporal data types have received much attention in the literature [4,5,18,21,3,14]. In [5], the type of *moving regions* is defined as a mapping from time to the type of complex regions. Let $[R]$ be the set containing all complex regions and τ be the set of all time instants (represented as real numbers), a moving region M is then defined as $M : \tau \rightarrow [R]$, with some restrictions. We denote the set of all moving regions as $[M]$. Moving regions are able to represent regions that change in shape and position over time; for example, an area representing the extent of hurricane force winds for a particular hurricane is naturally represented as a moving region.

At the discrete level, [9] represents a region as a collection of straight line segments that define the boundary of the region. Further, [9] defines a moving region as a collection of *moving segments*. A moving segment describes the motion of a line segment on the boundary of a region as it travels across a fixed time interval. Moving segments are typically represented as triangles in 3D space, showing a line segment as it contracts to a point over a time interval, or vice versa; we denote such a triangle a *delta triangle*. A collection of moving segments describing the motion of a region across a single time interval forms an *interval region*. Figure 1 shows an interval region defined by delta triangles. Because all edges of delta triangles of straight line segments moving segments do not change direction over the life of a delta triangle; thus, moving segments do not change direction within a single interval region. A collection of interval regions then forms a moving region. In this paper, we use the discrete model of moving regions, focusing on interval regions, to define temporal coverage aggregates.

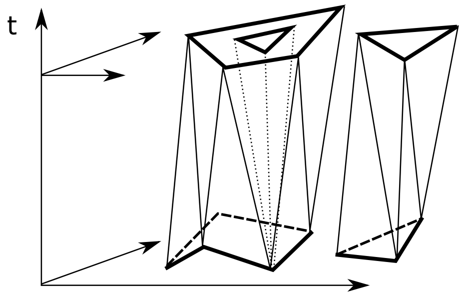


Fig. 1 An interval region represented as a collection of delta triangles. One face of the moving regions contains a hole that grows over the time interval. Delta triangles in the back of the volumes are not shown to reduce clutter. Delta triangles on the interior of the face (defining the hole) are dotted.

2.2 Aggregates

Aggregate operations have a long history of use and study in databases. The development of spatial aggregates is more recent, but has similarly received much attention. [10] presents a survey of spatiotemporal aggregate literature in which aggregates are categorized as spatial, temporal, or spatiotemporal. However, spatiotemporal aggregates in that paper focus mainly on the aggregation of attribute values related to spatiotemporal objects. Under that framework, spatiotemporal operations are used to collect the attributes of geometries that pertain to the result of the desired spatiotemporal operations, then aggregates are performed on those attribute; this differs from our focus on aggregating purely temporal information from moving regions.

Additional research on aggregate operations over spatiotemporal data includes [7,19,20]. [19] uses histograms to compute aggregates from a database, rather than streaming-style data. [20] treats aggregation of sequenced spatiotemporal data in the form of points and lines, but not regions. Kim [7] views the changes to regions as a continuously changing phenomenon with respect to time and generates a space-time volume from known discrete positions of the region, similar to the approach described in this paper. However, Kim does not perform aggregates with respect to temporal coverage.

A significant portion of the literature on spatial aggregates is devoted to mechanisms to support range queries, or box queries. Aggregate range queries perform some aggregate operation over spatial or spatiotemporal data that fall into a user specified area (the range or box), possibly over some specified time window [22,8,15]. Such aggregation mechanisms seem to stem from the support for range queries provided by spatial indexing methods such as R-trees and their variants [6,1,17], but much work has focused on developing data structures specific to aggregations. For these types of aggregates, the spatial portion of the aggregate tends to focus on discovering data objects that fall into the specified range, while the aggregate portion tends to be an aggregate operator over some attribute of the spatial objects that satisfy the range query. Thus, the aggregation portion of the operation behaves much like a traditional aggregate operator, and spatial operators are performed on spatial objects in isolation, rather than in aggregate, to compute an attribute value that is passed to the aggregate portion.

The algorithm presented in this paper requires geometric operations developed in the field of computational geometry; specifically, we must find a vertical line (in the temporal dimension) that intersects a volume that is maximized (or minimized) in terms of length. [2] describes computing the maximum distance between a pair of points and the minimum distance between a point and a parallel plane through another point contained within a point set in three-dimensional space. While this work is similar to the problem we must solve, it is too general for our specific problem. Instead, we take advantage of specific properties of moving regions to use more straightforward algorithms.

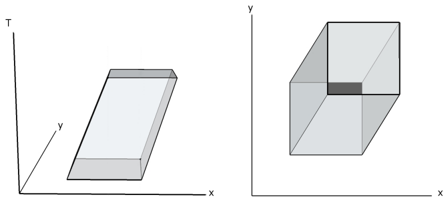
An early version of this work appears in [13]

3 Temporal Coverage Aggregates

In this section, we build a framework upon which the temporal coverage aggregates are defined. The advantage of this framework is that it can be extended to define additional aggregates. We begin with the concept of a *temporal coverage mapping* (TCM) operation. The TCM takes an interval region and returns a set of temporally static spatial objects paired with a *duration*, a time value indicating the duration that the area represented by the spatial object was covered by the interval region. For example, Figure 2 depicts an interval region consisting of a rectangle that translates in space over a time interval. In the top-down view of the interval region, the shaded portion indicates where rectangles at the two ends of the interval region overlap if they are projected out of the temporal dimension into the spatial dimensions. This area is clearly covered by the interval region over the entire time interval. Thus, the TCM for that interval region will contain a rectangle corresponding to that shaded area, with a duration indicating the length of the time interval. Furthermore, the interval region translates away from the leftmost line segment in Figure 2, causing that line segment (among others) to be covered for only an instant during the time interval; thus, the TCM will include a line segment, representing that line in space, and a duration indicating that it was covered for a single instant. Similarly, each spatial area covered by a distinct duration will exist in the TCM result set. More formally, let $[\alpha]$ be the set of all instances of complex points, lines, and regions, and $[D]$ be the set of all duration values (typically $[D] = \mathbb{R}$):

$$TCM : [M] \rightarrow 2^{[\alpha]\times[D]}$$

Temporal coverage operations (TCOs) are then defined by applying operations to the result of a temporal coverage mapping. Due to space limitations, we define two TCOs: *maxTCA* and *minTCA*. The *maxTCA* TCO returns the set of spatial areas covered for the maximum amount of time by an interval



(a) **(b)**

Fig. 2 (a) An interval region consisting of a rectangle that translates over a time interval. (b) a top-down view of (a); effectively the moving segments of (a) projected out of the temporal dimension.

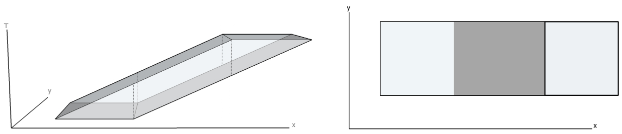
region; likewise, the *minTCA* operation returns the set of spatial areas covered for the minimum (non-zero) amount of time by an interval region. The syntax and semantics of the operations are as follows:

$$
\begin{aligned}
&maxTCA : 2^{[\alpha]\times[D]} \rightarrow 2^{[\alpha]}\\
&maxTCA(TCM(m)) \rightarrow \{x_1|\\
&(x_1, d_1) \in TCM(m) \wedge \forall(x_2, d_2) \in TCM(m) : d_1 \geq d_2\}\\
&minTCA : 2^{[\alpha]\times[D]} \rightarrow 2^{[\alpha]}\\
&minTCA(TCM(m)) \rightarrow \{x_1|\\
&(x_1, d_1) \in TCM(m) \wedge \forall(x_2, d_2) \in TCM(m) : d_1 \leq d_2\}
\end{aligned}
$$

3.1 Computing TOCs

The semantic expression of the TCM described above is not appropriate for computation of TOCs since it may result in an infinite mapping. For example, the result may contain an infinite series of lines that each exist for a single instant (for example, the area traversed by the leftmost moving segment in Figure 2 has a monotonically increasing amount of temporal coverage, resulting in an infinite set of lines each covered by a monotonically increasing amount of time at infinite temporal resolution). In this section, we describe an algorithm to compute the maxTCA and minTCA TOCs using well known geometric operations.

We frame the discussion of the computation of TOCs in terms of the maxTCA operation, since the minTCA operation follows the same template. Recall that moving segments describing an interval region form triangles in 3D space; thus, a moving segment is allowed to travel in a single direction, and at a constant rate, across a time interval. It follows that if any area is covered by an interval region for an entire interval, then that area will be covered by the regions at the interval boundaries. It further follows that the maximum time that an area can be covered by an interval region is limited to the duration of the time interval for which the interval region is defined. These observations lead to the easiest case of determining a maxTCA operation: the case when the regions at both ends of an interval region overlap when they are projected



(a) **(b)**

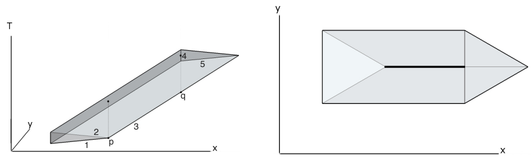
Fig. 3 A side (a) and top-down (b) view of an interval region.

out of the temporal dimension. Figure 2 depicts just such a case. Clearly, the dark shaded area in Figure 2b is covered by the interval region throughout the entire time interval, so it must be the area of maximum temporal coverage.

If the two regions at the temporal boundaries of an interval region are disjoint when projected out of the temporal dimension, the computation of TOCs becomes more difficult. The TCM produces a set of spatial objects (geometries) such that each entire object is covered by an interval region for the same duration; geometrically, this definition implies that the intersection of an interval region with any spatial object returned by a TCM that is extended infinitely in the temporal dimension is a 3-dimensional object g with constant height. In other words, the height of g is identical at every point covered by g in the spatial dimensions. For example, the shaded area shown in Figure 2b is the region of maximum temporal coverage, and thus the volume formed by extending that region in the temporal dimension across the duration of the interval region in Figure 2a will result in a 3-dimensional rectangle with constant height. Figure 3 shows a case in which the regions at the ends of the interval are disjoint. Figure 3b depicts a top down view of the interval region in Figure 3a. The shaded area in Figure 3b represents the region of maximum temporal coverage of the interval region in Figure 3a. Clearly, if the shaded region in Figure 3b is extended infinitely in the temporal dimension and intersected with the interval region in Figure 3a, the resulting volume will have constant duration for all points covered in the spatial dimension by the volume. The resulting volume is indicated by the dotted lines in Figure 3a. Note that the volume does not have constant values in the temporal dimension, but does have constant duration.

We make the observation that regardless of the dimension of the geometry of maximum coverage (ie., it may be a point, line, or region), the geometry of maximum coverage will always include a point on the boundary of at least one delta triangle; this follows from the fact that delta triangles are planar by definition. Furthermore, regardless of the dimension of the geometry of maximal coverage, each spatial object (point, line, or region) covered for the maximum amount of time by an interval region will be:

1. A point at which two boundaries of delta triangles intersect when projected out of the temporal dimension, denoted as *coverage points of interest* (CPOIs).
2. A line whose endpoints are CPOIs.
3. A region such that the end points of all line segments on the region's boundary are CPOIs.



(a) **(b)**

Fig. 4 A side (a) and top-down (b) view of an interval region.

We define the term *boundary points* of the geometry of maximum coverage to refer to the end points of line segments forming those geometries (or the point of maximal coverage in the case of a point geometry).

Theorem 1 *The boundary points of all geometries of maximum coverage that exist for an interval region are always CPOIs.*

Proof: In the case where the regions at either end of an interval overlap when projected out of the temporal dimension, it follows from the definition of regions that the geometry of maximum coverage is a region and that its boundary points are CPOIs. In the case where the regions at either end of the interval are disjoint, there are three cases.

i) *The geometry of maximum coverage is a point.* In this case, the point will correspond to the maximum distance between two 3-dimensional triangles that are non-parallel and overlap when projected out of the temporal dimension, or that meet at a point when projected out of the temporal dimension. If the triangles are non-parallel, and they are furthest apart at a single point, that point must be on the boundary of at least one of the triangles, and thus is a CPOI.

ii) *The geometry of maximum coverage is a line* (Figure 4). Again, since moving segments correspond to planar delta triangles in 3-dimensional space, if the geometry of maximum coverage is a line, then that line must lie on the boundary of at least one moving segment. Furthermore, that line will extend until the boundary of one triangle is no longer over the other, meaning that the boundaries of the triangles will intersect when projected out of time. Thus the boundary points of the line must be CPOIs.

iii) *The geometry of maximum coverage is a region.* This case can only occur when the triangles representing moving segments across an interval are parallel (since the triangles are planar). In such a case, the geometry of maximum coverage is a region that corresponds to the intersection of the projections of the triangle out of the temporal dimension, meaning its boundary points must be CPOIs. □

Theroem 1 implies that to discover the geometry of maximum coverage for an interval region, one must simply look at CPOIs. Once the CPOIs that are covered by an interval region for the longest amount of time are known, the task is simply to merge CPOIs into lines and regions as appropriate.

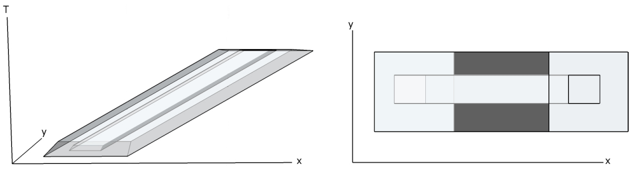
3.2 Merging CPOIs

The process of merging CPOIs into lines and regions of maximal temporal coverage is accomplished via a bookkeeping mechanism. To compute CPOIs, the edges of delta triangles are projected out of the temporal dimension, and the intersection of any of the edges results in a CPOI. Our approach to bookkeeping is to label each triangle edge; when two projected edges intersect, the CPOI is recorded once for each edge, and CPOI copies are labeled with the respective edge labels involved in the intersection. Once the maximal CPOIs are computed, CPOIs are sorted according to their label, and then their position along their assigned edge. If no holes are present in the interval region, adjacent CPOIs on an edge form a line segment. Line segments sharing end points are joined into lines. Finally, lines forming cycles are formed into regions. For example, consider Figure 4 in which some of the segments are labeled with numbers, and the points p and q are labeled. Segments 1, 2 and 3, when projected into the spatial dimensions, intersect at p; thus, the pairs $(p, 1), (p, 2)$, and $(p, 3)$ are recorded. Similarly, segments 3 and 4 will intersect at q, resulting in pairs $(q, 3)$ and $(q, 4)$ being recorded. The points are sorted first according to label, then according to position on the line: $(p, 1), (p, 2), (p, 3), (q, 3), (q, 4)$. p and q have the maximum time coverage, determined by extending p and q into the temporal dimension, and computing the length of intersection of the resulting lines with the volume induced by the interval region. Furthermore, p and q are adjacent on line segment 3, indicating that the entire line segment has maximal temporal coverage, and are joined into a line segment.

If an interval region contains holes, two additional steps are required. First, the adjacency of two points on a line segment does not guarantee that the line segment joining them has maximal temporal coverage by the interval region. For example, Figure 5 shows an interval region with a hole. In the top down view (Figure 5b), the left most CPOIs of the shaded region are all maximal coverage CPOIs that involve the same line segment, yet the hole causes a portion of the line segment to not be maximally covered. Thus, the adjacency of maximal coverage CPOIs is not sufficient to ensure a line segment is maximally covered. To verify, a single point along the original line between any two adjacent CPOIs must be tested for temporal coverage. If the coverage of that point is identical to the surrounding CPOIs on the line segment, then the line segment has maximal temporal coverage. Similarly, one cannot assume that a cycle of maximally covered line segments forms a maximally covered region. Instead, a point within such a cycle must be tested for identical time coverage as the cycle.

4 An Algorithm for Computing Temporal Coverage Aggregates

The complete algorithm to compute the maxTCO or minTCO is shown in Algorithm 1. In summary, the algorithm requires delta triangles to be projected into two dimensions (line 1). For an interval region with n moving segments,



(a) (b)

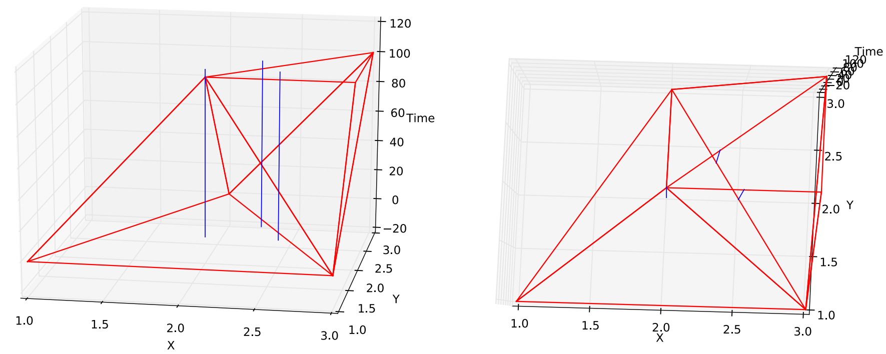
Fig. 5 A side (a) and top-down (b) view of an interval region.

Input: 3D line segments defining the edges of triangles that represent the moving segments defining an interval region.
Output: A set containing points, lines, and/or regions representing the areas covered for the largest amount of time by an interval region.

1 $S \leftarrow$ a set of line segments formed by projecting the 3D line segments forming edges of delta trinagles out of the temporal dimension;
2 $P \leftarrow$ the set of intersection points of all pairs $s_1 s_2 \in S$;
3 $P_l \leftarrow$ For each $p \in P$, create a labeled point for each segment involved in the intersection at p;
4 $S_e \leftarrow$ Extend each point $p \in P_l$ in the temporal dimension;
5 $P_m \leftarrow$ the points corresponding to each segment $s \in S_e$ whose intersection with the interval region volume has maximal duration;
6 Sort P_m by label, then by position along the line segment corresponding to the label;
7 $S_m \leftarrow$ create a set of segments from pairs of adjacent points on a line segment (using the labeled points). Keep only those such that a point on the segment in between the adjacent points has the same temporal coverage as the adjacent points;
8 $R_m \leftarrow$ create cycles from segments with shared end points in S_m. Keep a cycle if a point within the cycle has the same temporal coverage as the cycle boundary, otherwise it forms a hole in a region. Merge cycles that share edges into regions;

Algorithm 1: Algorithm for computing the maximum and minimum temporal coverage of an interval region.

this produces $O(n)$ line segments. The intersection of those projected line segments are computed to determine labeled CPOIs (lines 2-3). Line segment intersection is $O(n \lg n + k)$ where k is the number of intersection points (note that the intersections of line segments at end points will add the k in this case). The CPOIs are extended into the temporal dimension, and lengths for which they are contained within the interval region are recorded (lines 4-5). Because a volume representing a moving region is defined as a set of triangles (moving segments), this a segment/triangle intersection problem. Naively, this step takes $O(kn)$, but can be reduced using advanced data structures to $O(k \lg n)$. The CPOIs with maximal length are sorted (line 6) ($O(k \lg k)$). A point between any pair of CPOIs that are adjacent on a line segment is tested to see if the segment is maximally covered (line 7). Again, this is a segment/triangle intersection problem. Segments are formed into lines and cycles (using binary searches on a sorted list in $O(k \lg k)$ time. Finally, a point within each cycle is tested to see if the cycle is a maximally covered region (line 8) (another segment/triangle intersection problem). Thus, the algorithm has time complexity bounded by either $O(kn)$ if a naive segment/triangle intersection algorithm is used, or $O(k \lg k)$ otherwise.



a **b**

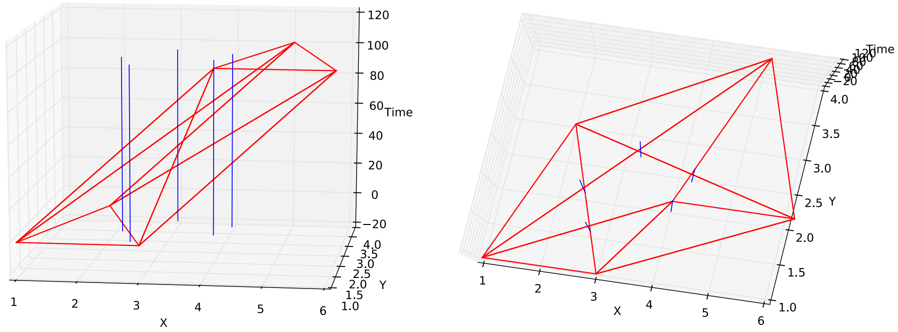
Fig. 6 An interval region in which the source and destination triangles overlap when projected out of time. Blue lines indicate CPOIs with nonzero duration projected through time.

5 Implementation

We have implemented the algorithm proposed in this paper as part of the Pyspatiotemporalgeom library [11, 12]. Pyspatiotemporalgeom is an open source python library providing spatiotemporal data types and operations. The code is not optimized, but is written as a proof of concept. Using the library, we verify the proposed algorithm over three scenarios. The first scenario consists of a triangle that moves and deforms over an interval such that the source and destination triangles overlap when projected out of the temporal dimension. Figure 6a shows a view of the scene from the side. The vertical blue lines in the scene indicate CPOIs that have nonzero temporal coverage. Clearly, the CPOIs forming the portion of the boundary regions that overlap when projected out of time intersect the volume for the entire duration. Figure 6b shows a top-down view of the same scene.

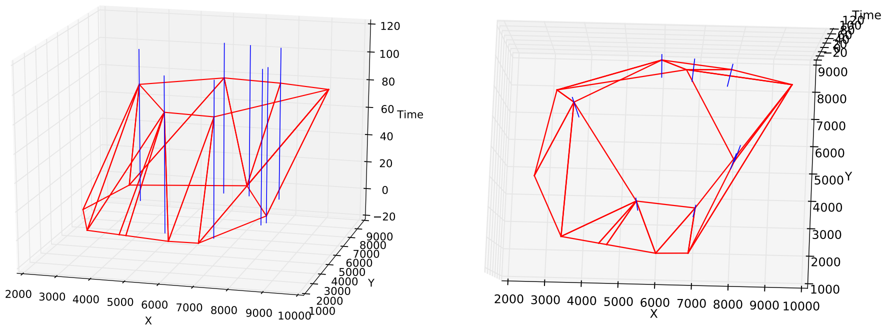
Figure 7 shows an example in which the source and destination regions do not overlap when projected out of time. In this case, there are two points of maximum coverage: $(2.7, 1.57)$ and $(4, 2)$. These two points occur on the line traversed by the point $(1, 1)$ in the source region as it travels to point $(4, 2)$ in the destination region. Because these points are adjacent on a line segment (the line segment created by that traversal), they form the end points of a line segment of maximal temporal coverage; this is visually verified in the image. Other CPOIs covered by nonzero duration are also indicated.

Figure 8 depicts a more complex example in order to verify the algorithm for non-trivial input. The source and destination regions are randomly generated, and the algorithm is run using the resulting interval region. Again, blue lines indicate the CPOIs with nonzero duration projected through time. In this case, the CPOIs of maximum coverage identify the portions of the regions that overlap when projected out of time.



a **b**

Fig. 7 An interval region in which the source and destination triangles do not overlap when projected out of time. Blue lines indicate CPOIs with nonzero duration projected through time.



a **b**

Fig. 8 An interval region in which the source and destination regions are randomly generated. Blue lines indicate CPOIs with nonzero duration projected through time.

6 Extending to Moving Regions

Theorem 1 relies on the fact that delta triangles are planar, meaning that a line segment does not change direction as it travels across an interval region. To expand the aggregates to moving regions made up of collections of interval regions, the problem emerges that a point covered by maximum, or minimum duration may be covered by multiple interval regions; the implication is that we cannot examine interval regions in complete isolation to arrive at the final answer. Theorem 1 still applies in that the point or points of maximum coverage will still occur on CPOIs, and the geometry of maximum/minimum coverage can still be computed using Algorithm 1, however, CPOIs must be computed using all line segments and all delta triangles in the moving region. Furthermore, all delta triangles from all interval regions must be tested against all CPOIs. Again, indexing and pruning techniques my be effectively employed, but the number of CPOIs and delta triangles involved can easily grow large.

7 Conclusion

In this paper, we formally defined temporal coverage aggregates, provided an algorithm to compute the operations, and showed the correctness of the algorithm. Furthermore, we discussed an implementation of the algorithm and used to to provide examples. The algorithm can be implemented in linearithmic time using appropriate data structures and techniques, or more straightforwardly in quadratic time.

Future work in this area includes the further identification of spatiotemporal aggregates that are geometrically based. A complete set of aggregate operations over moving objects would be a useful extension to moving objects algebras.

References

1. Beckmann, N., Kriegel, H., Schneider, R., Seeger, B.: The r*-tree: an efficient and robust access method for points and rectangles. SIGMOD Rec. **19**(2), 322–331 (1990)
2. Chazelle, B., Edelsbrunner, H., Guibas, L., Sharir, M.: Diameter, width, closest line pair, and parametric searching. In: Symposium on Computational Geometry, pp. 120–129. ACM (1992)
3. Cotelo Lema, J.A., Forlizzi, L., Güting, R.H., Nardelli, E., Schneider, M.: Algorithms for moving objects databases. The Computer Journal **46**(6), 680–712 (2003). DOI 10.1093/comjnl/46.6.680. URL http://comjnl.oxfordjournals.org/content/46/6/680.abstract
4. Forlizzi, L., Güting, R.H., Nardelli, E., Schneider, M.: A data model and data structures for moving objects databases. SIGMOD Rec. **29**(2), 319–330 (2000). DOI 10.1145/335191.335426. URL http://doi.acm.org/10.1145/335191.335426
5. Güting, R.H., Böhlen, M.H., Erwig, M., Jensen, C.S., Lorentzos, N.A., Schneider, M., Vazirgiannis, M.: A foundation for representing and querying moving objects. ACM Trans. Database Syst. **25**(1), 1–42 (2000)
6. Guttman, A.: R-trees: a Dynamic Index Structure for Spatial Searching. In: SIGMOD '84: Proceedings of the International Conference on Management of Data, pp. 47–57. ACM, New York, NY, USA (1984). DOI http://doi.acm.org/10.1145/602259.602266
7. Kim, K.S., Zettsu, K., Kidawara, Y., Kiyoki, Y.: Moving phenomenon: Aggregation and analysis of geotime-tagged contents on the web. In: Web and Wireless Geographical Information Systems, pp. 7–24. Springer (2009)
8. Lazaridis, I., Mehrotra, S.: Progressive approximate aggregate queries with a multi-resolution tree structure. SIGMOD Rec. **30**(2), 401–412 (2001)
9. Lema, J., Forlizzi, L., Güting, R.H., Nardelli, E., Schneider, M.: Algorithms for moving objects databases. The Computer Journal **46**(6) (2003)
10. Lopez, I., Snodgrass, R., Moon, B.: Spatiotemporal aggregate computation: a survey. IEEE Transactions on Knowledge and Data Engineering **17**(2), 271–286 (2005)
11. McKenney, M.: Pyspatiotemporalgeom package. https://pypi.python.org/pypi/pyspatiotemporalgeom/ (2016). Version 0.2.7.1, Accessed: 2016-01-14
12. McKenney, M.: Pyspatiotemporalgeom source code. https://bitbucket.org/marmcke/pyspatiotemporalgeom/ (2016). Version 0.2.7.1, Accessed: 2016-01-14

13. McKenney, M., Frye, R., Benchly, Z., Maughan, L.: Temporal coverage aggregates over moving region streams. In: Proceedings of the 5th ACM SIGSPATIAL International Workshop on GeoStreaming, IWGS '14, pp. 21–24. ACM, New York, NY, USA (2014). DOI 10.1145/2676552.2676555. URL http://doi.acm.org/10.1145/2676552.2676555
14. McKenney, M., Viswanadham, S.C., Littman, E.: The cmr model of moving regions. In: Proceedings of the 5th ACM SIGSPATIAL International Workshop on GeoStreaming, pp. 62–71. ACM (2014)
15. Papadias, D., Kalnis, P., Zhang, J., Tao, Y.: Efficient olap operations in spatial data warehouses. In: Proceedings of the International Symposium on Advances in Spatial and Temporal Databases (2001)
16. Schneider, M., Behr, T.: Topological relationships between complex spatial objects. ACM Trans. Database Syst. **31**(1), 39–81 (2006)
17. Sellis, T., Roussopoulos, N., Faloutsos, C.: The r+-tree: A dynamic index for multi-dimensional objects. In: Proceedings of the International Conference on Very Large Data Bases (1987)
18. Sistla, A.P., Wolfson, O., Chamberlain, S., Dao, S.: Modeling and querying moving objects. In: Proceedings of the Thirteenth International Conference on Data Engineering, ICDE '97, pp. 422–432. IEEE Computer Society, Washington, DC, USA (1997). URL http://dl.acm.org/citation.cfm?id=645482.653301
19. Tao, Y., Papadias, D.: Historical spatio-temporal aggregation. ACM Transactions on Information Systems (TOIS) **23**(1), 61–102 (2005)
20. Timko, I., Böhlen, M., Gamper, J.: Sequenced spatiotemporal aggregation for coarse query granularities. The VLDB Journal **20**(5), 721–741 (2011)
21. Worboys, M.F.: A unified model for spatial and temporal information. The Computer Journal **37**(1), 26–34 (1994). DOI 10.1093/comjnl/37.1.26. URL http://comjnl.oxfordjournals.org/content/37/1/26.abstract
22. Zhang, D., Tsotras, V.J.: Improving min/max aggregation over spatial objects. In: Proceedings of the 9th ACM international symposium on Advances in geographic information systems, GIS '01 (2001)