Modeling and Representing Real-World Spatio-Temporal Data in Databases

José Moreira[1]
Department of Electronics, Telecommunications and Informatics, University of Aveiro,Portugal
Institute of Electronics and Informatics Engineering of Aveiro, University of Aveiro, Portugal
jose.moreira@ua.pt

José Duarte
Institute of Electronics and Informatics Engineering of Aveiro, University of Aveiro, Portugal
hfduarte@ua.pt

Paulo Dias
Department of Electronics, Telecommunications and Informatics, University of Aveiro,Portugal
Institute of Electronics and Informatics Engineering of Aveiro, University of Aveiro, Portugal
paulo.dias@ua.pt

Abstract

Research in general-purpose spatio-temporal databases has focused mainly on the development of data models and query languages. However, since spatio-temporal data are captured as snapshots, an important research question is how to compute and represent the spatial evolution of the data between observations in databases. Current methods impose constraints to ensure data integrity, but, in some cases, these constraints do not allow the methods to obtain a natural representation of the evolution of spatio-temporal phenomena over time.

This paper discusses a different approach where morphing techniques are used to represent the evolution of spatio-temporal data in databases. First, the methods proposed in the spatio-temporal databases literature are presented and their main limitations are discussed with the help of illustrative examples. Then, the paper discusses the use of morphing techniques to handle spatio-temporal data, and the requirements and the challenges that must be investigated to allow the use of these techniques in databases. Finally, a set of examples is presented to compare the approaches investigated in this work. The need for benchmarking methodologies for spatio-temporal databases is also highlighted.

2012 ACM Subject Classification Information systems → Spatial-temporal systems; Information systems → Data model extensions; Computing methodologies → Shape modeling

Keywords and phrases spatio-temporal databases, region interpolation problem, moving regions, morphing techniques

Digital Object Identifier 10.4230/LIPIcs.COSIT.2019.6

Category Vision Paper

Funding This work is partially funded by National Funds through the FCT (Foundation for Science and Technology) in the context of the projects UID/CEC/00127/2013 and POCI-01-0145-FEDER-032636.
José Duarte: has a research grant awarded by the FCT with reference PD/BD/142879/2018.

1 Introduction

Nowadays, there are many technologies and devices that capture large amounts of data about the position, shape and size of spatial phenomena over time. Although there are several commercial and open-access tools for storing and processing spatial data, few exist to deal

[1] Corresponding author

14th International Conference on Spatial Information Theory (COSIT 2019).
Editors: Sabine Timpf, Christoph Schlieder, Markus Kattenbeck, Bernd Ludwig, and Kathleen Stewart; Article No. 6; pp. 6:1–6:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

with the evolution of spatial data over time. There are two main approaches to handle spatio-temporal data called the discrete model and the continuous model. In the discrete model spatio-temporal data are represented as sets of cells (points) in time (1D) and space (2D). This approach is simple, but the resolution of the model depends on the size of the cells and the computational cost can be cumbersome when handling large amounts of data.

In the continuous model data are represented in vector mode using abstract data types usually referred to as moving objects, such as, moving points, moving lines, and moving polygons (also called moving regions). These are represented as ordered sequences of points, lines or polygons (observed values), and functions describing their spatial transformations (e.g., translation, rotation, scaling and deformation) during the time interval between two consecutive observations [4, 6]. Data models and query languages exist that largely meet the needs identified in the literature to model the evolution of spatial data over time, and there are also prototype systems such as Secondo [8] and Hermes [19] which have been developed manly for research and teaching. However, their use in real case studies is almost non-existent.

Reconstructing data between observations is not trivial. Current solutions focus on the creation of interpolations that are robust and valid, but do not create realistic approximations of the evolution of spatio-temporal phenomena between observations in some circumstances. In fact, the search for methods capable of obtaining interpolations that are robust and realistic at the same time, compatible with the requirements usually imposed to represent spatial and temporal data in databases, remains a research topic.

In this paper, our goal is to show that morphing techniques, used to create video animations among other applications, can be investigated to represent the evolution of spatio-temporal data in databases. We are particularly interested in modeling spatio-temporal data using continuous models of time and space. We show that morphing techniques can obtain interpolations that are closer to the actual evolution of real-world phenomena than the interpolations created using the methods proposed in the spatio-temporal databases literature, and we identify the constraints and the issues that must be studied to enable the use of morphing techniques to create spatio-temporal data. The aim is to find solutions to represent data with small errors of approximation, so that they can be used in scientific and engineering applications, and that can be used to develop methods and algorithms to perform spatio-temporal operations in databases.

The remainder of this paper is organized as follows. Section 2 presents an overview on modeling and querying spatio-temporal data in databases. Emphasis is given to the methods proposed in the literature to create spatio-temporal data from observations. Section 3 introduces the use of morphing techniques in the context of spatio-temporal databases. Two main issues are covered: polygon matching and the representation of the evolution of spatial data between known observations. Section 4 presents examples comparing the results obtained when using one of the main methods proposed in the spatio-temporal databases literature and a morphing technique. Section 5 concludes this work and presents guidelines for future research.

2 The region interpolation problem in spatio-temporal databases

2.1 Spatio-temporal data models and query languages

A moving object is a triple (T, S, f), where $T \subset \mathbb{R}$ is the time domain, $S \subset \mathbb{R}^2$ is the spatial domain and $f : \mathbb{R} \times \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is a function that gives the transformation of S during T, such that $(t, x, y) \in \mathbb{R} \times \mathbb{R}^2 \mid (\exists x')(\exists y')(\, t \in T \wedge (x', y') \in S \wedge (x, y) = f(t, x', y')\,)$ [10]. This is an abstract definition that must be transposed into a discrete (finite) model suitable for implementation in a database system. Several data models and query languages have been proposed, but the most complete, and also the most successful, is, by far, the approach based on abstract data types [4, 6, 9]. This data model allows the representation of objects such as moving points, moving lines and moving regions that may have complex shapes, e.g., regions with holes. The authors also propose a comprehensive set of spatio-temporal operations, such as, projections (e.g., the shape of a moving region at a given time instant, the footprint of a moving region during a time interval or numerical measures, such as, the velocity or the size of a moving region over time), topological operations to evaluate the interaction of a moving object with other moving or static objects, distance operations and predicates. In addition, abstract data types can be smoothly integrated in the database management systems currently in use, which has also contributed to the success of this approach over other interesting proposals, namely, spatio-temporal constraints databases [3, 7].

One of the most interesting features of the data model proposed in [6] is the sliced representation (Figure 1).

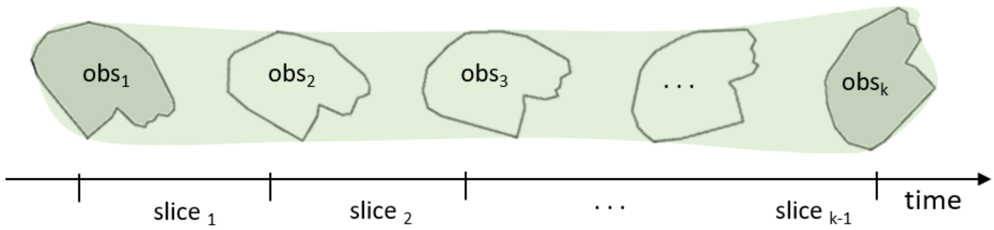

Figure 1 Sliced representation of a moving region.

The example shows the development of a moving region from left to right. Each polygon represents an observed value, i.e., the known shape of a moving region at a specific time instant. Each slice represents the development of the moving region between two consecutive observations. The definition of a slice includes, at least, the position and the shape of the moving object at a given time, and a function used to represent the spatial transformations (translation, rotation, skewing, etc.) of the object between consecutive observations. Moreover, constraints on the continuity of consecutive slices must be imposed.

A major challenge is to find a function that represents the spatio-temporal behavior of moving objects between observations as closely as possible. This problem is particularly complex when handling moving regions and moving lines, and it is often referred to as the Region Interpolation Problem. In this paper, this problem is discussed from two different points of view: special-purpose solutions proposed in the databases research community (Section 2.2) and general-purpose morphing techniques (Section 3).

2.2 Creating spatio-temporal data from observations

The region interpolation problem in spatio-temporal databases has been studied for the first time in [22]. The main contribution of this work is the so-called rotating-plane algorithm, that is used in subsequent works, which allows moving regions to be created from an ordered sequence of observations. Figure 2 illustrates the interpolation between a source (P) and a target (Q) region obtained using this algorithm.

The algorithm scans the line segments in P and Q one by one (e.g., in counter-clockwise order), starting with the ones having the smallest angle relatively to the x-axis in each shape ($\bar{p}_1$ and $\bar{q}_1$, respectively). The movement from P to Q is described by linear equations.

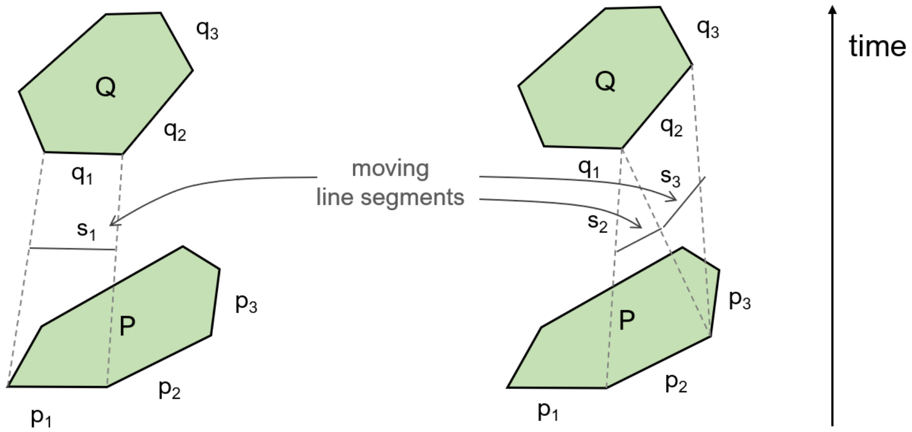

Figure 2 Rotating-plane algorithm: when the line segments (e.g., $\bar{p}_i$ and $\bar{q}_j$) are parallel (left-hand side); when the angle of the line segments (e.g., $\bar{p}_{i+1}$ and $\bar{q}_{j+1}$) is different (right-hand side).

When the angles of the selected line segments in P and Q are equal (e.g., $\bar{p}_1$ and $\bar{q}_1$), a linear transformation of $\bar{p}_1$ into $\bar{q}_1$ is performed (see the moving line segment $\bar{s}_1$), and the algorithm goes to the next segment in both shapes ($\bar{p}_2$ and $\bar{q}_2$, respectively). When the angle of the selected segment in P is smaller than the angle of the selected segment in Q (e.g., $\bar{p}_2$ and $\bar{q}_2$), the first ($\bar{p}_2$) degenerates progressively (see the moving line segment $\bar{s}_2$) into the first point (in counter-clockwise order) of the other ($\bar{q}_2$) and the algorithm goes to the next segment ($\bar{p}_3$) of P. The procedure is equivalent when the angle of the selected segment in P (e.g., $\bar{p}_3$) is greater than the angle of the selected segment in Q (e.g., $\bar{q}_2$). In this case, a point in P becomes a line segment in Q (see the moving line segment $\bar{s}_3$). So, as depicted in the right-hand side of the figure, when the angle of the selected line segments in P and Q is different, the movement of a line segment from the source to the target is given by the movement of two (or more) line segments ($\bar{s}_2$ and $\bar{s}_3$).

This algorithm does not allow line segments to rotate, which is interesting to avoid invalid interpolations. An interpolation is invalid if, for example, there are line segment intersections during interpolation. This choice makes the implementation of spatio-temporal operations easier, because the movement of the vertices is described by linear equations, and it ensures that the 3D representation of the resulting moving region in (x, y, t)-space is a polyhedron. It also helps keeping compatibility with existing spatial DBMS, which, in most cases, are not able to handle curves. However, the approximation of the evolution of moving regions that rotate is poor, as illustrated in Figure 3.

This example shows a polyhedron representing a fixed moving region that rotates approximately 45 degrees. Although the shapes of the source and target are equal, we observe that the intermediate shape of the moving object estimated using the rotating-plane algorithm at the middle of the interpolation differs greatly from P and Q. This problem arises because, as exemplified in Figure 2, the rotation of a line segment is represented implicitly by the linear movement of two (or more) line segments.

As the rotating-plane algorithm is only able to handle convex shapes, [22] proposes to split non-convex shapes into convex features and to organize them in a convex hull tree. Each node (feature) is a hole in the convex hull of the shape in the parent node. Finding a correspondence between the features in the two convex hull trees is difficult. This issue is approached superficially in [11, 22].

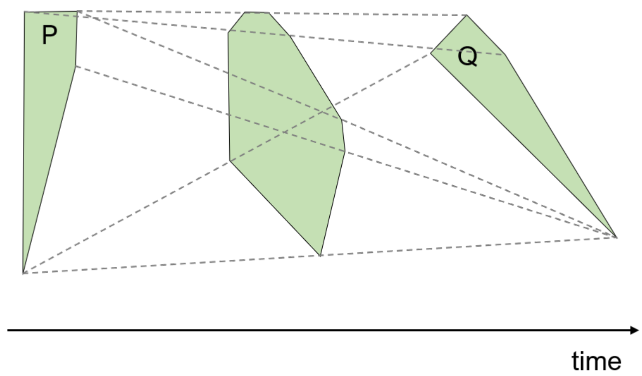

Figure 3 Interpolation created using the rotating-plane algorithm.

In [16] a counter-example demonstrating that the rotation-plane algorithm is not robust is presented. The authors also argue that it is not always possible to create a single interpolation between a source and a target that is valid at all times. So, they propose an algorithm to split an interpolation into three parts at most to avoid line segment intersections during interpolation. Concavities are collapsed into or expanded from a single point (depending on whether the concavities are in the source or in the target, respectively), and intersecting concavities are detected and removed using a process called evaporation (concavities disappear), and condensation (concavities appear later in the interpolation). This can cause an anomalous deformation of the moving region during interpolation, but it has the advantage that the interpolations are always valid. This is demonstrated using an example involving highly complex (snail-shaped) shapes.

This work is extended in [15] to handle moving regions with holes and with a variable number of components (multi-regions). This includes dealing with transformations, such as, splitting and merging regions during interpolation. Almost at the same time, [11] revisited the work in [22] to make it robust by using a strategy similar to the one used in [16].

All the methods mentioned above are based on the rotating-plane algorithm. Yet, this algorithm has well-known issues, particularly, when representing moving regions that rotate and when representing concavities. For that reason [12] presents a different approach to handle moving regions with fixed shape. It presents a data model that can handle curves, and algorithms to compute spatio-temporal operations, namely, the spatial footprint of a moving region, the intersection of a moving region with a static object and the intersection of a moving region with a moving point. Moving segments are allowed to rotate, but the shape of the region is fixed, i.e., the only spatial transformations allowed are translation and rotation.

3 Using morphing techniques in spatio-temporal databases

3.1 Creating interpolations using morphing techniques

The transformation of a source into a target is a problem that has been studied since the beginning of the 1990s in areas such as video animation, gaming and medical imaging. Many techniques were proposed for the interpolation of 2D images, free-form curves, planar shapes (e.g., polygons and polylines) and volumetric representations (3D objects) [20]. The morphing of free-form-curves and planar shapes shares some similarities with the region interpolation problem in spatio-temporal databases. The main objective is to obtain an interpolation that is smooth and realistic, providing visually appealing animations.

Some morphing techniques use iterative methods, which allow for sophisticated interpolations, but the computational costs are high and therefore they would not be suitable for processing queries on large datasets. Yet, there are approaches to estimate the shape of a moving region between observations using formulas. This is the case of the approach proposed in [1], which takes as input two meshes created using a compatible triangulation algorithm, e.g., [14, 21]. Thus, the source and the target meshes have the same number of triangles and there is a one-to-one correspondence between them. The interpolation is given by the affine transformation of each triangle of P into the corresponding triangle in Q and is obtained using Single Value Decomposition. Since the transformation of each triangle is independent of the transformation of its neighbors, it is necessary to calculate a unique position for the shared vertices. Simple solutions, such as, computing the midpoint or using a least squares formulation, can be used for that purpose. A reformulation of this problem using normal equations is presented in [2]. Figure 4 displays an interpolation created using this approach.

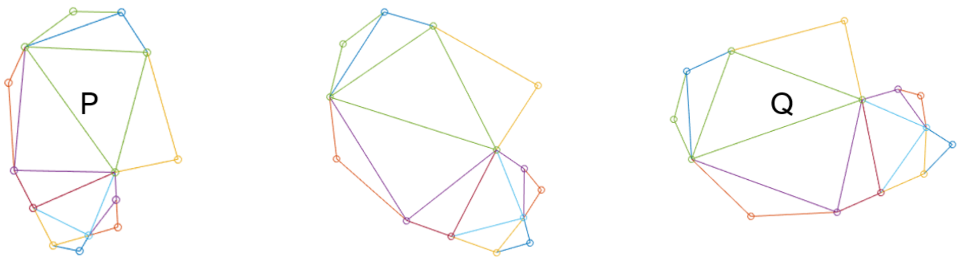

Figure 4 Interpolation created using the method proposed in [1, 2].

The results presented in [1, 2] show that it is possible to obtain realistic interpolations even when the shapes are very different and complex (Figure 5).

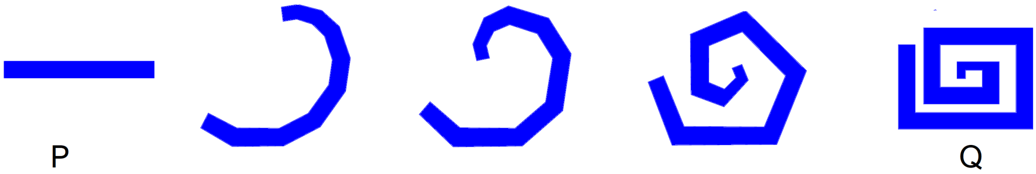

Figure 5 Another interpolation created using the method proposed in [1, 2].

The representation of a complex shape as a mesh of triangles allows using divide-and-conquer strategies to implement complex operations. For instance, two moving regions intersect if any triangle of P intersects at least one triangle of Q during the interpolation. In addition, there are already several algorithms to deal with triangles in computational geometry that may be useful to implement spatio-temporal operations. Finally, since the interpolation is given by a transformation matrix, the shape of a moving region can be estimated at any given time using a single formula.

3.2 Polygon matching

When transforming a shape into another shape some notion of correspondence or matching between the two shapes must be defined. This leads to another problem known as the vertex correspondence problem or polygon matching. Polygon matching consists of finding a correspondence between the elements (e.g., vertices or edges) of two shapes. This paper outlines two representative approaches to solve this problem.

The first uses the concept of feature points [13, 17], which are a subset of vertices that best represent the shape of the objects (e.g., the numbered vertices in Figure 6). A feature point is described using measures, such as the angle and the distance to its neighbors, which ideally should be invariant under translation, rotation and scaling. The mapping between the features points in a source and a target is done using similarity functions. If the number of vertices between two features points in the source is different from the number of vertices between the two corresponding feature points in the target, new vertices are added. This creates a one-to-one correspondence between all vertices in the source and the target, as required by many interpolation algorithms.

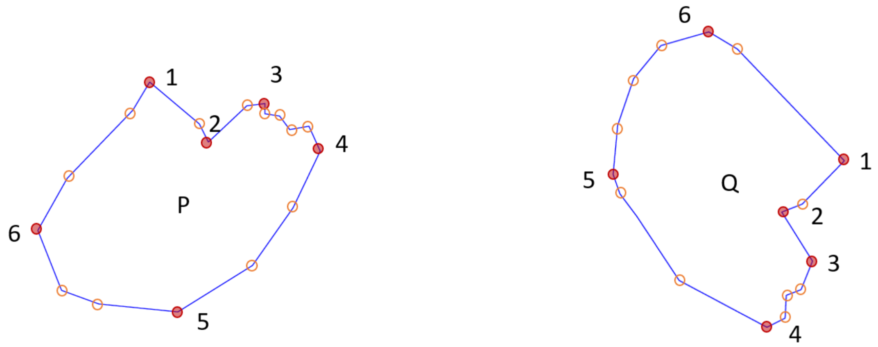

Figure 6 Polygon matching using feature points.

The second approach uses turning functions to find a correspondence between the vertices of two shapes. The method consists of representing the length of the edges (d) and the turning angles α of a source and a target in a two-dimensional chart (Figure 7).

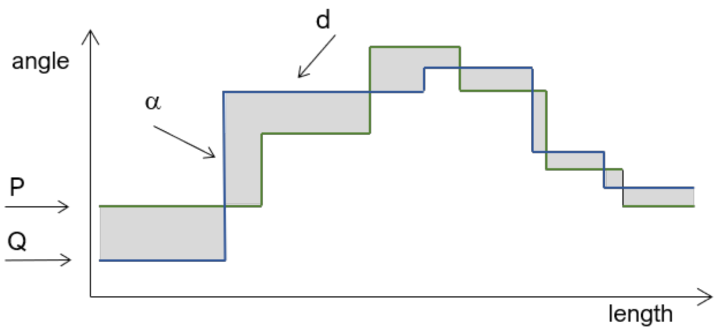

Figure 7 Polygon matching using turning functions.

The sum of the lengths of the segments in each polygon must be equal to 1. The algorithm consists of shifting edges from left to right or vice-versa, to find the mapping of the vertices in P and Q that minimizes the sum of the colored areas (gray rectangles) in the figure.

4 Representation of spatio-temporal data using morphing techniques

4.1 Examples

This section compares the methods discussed in section 2 with the morphing technique presented in Section 3, focusing on the representation of real-world phenomena in spatio-temporal databases. Since the representation of spatial transformations such as translation and scaling is easy, the emphasis is given to moving regions rotation and morphing. The data used in the examples were extracted from satellite images monitoring the movement of two large blocks of an iceberg in the Antarctic region. The images were segmented to extract the shape of the icebergs at different dates, and the correspondence between the shapes of consecutive pairs of observations of an iceberg was obtained using the method proposed in [13].

The example in Figure 8 shows the evolution of the shape of an iceberg between two consecutive observations (P and Q). The three snapshots in the middle were estimated using the method proposed in [11]. The predominant spatial transformation is rotation and the shape of the iceberg has a large concavity.

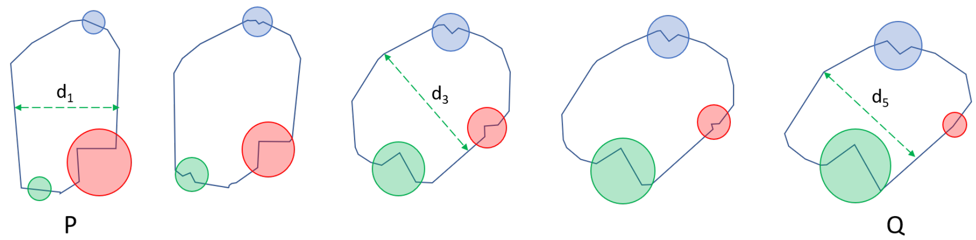

Figure 8 Interpolation created using the method proposed in [11].

This example highlights the main issues with the methods presented in section 2 when considering deformable moving regions. First, it is observable that the shape of the moving region tends to inflate (expand) until it reaches the middle of the interpolation and has an opposite behavior during the second half of the interpolation (the length of d_3 is greater than the lengths of d_1 and d_5). This is also observable in Figure 3. This means that numerical measures, e.g., the area, tend to increase and decrease during the interpolation, which usually is not the expected behavior in real-world phenomena. The anomalous deformation is due to the constraints imposed by the rotating-plane algorithm, which does not allow line segments to rotate (the rotation of a line segment is simulated by the movement of two or more line segments that move linearly). This algorithm is also used in [15] and [11, 22], and so, the results are similar. It is also important to note that the polygon matching step is not performed in the rotating-plane algorithm: the first pair of line segments to be processed is chosen using heuristics, e.g., the smallest angle relatively to the x-axis, which may not be a good choice in many cases.

This figure also highlights issues on handling concavities. The light blue circles show a normal case where the concavity at the top appears progressively during the interpolation. However, the concavity on the bottom right does not go along with the rotation of the object from P to Q. Instead, it is artificially divided in two: the concavity in P, which disappears progressively (marked by the light red circles), and the concavity in Q, which appears progressively (marked by the light green circles) during the interpolation[2]. This is caused by the method proposed in [22] to map the concavities in the convex-hull trees of the source and target. The mapping is based on heuristics, such as, the distance between centroids or the percentage of overlapping between features (in this context, a feature is a convex-polygon representing a hole), but, as shown in [18], this method is not safe, particularly with noisy data.

[2] Best illustrated in the video animation accessible from `http://most.web.ua.pt/resources/cosit2019/concavities.mp4`

When using the method proposed in [16], concavities tend to vanish because each concavity is mapped to a point in the convex-hull of the other shape, and they reappear during the second half of the interpolation. This is a good strategy to guarantee that interpolations are robust, but may cause important deformations as shown in Figure 9.

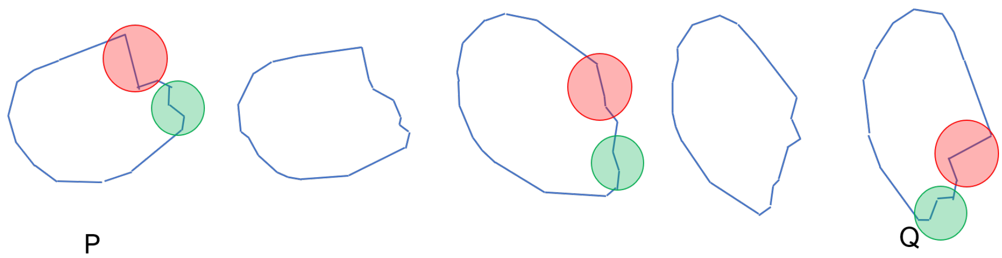

Figure 9 Interpolation created using the method proposed in [16].

Figure 10 displays an interpolation created using compatible triangulation [14] and the morphing technique proposed in [1, 2].

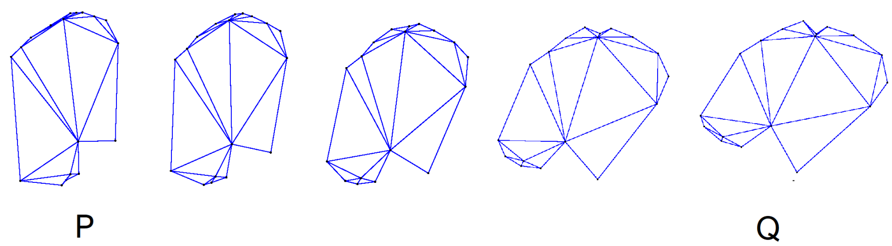

Figure 10 Interpolation created using the morphing algorithm proposed in [1, 2].

Each triangle in P is transformed into the corresponding triangle in Q. Because there is a prior correspondence between the vertices of P and Q, obtained using for example, a method based on feature points, the mapping between the features in P and Q, including concavities, is better than in the previous examples. In addition, the method tries to preserve the rigidity of the object. As a consequence, the moving region does not tend to inflate or deflate artificially during the interpolation.

Experiments were carried out to quantify the error associated with the interpolation method proposed in [1, 2], however, it was difficult to quantify the weight of the noise in the results. In some cases, errors caused by noise prevail over those of the interpolation method, and so, it was difficult to make convincing conclusions [5].

The algorithms used to create the examples based on morphing techniques [1, 2] and on the method proposed in [16, 15] were implemented by the authors of this paper, following the specifications presented in the original papers. Since we could not identify clearly the strategy used in [16, 15] to choose the point in the convex-hull of a moving object to which a concavity should converge to, we evaluated several alternatives. The results were slightly different for each choice, but, as they do not influence the discussion and the conclusions of this work, we present only the results of one of the alternatives. The examples concerning the methods proposed in [11, 12] were created using the tools in the virtual machine provided by the authors at the Secondo website[3].

[3] `http://dna.fernuni-hagen.de/secondo/`

4.2 Research issues

As we have seen in Section 2, the solutions proposed in the literature impose constraints on the movement and on the morphing of the objects to make the implementation of spatio-temporal operations easier. Removing these constraints, e.g., allowing moving segments to rotate and the shape of the region to change, simultaneously, is challenging. [12] proposes a method that allows moving segments to rotate, but handles only regions with fixed shape. The authors observe that, under these circumstances, the movement of the vertices of a moving region can be represented by trochoids and the curves traced by the segments can be represented by ravdoids. The parametric equations of these curves are then used to implement spatio-temporal operations. Figure 11 displays the curves (trochoids) defined by two vertices on the boundary of a moving object (the region in gray) whose center of rotation moves horizontally.

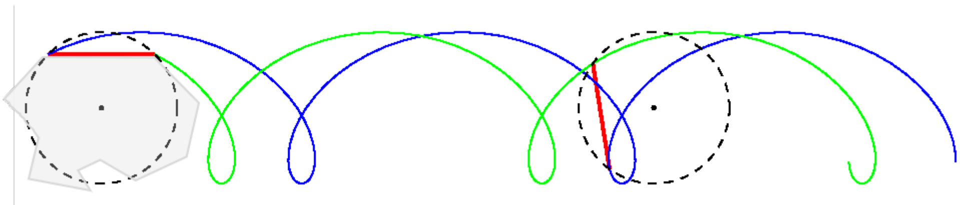

Figure 11 Path of two vertices in the boundary of a rigid moving object.

Extending this idea to handle regions with non-fixed shape is challenging because the parametric equations that represent the paths traced by the vertices and the segments of the region are no longer known. The new resulting curves are exemplified in Figure 12.

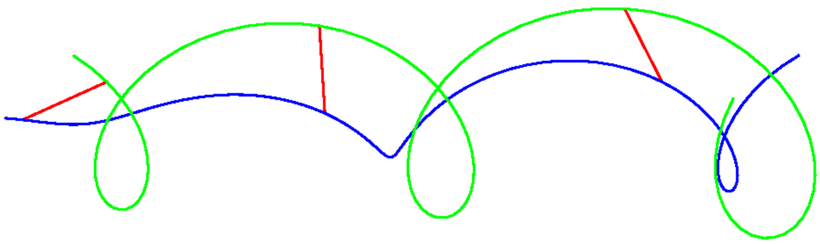

Figure 12 Path of two vertices on the boundary of a deformable moving object.

Moreover, the path traced by the vertices and the segments of the moving regions with fixed shape is independent of the interpolation method used as long as it preserves the rigidity of the object during interpolation. This is no longer true when the moving region is deformable, because the trace of the vertices and the segments generated using two interpolation methods will probably differ. Whether or not these curves are related or there exists a family of curves that can be used in this case is a subject for further investigation. There at least two paths for investigation here. Finding a general solution that is independent of the interpolation method used or finding a solution for a specific method. In both cases, after the parametric equations have been found, the implementation of spatio-temporal operations should follow similar strategies.

Ideally, we should find the parametric equations of these curves and use analytical methods to implement spatio-temporal operations. This means that it would be possible to obtain exact and computationally efficient solutions. An alternative is to use morphing techniques to obtain the best possible representation of the evolution of the spatio-temporal phenomena and then use approximation functions, e.g., splines, to store that information. In this way, it would be possible to have simpler equations and the algorithms developed to implement spatio-temporal operations would be independent of the interpolation method used. However, the effect of the approximation on the characteristics of the interpolation, e.g., robustness and computational costs, must be minimized.

Another solution is to use numerical (iterative) methods to compute spatio-temporal operations. There are already solutions proposed in the literature, but this topic is not discussed in this paper.

4.3 Discussion

The examples above show that morphing techniques can give important insights and well-established solutions to the modeling and representation of spatio-temporal data in databases. However, the use of morphing techniques in databases raises new challenges that need investigation.

First, unlike the methods presented in Section 2, where rotation is represented implicitly (i.e., it is simulated by splitting line segments into parts that move linearly), morphing techniques allow rotation to be represented explicitly. This allows to obtain closer representations of the real evolution of the phenomena, but it also increases the complexity of the algorithms, because it becomes necessary to deal with curves. This problem is partially investigated in [12], but the proposed methods only apply to moving regions with fixed shape and the algorithms presented implement only a subset of the operations that should be provided by a spatio-temporal database. For instance, finding whether two rigid moving regions intersect is an open issue. On the other hand, using triangulation-based methods that allow decomposing complex shapes into triangles can make the development of algorithms easier.

Second, the focus of research in the spatio-temporal databases literature to solve the region interpolation problem is on creating efficient and robust interpolations, to ensure that the shape of the moving regions is always valid during the interpolation. Conservative approaches are followed, but the constraints imposed to the methods may cause unnatural deformations of the objects during the interpolation. This is an important issue when dealing with real-world data. On the other hand, morphing techniques are less conservative with respect to robustness and few studies exist on how to deal with complex transformations such as merging and splitting moving regions.

It is important to note that robustness is commonly evaluated using high complex examples, but no formal proof exist that the methods are robust. The proof that a method is not robust is usually given by counter-example. So, we also need to develop methods to detect invalid interpolations to ensure that all data in the database are valid.

Third, creating spatio-temporal data, such as moving regions (e.g., icebergs) and moving lines (e.g., the front-line of a forest fire), from raw data (e.g., an ordered sequence of images, videos or time-lapse videos) is difficult and time-consuming. Previous work is evaluated using synthetic data and so, it is difficult to conclude whether an interpolation represents well a given phenomenon, because one does not know how the shape should be between the source and the target observations. The alternative is to use a ground truth. For instance, given an ordered sequence of observations it is possible to create a sample (e.g., the observations in odd positions) and compare the results of the interpolation created using the sample with the 'observations in the even positions [18]. However, the creation of spatio-temporal data goes through several stages, for example, segmentation to extract the shape of the object from a sequence of images, simplification to reduce the number of vertices, polygon matching to establish a correspondence between the source and target shapes, and finally, the interpolation of the shapes. Thus, the errors measured when running the interpolation algorithms may have been partially caused at the previous stages. Consequently, a ground truth and a benchmarking methodology are needed to measure and compare the accuracy of the interpolation methods.

5 Conclusion

This article deals with the representation of moving objects in spatio-temporal databases using continuous models in time and space, i.e., spatial data are represented in vector format, as well as their evolution over time. Despite advances in this field, there are open issues that need to be investigated so that spatio-temporal databases can be effectively used in real-world problems. A notable example is the region interpolation problem. [12] has recently proposed a new approach to interpolate moving regions of fixed shape, which is a step forward relatively to previous work. Nevertheless, the representation of general-purpose moving regions remains an open issue.

In this paper, we argue that there are topics studied in the field of morphing techniques that may help to solve important issues in spatio-temporal databases research. However, even though these techniques are widely used in visualization, their use in databases is unusual or nonexistent. Several challenges must be investigated to allow the use of morphing techniques in the context of databases:

- **(a)** Robustness – additional constraints must be set to enforce data integrity in the database (there are constraints on the representation of spatial and spatio-temporal data that, in general, are not considered in animation and visualization);
- **(b)** Operability – the use of new interpolation methods, data structures (e.g., triangle meshes) and models, requires developing new methods and computational geometry algorithms to implement spatio-temporal operations, such as, projection, distance and topological relationships;
- **(c)** Optimization – the development of new data structures and algorithms may require the development of new optimization techniques to provide fast response times when querying large datasets;
- **(d)** Context-awareness – there are static and dynamic factors that may affect the evolution of the spatio-temporal phenomena between observations (e.g., wind direction and speed, and the slope of the terrain affect the movement of the front-line of forest fires). Currently, the methods proposed in the literature to solve the region interpolation problem and the morphing techniques ignore these contextual factors.

The strategies presented in this article are being investigated in a research project financed by national funds, that started in 2018. The aim is to develop methods and tools to enable quantitative analysis of spatio-temporal data, guaranteeing levels of objectivity, precision and reproducibility compatible with the completion of scientific and engineering work.

Two case studies will be considered. The propagation of controlled forest fires and the morphological changes of living cells. The objective of the first is to compute the emissions of gases to the atmosphere using the representation of the propagation of the fire-line provided by a database. The data will be captured using drones and a meteorological station that will measure the gases emissions to the atmosphere. The objective of the second case study is to represent the continuous evolution of living cells in a database. The data will be captured using electronic microscopes and recorded as time-lapse videos. These case studies will provide real data and requirements that involve the modeling of several spatio-temporal features. This is an important input to create a ground truth or a benchmark to evaluate the methods proposed in the spatio-temporal databases research community, thus addressing an important limitation found in previous work, where only synthetic data are used.

References

1 Marc Alexa, Daniel Cohen-Or, and David Levin. As-rigid-as-possible shape interpolation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH '00*, pages 157–164, 2000.

2 William Baxter, P Barla, and K Anjyo. Rigid shape interpolation using normal equations. In *NPAR '08 Proceedings of the 6th international symposium on Non-photorealistic animation and rendering*, pages 59–64, 2008.

3 Jan Chomicki, Yuguo Liu, and Peter Z. Revesz. Animating Spatiotemporal Constraint Databases. In *Proceedings of the International Workshop on Spatio-Temporal Database Management*, STDBM '99, pages 224–241, London, UK, 1999. Springer-Verlag.

4 J Cotelo Lema, Luca Forlizzi, Ralf Hartmut Guting, Enrico Nardelli, and Markus Schneider. Algorithms for Moving Objects Databases. *The Computer Journal*, 46(6):680–712, 2003.

5 José Duarte, Paulo Dias, and José Moreira. An Evaluation of Smoothing and Remeshing Techniques to Represent the Evolution of Real-World Phenomena. In *Advances in Visual Computing - 13th International Symposium, ISVC 2018, Las Vegas*, pages 57–67, 2018. doi:10.1007/978-3-030-03801-4_6.

6 Luca Forlizzi, Ralf Haxtmut Giiting, Enrico Nardelli, and Markus Schneider. A Data Model and Data Structures for Moving Objects Databases. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 319–330, 2000.

7 Stephane Grumbach, Manolis Koubarakis, Philippe Rigaux, Michel Scholl, and Spiros Skiadopoulos. *Chapter 5: Spatio-temporal Models and Languages: An Approach Based on Constraints*, pages 177–201. Springer, September 2003.

8 Ralf Hartmut Güting, Thomas Behr, and Christian Düntgen. SECONDO : A Platform for Moving Objects Database Research and for Publishing and Integrating Research Implementations. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 33(2):56–63, 2010.

9 Ralf Hartmut Güting, Michael H Böhlen, Martin Erwig, Christian S Jensen, Nikos A Lorentzos, Markus Schneider, and Michalis Vazirgiannis. A Foundation for Representing and Querying Moving Objects. *ACM Trans. Database Systems*, 25(1):1–42, 2000.

10 S. Haesevoets and B. Kuijpers. Closure properties of classes of spatio-temporal objects under Boolean set operations. In *Proceedings Seventh International Workshop on Temporal Representation and Reasoning. TIME 2000*, pages 79–86, July 2000.

11 Florian Heinz and Ralf Hartmut Güting. Robust high-quality interpolation of regions to moving regions. *GeoInformatica*, 20(3):385–413, 2016.

12 Florian Heinz and Ralf Hartmut Güting. A data model for moving regions of fixed shape in databases. *International Journal of Geographical Information Science*, 32(9):1737–1769, 2018.

13 Ligang Liu, Guopu Wang, Bo Zhang, Baining Guo, and Heung-Yeung Shum. Perceptually based approach for planar shape morphing. In *Proceedings of the 12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004*, pages 111–120, October 2004.

14 Zhiguang Liu, Howard Leung, Liuyang Zhou, and Hubert P. H. Shum. High Quality Compatible Triangulations for 2D Shape Morphing. In *Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology*, VRST '15, pages 85–94, New York, NY, USA, 2015. ACM. doi:10.1145/2821592.2821594.

15 Mark Mckenney and Roger Frye. Generating Moving Regions from Snapshots of Complex Regions. *ACM Trans. Spatial Algorithms Systems*, 1(1):1–30, 2015.

16 Mark Mckenney and James Webb. Extracting Moving Regions from Spatial Data. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 438–441, San Jose, 2010.

17 Q Meng, Z Fu, Y Huang, and M Shen. A FAST MATCHING APPROACH OF POLYGON FEATURES. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, I-2:45–49, July 2012.

18 J. Moreira, P. Dias, and P. Amaral. Representation of continuously changing data over time and space: Modeling the shape of spatiotemporal phenomena. In *Proceedings of the 2016 IEEE 12th International Conference on e-Science, e-Science 2016*, 2017.

19 Nikos Pelekis, Yannis Theodoridis, Spyros Vosinakis, and Themis Panayiotopoulos. Hermes - A Framework for Location-Based Data Managment. In *EDBT'06 Proceedings of the 10th international conference on Advances in Database Technology*, pages 1130–1134, Munich, 2006.

20 Majken K. Rasmussen, Esben W. Pedersen, Marianne G. Petersen, and Kasper Hornbæk. Shape-changing Interfaces: A Review of the Design Space and Open Research Questions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 735–744, New York, NY, USA, 2012. ACM.

21 Vitaly Surazhsky and Craig Gotsman. High quality compatible triangulations. *Engineering with Computers*, 20:147–156, 2002.

22 Erlend Tøssebro and Ralf Güting. Creating Representations for Continuously Moving Regions from Observations. In *Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases*, pages 321–344. Springer-Verlag Berlin Heidelberg, 2001.