Compatible Time-Aware Shape Simplification

First Author Name[1][a], Second Author Name[1][b] and Third Author Name[2][c]
[1]*Institute of Problem Solving, XYZ University, My Street, MyTown, MyCountry*
[2]*Department of Computing, Main University, MySecondTown, MyCountry*
{f_author, s_author}@ips.xyz.edu, t_author@dc.mu.edu

Keywords: Polygon generalization, polygon simplification, spatio-temporal data.

Abstract: Current research has shown good interest in operations in spatiotemporal data (Duarte et al., 2018). Part of this research is focused on representing and executing operations on moving deformable regions (McKenney and Webb, 2010) (Duarte et al., 2018) (Tøssebro and Güting, 2001). Acquiring spatio-temporal data is still an open topic. One of the steps consists of simplifying the boundary of the region of interest after image segmentation. The boundary simplification is usually performed individually in each contour detected. In this paper, we propose a novel methodology for simplification that accounts for temporal variation in order to attempt to improve following operations and interpolation.

1 Introduction

Over the last few decades, the amount of geographical data that has been generated is increasing at great speed. Developments in new sensor techniques and processing algorithms have supported an explosion on the availability of geographical data, for very different purposes. These purposes can be as diverse as iceberg tracking (Silva and Bigg, 2005), wildfire tracking (Sifakis et al., 2011), land cover/use (Johnsson, 1994), coastal lines (Sesli et al., 2009) and many more.

There has been great interest in managing spatial data, leading to developments of spatial databases (Piórkowski et al., 2011), now broadly available. However, the subject of data acquisition is still an open research topic, with the current developments based mostly on object based image analysis (Chen et al., 2018).

During the acquisition process, the steps are usually comprised of image segmentation, polygon extraction, feature extraction and classification(Chen et al., 2018). The image segmentation algorithms usually provide the pixels corresponding to the contour, and to extract the polygons some sort of simplification is used. The most common simplifications used are Douglas-Peucker (Wu and Marquez, 2003) (Douglas and Peucker, 1973) and Visvalingam-Whyatt (Visvalingam and Whyatt, 1993).

With the growing body of historical data, there is an increase in the demand for studies involving spatio-temporal data, such as moving and deforming linestrings and moving and deforming regions, polygons and shapes. These shapes represent the temporal evolution, including movement and deformation, of phenomena as distinct as tracking icebergs, wildfires and shorelines. However, most of the studies on spatio-temporal data focus on the interpolation process (Tøssebro and Güting, 2001) (McKenney and Webb, 2010) or on the database operations (Tøssebro and Güting, 2001) (Duarte et al., 2018). These techniques are based on being provided an already-simplified polygon for interpolation.

However, the simplification process might introduce discrepancies or issues in the interpolated regions. The simplification process and its impacts in proper shape morphing has not been widely studied. The image segmentation from real-world geographical data provides the identification of contour pixels on an image. These pixels are then simplified to a polygon, according to some technique or some metric. The most used simplifications are Douglar-Peucker (and variations) (Wu and Marquez, 2003) and Visvalingam-Whyatt (Visvalingam and Whyatt, 1993). Both families of simplification techniques are designed for single shapes, without regard to temporal evolution of the shapes.

In this paper, we present a novel technique for

[a] https://orcid.org/0000-0000-0000-0000
[b] https://orcid.org/0000-0000-0000-0000
[c] https://orcid.org/0000-0000-0000-0000

compatible simplification between two shapes of the same object, with the intent focus on preserving points that will be important to the time-matched shape and would be discarded otherwise. The simplification technique should provide polygons from extracted dense contours obtained from image segmentation. We also perform a visual evaluation of the importance of simplification for shape morphing.

This paper is organized as follows. Section 2 presents an overview on related work on simplification techniques and their applications. Section 3 details the proposed method. Section 4 discuss some aspects of the proposal, using some examples to compare to standard techniques. Section **??** present some results for compatible shape simplification. Finally, Section 5 presents the conclusion and guidelines for future work.

2 Related work

The process of generalization of maps consists of the selection and simplified representation of detail appropriate to the scale and/or purpose of the map. The generalization of lines can be considered as started by the Ramer-Douglas-Peucker (RDP) algorithm, published in 1972 and 1973 (Ramer, 1972) (Douglas and Peucker, 1973).

In the general RDP algorithm, a line is defined between the first and last points of a line segment to be simplified (Douglas and Peucker, 1973). Then, the point farthest from this line is included in the simplified polyline - as long as the point distance is below an accepted threshold level (ξ). This process is now applied recursively on all sub lines on the polyline until there are no points over ξ distance from a line. In order to apply this process to a polygon, we need to pick two vertices as reference points for the polyline.

In the Visvalingam-Whyatt (VW) algorithm (Visvalingam and Whyatt, 1993), for every point on the line or polygon we build a triangle between this point and the previous and next points. Then, the area of the triangle is calculated, and all central points of triangles with an area below a threshold ξ are removed, and the area of the 2 adjacent vertices is recalculated (Shi and Cheung, 2006).

These techniques are characteristic in that they are suited for simplification of a single polygon or polyline. Since they are heavily used on applications like cartography, their focus is on simplification a single polygon with minimal information loss. Modern technologies enabled us to obtain high-resolution data using satellites or aerial photos. In these scenarios, one can use image segmentation in order to extract the boundaries of objects or regions of interest. Image segmentation provides a definition of all pixels on the boundaries, however it can create additional problems due to the size of the polygons involved, which can be over hundreds or thousands of vertices, and because of the natural noisy borders on image acquisition and processing. The amount of vertices can be detrimental to the performance of the vertex correspondence problem, needed to identify the corresponding vertices between two shapes. An excess amount of vertices are also detrimental to the interpolation algorithms performance. Finally, noisy borders can create artifacts during shape interpolation for real world phenomena that are unintended.

Accounting that the morphing or interpolation happens between two different shapes, one can use this additional information to create different simplifications. Baxter et al. (Baxter III et al., 2009) proposed a method for compatible embedding of two shapes, and later generalized the approach for multiple shapes (Baxter et al., 2009). The proposed method reduces the number of points on a polygon, however due to the necessity of keeping the textures inside the polygon for the morphing process, the reduction of points aims at embedding and not at generalization of the polygon. While suited for morphing animations, embedding can create anomalies when used on real world application data like iceberg tracking, where one would rather be close to the original polygons than ensure embedding on a bigger polygon that loses original shape information (like boundary characteristics or polygon area).

Since the existing line generalization algorithms are focused more on the simplification of non-temporal shapes (mostly polylines and polygons), two implications arise that might affect future interpolation methods. The first issue is related to simplification of smooth sides, where the algorithm will pick different points on the curve to represent it at each time frame, without consistency. This can be seen on Figure 4, where we have two pictures of iceberg B-15a. Those figures were simplified using Douglas-Peucker. As it can be seen on the highlighted detail, the smooth side of the iceberg is represented by different points on different time frames. This can have a side effect of internal deformation, due to a big displacement if those points are matched. Using some figures from the dataset provided by Brown (http://vision.lems.brown.edu/content/available-software-and-databases), we can also see on Figure 3 in the darker highlights this issue.

The second issue that arises is the mapping of similar areas with different resolutions. Due to the property of the algorithms, it is possible that equivalent parts of the curve are mapped in each simplification with a different number of points, so that a matching has to occur again on points that are geodesically distant.

3 Proposed method

Since the quality of the vertex matching is of utmost importance for the interpolation results (Duarte et al., 2018), we propose a method of compatible time-aware shape simplification.

Our method relies on implicit information gathered from knowing that we are simplifying a pair of shapes, instead of a single shape. We aim to balance removing points not representative of the individual shape, while keeping points that represent distinct features on a shape and points on the other shape that should represent that distinct feature on the matched shape. Our method can also operate before the vertex correspondence problem, avoiding the performing of expensive optimization processes on matching a huge set of vertexes.

Avoiding to remove points that will match future features on morphing should expect a more natural interpolation result for a pair of shapes. A simplification process with similar and closer points should naturally lead to less events of high points displacement or high deforming of internal triangle areas during interpolation.

On the usual approach (Duarte et al., 2018), the workflow that prepares spatio-temporal data for interpolation can be described as 3 main steps:

- Segmentation: extract the shape or region-of-interest from each single raster image separately.
- Simplification: simplify each individual curve generated.
- Matching: match two different shapes vertices, adding vertices on contours if needed.

In this workflow, currently only the third step takes into account more than one shape. We aim to improve this process by extending time-awareness into the simplification stage. It is important to note that the points which might be added on the matching step could have been points that were removed in the simplification step, since they were pixels on the original raster image.

Our method has two main objectives on simplification, and one minor objective. The first major objective that we have is to simplify a polygon in a way that will not require point addition on the matching step. Any point on the lines of the contour were originally points of the real object, and thus removing the point and later picking it on a line ensures local information loss. The proposed method strives to keep points that will be representative on future or past shapes by design.

The second main objective is to allow simpler matching. Providing matching algorithms with locally-aware points in order to reduce vertex matching complexity. Since matching algorithms can work with location data in order to help match vertices (Van Kaick et al., 2011), providing the algorithms with better locality should help the matching process.

In order to keep compatible points and to allow better matching, similar regions should be represented with similar resolution or density of points. This minor objective follows from the two main objectives.

For the purposes of this section, we consider the polygon to have a point removed as P and the matched polygon as Q, with $p \in P$ and $q \in Q$ as points on the polygons. We start by defining a cost function for each vertex on the shape. This cost represents the loss of information on the pair of shapes. This can be seen on Equation 1. It introduces a new parameter, time_factor, representing the preference of the user between keeping temporal information or single shape features. This parameter can be varied according to bigger or smaller time intervals between the shapes, for example.

$$cost_p = max(cost\_single_p, cost\_matched_p * time\_factor) \quad (1)$$

In order to consider both the loss of information on the current shape and on the morphing, we assume the cost to be greatest cost between considering the cost for a single shape and the cost for loss of feature representation on the matched shape.

We then define the cost for a single shape as the area of the triangle between p, $p-1$ and $p+1$, similar to VW algorithm. This is represented on Equation 2. For this function, one could use any measurement, like the distance between p and the line connecting $p+1$ and $p-1$, which would lead to a simplification closer to RDP algorithm.

$$cost\_single_p = area\_triangle(p, p+1, p-1) \quad (2)$$

The cost for the matched shape was chosen to represent two main scenarios. The first scenario is where there exists a feature in P that does not exist in Q. We understand this point as a significant point, and so we define the Considering the current polygon as P and the matched polygon as Q, we then define the cost for unique feature as seen on Equation 3. This cost represents the minimum distance between p and any point in Q, thus representing that p is a significant feature of P.

$$cost\_unique\_feature_p = min(d_{pq}) \forall q \in Q) \quad (3)$$

We also define a cost for a matched feature. This measure is the complement of the cost defined on Equation 3. The cost for a matched feature can be seen on Equation 4. The cost for a matched feature is the furthest distance between p and any point $q \in Q$, provided that p is still the closest point in P to q. This suggests that we might have p morphing into q on future steps.

$$\begin{aligned} cost\_matched\_feature_i = max(d_{pq}) \forall q \in Q| \\ d_{pq} = min(d_{kq}) \forall k \in P) \end{aligned} \quad (4)$$

Finally, we define loss of information of point p in the matching to Q as the greater between the two costs, as seen on Equation 5.

$$\begin{aligned} cost\_time_i = max(cost\_unique\_feature_p, \\ cost\_matched\_feature_i) \end{aligned} \quad (5)$$

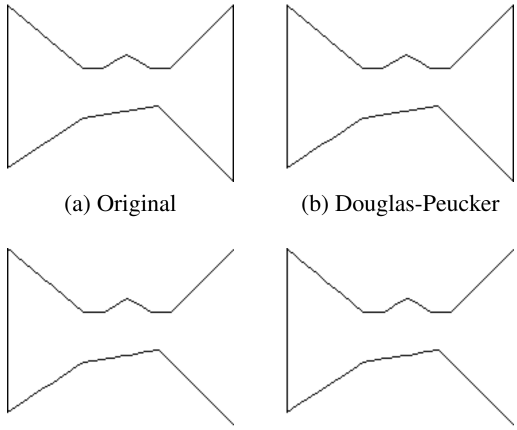
Given this definition of cost, our *SIMPLIFY* function starts by removing the lowest cost point in P, then removing the lowest cost point in Q, iterating as many times as necessary to achieve the desired amounts or vertices on the simplified polygon.

```
function SIMPLIFY(P, Q, size)
    while ||P|| > size ∨ ||Q|| > size do
        if ||P|| > size then
            r ← p ∈ P|cost(p) = min(cost(k)∀k ∈ P)
            P ← P − r
        end if
        if ||Q|| > size then
            r ← q ∈ Q|cost(q) = min(cost(k)∀k ∈ Q)
            Q ← Q − r
        end if
    end while
end function
```

4 Discussion

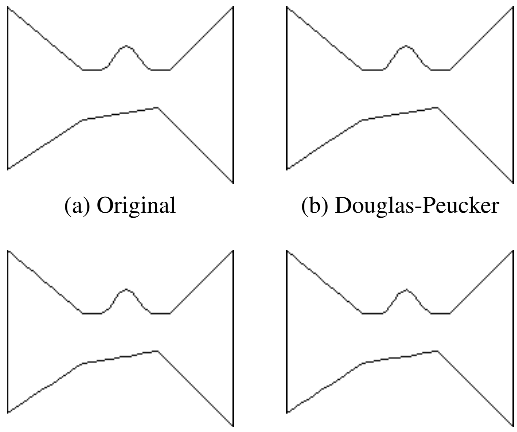
In order to test our method compared to the more traditional methods, we evaluated it on Brown University Binary Image dataset, available at http://vision.lems.brown.edu/content/available-software-and-databases.

Performing the compatible simplification between arb01 (Figure 1) and arb02 (Figure 2) we can see that all methods cannot be visually distinguished, leading to very similar results of contour features.



(a) Original (b) Douglas-Peucker (c) Visvalingam-Whyatt (d) Proposed method

Figure 1: Comparison of arb01 contour for the 4 algorithms, with 95% points reduction



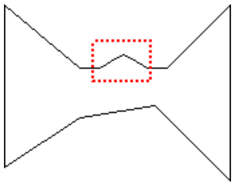
(a) Original (b) Douglas-Peucker (c) Visvalingam-Whyatt (d) Proposed method

Figure 2: Comparison of arb02 contour for the 4 algorithms, with 95% points reduction

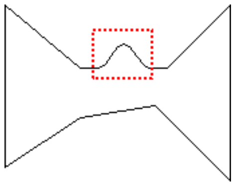
However, once we examine the points kept on the contours, we can verify that each algorithm led to a different set of points. We can verify that for arb01, the feature highlighted in Figure 3 can be represented by very few points, since it is a triangular shape. However, in the arb02 (Figure 3b), in order to represent the same feature with a round curve more points are needed.

It would be important to keep some points on arb01 so that a point-to-point matching would be easier. This can be seen on Figure 3, where the generalization algorithms can be compared side-by-side and the respective point density can be seen.

This result can also be seen on real world datasets.



(a) arb01

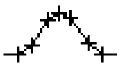


(b) arb02

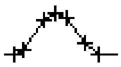
(c) VW on arb01

(d) DP on arb01

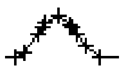
(e) Our method on arb01



(f) VW on arb02



(g) DP on arb02



(h) Our method on arb02

Figure 3: Highlight on the feature-area representation to all simplification algorithms

(a) ice01

(b) ice02

Figure 4: Original iceberg photos

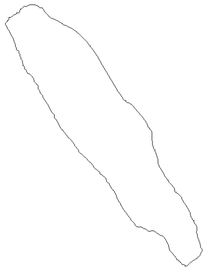
On this paper, we used images of fragments from Iceberg B-15, obtained at https://visibleearth.nasa.gov/. For the purposes of this paper, images we used two images of the same iceberg taken at different times, as they can be seen on Figure 4.

Similarly, our proposed method can be shown to work better on the highlights, as can be seen on Figure 7.

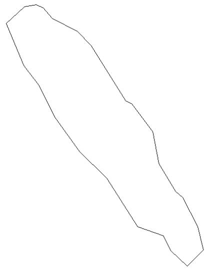
This dataset is interesting because icebergs are inherently deformable moving regions, due to natural movements of translation, rotation and deformation. In our method we consider only simple regions, i.e., regions with only one face and no holes.

5 Conclusion

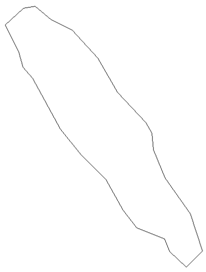
We presented an algorithm for compatible time-aware simplification of 2D shapes. The simplification is an important part of applying morphing methods for real-world interpolation. Obtained data has noisy borders, and is converted to a vector shape.



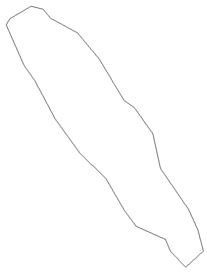
(a) Original



(b) Douglas-Peucker

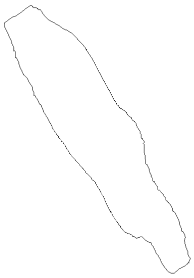


(c) Visvalingam-Whyatt

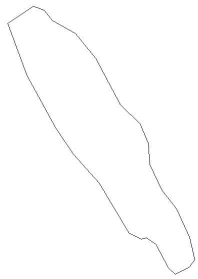


(d) Proposed method

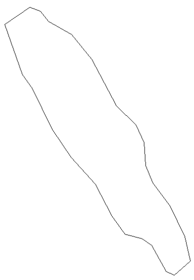
Figure 5: Comparison of ice01 contour for the 4 algorithms, with 95% points reduction



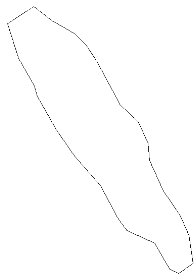
(a) Original



(b) Douglas-Peucker

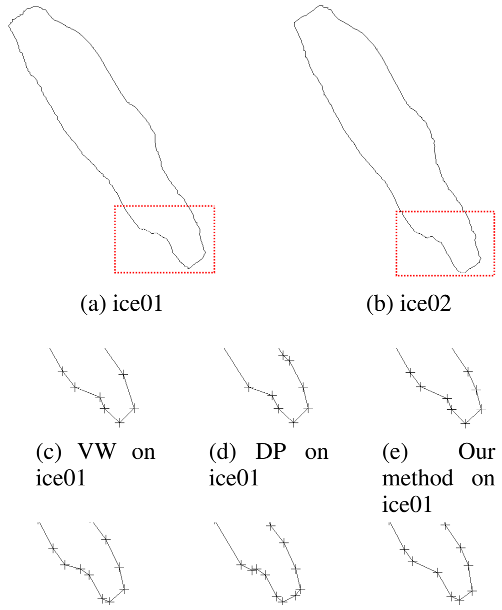


(c) Visvalingam-Whyatt



(d) Proposed method

Figure 6: Comparison of ice02 contour for the 4 algorithms, with 95% points reduction



(a) ice01 (b) ice02

(c) VW on ice01 (d) DP on ice01 (e) Our method on ice01

(f) VW on ice02 (g) DP on ice02 (h) Our method on ice02

Figure 7: Highlight on the feature-area representation to all simplification algorithms

The proposed method keeps similarity between snapshots of shapes in different times. Figures 3 and 7 show how it retains better geometry on borders. Figure 3 shows the effectiveness of our method on features requiring different resolutions. The major advantage of our method is that it simplifies the shapes taking into account needed features on future or previous snapshots.

This problem is part of our full vision of a process that encompasses spatio-temporal data acquisition, preparation and use of morphing techniques to represent real world phenomena. In this paper, we focused on the geometric aspects of the simplification problem. The proposed method should allow for easier vertex matching and for better morphing results.

ACKNOWLEDGEMENTS

We would like to thank the project

REFERENCES

Baxter, W., Barla, P., and Anjyo, K. (2009). N-way morphing for 2d animation. *Computer Animation and Virtual Worlds*, 20(2-3):79–87.

Baxter III, W. V., Barla, P., and Anjyo, K.-i. (2009). Compatible embedding for 2d shape animation. *IEEE Transactions on Visualization and Computer Graphics*, 15(5):867–879.

Chen, G., Weng, Q., Hay, G. J., and He, Y. (2018). Geographic object-based image analysis (geobia): Emerging trends and future opportunities. *GIScience & remote sensing*, 55(2):159–182.

Douglas, D. H. and Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization*, 10(2):112–122.

Duarte, J., Dias, P., and Moreira, J. (2018). An evaluation of smoothing and remeshing techniques to represent the evolution of real-world phenomena. In *International Symposium on Visual Computing*, pages 57–67. Springer.

Johnsson, K. (1994). Segment-based land-use classification from spot satellite data. *Photogrammetric Engineering and Remote Sensing*, 60(1):47–53.

McKenney, M. and Webb, J. (2010). Extracting moving regions from spatial data. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, pages 438–441, New York, NY, USA. ACM.

Piórkowski, A. et al. (2011). Mysql spatial and postgis–implementations of spatial data standards. *EJPAU*, 14(1):03.

Ramer, U. (1972). An iterative procedure for the polygonal approximation of plane curves. *Computer graphics and image processing*, 1(3):244–256.

Sesli, F. A., Karsli, F., Colkesen, I., and Akyol, N. (2009). Monitoring the changing position of coastlines using aerial and satellite image data: an example from the eastern coast of trabzon, turkey. *Environmental Monitoring and Assessment*, 153(1-4):391–403.

Shi, W. and Cheung, C. (2006). Performance evaluation of line simplification algorithms for vector generalization. *The Cartographic Journal*, 43(1):27–44.

Sifakis, N. I., Iossifidis, C., Kontoes, C., and Keramitsoglou, I. (2011). Wildfire detection and tracking over greece using msg-seviri satellite data. *Remote sensing*, 3(3):524–538.

Silva, T. A. and Bigg, G. R. (2005). Computer-based identification and tracking of antarctic icebergs in sar images. *Remote Sensing of Environment*, 94(3):287–297.

Tøssebro, E. and Güting, R. H. (2001). Creating representations for continuously moving regions from observations. In Jensen, C. S., Schneider, M., Seeger, B., and Tsotras, V. J., editors, *Advances in Spatial and Temporal Databases*, pages 321–344, Berlin, Heidelberg. Springer Berlin Heidelberg.

Van Kaick, O., Zhang, H., Hamarneh, G., and Cohen-Or, D. (2011). A survey on shape correspondence. In

Computer Graphics Forum, volume 30, pages 1681–1707. Wiley Online Library.

Visvalingam, M. and Whyatt, J. D. (1993). Line generalisation by repeated elimination of points. *The cartographic journal*, 30(1):46–51.

Wu, S.-T. and Marquez, M. R. G. (2003). A non-self-intersection douglas-peucker algorithm. In *16th Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2003)*, pages 60–66. IEEE.