KN - Journal of Cartography and Geographic Information (2022) 72:201–212
https://doi.org/10.1007/s42489-022-00106-6

Visualizing the Off-Screen Evolution of Trajectories

Axel Forsch[1] · **Friederike Amann**[1] · **Jan-Henrik Haunert**[1]

Received: 20 January 2022 / Accepted: 11 April 2022 / Published online: 18 May 2022

Abstract
In the context of volunteered geographic information, large sets of trajectories of humans and animals are collected. Analyzing these trajectories visually is often complicated due to limited display sizes. For instance, when a user chooses a large map scale to inspect the details of a trajectory, only a small part of the trajectory is visible in the map. Therefore, in this article, we present an approach for visualizing the off-screen evolution of trajectories, i.e., their continuation outside of the displayed map. We propose visual cues in the form of glyphs that are displayed at the map's boundary and that consist of one or multiple disk sectors of varying size and opening angle. These glyphs indicate the direction and variability of direction of a trajectory's continuation outside the map frame. We present an algorithm for computing the glyphs efficiently and evaluate them in a user study. The results show that the glyphs are intuitive to understand even without explanation. We further present suggestions for improving the glyph design based on the results.

Keywords Trajectories · Glyphs · Visual cues · Volunteered geographic information

Visualisierung der Fortsetzung von Trajektorien außerhalb des Kartenausschnitts

Zusammenfassung
Im Kontext freiwillig erhobener geografischer Informationen werden die Bewegungen von Menschen und Tieren in Form großer Mengen von Trajektorien erfasst. Die visuelle Analyse dieser Trajektorien ist aufgrund von begrenzten Bildschirmgrößen oft schwierig. Wählt ein Benutzer beispielsweise einen großen Kartenmaßstab, um die Details einer Trajektorie zu untersuchen, ist nur ein kleiner Teil der Trajektorie auf der Karte sichtbar. Daher stellen wir in diesem Artikel einen Ansatz zur Visualisierung der Fortsetzung von Trajektorien außerhalb des angezeigten Kartenausschnitts vor. Dazu nutzen wir Signaturen, die am Rand der Karte angezeigt werden und aus einem oder mehreren Kreissektoren unterschiedlicher Größe und Öffnungswinkel bestehen. Diese Signaturen zeigen die Richtung und Richtungsvariabilität der Fortsetzung einer Trajektorie außerhalb des Kartenausschnitts an. Wir stellen einen Algorithmus zur effizienten Berechnung der Signaturen vor und evaluieren ihn in einer Anwenderstudie. Die Ergebnisse zeigen, dass die Signaturen auch ohne Erklärung intuitiv zu verstehen sind. Des Weiteren präsentieren wir basierend auf diesen Ergebnissen Vorschläge zur Verbesserung der Signaturen.

1 Introduction

Volunteered geographic information (VGI) provides a rich source for spatial analysis (Goodchild 2007). In the field of movement pattern analysis, for example, researchers profit from large sets of crowd-sourced animal and human trajectories (Huang et al. 2013). For instance, we imagine a city planner working with a map that displays a set of trajectories of cyclists as well as information on the geographical context. A challenge arises if the city planner zooms in to a large map scale, e.g., to inspect a local area for planning reasons. At this large map scale, only small parts of the trajectories can be shown and the overall context gets lost. Therefore, our aim is to augment the map with visual cues about how the trajectories evolve outside of the map frame. With this we intend to provide the city planner with wider-area information about the movement of the cyclists.

✉ Axel Forsch
forsch@igg.uni-bonn.de

✉ Jan-Henrik Haunert
haunert@igg.uni-bonn.de

[1] Institute of Geodesy and Geoinformation, University of Bonn, Meckenheimer Allee 172, 53115 Bonn, Germany

Apart from movement pattern analysis, the visualization of trajectories also matters in the context of navigation systems and location-based services (Narzt et al. 2004). For example, a hiker may wish to follow a route computed by an algorithm or a GPS track that has been recorded and voluntarily shared by a different user. In this application, the trajectory needs to be visualized on a mobile device, such as a smartphone or a hand-held navigation system. The mobile device shows a map providing a local view of the track to allow the hiker to decide on where to turn. In addition, the hiker is interested in knowing how the trajectory will evolve in the next few minutes. However, due to the limited size of the device, this part of the trajectory cannot be displayed on the currently visible part of the map. Thus, essentially, the hiker is facing the same problem as the city planner. Using visual cues to indicate the off-screen evolution of trajectories could ease this problem.

In this article we follow this idea using glyphs indicating the off-screen evolution of trajectories (i.e., their continuation outside the displayed map). In particular, we aim at visualizing an initial section of the trajectories' off-screen evolution to provide a more global comprehension of the trajectories' direction in addition to the mere local view. Figure 1 shows an example with three trajectories (red). The glyphs (blue) are computed for each trajectory leaving the map frame and are displayed as a map overlay within a margin along the currently displayed map extent (gray). To be able to distinguish different directions and shapes of the off-screen parts of the trajectories, the glyphs are varied in length, opening angle and orientation.

In detail, each glyph is computed by searching the *disk sector* with the smallest perimeter that covers the off-screen part of the trajectory. For most applications, not the evolution of the trajectory as a whole is of interest, but only of the neighboring off-screen section. The dimension of this considered section, the *look-ahead value*, is part of the parametrization of the glyphs. To account for possible outliers and to filter out local protrusions of the trajectory recordings, the disk sector must only cover a certain share of the off-screen section. To fit onto the overlay area, the initially calculated disk sector is scaled down with a factor that is constant for all glyphs (Fig. 2). Next to this basic glyph, we developed two additional glyphs that use multiple disk sectors. These advanced versions of the glyph allow for a more detailed preview of the trajectory at the cost of a higher complexity. To analyze whether this increase in complexity is warranted, we conducted a user study.

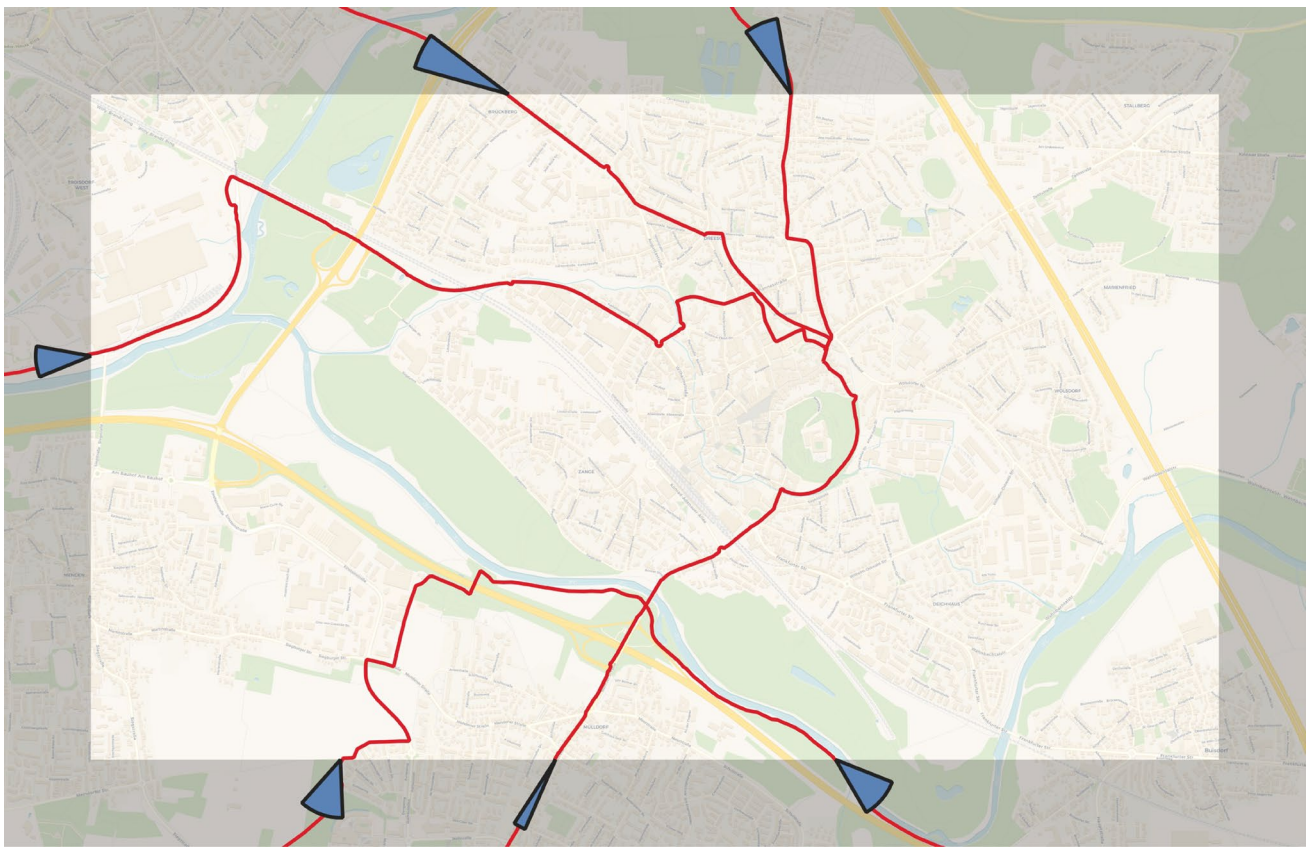


Fig. 1 Glyphs (blue) indicate the direction in which the displayed trajectories (red) will evolve outside of the map frame. They are displayed within an overlay area along the displayed map extent (gray) (color figure online)

Fig. 2 Basic glyph with displayed map extent (right of dashed line) and off-screen part of the map (left of dashed line). The glyph (solid blue) is a scaled-down version of the minimal disk sector covering a certain share (here: 95%) of the neighboring off-screen trajectory points (light blue) such that it fits on the overlay area (gray) (color figure online)

The article is organized as follows. In Sect. 2 we introduce related work. In Sect. 3 we present our methodology for generating the glyphs. Section 4 deals with the user study. We conclude the article in Sect. 6. The source code for generating the glyphs can be accessed at https://gitlab.vgiscience.de/forsch/offscreen-evolution

2 Related Work

The visualization of trajectory data has two main applications: user navigation and the visual analysis of large sets of trajectory data in expert user systems. In the context of user navigation, the visualization on small-screen devices has received considerable attention during the last decades. A lot of research has been done on visualizing point features that are outside of the screen, while visualizing off-screen line features has not been considered yet. At the same time, visual exploration techniques for trajectory data have been developed. In the following we review previous work on these topics as our work aims at supporting both of these tasks.

2.1 Visualization on Small-Screen Devices

With the advent of small mobile devices new challenges in cartography have emerged. To deal with a limited storage capability and a low resolution, Sester and Brenner (2005) presented a progressive generalization approach to transmit both: detail and overview information. Yoo and Cheon (2006) introduced the fisheye view as a visualization method that effectively layouts information on the screen. Carmo et al. (2007) integrate filtering mechanisms which are based on semantic criteria, relevance of the object and number of displayed objects to visualize icons superimposed on a map. Gedicke et al. (2021) worked on label placement on small screens. Their method is designed to reduce the necessity to zoom the map while exploring the data to ensure a high level of detail while keeping the global context.

2.2 Visual Exploration of Trajectory Data

In the context of large sets of trajectories Andrienko and Andrienko (2010) propose aggregation and summarization methods. In view of the increasing amount of available urban data Ferreira et al. (2013) target the visual exploration of the data. Their model enables users to visually query taxi trips and supports origin-destination and spatio-temporal queries.. More recently, Custers et al. (2021) presented a different approach for identifying mobility patterns based on simultaneous aggregation and simplification of both the trajectories and the underlying network. Visual exploration has also been studied in the context of large groups of moving entities. Scheepens et al. (2014) developed non-overlapping

glyphs representing aggregated groups of moving objects. Cakmak et al. (2020) present *MotionGlyphs* as a visualization technique for dense spatio-temporal networks. They use these glyphs to analyze animal behavior. Similarly, Demšar et al. (2015) perform movement analysis based on large animal movement data. They use a trajectory's segmentation into pieces fulfilling geometric criteria and the identification of a representative path for a certain set of tracks. The authors, moreover, apply several multi-dimensional visualization methods such as the space–time cube. The space–time cube, initially introduced by Hägerstrand (1970), is one of the most frequently used visualizations for spatio-temporal data. The data is visualized in 3-D using two dimensions for the spatial information, while the last dimension represents time (Kraak , 2003; Filho et al. 2020; Ssin et al. 2019). Visual exploration of trajectories in the context of transportation is investigated by Markovic et al. (2020). The authors create heat maps of important waypoints and published several interactive online animations.

2.3 Visualizing Off-Screen Landmarks

Several studies focus on the visualization of off-screen objects by glyphs displayed at the boundary of the currently displayed map extent. Baudisch and Rosenholtz (2003) present *Halo*, an approach that uses an arc to let the user infer the direction and distance to the off-screen object: A circle surrounds the object and reaches into the border region on the displayed map. The user is able to understand the distance information without needing a legend based on *amodal perception*, which allows the user to automatically complete shapes that are only partially visible. At the same time, Zellweger et al. (2003) presented *City Lights*, which is a fisheye technique, where size and distance information are conveyed using different colors.

Gustafson et al. (2008) concentrated on a *Wedge* approach using acute isosceles triangles to achieve a clutter-free visualization. The tip of the Wedge coincidences with the off-screen object, while the base and the legs are displayed on screen. Similar to the Halo approach, the user then mentally triangulates the tip beyond the screen. Overlap and clutter between different Wedges is avoided by changing rotation or intrusion on the display window. An evaluation showed that users were significantly more accurate when using Wedges than Halos.

Li and Zhao (2017) measured the perception of distance on resized icons representing off-screen landmarks. The further away a landmark is, the smaller the corresponding icon is displayed. The authors conducted a study that revealed possible misinterpretations of the off-screen icon's location and its size which is meant to represent the actual distance.

Burigat et al. (2006) performed a comparative evaluation of the *Halo* and *City Lights* approaches and approaches using arrows. The arrow approaches use small arrow symbols indicating the direction and distance to a certain object using varying arrow orientation and size. The larger the arrow is the closer is the off-screen object. One central finding is that the arrow approaches are more effective than the Halo approach when the object configuration would cause cluttering on the small screen.

A further study of Burigat and Chittaro (2011) compared the aforementioned approaches to *Overview+Detail*. The latter is not based on contextual cues but shows the map with an additional detailed view of an area of interest. Especially on small mobile devices this approach has limitations due to readability reasons.

2.4 Information Visualization

The visual exploration of data has also been studied in the context of statistics. Tukey (1977) introduced the term exploratory data analysis for techniques to discover patterns in data sets. Most related to our work are box-and-whisker plots which graphically indicate the locality and spread of the data points, by showing a box containing all points between a lower and an upper percentile. Analogously, we use contextual cues shaped as disk sectors which contain a given percentile of the off-screen trajectory points.

The studies discussed above present methods to handle huge sets of trajectories and to display information on small-screen devices. They also introduce cues to convey information on off-screen landmarks. Nonetheless, visualizing off-screen trajectories has not been considered in previous work. In this article we fill this gap by combining these approaches.

3 Methodology

In this section, we present the design of the basic glyph and outline our algorithm to compute it. We then extend this definition to generate more advanced versions of the glyph.

We start by introducing the terminology and by formalizing the problem. We are given a map frame of fixed extent and one or multiple trajectories passing through it. Following previous work on off-screen visualization we define an *overlay area* of fixed width b along the map display's boundary. This area is used to display the glyphs. The rest of the map frame is the *main area*. Everything outside of the main area we call *off-screen* (even though the overlay area is still technically on the current map extent). In case that a trajectory enters the map frame more than once we consider each on-screen part as a separate trajectory. Each trajectory that passes the main area has zero, one or two off-screen parts, as both of its ends can potentially be off-screen. Each off-screen part is of arbitrary length and not all of it might be of

interest for the application. Therefore, only an initial section of each off-screen part of a given length defined by a *look-ahead value* ℓ is considered, the *off-screen section*. The start point of each off-screen section is the crossing point of the trajectory with the overlay area's inner boundary.

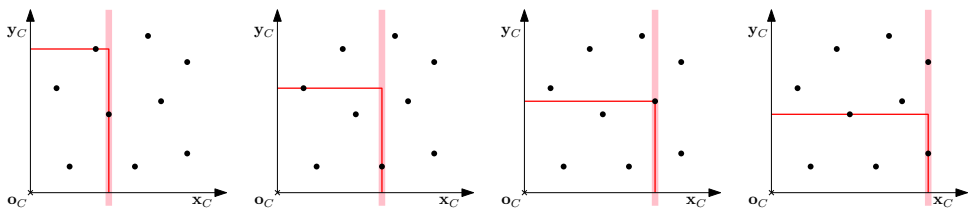
The problem at hand is to find a suitable visualization for the off-screen section. The key properties of the off-screen section we want to communicate are its direction and its variability of direction. Additionally, we try to keep the visualization as simple as possible. We solve this task using glyphs in the form of disk sectors that we display on the overlay area of the map. Disk sectors are suitable for this task as their orientation encodes the direction, the opening angle encodes the variability of direction, and the radius encodes the velocity. To formalize this, the idea is to find a disk sector attached to the off-screen section's start point that covers the points of the off-screen section and that is minimal with respect to a prescribed objective function f. To account for possible outliers and to filter out local protrusions in the trajectory we soften this definition such that only a share κ of the points must be covered. To fit on the overlay area, the glyphs are scaled down with a constant factor.

The approach for computing a glyph can be broken down into two parts. The first part is finding the minimal disk sector given the orientation of its right boundary (i.e., the line segment bounding the disk sector to the right). For this we developed an algorithm that computes a minimal enclosing rectangle; by transforming the input points to polar coordinates we can apply this algorithm to find the minimal disk sector. The second part is finding the orientation of the right edge that minimizes the disk sector.

3.1 Finding a Minimal k-Rectangle

Given a coordinate frame C and a point set $P = \{p_1, p_2, \ldots, p_n\}$ of n input points with $x_i \geq 0$ and $y_i \geq 0$ for $i = 1 \ldots n$, we search for a rectangle with one corner at the point of origin $\mathbf{o}_C$ of C and edges parallel to the coordinate axes of C, such that it covers at least $k = \lceil \kappa \cdot n \rceil$ of the input points. We call such a rectangle a *k-rectangle*. With MinimalKRectangle we refer to the problem that asks for a k-rectangle minimizing f. Our assumptions with respect to f are that it is monotonically increasing in both the width and the height of a k-rectangle and that it can be evaluated in constant time for a given k-rectangle. This assumption holds, for example, if we define f as the perimeter or the area of a k-rectangle.

A k-rectangle is *non-dominated* if there is no other k-rectangle that is smaller in one of the two dimensions, width or height, and at most as large in the other dimension. We solve MinimalKRectangle by enumerating the set of all non-dominated k-rectangles and taking one that is optimal with respect to f. Using a sweep-line approach we enumerate the non-dominated k-rectangles in increasing order of their widths. This yields an algorithm with running time in $\mathcal{O}(n \log n)$. Due to the monotonicity of f, any optimal solution has to be non-dominated. Hence, the algorithm is correct. A visualization of this approach is given in Fig. 3. Pseudo-code for the algorithm is given in Algorithm 1.



(a) $f = 66$ **(b)** $f = 64$ **(c)** $f = 70$ **(d)** $f = 72$

Fig. 3 Visualization of the algorithm for $k = 4$. All four non-dominated k-rectangles are displayed in **a**–**d** with the sweep line shown in light red; f corresponds to the area. The optimum with respect to f is found in step (**b**) (color figure online)

Algorithm 1 MinimalKRectangle

Input: point set P, number of points k, objective function f
Output: width and height of a k-rectangle minimizing $f(width, height)$

$X[1 \dots n], Y[1 \dots n] = P$ sorted by x-/y-coordinate respectively
$i_X = k$ ▷ index in X
$i_Y =$ index of point with maximum y of first k points in X ▷ index in Y
$\bar{x} = X[i_X].x$, $\bar{y} = Y[i_Y].y$
$f_{\text{best}} = f(\bar{x}, \bar{y})$ ▷ initial solution defined by the k left-most points
for increment i_X until $\|P\|$ **do**
  if $X[i_X].y \geq \bar{y}$ **then**
    continue ▷ skip point as its y-value is higher than current y
  end if
  $\bar{x} = X[i_X].x$
  for decrement i_Y until k **do**
    if $Y[i_Y].x > \bar{x}$ **then**
      continue ▷ skip point as its x-value is higher than current x
    end if
    $\bar{y} = Y[i_Y].y$
    $f_{\text{curr}} = f(\bar{x}, \bar{y})$
    if $f_{\text{curr}} < f_{\text{best}}$ **then**
      update f_{best} and remember solution as $(\hat{x}, \hat{y})$
    end if
    break
  end for
end for
return solution $(\hat{x}, \hat{y})$ corresponding to f_{best}

3.2 Computing Minimal Disk Sectors

The algorithm for solving MinimalKRectangle is based on the fact that the parameters of a rectangle, its width and height, are equivalent to the x- and y-values of its spanning points. In case of disk sectors, the defining parameters are the opening angle θ and the radius r. Therefore, we can transform the points of the off-screen section from cartesian coordinates (x, y) to polar coordinates (φ, ρ) and use these as input for the algorithm. The range ρ is equivalent to the radius r and the direction φ is equivalent to the opening angle θ of the disk sector. The radius is always positive and the direction can be normalized to the interval $[0, 2\pi)$ such that the requirements on the input points for MinimalKRectangle are fulfilled. The objective functions for minimizing area or perimeter are the following:

$$f_{\text{area}} = \frac{1}{2}\theta r^2 \tag{1}$$

$$f_{\text{perimeter}} = 2r + \theta r \tag{2}$$

For computing the direction a reference direction ϕ_0 is needed. This reference defines the right boundary of the resulting disk sector. When searching for the smallest disk sector enclosing a given set of points, one of the points must be located on the right boundary of the disk sector, otherwise we can shrink the disk sector on this side to obtain a smaller one containing the same number of points. Therefore, we apply the polar version of MinimalKRectangle once for each input point p_i, using the direction to point p_i as ϕ_0. Finally, we select the minimal disk sector as solution. Applying the algorithm for MinimalKRectangle n times results in an overall running time of $\mathcal{O}(n^2 \log n)$. Sorting the input points only once for each of the two dimensions reduces the running time to $\mathcal{O}(n^2)$.

3.3 Different Glyph Designs

Based on the described algorithm we have developed three different glyphs of increasing complexity. The *basic glyph* is made of a single disk sector. It is parameterized by the look-ahead value ℓ and the inlier-ratio κ. This glyph is kept simple to make it easy to read and understand. As a

drawback the information the glyph provides is limited as the considered off-screen section is summarized into a single disk sector.

To tackle this issue, the *staggered glyph* is made of multiple disk sectors each having different look-ahead values. The number of disc sections n is an additional parameter for this type. The individual disk sectors are layered in order of the look-ahead value with the largest look-ahead value being in the back. Figure 4a shows the staggered design. The largest disk sector in this glyph is identical to the basic glyph. The staggered glyph allows for a finer differentiation of the trajectory's off-screen section's shape as turns at more distant look-ahead values are only visible in the larger disk sectors.

In the staggered variant the neighboring off-screen section is part of all disk sectors. To counteract this, the *consecutive glyph* decouples the disk sectors by calculating them for consecutive subsections of the off-screen section. Each disk sector only consideres points between two given look-ahead values $\ell_{\min}$ and $\ell_{\max}$. Figure 4b shows the consecutive glyph. Due to the inlier ratio not all considered points need to be covered by the individual disk sector. Therefore, the disk sectors can be disjoint. To ensure that a glyph is a contiguous region and thus to improve readability, the look-ahead ranges of two consecutive disk sectors may overlap ($\ell_{\max,i} > \ell_{\min,(i+1)}$). This is comparable to a sliding average filter.

3.4 Glyph Scaling

The glyps need to be scaled down to fit on the overlay area. By construction, the glyphs cover the off-screen section of the trajectory which has a length ℓ. Considering the extreme case that the off-screen section is a straight line, the maximum possible radius $r_{\max}$ for a glyph is ℓ. This glyph needs to fit on an overlay area of width b. To ensure that all glyphs fit on the overlay area we scale each disk sector's radius with a scale factor $s = \frac{b}{\ell}$.

4 Experimental Evaluation

In this section, we evaluate the glyphs. Specifically, we want to analyze whether the glyphs are intuitive to understand and convey a correct impression of the off-screen evolution of the trajectory. Special emphasis lies on the comparison of the glyph types.

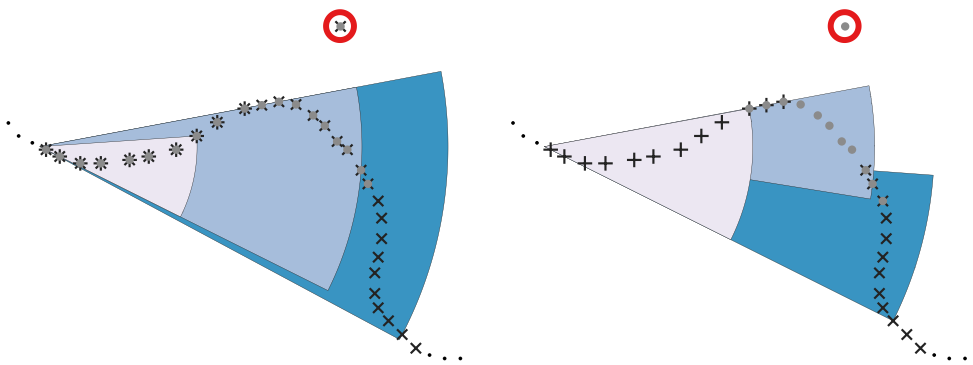
For the evaluation, we performed an online user study. The details of the study design are given in Sect. 4.1. In Sect. 5 we present a detailed discussion of our results.

4.1 Experimental Setup

In our user study, participants were asked questions on the off-screen evolution of trajectories based on the glyphs. The study is separated into two different tasks:

1. *Interpretation* Given a real-world trajectory and the corresponding glyph, the participant has to select the correct off-screen evolution out of a selection of three options.
2. *Preference* The three glyph types and the corresponding on- and off-screen trajectory are shown. The participant is asked to pick the glyph that they prefer for the given situation.

Figure 5 shows two examples for these task types. Task 1 (Fig. 5a) features multiple choice questions, where exactly one option is correct. The answers to this task give information on the interpretability of the glyphs. We record the given answer as well as the time each participant needs to answer. All options in Task 2 (Fig. 5b) are correct, they only differ in the type of the displayed glyph. The aim of this task is to get a subjective opinion of the participant on which glyph is matching the situation best. In the case that none of the glyphs pleases the participant, a fourth option is provided. In the study, participants receive six questions for Task 1 and three questions for Task 2. The complete study



(a) Staggered glyph **(b)** Consecutive glyph

Fig. 4 Considered vertices of the first disk sector (black plus), the second disk sector (gray dots) and the third disk sector (black cross) for the advanced glyphs. As not all of the vertices need to be covered by the disk sectors, the outlier (marked red) is not included into the glyphs (color figure online)

(a) Task 1: Interpretation **(b)** Task 2: Preference

Fig. 5 Illustration of the tasks. Task 1 tests if the glyph comprehensively indicates the further evolution of the track. The correct trajectory continuation has to be selected. Task 2 allows to subjectively choose the most appropriate glyph. All three variants fit to the trajectory on the right side, but the participant can select his preferred glyph type

can be examined at https://www.geoinfo.uni-bonn.de/offscreen-evolution-study/.

4.2 Question Layout

To avoid a learning effect in the study we separate the participants for Task 1 into three groups *A*, *B* and *C* and generated six question instances. An instance corresponds to a map frame with a single real-world trajectory and three possible off-screen evolutions of which only one is the original, correct one. All three different glyph types are shown to each of the groups so that answers for each glyph type are present for each instance. The instances are given sorted by glyph type to each group, starting by the basic glyph, continuing with the staggered one and ending with the consecutive one. We choose this order to present the glyphs in order of rising complexity. An overview of the study layout for Task 1 is given in Table 1.

Table 1 Task 1: each group (*A*, *B* and *C*) receives six questions: two per glyph type. The question order is indicated by the subscript

Task 1	Glyph Type		
	Basic	Staggered	Consecutive
Instance 1	A_1	B_3	C_5
Instance 2	C_1	A_3	B_5
Instance 3	B_1	C_3	A_5
Instance 4	A_2	B_4	C_6
Instance 5	B_2	A_4	B_6
Instance 6	C_2	C_4	A_6

The study starts with a short tutorial introducing the two tasks. For each task, one introductory instance is presented. The tutorial guarantees that the timing of the answers is not skewed at the beginning. We deliberately refrain from giving an explanation of the glyphs to evaluate their intuitiveness.

4.3 Instance Generation

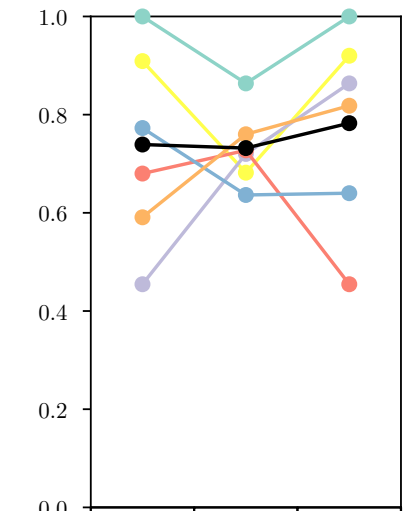
To simulate a real-world example as close as possible we use real-world trajectories recorded from cyclists in the metropolitan area of Cologne and Bonn to generate the instances. The map frames are selected randomly along the trajectories. To ensure equal preconditions, the map frames all have the same extent and are rotated such that the trajectory leaves the map at the right-hand side boundary. The wrong options for the off-screen evolution used for Task 1 are generated by selecting an alternative destination outside the displayed map and using an online bicycle routing service[1] to compute an optimal cycling path to this destination.

[1] http://brouter.de/brouter-web/.

The parameters for the glyphs are identical across all glyph types and instances. We selected the parameters according to the setting of bicycle navigation. In specific, we limited the maximum look-ahead value $\ell_{\max}$ to 1000 meters as this relates to 3–4 min of cycling, which seems sufficient for general usage. Furthermore, we selected the inliner-ratio $\kappa = 0.90$ and the width of the overlay area $b = 200$m. For the staggered and consecutive glyph, additional look-ahead values for a second and third disk sector of $\ell_0 = 500\,m$ and $\ell_1 = 750\,m$ and an overlap of 50% of the look-ahead ranges have been chosen, resulting in the ranges $\mathcal{I}_1 = [0, 500], \mathcal{I}_2 = [250, 750]$ and $\mathcal{I}_3 = [500, 1000]$ (all in meters). All disk sectors are minimized regarding their perimeter using $f_{\text{perimeter}}$ as objective function. We refrain from using the f_{area} as initial testing showed that this favors narrow disk sectors which are hard to recognize on a map. Using the perimeter instead creates compact disk sectors. Using our implementation, the computation of individual glyphs took less than 0.1 s. Further testing showed that glyphs for large instances with $n = 2000$ points and an inlier-ratio of 50% could be computed in under 0.3 s.

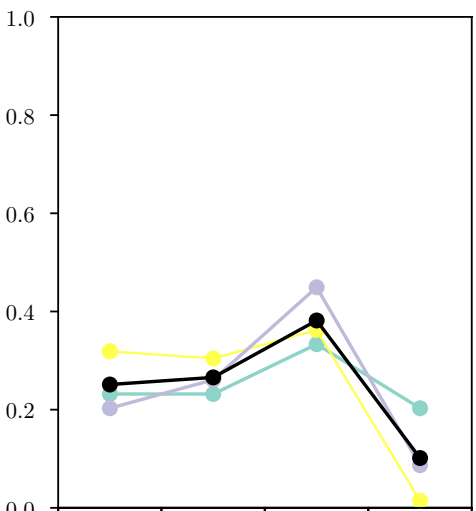
5 Results and Discussion

The study was conducted as an online study and was distributed via social media, the student association and by word-of-mouth recommendation. In total, 72 participants volunteered for the study, who randomly got assigned to one of the groups, resulting in the following group sizes: $\|A\| = 26, \|B\| = 22, \|C\| = 24$. The participants were asked for basic demographic and social statistics. Most of the



	Basic	Staggered	Consecutive
Instance 1	100.0%	86.4%	100.0%
Instance 2	90.9%	68.2%	92.0%
Instance 3	45.5%	72.0%	86.4%
Instance 4	68.0%	72.7%	45.5%
Instance 5	77.3%	63.6%	64.0%
Instance 6	59.1%	76.0%	81.8%
Total	**73.9%**	**73.2%**	**78.3%**

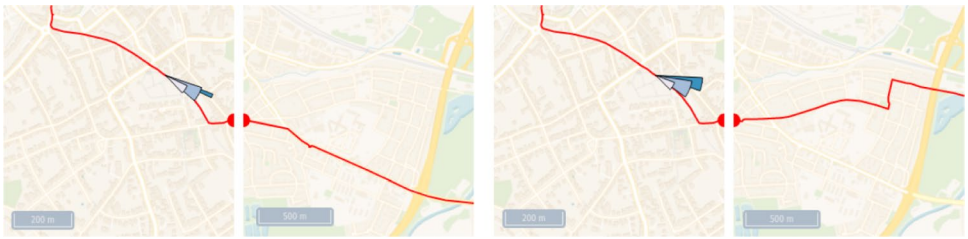
(a) Correctness



	Basic	Staggered	Consecutive	None
Instance 7	23.2%	23.2%	33.3%	20.3%
Instance 8	31.9%	30.4%	36.2%	1.4%
Instance 9	20.3%	26.1%	44.9%	8.7%
Total	**25.1%**	**26.6%**	**38.2%**	**10.1%**

(b) Preference

Fig. 6 **a** Share of correct answers per instance and glyph type **b** stated preference by instance and glyph type



(a) Given glyph **(b)** Option B (correct) **(c)** Glyph for option A **(d)** Option A (wrong)

Fig. 7 **a** Instance 4, consecutive glyph: **d** many participants selected answer A, **b** but B is the correct one. **c** Glyph if option A was correct

participants were in the age range of 16–35 years (≈ 74%), approximately 54% of the participants stated male as their gender, while 36% stated female.

5.1 Quantitative Analysis

Task 1 allows us to analyze the interpretability of the glyphs. For each question a participant scored one point if they selected the correct answer and zero otherwise. Divided by the number of questions the score gives information about the share of correct answers. Figure 6a displays the share of correct answers, differentiated by instance and glyph type. Overall, 75.2% of the answers were correct. However, there are substantial differences when comparing individual instances to each other. For example, Instance 1 has an correctness of 94.4%, while Instance 4 has only been answered correctly 61.8% of the cases. While all three glyph types have a similar correctness between 73.2% and 78.3%, the results for a single instance differ substantially for different types. For example, the consecutive glyph has nearly twice as much correctness as the basic one in Instance 3. These observations suggest that the interpretability of the glyph highly depends on the individual instance. The length of the look-ahead ranges defines the resolution of the glyph. Thus, glyphs for very curvy trajectories, where the lengths of the curves are shorter than the considered look-ahead ranges, are hard to interpret, because this resolution is too low.

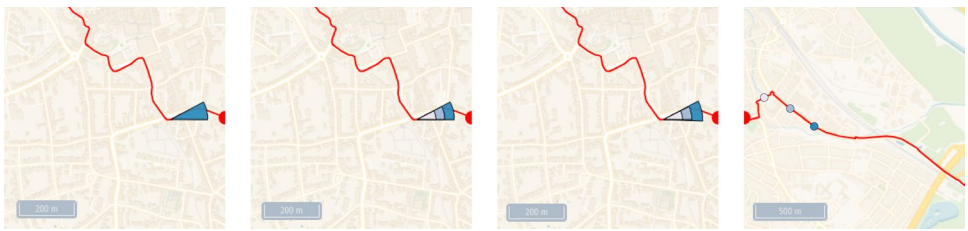
In addition to the score we recorded the answering times for each question to draw conclusions on how intuitive the glyphs are: assuming the faster a participant could answer, the more intuitive the glyph is. The average answering times for each glyph type were $t_{\text{basic}} = 29.6\text{s}$, $t_{\text{staggered}} = 22.5\text{s}$, $t_{\text{consecutive}} = 17.9\text{s}$. While this implies that the more complex glyphs are more intuitive, we refrain from putting any importance on these results, as the types occurred in order basic → staggered → consecutive during the study. Thus, the speed-up might be related to learning effects or subsiding diligence towards the end of the study. Future study designs should account for such effects.

Figure 6b shows the glyph preference as stated by the participants in Task 2. The consecutive glyph is preferred by more users than the other glyph types for all instances. Nonetheless, the preferred types are quite balanced with the lowest preference still being above 20%. Noticeable is the large share (22.2%) of participants that found neither of the shown glyphs fitting for Instance 7. This coincides with the previous observation that the instance has a large impact on the quality of a glyph.

5.2 Qualitative Analysis

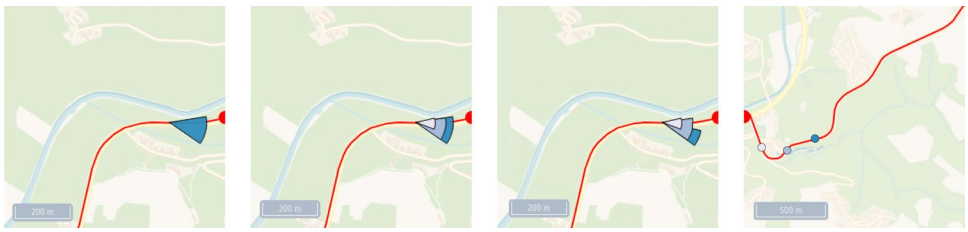
The quantitative analysis of the result revealed that a large factor determining the readability of the glyphs is the individual configuration of an instance. Therefore, we take a closer look at instances that resulted in ambiguous results in our study.

Even though the consecutive glyph achieved the overall best results, its accuracy in Instance 4 is only 45.5% (Fig. 6a). Out of the 22 answers, 12 participants decided for the wrong option *A* instead of the correct option *B*. Option *C* was not selected at all. The glyph thus seems to be ambiguous for the first two options. Figure 7 shows Instance 4, with the given glyph shown in Fig. 7a. The narrow disk sector for the largest look-ahead value might convey the wrong



(a) Basic **(b)** Staggered **(c)** Consecutive **(d)** Evolution

Fig. 8 Instance 5: all three variants **a** basic, **b** staggered and **c** consecutive look very similar due to the curvy evolution of the trajectory in close vicinity to the cut-off point. **d** The off-screen evolution with loook-ahead values indicated by blue markers (color figure online)



(a) Basic **(b)** Staggered **(c)** Consecutive **(d)** Evolution

Fig. 9 Instance 7: participants cannot decide on a preferred glyph for the given situation. **d** The look-ahead distances are not obvious due to the curves in the off-screen part

perception of an evolution more to the top of the map as seen in option *A* (Fig. 7d) compared to option *B* (Fig. 7b). For comparison, the correct glyph for the off-screen trajectory in option *A* is shown in Fig. 7c. Comparing the two glyphs, the disk sectors for the first two look-ahead values are nearly identical, but the last one changes significantly. A reason for the high share of wrong answers might be the arrangement of the options in the study (see Fig. 5a): only option *B* is placed right next to the visible map frame, for the other options the perspective might be a bit skewed and the directions might not be easy to identify.

In Instance 5 all glyph types score a similar correctness of 64.0–77.3%. Figure 8 shows the glyphs with the off-screen evolution. All three variants look quite similar. Reason for this is the curvy off-screen section of the trajectory in close vicinity to the cut-off point in conjunction with the large overlap in the look-ahead intervals. Figure 8d shows the look-ahead values 500 m, 750 m and 1000 m. As these positions are derived following the trajectory and due to the curvy nature of the off-screen section, these positions are all very close to the map boundary. The visualization can be improved by selecting the look-ahead position based on a *radius* instead of along the trajectory. Another improvement can be achieved by reducing the overlap parameter. Both of these measures decrease the shared information between the disk sectors and thus increase the information conveyed by the glyph.

In Task 2, Instance 7 (Fig. 9) many participants could not decide for a preferred glyph. Similar to the previous example the near off-screen evolution is quite curvy and thus the maximum look-ahead value is close to the map boundary. Again, we suggest using a fixed radius as look-ahead value to improve the visualization.

6 Conclusions

In this article we have proposed using glyphs in the form of disk sectors to indicate how trajectories evolve outside of the visible map frame. We have outlined an algorithm that computes the glyphs in $\mathcal{O}(n^2)$ time under relatively mild assumptions regarding the optimization objective. Future research may deal with finding an algorithm of sub-quadratic time for specific objective functions, which has been achieved for related algorithmic problems (Eppstein and Erickson , 1994).

We have opted for the perimeter of a glyph as the minimization objective as minimizing the area of a glyph resulted in very narrow glyphs. We evaluated the approach in an online user study. The results show that the glyphs are simple and informative. Nonetheless, the trajectories' geometry has a large impact on the readability of the glyphs and, for some cases, the tested glyphs do not seem to convey the intended information. In our evaluation we identified the parameter settings of the glyphs used in our study as one reason why this is the case. For future applications, we, therefore, recommend to reduce the overlap of the look-ahead intervals for glyphs. Additionally, we recommend using a fixed radius to determine the off-screen sections instead of a distance along the trajectory.

In our evaluation, we focused on the efficacy of single glyphs. For the visual exploration of large sets of trajectories many glyphs must be displayed on the same map (see Fig. 1). Challenges such as the overlap between multiple glyphs, therefore, remain to be addressed in future research.

Funding Open Access funding enabled and organized by Projekt DEAL. This research was supported by the German Research Foundation (DFG), grant Ha 5451/7-1, within the DFG priority program, grant 1894.

Declarations

Conflict of interest All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

References

Andrienko G, Andrienko N (2010) A general framework for using aggregation in visual exploration of movement data. Cartogr J 47(1):22–40. https://doi.org/10.1179/000870409X12525737905042

Baudisch P, Rosenholtz R (2003) Halo: a technique for visualizing off-screen objects. Proc SIGCHI Conf Hum Factors Comput Syst (CHI 03). https://doi.org/10.1145/642611.642695

Burigat S, Chittaro L (2011) Visualizing references to off-screen content on mobile devices: a comparison of arrows, wedge, and overview + detail. Interact Comput 23(2):156–166. https://doi.org/10.1016/j.intcom.2011.02.005

Burigat S, Chittaro L, Gabrielli S (2006) Visualizing locations of off-screen objects on mobile devices: a comparative evaluation of three approaches. Proc Conf Hum-Comput Interact Mob Dev Serv (MobileHCI 06). https://doi.org/10.1145/1152215.1152266

Cakmak E, Schäfer H, Buchmüller J et al (2020) Motionglyphs: visual abstraction of spatio-temporal networks in collective animal behavior. Comput Graph Forum 39(3):63–75. https://doi.org/10.1111/cgf.13963

Carmo MB, Afonso AP, Matos PP (2007) Visualization of geographic query results for small screen devices. Proc ACM Workshop Geograph Inf Retr (GIR 07). https://doi.org/10.1145/1316948.1316965

Custers B, Meulemans W, Speckmann B et al (2021) Coordinated schematization for visualizing mobility patterns on networks. Proc Int Conf GIS (GIScience 21) 208:1–16. https://doi.org/10.4230/LIPIcs.GIScience.2021.II.7

Demšar U, Buchin K, Cagnacci F et al (2015) Analysis and visualisation of movement: an interdisciplinary review. Mov Ecol 3(1):1–24. https://doi.org/10.1186/s40462-015-0032-y

Eppstein D, Erickson J (1994) Iterated nearest neighbors and finding minimal polytopes. Discrete Comput Geom 11(3):321–350. https://doi.org/10.1007/BF02574012

Ferreira N, Poco J, Vo HT et al (2013) Visual exploration of big spatio-temporal urban data: a study of New York city taxi trips. IEEE Trans Vis Comput Graph 19(12):2149–2158. https://doi.org/10.1109/TVCG.2013.226

Filho JAW, Stuerzlinger W, Nedel L (2020) Evaluating an immersive space-time cube geovisualization for intuitive trajectory data exploration. IEEE Trans Visu Comput Graph 26(1):514–524. https://doi.org/10.1109/TVCG.2019.2934415

Gedicke S, Jabrayilov A, Niedermann B et al (2021) Point feature label placement for multi-page maps on small-screen devices. Comput Graph 100:66–80. https://doi.org/10.1016/j.cag.2021.07.019

Goodchild MF (2007) Citizens as sensors: the world of volunteered geography. GeoJournal 69(4):211–221. https://doi.org/10.1007/s10708-007-9111-y

Gustafson S, Baudisch P, Gutwin C et al (2008) Wedge: clutter-free visualization of off-screen locations. Proc SIGCHI Conf Hum Factors Comput Syst (CHI 08). https://doi.org/10.1145/1357054.1357179

Hägerstrand T (1970) What about people in regional science? Pap Reg Sci Assoc 24(1):7–24. https://doi.org/10.1007/BF01936872

Huang R, Huang C, Shan J et al (2013) Evaluation of GPS trajectories on VGI and social websites. Int Conf Geoinform. https://doi.org/10.1109/Geoinformatics.2013.6626121

Kraak MJ (2003) The space-time cube revisited from a geovisualization perspective. In: Proceedings of 21st International Cartographic Conference, pp 1988–1996

Li R, Zhao J (2017) Off-screen landmarks on mobile devices: levels of measurement and the perception of distance on resized icons. KI - Künstliche Intell 31(2):141–149. https://doi.org/10.1007/s13218-016-0471-7

Markovic N, Miller S, Laan ZV et al (2020) Visual exploration of Utah trajectory data and their applications in transportation. Tech Rep 5:5. https://doi.org/10.15760/trec.244

Narzt W, Pomberger G, Ferscha A et al (2004) A new visualization concept for navigation systems. In: Stary C, Stephanidis C (eds) User-centered interaction paradigms for universal access in the information society. Springer, Berlin, pp 440–451. https://doi.org/10.1007/978-3-540-30111-0_38

Scheepens R, Wetering Hvd, Wijk JJV (2014) Non-overlapping aggregated multivariate glyphs for moving objects. IEEE Pac Vis Symp. https://doi.org/10.1109/PacificVis.2014.13

Sester M, Brenner C (2005) Continuous generalization for visualization on small mobile devices. Developments in spatial data handling. Springer, Berlin, pp 355–368. https://doi.org/10.1007/3-540-26772-7_27

Ssin SY, Walsh JA, Smith RT et al (2019) GeoGate: correlating geo-temporal datasets using an augmented reality space-time cube and tangible interactions. IEEE Conf Virtual Real 3D User Interfaces (VR). https://doi.org/10.1109/VR.2019.8797812

Tukey JW (1977) Exploratory data analysis, vol 2. Addison-Wesley, Boston

Yoo HY, Cheon SH (2006) Visualization by information type on mobile device. Proc Asia-Pac Symp Inf Vis (APVis 06). https://doi.org/10.5555/1151903.1151925

Zellweger PT, Mackinlay JD, Good L et al (2003) City lights: contextual views in minimal space. Ext Abstr Hum Factors Comput Syst (CHI EA 03). https://doi.org/10.1145/765891.766022