Median Trajectories

Kevin Buchin
TU Eindhoven

Maike Buchin
TU Eindhoven

Marc van Kreveld
Utrecht University

Maarten Löffler
University of California at Irvine

Rodrigo I. Silveira
Universitat Politècnica de Catalunya

Carola Wenk
University of Texas at San Antonio

Lionov Wiratma
Parahyangan Catholic University

Abstract

We investigate the concept of a *median* among a set of trajectories. We establish criteria that a "median trajectory" should meet, and present two different methods to construct a median for a set of input trajectories. The first method is very simple, while the second method is more complicated and uses homotopy with respect to sufficiently large faces in the arrangement formed by the trajectories. We give algorithms for both methods, analyze the worst-case running time, and show that under certain assumptions both methods can be implemented efficiently. We empirically compare the output of both methods on randomly generated trajectories, and evaluate whether the two methods yield medians that are according to our intuition. Our results suggest that the second method, using homotopy, performs considerably better.

1 Introduction

A relatively new type of geometric data that is being collected and analyzed more and more often is the *trajectory*: a path through space and time that a certain object traverses. This is due to technological advances like GPS, RFID tags, and mobile phones, and has caused an increase in demand for analysis possibilities. New analysis methods for trajectory data have been developed in the last few years, but a number of basic concepts are still lacking a satisfactory study. One of these concepts is the *median trajectory* for a given collection of trajectories. Intuitively, a median trajectory is a trajectory that uses pieces of the trajectories of the collection and is somehow in the middle. However, it is not clear how this concept should be defined. In this paper we establish criteria that we believe a median trajectory should meet, and we develop two median definitions that meet these criteria. Furthermore, we give algorithms to compute the median trajectory according to these definitions, and analyze experimentally whether our definitions give useful output.

1.1 Trajectories

Trajectories are a type of geographic data that have a temporal and a spatial component. Trajectories describe the locations over time of an entity that can move. The entity can be a person, animal, vehicle, hurricane (eye of), shopping basket (with an RFID tag), or any other moving object. We assume that the movement is continuous, but is measured at a discrete set of times.

Formally, a trajectory is the time-stamped path taken by a moving object, and is typically represented by a sequence of $n+1$ tuples of points and time stamps, $(p_0, t_0), \ldots, (p_n, t_n)$, which are points in space-time, where space is two- or three-dimensional. In this paper space is always two-dimensional. A collection of m trajectories $\tau_1, \ldots, \tau_m$ therefore gives rise to an input size of $\Theta(nm)$. In some applications, the time stamps of the m trajectories are exactly the same, while in other applications they are different. In general, trajectories can be collected with different or irregular sampling rates, at different times, and data can be missing. In between time stamps, we have no knowledge of the movement of the entity. The standard assumption is that the moving object moves with constant velocity from a time-stamped point to the next time-stamped point. Therefore, the path of a trajectory is a polygonal curve with n edges that can self-intersect, and can have repeated vertices if the entity does not move. Often, the number of edges of a single trajectory is much larger than the number of trajectories in a set, that is, $n \gg m$.

1.2 Trajectory analysis

Analysis methods for trajectories have been developed in Geographic Information Science and in Data Mining. Sets of trajectories can be analyzed in a variety of ways. They can be clustered into a collection of subsets that have a high within-subset similarity and a low across-subset similarity (e.g. [19, 28] and many more). They can be classified if a clustering is given [29]. Also, movement patterns on them can be computed [**?**, 20, 27]. Movement patterns that have been defined and for which algorithms have been suggested are flocking, convoys, herds, leadership, commuting, encounter, and various others. These intuitively represent similar movement in a group (same location), similar movement over a time span (same heading), or movement to the same position (same destination). An overview and classification of movement patterns relevant to trajectory data can be found in [13].

Several analysis tasks require a definition of (and algorithms for) similarity of trajectories. For instance, a simple similarity measure for trajectories is the average distance at corresponding times. With a similarity measure, or its inverse, a distance measure, clustering methods can easily be given. Single linkage and complete linkage clustering need a similarity measure only, and that defines the clustering. On the other hand, k-means and k-medoids clustering requires a definition of the mean and the median, respectively, regardless of whether the data are numbers or trajectories.

1.3 Mean and median trajectories

The intuition behind a mean trajectory is that it averages locations, one of each trajectory, like a center of gravity. In contrast, the intuition behind a median trajectory is that it always is central with respect to the number of given trajectories. Imagine a collection of GPS tracks from hikes by different people on different days. The hikers may have followed the same route globally, but there may have been options like going left around a lake or right, or taking a detour to a viewpoint. From their tracks, we want to extract a good global route. Notice that if in such a data set, seven hikers went left around a lake and three went right, then a mean trajectory would actually go through the lake, whereas a median trajectory would go with the group of seven. Similarly, with a side path to a viewpoint and back, if the majority goes to the viewpoint, then a median trajectory should do so as well. A mean trajectory might go partially to the viewpoint, which does not make sense in this context.

1.4 Overview of results

In Section 2 we discuss the idea of median trajectories. No definition has been suggested yet, so we investigate properties that a suitable median should have. We first propose a simple definition (*simple median*) that directly follows the definition of a *level* in an *arrangement of lines*, a standard concept in computational geometry [17]. We also propose a more refined definition (*homotopic median*) that uses geometric and topological concepts, and may be better suited to most applications that involve trajectories. Then we discuss the maximum combinatorial complexity of a median according to these definitions.

In Section 3 we prove that both median definitions have the property that it is always in the middle (in a sense to be defined later).

In Section 4 we present algorithms that compute median trajectories according to the two definitions. We can compute the simple median in $O((nm)^2)$ time, and the homotopic median in $O((nm)^{2+\epsilon})$ time for any $\epsilon > 0$, in the worst case. Here, m is the number of given trajectories and n is the maximum complexity of any trajectory. We improve our algorithms for practical situations. We can compute the simple median in $O((nm + k)\alpha(nm)\log(nm))$ time, where α is the inverse Ackermann function and k is the number of vertices of the median, i.e., the output complexity. Under certain assumptions related to the sampling of the trajectory, we can compute the homotopic median in $O(nm\log^2(nm) + k\alpha(nm)\log(nm))$ time. We note that $k = O((nm)^2)$ in the worst case, as we show in Section 2.2. One would expect that typically, $k = \Omega(n)$ and $k = O(nm)$.

In Section 5 we give results of tests that we obtained from an implementation. We use a random similar trajectory generator and analyze the length, total angular change, and description size of median trajectories according to both definitions. In Section 6 we discuss our results and suggest directions for further research.

2 On the definition of a median trajectory

The *mean* and the *median* are notions that define a "middle" of a set of data of a certain type. For sets of numbers, their definitions are clear, but for points in the plane, for instance, there are already many different possibilities. Existing notions of middle include the center of gravity, the center of the smallest enclosing disk (also known as *Gaussian center* or *Steiner center* [14, 15]), the point that minimizes the sum of distances to the other points (the *Weber point* [16]), or the *center point* [6]. A center point is a point p that maximizes the minimum number of points of the set in any closed half-plane whose bounding line passes through p. This minimum number is known to be at least $n/3$ for any set of points in the plane [6]. The one-dimensional equivalents of these notions correspond to the mean or the median. Observe that for some point sets, such as a set of points in convex position, no point from the set would intuitively be the mean or median of the set. For a set of m real-valued functions the *median function* can be defined by taking the median function value (image) at each value of the domain. If the functions are continuous, and assuming m to be odd for ease of description, this corresponds exactly to the $\lceil m/2 \rceil$*-level* in the arrangement of the graphs of the functions [17]. Figure 1 (left) shows an example for lines. The worst-case complexity in this setting (over all sets of m lines) is unknown; it is $\Omega(m \log m)$ and $O(m^{4/3})$ [12, 17].

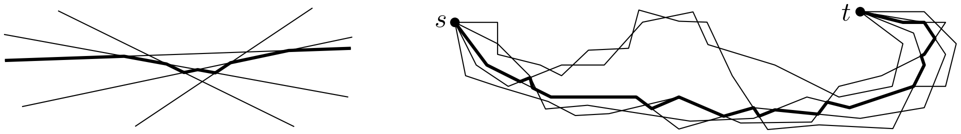

Fig. 1: The median level in an arrangement of lines (left). A median trajectory from s to t that always switches trajectory at every intersection (right).

Let us consider median trajectories. Trajectories include a temporal component as well as a spatial component, but it is not clear whether a median can take the temporal component into account in a useful way. We discuss some examples. Suppose the trajectories came from a group of animals that were traveling in a herd. Then we can use the temporal component because we know that the animals were together at any point in time. Next, suppose that the animals were traveling solitary, according to a similar route. Then they traveled on different days or months, and we cannot use the temporal component. Even if the animals had the same starting location of the route, we cannot simply align the starting times of the travel, because one animal may have been held up due to a predator, which upsets the time correspondence that we assumed at the start. The same is true for trajectories of cars with the same origin and destination: an initial time correspondence may easily be upset due to traffic lights or traffic conditions.

Thus, in many situations we want a median trajectory that does not take the temporal component into account. A similar motivation was given for trajectory similarity measures: many of these are partly shape-based, like dynamic time warping or largest common subsequence [26], or fully shape-based (ignoring the temporal component), like Hausdorff distance [4] or Fréchet distance [5]. Hence, we will concentrate on medians of trajectories based on the path of the trajectory. The median that we will define and compute will therefore be the path of a median trajectory. With slight abuse of terminology, we will just write "median trajectory" for brevity.

We note that with a temporal component, some research on modeling motion and kinetic data structures is related to the median (or mean) trajectory (e.g. [1–3]).

2.1 Requirements for a median trajectory

Let a set $T = \{\tau_1, \ldots, \tau_m\}$ of m trajectories be given, each containing n vertices. We assume for convenience that all trajectories start at the same point s and end at the same point t; this is a strong and unrealistic assumption, but we are interested in a clean definition of the median where behavior at the ends is not considered important. We also assume that no trajectory passes through

s or t a second time and that s and t are incident to the unbounded face of the arrangement of curves corresponding to the trajectories. We need the latter assumption to decide which trajectory is in the middle at the start and end. Note that we also assume a direction on the trajectories, namely from s to t, and for convenience we assume that m is odd. Finally, we assume that the curves do not touch or coincide with each other or themselves at any point, unless they cross, and no three trajectories pass through a common point. Several of these assumptions can be removed, but they make the description easier.

We list several required properties of the median trajectory:

1. The median trajectory is a polygonal curve from s to t.
2. Any point on the median trajectory lies on some trajectory of the input.
3. Any edge of the median trajectory has the same direction as the edge of the input trajectory on which it lies.
4. For any point p on the median trajectory, the minimum number of distinct trajectories that p must cross to reach the unbounded face (including the one(s) on which p lies) is $(m+1)/2$.

Besides these requirements, a number of desirable properties of the median trajectory can be given: Its length, total angular change, and number of vertices should be about the same as in the input trajectories. Finally, the median trajectory should be robust with respect to outliers: if ten trajectories follow the same route but one or two are completely different, the presence of these two outliers should not influence the median trajectory much. In particular, in the presence of outlier trajectories, the last property should be restated with m defined as the number of non-outlying trajectories instead of the total number of trajectories.

Let $\mathcal{A}$ be the arrangement formed by the (paths of the) trajectories in T. It is composed of $O(nm)$ line segments and therefore it may have complexity up to $\Theta((nm)^2)$. The median trajectory is a path that follows edges of this arrangement. In the immediate neighborhood of s, it is clear how the median trajectory leaves s: Since we assume that s is on the outer face, exactly one face incident to the start point is the unbounded face. Then we can order the m edges adjacent to s with the first and last edge adjacent to the outer face. Then the edge the median starts on is simply the $\lceil m/2 \rceil$-nd edge in the order.

2.1.1 Simple definition

Inspired by the median level in an arrangement of lines, we can give a very simple definition of the median: It is the trajectory obtained after leaving s in the only possible way while satisfying the last property, and then switching the trajectory at every intersection point, following the next trajectory in the forward direction (see Figure 1 (right)). Note that if the trajectories are x-monotone, then this definition gives the same result as the $\lceil m/2 \rceil$-level or median function given before. We refer to a median by this definition as a *simple median*.

Lemma 2.1. *The simple median satisfies the four required properties for a median.*

Proof. Properties 1, 2 and 3 are clearly true. Property 4 follows from Lemma 3.1, which we prove in the next section together with the proof of this property for the other median trajectory definition. □

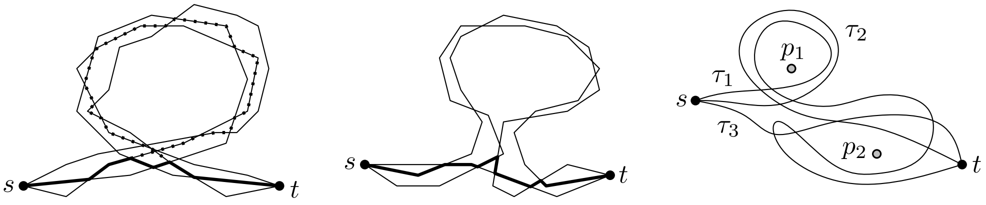

Fig. 2: A loop in all trajectories makes the outcome (bold) of always switching trajectory undesirable (left). Undesirable outcome can result even when no trajectory has self-intersections (middle). Homotopy with respect to two poles (right).

Although this definition often gives desirable results, it can behave badly when trajectories are self-intersecting, see Figure 2 (left). All three trajectories make the loop, but the simple median does not. Similarly, when the trajectories are not self-intersecting, the majority route can also be missed. In Figure 2 (middle), two trajectories go up and make a detour, while one trajectory stays low. The simple median in this situation also stays low.

2.1.2 Homotopy definition

To obtain a more natural median trajectory definition, we identify parts of the plane with respect to which the median should behave the same as most of the input trajectories. In both examples of Figure 2, we have a region in the plane that is a bounded face of the arrangement $\mathcal{A}$ that is relatively large. We propose placing *poles* in such large faces and require that the median goes in the same way around the poles as the input trajectories, using the concept of *homotopy*.

We make this more precise. Let $T = \{\tau_1, \ldots, \tau_m\}$ be the input trajectories and let $P = \{p_1, \ldots, p_h\}$ be a set of h poles which are assumed to not lie on any trajectory. Since the trajectories all go from s to t, we can use deformability of the trajectories into each other in the punctured plane [23]. Two trajectories τ_i and τ_j are *homotopic* if one can be deformed continuously into the other without passing over any pole, and while keeping s and t fixed. In Figure 2 (right), τ_1 and τ_2 are homotopic to each other, while τ_3 is not homotopic to τ_1 or τ_2. Homotopy is an equivalence relation.

We first discuss how we define the median when all trajectories in T are homotopic with respect to P. We use a variation of the trajectory switching approach: follow the median over the correct edge at s. Assume we have followed the median and we are on an edge of trajectory τ_i, and we encounter an intersection v with a trajectory τ_j, with $1 \leq i, j \leq m$. If the median so far, concatenated with trajectory τ_j from v until t, has the same homotopy type as the input trajectories, then we switch to τ_j, otherwise we ignore the intersection and stay on τ_i. This approach maintains the invariant that if we would simply stay on the current trajectory until t, the homotopy type of the median is correct. In fact, this method is equivalent to computing the simple median, but in a universal cover space based on the poles, where only the intersections with the correct homotopy remain; details about this are given in Section 3.2. A median by this definition is referred to as a *homotopic median*. In Figure 2 (left) the dotted loop would be included in the homotopic median.

Lemma 2.2. *The homotopic median satisfies the required properties for a median.*

Proof. Properties 1, 2, and 3 are clearly true. Property 4 follows from Lemma 3.2, proved in the next section. □

The remaining question is how to place a set of poles P such that ideally, the trajectories in T are homotopic with respect to it. A number of different strategies for this are conceivable. We choose to use a simple approach that places a pole in a face of $\mathcal{A}$ whenever it is larger than r, i.e., a disk of size r fits in the face, for some value of r to be determined later. This is motivated by the fact that large faces are more likely to be important. This choice gives no guarantee that the trajectories will be homotopic, however. To solve this we could either increase r or allow some of the trajectories to have a different homotopy, and instead compute the median for a subset $T' \subset T$, i.e., effectively treating the remaining trajectories as outliers.

We note that homotopy has been used recently to define the similarity of two curves. In an environment without obstacles, the Fréchet distance is a good measure for the dissimilarity of two curves, but when there are obstacles between the curves, an extension is needed. This has led to the definition of homotopic Fréchet distance [10], which also uses the punctured plane.

2.2 The complexity of median trajectories

Since the arrangement $\mathcal{A}$ formed by the m trajectories has complexity $O((nm)^2)$, the median trajectory cannot have more edges than that. We show that it is actually possible for the median to have complexity $\Omega((nm)^2)$ for both definitions. We show that this lower bound holds even for m non-self-intersecting trajectories.

We first give an example that shows that even a single trajectory can give rise to a median that has $\Theta(n^2)$ complexity. Consider the simple definition or assume that r is so large that there are no poles. Then at every intersection point, we switch the trajectory.

The lower bound construction, illustrated in Figure 3 (left), is based on a set of long vertical line segments that are directed (from left to right) up, down, down, up, up, down, down, and so on, and a set of long horizontal line segments that are directed (from bottom to top) rightward, leftward, rightward, leftward, and so on. Every long vertical line segment intersects every long horizontal line segment. We will connect these line segments to form one trajectory that (i) respects the directions, (ii) places the source and target on the outer face, and (iii) induces a median that turns at every intersection point of the grid of long line segments. We note that a simpler example exists if we do not require that the source and target are incident to the outer face.

Consider the resulting median trajectory. First, the trajectory zigzags using slightly more than half of the long vertical line segments, then it goes back and zigzags again using the other slightly less than half of the long vertical line segments. After having used all long vertical line segments, the trajectory zigzags from bottom to top using all long horizontal line segments, the bottommost one directed from right to left. The trajectory and the start of the median are indicated in the figure. It is easy to see that the median of one trajectory with n edges has complexity $\Omega(n^2)$.

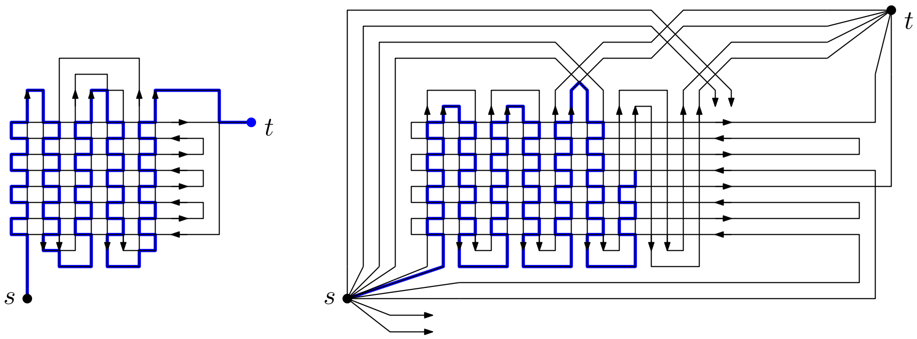

Fig. 3: The median of a single trajectory can have quadratic size (left). The median of m trajectories with n edges each can have size $\Omega((nm)^2)$ (right).

Essentially the same construction can be used to show that m trajectories with n edges each can give rise to a median of complexity $\Omega((nm)^2)$. We start with similar sets of long directed vertical and horizontal line segments, $\Omega(nm)$ of each type, see Figure 3 (right). We use $\lceil m/2 \rceil$ trajectories to connect the long vertical line segments into trajectories. We use pairs of trajectories that together essentially look like the first half of the trajectory in the single-trajectory example. They are placed next to each other. We use the other $\lfloor m/2 \rfloor$ trajectories to connect the long horizontal line segments. Using suitable intersections between the trajectories to make the median proceed its structure of zigzagging up and zigzagging down alternatingly, we obtain the lower bound of $\Omega((nm)^2)$ for the median complexity. Notice that, in constrast to the single trajectory construction, the input trajectories do not self-intersect.

Finally, we also note that for x-monotone trajectories, using the median level lower bound for arrangements of lines immediately leads to an $\Omega(nm \log m)$ lower bound on the complexity of the median in this case [31].

3 Staying in the middle

In this section, we prove that our definitions and algorithms give medians that "stay in the middle", as required, that is, we prove Property 4. This completes the proofs of Lemmas 2.1 and 2.2. We will first consider the simple setting without poles, corresponding to Section 2.1.1; after that we will show how to deal with poles.

3.1 Simple setting

Let E be a space that is homeomorphic to an open disk (for example, $\mathbb{R}^2$), let s and t be two different given points in E and let $T = \{\tau_1, \tau_2, \ldots, \tau_m\}$ be a collection of $m > 2$ trajectories starting at s and ending at t. We assume that the trajectories do not touch or coincide with each other or themselves at any point, unless they cross. We also assume that no three trajectories (or pieces of the same trajectory) cross in one point. This means that the trajectories induce an arrangement $\mathcal{A}$ on E whose vertices of degree > 2 are s, t, and all crossings between the trajectories in T, while the edges are pieces of the trajectories between two vertices of degree > 2. The trajectories define an orientation on the edges of the arrangement. Figure 4 (left) shows such an arrangement for a set of trajectories in $\mathbb{R}^2$.

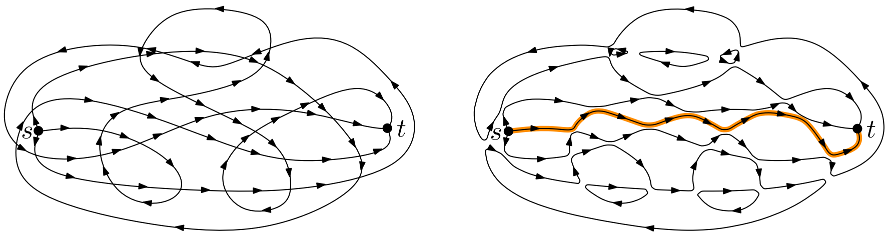

Fig. 4: A (not so natural) configuration of 3 trajectories from s to t (left). The resulting rearranged trajectories and some isolated cycles (right). The median is marked.

The vertices of $\mathcal{A}$ arising from intersections all have degree 4, having 2 incoming and 2 outgoing edges, with the two incoming edges being adjacent and the two outgoing edges as well. Vertex s has indegree 0 and outdegree m, and vertex t has indegree m and outdegree 0.

Let f be a face of $\mathcal{A}$, and let p be a point in the interior of f. Consider a ray starting at p and going to infinity that does not go through any vertex of $\mathcal{A}$ and is not tangent to any edges of $\mathcal{A}$. Let l be the number of crossings of this ray with the edges of $\mathcal{A}$ where the edge crosses the ray from right to left, and let r be the number of such crossings where it crosses from left to right. We define the *order* of f to be $(r - l) \mod m$. Note that there can be m different orders. Rotating the ray continuously does not change the order. Thus, the order is unique no matter in which direction we shoot the ray. The outer face has order 0. Also note that there are m faces adjacent to s (or t), and that they all have different orders. We will say that the outgoing edge from s between the face of order 0 and the face of order 1 is the *first* edge, the edge between the faces of order 1 and 2 is the second edge, etc., until the m-th edge between the faces of order $m - 1$ and order 0.

Now we can state formally what we want to show.

Lemma 3.1. *Let E, s, t, and T be as described above. A path π leaving s on the k-th edge, switching in the forward direction at every intersection, will end at t arriving on the k-th edge. Furthermore, at any point on π, we have the property that leaving on the left side of π and without crossing π again, at least $k - 1$ distinct trajectories from T must be crossed to reach the outer face, and leaving on the right side of π, at least $m - k - 1$ distinct trajectories must be crossed.*

Proof. A path that starts on the k-th edge is an edge between a face of order $k - 1$ and a face of order k. Each vertex of $\mathcal{A}$ other than s and t has one face between its two incoming edges and one face between its two outgoing edges, and these have the same order. Whenever we reach a vertex (intersection) of $\mathcal{A}$, according to our procedure, we "switch to the other trajectory". Suppose that the other trajectory is coming from the left. This means that we switch to the right, stay incident to the face of order $k - 1$ we were incident to before, and switch to another face of order k. We will always stay between a face of order $k - 1$ and one of order k, therefore we end on the k-th edge. Furthermore, since the order difference between adjacent faces is always 1, at least k or $m - k$ trajectories must be crossed to get from a face of order k to the outer face (which has order 0). □

We can, in fact, define a new set T' of m trajectories going from s to t to be the m paths that the simple switching algorithm produces when leaving s on its m different incident edges. We remove every vertex of $\mathcal{A}$ (except for s and t) by connecting each incoming edge to its adjacent outgoing edge, joining two pairs of edges. We shortcut slightly in an ε-neighborhood of the vertex, merging one pair of faces. After this, the arrangement has been decomposed into m simple trajectories that go from s to t and possibly a number of isolated closed loops, none of which intersect each other. Figure 4 (right) shows an example.

Finally, observe that if s and t are incident to the outer face, then the $\lceil m/2 \rceil$-th face is what we would intuitively call the "middle".

3.2 Homotopic curves amidst poles

Let $P = \{p_1, \ldots, p_h\}$ be a set of h poles in the plane; the poles act as point obstacles. We will consider the domain $D = \mathbb{R}^2 - P$, which is a punctured plane, and see how to define medians of trajectories that live in D.

It is well-known that a closed curve in D has a homotopy class which can be described as a reduced sequence of elements of the generating set of its fundamental group and their inverses (see, e.g., [7]). We can choose as generating set the set of h counterclockwise curves around each of the poles in P rooted at some base point (not in P). In other words, each closed curve can be decomposed into a number of these cycles and their inverses.

Two trajectories from a starting point s to an endpoint t are homotopic if the closed curve obtained by gluing them together can be contracted to a point, or equivalently, is in the homotopy class 0. One of the consequences of this definition is that the points s and t play no special role. In particular, it does not make a difference if a trajectory makes an extra loop around one of them or not.

Now, let T be a set of m trajectories from s to t that are all homotopic to each other in D. We will show that similar properties as in the simple setting still hold for T. We follow the description in [9] and argue based on a universal cover U of D.

A universal cover is a topological space that locally looks like the original space but is simply-connected (see e.g. [30] for a formal description). In our setting, it can be described as follows: we cut up D along vertical rays starting at each point of P, resulting in vertical slabs. We now glue together copies of these vertical slabs at the boundary rays, in such a way that no cycle is formed. Thus, if one stands in a point of one of the copies and starts walking, and walks around one of the poles in P, one will end up in a different copy of D, even though the projection will be the same.

This space U is homeomorphic to an open disk. Note that a closed curve in D is closed in U if and only if its homotopy class is 0. Therefore, all our trajectories in T have the same start- and endpoint in U as well. And as a result, Lemma 3.1 still applies in this case. One way to visualize this is by taking a bounded subset of U without any poles that contains all trajectories of T. Such a set can be viewed as a projection of a surface in $\mathbb{R}^3$(see also [18]). Figure 5 (left) shows a set of homotopic curves in such a subset.

Lemma 3.2. *Let s, t, T and P be as defined above. A path π starting on the k-th edge leaving from s and switching at every intersection with the correct homotopy type will end on the k-th edge of t.*

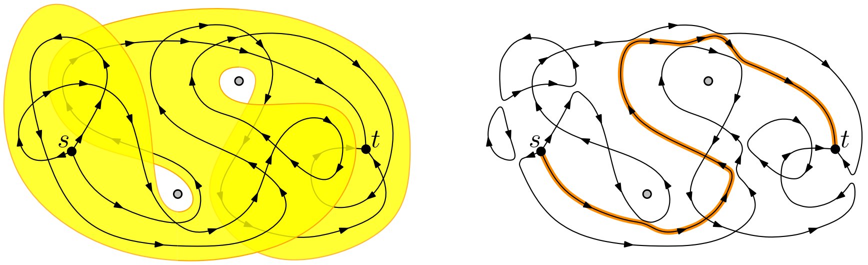

Fig. 5: A set of three trajectories from s to t that are homotopic with respect to two poles (left). We can visualize a subset of U that is homeomorphic to a disk, in which the trajectories are embedded. The resulting rearranged trajectories and some isolated cycles (right). The median is marked.

Furthermore, at any point on π, we have the property that leaving on the left side of π and without crossing π again, at least $k-1$ distinct trajectories from T must be crossed to reach the outer face, and leaving on the right side of π, at least $m-k-1$ distinct trajectories must be crossed.

Proof. Let $\mathcal{A}$ be the arrangement of the trajectories in D, and let $\mathcal{B}$ be the arrangement of the same trajectories lifted into U. All vertices of $\mathcal{B}$ are also vertices of $\mathcal{A}$. On the other hand, a vertex v of $\mathcal{A}$ of degree 4 is a vertex of $\mathcal{B}$ of degree 4 only if the two partial trajectories from s to v involved in the crossing have the same homotopy in D (or equivalently, if their continuations from v to t have the same homotopy). The homotopy definition for a median switches only at a vertices that are present in $\mathcal{B}$. Hence, the lemma follows from the same argumentation as used for Lemma 3.1 (but now in $\mathcal{B}$). □

As before, we can define a set T' of the m trajectories that the strategy of switching at every intersection with the correct homotopy type produces. Figure 5 (right) shows an example. Note that the crossings among the trajectories in T' correspond exactly to those vertices of $\mathcal{A}$ that are not in $\mathcal{B}$.

Also observe that, again, if s and t are incident to the outer face of the arrangement $\mathcal{A}$ of the trajectories in D, then they are also incident to the outer face of $\mathcal{B}$, so the "middle" is again what we intuitively expect it to be.

4 Algorithms to compute a median trajectory

In this section we show that both simple medians and homotopic medians can be computed efficiently. The simple median can be computed in $O((nm)^2)$ time in the worst case, while the homotopic median can be computed in $O((nm)^{2+\epsilon})$ time in the worst case for any $\epsilon > 0$. This is (close to) quadratic in the input size. In practice, running times will be faster because they depend on complexities of intermediate results that should be much less than the worst-case situations. In particular, let A be the complexity of the arrangement formed by the nm edges of the trajectories. Although $A = O((nm)^2)$ and this is tight in the worst case, for trajectories we typically expect it to be much smaller. Therefore, making the time bound depend on A instead of $(nm)^2$ is desirable.

Similarly, let h be the number of poles. Then $h = O(A) = O((nm)^2)$, but only large enough faces give rise to a pole so h is typically much smaller than A. Finally, the complexity of the median itself, the output size k, is $O(A) = O((nm)^2)$, but typically we expect it to be smaller. Notice that h or k can be large when the other is small. We will show that the simple median can be computed in an output-sensitive manner as well, and the homotopic median can be computed more efficiently when the number of poles and the output size are not very large. With a natural sampling assumption, we can remove the dependency on the number of poles.

4.1 Computing the simple median

A simple algorithm to compute the median with the simple definition is via the construction of the arrangement $\mathcal{A}$. The arrangement can be constructed in $O(nm \log(nm) + A)$ time, where A is the complexity of the arrangement [21]. Then we simply follow the median trajectory through this arrangement, taking $O(1)$ time at every intersection point or trajectory vertex. Every intersection point and trajectory vertex will be a vertex of the median we compute. Hence, this algorithm takes $O(nm \log(nm) + A) = O((nm)^2)$ time.

The complexity A of the arrangement is not directly related to the complexity of the median. For instance, it can happen that $A = \Theta((nm)^2)$ even when the median never switches trajectories. For an output-sensitive algorithm, we use Har-Peled's randomized algorithm for an on-line walk in a planar arrangement [22]. This algorithm allows to compute the part of an arrangement that is found when "walking" on it. In other words, the part intersected by a curve not known in advance, with the walking direction potentially changing every time an edge is crossed. In our case, the walk is precisely the median trajectory we want to compute. The expected runtime of this algorithm is $O((nm + I)\alpha(nm + I)\log(nm))$ in an arrangement of nm line segments, where I is the number of intersections between the walk and the arrangement, and α denotes the inverse Ackermann function. In our case, $I = k$ is the complexity of the median, because the median switches trajectories at every intersection. Furthermore, $I = O((nm)^2)$ and then $\alpha(nm + I) = O(\alpha(nm))$.

Theorem 4.1. *The simple median of m trajectories with n edges can be computed in $O((nm)^2)$ time or in $O((nm + k)\alpha(nm)\log(nm))$ expected time, where k is the size of the output.*

4.2 Computing the homotopic median

We give an algorithm that computes the homotopic median for a given value of r. If for the resulting poles not all trajectories are homotopic, we compute the median trajectory of the largest homotopy class with respect to these poles. The algorithm consists of three steps.

1. Compute poles, one for each face in which a disk of radius r fits.
2. Compute the homotopy type of each trajectory, and determine the type that occurs most often. Remove all trajectories that do not have this type.
3. Follow the median from s: at every intersection, determine whether the continuation on the new trajectory yields the correct homotopy type (when the new trajectory is followed to t).

4.2.1 Step 1

Step 1 can be performed in $O(nm\log(nm) + A)$ time by constructing the arrangement [21] and then computing the medial axis in each face in linear time [11]. The medial axis of a polygon P contains all points p such that there is a circle centered at p that lies inside P but touches P in at least two points. The vertices of the medial correspond to circles that touch P in at least three points. The number of vertices of P is $O(|P|)$. Since there must be a largest circle that fits inside P that touches it in at least three points, we can find this by checking all vertices of the medial axis.

4.2.2 Step 2

Step 2 can be performed using an algorithm of Cabello et al. [9]. They showed that deciding whether two paths are homotopic takes $O(n\sqrt{h}\log h)$ time, assuming the two paths have n edges and there are h poles. We show how to use this algorithm in our setting.

The algorithm of Cabello et al. [9] lifts the paths to a *universal cover* of the plane without the poles. A possible way to construct a universal cover is the following: Take a spanning tree of the poles, and extend the tree by a ray to the left from the left-most pole, and to the right from the right-most pole. This subdivides the plane into two parts B_0 and B_1. Now glue together copies of B_0 and B_1 in such a way that if paths cross a different edge of the extended spanning tree, they move into different copies of B_0 or B_1.

Two paths are homotopic in the original space if and only if they start and end in the same points when lifted to the universal cover. To check for homotopy we construct the portion of the cover space in which the paths live. More specifically, we construct the corresponding part of the *dual tree* of the universal cover. In the dual tree we have a node for each copy of B_0 and B_1, and an arc between two nodes if they are glued together, i.e., a path goes from one to the other through an edge of the extended spanning tree of the poles. We label this arc by the corresponding edge of the spanning tree. The dual tree is infinite, but we use only the relevant parts, depending on the trajectories.

Instead of two infinite rays, the algorithm in [9] uses a bounding box and connects the spanning tree to a point on the left and on the right boundary. Furthermore, the extended spanning tree is replaced by a simple path that traces the spanning tree using perturbation. We will nonetheless refer to it as the spanning tree of the poles. As a result, B_0 and B_1 are simple polygons which are preprocessed for efficient ray shooting. The algorithm uses a spanning tree on the h poles with stabbing number $O(\sqrt{h})$ for any line, which can be constructed in $O(h^{1+\epsilon})$ time for any $\epsilon > 0$. The algorithm proceeds by concatenating two paths to a loop, traverses the loop to construct the part of the universal cover of the plane without the poles in which the loop lives, and checks whether the loop starts and ends at the same point in the universal cover (refer to [9, Lemma 11]).

For our algorithm, we need to check the homotopy types of more trajectories in order to find the largest homotopy class. We do not concatenate every pair of trajectories, but trace each from the same starting point in the dual tree and find the node where it ends. The node with the highest count gives the largest homotopic subset. Tracing one trajectory takes $O(n\sqrt{h}\log h)$ time, where the $O(\sqrt{h})$ factor comes from the fact that each line segment intersects $O(\sqrt{h})$ spanning tree edges, and the $O(\log h)$ factor comes from ray shooting in B_0 and B_1 [24]. The total running time of

Step 2 is $O(mn\sqrt{h}\log h + h^{1+\epsilon})$ for any $\epsilon > 0$. With each node in the dual tree of the cover space we also store the parts of trajectories in the corresponding part of the cover space, which is needed for Step 3.

4.2.3 Step 3

Step 3 can be performed by explicitly computing the arrangement of the trajectories. We trace the median, switching between two intersecting trajectories only if this intersection also occurs in the universal cover, or equivalently, if the two trajectories are at the same node of the dual tree. Since the intersection lies in the same node of the dual tree, the parts before the intersection are homotopic and therefore the parts after the intersection are homotopic as well, since the full trajectories are homotopic. The running time of this step would then consist of $O(nm\log(nm) + A)$ time to compute the arrangement and $O(mn\sqrt{h} + A)$ time to trace the median. To obtain an output-sensitive algorithm we again use Har-Peled's algorithm, but now we walk only in the arrangements of the trajectories restricted to the corresponding part of the universal cover. More specifically, we trace the lifted median in the universal cover using the dual tree. With each node of the tree we have stored the corresponding parts of the trajectories, and at every node of the tree we will use only these parts. While we trace the lifted median, it may return to the same node of the dual tree several times and therefore go through the same arrangement several times. But since we leave/enter the corresponding part of the universal cover through the same edge of the spanning tree of the poles, we can simply concatenate the walks without adding intersections. If N_i denotes the number of line segments in the i-th node and k_i denotes the complexity of the median restricted to the i-th node, the expected running time of the walk is $O(\sum_i (N_i + k_i)\alpha(N_i + k_i)\log(N_i))$ which is bounded by $O((nm\sqrt{h} + k)\alpha(nm)\log(nm))$ since $\sum_i N_i = O(nm\sqrt{h})$ and $\sum_i k_i = k + O(nm\sqrt{h})$.

Theorem 4.2. *The homotopic median of m trajectories with n edges can be computed in $O((nm)^{2+\epsilon})$ time or in $O((nm\sqrt{h} + k)\alpha(nm)\log(nm) + h^{1+\epsilon} + A)$ expected time for any $\epsilon > 0$, where h is the number of poles, A is the arrangement size and k is the output size.*

4.2.4 Sampling assumption

It seems reasonable to assume that the size of faces that are relevant and receive a pole, represented by r, is not much less than the length s of the longest edge in any trajectory. Suppose for instance that $r \leq s/6$. Then the trajectories are sampled so sparsely that it can happen that two trajectories have exactly the same edge of length $2r$, whereas one traversed in that time unit a distance $6r$ using three sides of a square of side length $2r$ instead of one side; note that such a square contains a disk of radius r and therefore should contain a pole. However, we cannot know that the trajectories were very different, although the choice of r suggests that this is relevant. This makes the assumption $r = \Omega(s)$ reasonable; we refer to it as the *sampling assumption.* Under this assumption, we can improve the running times of our algorithms.

For Step 1, we determine all faces that receive a pole without constructing the arrangement. Let D be a disk of radius r centered at the origin. Take the Minkowski sum of every edge of every trajectory and D, and compute the union of these $O(nm)$ "race tracks". The complement of the union contains parts of all faces that are large enough, although the same face of $\mathcal{A}$ may appear as several faces in the complement of the union. Notice that homotopic equivalence is not influenced if any face of $\mathcal{A}$ has more than one pole. The sampling assumption implies that all race tracks are *fat objects* of similar size, and hence the union complexity is bounded by $O(nm)$ [32]. We use the algorithm of Kedem et al. [25] to construct it in $O(nm \log^2(nm))$ time. This gives us a set of $h = O(nm)$ poles.

For efficiency reasons in Steps 2 and 3, we must avoid having two poles closer than $2r$. Two such poles would lie in the same face of $\mathcal{A}$, so we can remove either one. We determine a subset of the poles such that every big face of $\mathcal{A}$ has at least one pole in the subset, and any two poles in the subset are at least $2r$ apart. We do this by computing the Delaunay triangulation of the poles. We then remove all edges that have length more than $2r$, and choose one pole per connected component in our subset. Let P be this subset of poles. The idea is to construct a spanning tree on P with stabbing number $O(1)$ for line segments of length at most s.

We do this as follows: take a set of vertical lines that are exactly r apart. Within each vertical slab, connect the poles by y-coordinate. Across vertical slabs, consider every two consecutive non-empty slabs; there may be empty slabs in between. We connect the rightmost point of the left slab with the leftmost point of the right slab. Any segment of length at most s crosses $O(1)$ slabs, and due to vertical spacing within a slab, it intersects $O(1)$ spanning tree edges per slab.

Lemma 4.3. *The spanning tree on P has $O(1)$ stabbing number for line segments of length at most s, in particular, for the edges of the trajectories.*

Now we again use the same algorithm as above, but replace the spanning tree with stabbing number $O(\sqrt{h})$ by this spanning tree with $O(1)$ stabbing number. The resulting expected running time for Step 3 is $O((nm + k)\alpha(nm) \log(nm))$.

Theorem 4.4. *The homotopic median of m trajectories with n edges can be computed in $O(nm \log^2(nm) + k\alpha(nm) \log(nm))$ expected time, where k is the size of the output, if the sampling assumption is satisfied.*

5 Experimental results for median trajectories

In this section we present experimental results that aim at analyzing the quality of the medians generated by our two definitions. The experiments compare the definitions quantitatively, with respect to the desirable properties mentioned in Section 2—number of vertices, total length, and total turning angle—and also qualitatively, by analyzing visually in which cases one or the other method produces counterintuitive results.

5.1 Experimental set-up

To test our two definitions and compare them, we implemented a random trajectory generator that generates sets of "similar trajectories". For each of these sets of trajectories, the medians for both definitions were computed and analyzed.

Tab. 1: Average length, average angular change, and average number of vertices for simple (S) and homotopic (H) median trajectories over 100 tests. The average of these measures for the input trajectories was normalized to 1. The standard deviation is also shown.

	length				angular change				no. of vertices			
	μ_S	σ_S	μ_H	σ_H	μ_S	σ_S	μ_H	σ_H	μ_S	σ_S	μ_H	σ_H
1a	0.961	0.17	0.995	0.02	5.743	1.02	5.052	0.88	3.660	0.71	3.364	0.53
1b	0.940	0.19	0.995	0.03	6.058	1.33	5.397	0.84	3.684	0.82	3.477	0.54
2a	0.506	0.26	0.955	0.11	4.107	2.26	4.992	0.84	2.139	1.14	3.625	0.63
2b	0.493	0.24	0.923	0.13	4.213	2.13	4.912	1.19	2.118	0.99	3.610	0.84

The random trajectory generator starts by generating a given number of waypoints uniformly distributed at random inside a rectangle, with the restriction that for three consecutive waypoints, the angle between them is not too small (to avoid U-turns). Given these waypoints, a given number of trajectories is generated that all follow the sequence of waypoints. For each trajectory, edges are generated that have a variation in length and in heading, but always somewhat towards the next waypoint. A waypoint is considered reached if a trajectory has a vertex within a certain distance from it, and then the next edge proceeds towards the next waypoint. The longest-to-shortest edge length ratio is less than 2, and the heading is up to 30° off from being directed to the next waypoint. All trajectories start at the first waypoint and end at the last one. Two examples are shown in Figure 6.

5.2 Data set

We generated many sets of 8 waypoints and then 9 trajectories for each of them. A set was accepted if at least 6 out of 9 trajectories were homotopic for a fixed value of r. For each accepted set, the medians according to both definitions were computed, giving the length, angular change, and number of vertices for both. We distinguished four types of waypoint sets, depending on two properties of the polygonal line implied by the sequence of 8 waypoints: we compare no self-intersections (1) to self-intersections (2), and low angular change (a) to high angular change (b) (angles below or above 4π). We repeatedly generated sets until we had 100 sets in each of the four classes. Note that the classes refer to the properties of the waypoints, not the trajectories themselves (trajectories may self-intersect even if the waypoint sequence does not).

5.3 Results

We summarize the results in Table 1. Observe that with self-intersections the simple median method gives medians that are on average significantly shorter than the input trajectories. The other tables also indicate that often the simple method does not deal well with self-intersecting trajectories. Regarding the angular change and the number of vertices, we see that both medians have much higher values than the average of the input. The fact that the number of vertices of the medians is higher is not surprising because intersections become vertices, and this explains why the angular change is higher as well. This higher angular change is undesirable, because it can make the median trajectory appear very different from the input trajectories. Comparing the definitions, the lower standard deviation suggests that the results for the homotopic median are more consistent.

Visual inspection showed that the simple median occasionally made "errors" (against intuition) even for inputs without self-intersections, and nearly always for inputs with self-intersections, see Figure 6. The homotopic median nearly always gave intuitive results, although an occasional "error" could be observed, due to the absence of poles in small or narrow regions. In the bottom figure in Figure 6, the homotopic median misses the loop at waypoints 5 and 6. In the top figure, the homotopic median is mostly correct but it makes extra zigzags, best visible near waypoints 6 and 7. This explains the high angular change of the median.

We also tested how the number of vertices of the median is influenced by the number of trajectories. There seems to be a linear dependence, suggesting that the output size $k = \Theta(mn)$, but this observation is highly dependent on our random trajectory generator. Hence we did not test this further.

In summary, except for the high angular change and occasional missing parts, the homotopic median appears to be a good definition for sets of trajectories generated by our generator, even for intersecting trajectories. It gives much better results than simple medians.

6 Discussion and future research

We discussed the fundamental—but up to now missing—concept of the median of a set of trajectories. We make a first step in this direction by proposing necessary and desirable conditions that, intuitively, a trajectory median should satisfy. Based on them, we presented two definitions of the path of a median trajectory of a set of trajectories, together with efficient methods to compute them. We also proved properties of the resulting medians and analyzed them experimentally.

Given the importance of the concept of a median trajectory and its novelty, we believe this paper opens up many venues of further research.

We made several restrictions in this paper that may be unrealistic. We assumed the start and end points of all trajectories to coincide and to lie in the unbounded face. We also assumed that most trajectories are similar enough; the homotopy method can deal with some trajectories that are outliers, but it cannot deal properly with the situation where *parts* of many trajectories are outliers. This can result in a situation where the largest homotopy class has only one trajectory, whereas an intuitively correct median may still exist. We also assumed the parameter r to be given; it would be desirable to choose it automatically in an efficient manner.

We have not addressed the question how to assign time stamps to the median or how to use the time stamps of the input to guide the computation. Of course it may be possible to define a median that has good properties in a completely different way, with or without using the time stamps. Finally, it would be interesting to test the definitions of medians for various types of real-world data, instead of generated data.

Acknowledgements

This research has been supported by the Netherlands Organisation for Scientific Research (NWO) under BRICKS/FOCUS grant number 642.065.503, under the project GOGO, and under project no. 639.022.707. M. B. is supported by the German Research Foundation (DFG) under grant number BU 2419/1-1. M. L. is further supported by the U.S. Office of Naval Research under grant N00014-08-1-1015. R. I. S. is also supported by the Netherlands Organisation for Scientific Research (NWO). C. W. is supported by the National Science Foundation grant NSF CCF-0643597.

References

[1] P. Agarwal, M. de Berg, J. Gao, and L. Guibas. Staying in the middle: Exact and approximate medians in R1 and R2 for moving points. In *Proc. 17th Canadian Conf. on Comput. Geom.*, pages 43–46, 2005.

[2] P. Agarwal, J. Gao, and L. Guibas. Kinetic medians and kd-trees. In *Proc. 10th European Symp. on Algorithms*, pages 5–16, 2002.

[3] P. Agarwal, L. Guibas, J. Hershberger, and E. Veach. Maintaining the extent of a moving point set. *Discrete Comput. Geom.*, 26:353–374, 2001.

[4] O. Aichholzer, H. Alt, and G. Rote. Matching shapes with a reference point. *Int. J. Comput. Geom. Appl.*, 7:349–363, 1997.

[5] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *Int. J. Comput. Geom. Appl.*, 5:75–91, 1995.

[6] N. Amenta, M. Bern, D. Eppstein, and S.-H. Teng. Regression depth and center points. *Discrete Comput. Geom.*, 23:305–323, 2000.

[7] M. Armstrong. *Basic Topology.* Springer, 1979.

[8] K. Buchin, M. Buchin, J. Gudmundsson, M. Löffler, and J. Luo. Detecting commuting patterns by clustering subtrajectories. *Int. J. Comput. Geom. Appl.*, To Appear.

[9] S. Cabello, Y. Liu, A. Mantler, and J. Snoeyink. Testing homotopy for paths in the plane. *Discrete Comput. Geom.*, 31:61–81, 2004.

[10] E. Chambers, E. Colin de Verdière, J. Erickson, S. Lazard, F. Lazarus, and S. Thite. Homotopic Fréchet distance between curves or, walking your dog in the woods in polynomial time. *Comput. Geom.*, 43(3):295–311, 2010.

[11] F. Chin, J. Snoeyink, and C. Wang. Finding the medial axis of a simple polygon in linear time. *Discrete Comput. Geom.*, 21:405–420, 1999.

[12] T. Dey. Improved bounds for planar k-sets and related problems. *Discrete Comput. Geom.*, 19:373–382, 1998.

[13] S. Dodge, R. Weibel, and A.-K. Lautenschütz. Towards a taxonomy of movement patterns. *Information Visualization*, 7(3-4):240–252, 2008.

[14] S. Durocher and D. Kirkpatrick. The Steiner centre of a set of points: Stability, eccentricity, and applications to mobile facility location. *Int. J. Comput. Geom. Appl.*, 16:345–372, 2006.

[15] S. Durocher and D. Kirkpatrick. Bounded-velocity approximation of mobile Euclidean 2-centres. *Int. J. Comput. Geom. Appl.*, 18:161–183, 2008.

[16] S. Durocher and D. Kirkpatrick. The projection median of a set of points. *Comput. Geom.*, 42:364–375, 2009.

[17] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer, 1987.

[18] D. Eppstein and E. Mumford. Self-overlapping curves revisited. In *Proc. 20th Symp. on Discrete Algorithms*, pages 160–169, 2009.

[19] S. Gaffney and P. Smyth. Trajectory clustering with mixtures of regression models. In *Proc. 5th Int. Conf. on Knowledge Discovery and Data Mining*, pages 63–72, 1999.

[20] J. Gudmundsson, M. van Kreveld, and B. Speckmann. Efficient detection of patterns in 2D trajectories of moving points. *GeoInformatica*, 11:195–215, 2007.

[21] D. Halperin. Arrangements. In J. Goodmann and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, pages 529–562. Chapman & Hall/CRC, 2004.

[22] S. Har-Peled. Taking a walk in a planar arrangement. *SIAM J. Comput.*, 30(4):1341–1367, 2000.

[23] J. Hershberger and J. Snoeyink. Computing minimum length paths of a given homotopy class. *Comput. Geom.*, 4:63–97, 1994.

[24] J. Hershberger and S. Suri. A pedestrian approach to ray shooting: Shoot a ray, take a walk. *J. Algorithms*, 18(3):403–431, 1995.

[25] K. Kedem, R. Livne, J. Pach, and M. Sharir. On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles. *Discrete Comput. Geom.*, 1:59–70, 1986.

[26] G. Kollios, M. Vlachos, and D. Gunopulos. Discovering similar trajectories. In S. Shekhar and H. Xiong, editors, *Encyclopedia of GIS*, pages 1168–1173. Springer, 2008.

[27] P. Laube and R. Purves. An approach to evaluating motion pattern detection techniques in spatio-temporal data. *Computers, Environment and Urban Systems*, 30:347–374, 2006.

[28] J. Lee, J. Han, and K.-Y. Whang. Trajectory clustering: a partition-and-group framework. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 593–604, 2007.

[29] J.-G. Lee, J. Han, X. Li, and H. Gonzalez. *TraClass*: Trajectory classification using hierarchical region-based and trajectory-based clustering. In *PVLDB '08*, pages 1081–1094, 2008.

[30] J. Munkres. *Topology: A first course.* Prentice Hall, 1975.

[31] G. Tóth. Point sets with many k-sets. *Discrete & Computational Geometry*, 26:187–194, 2001.

[32] A. van der Stappen, D. Halperin, and M. Overmars. The complexity of the free space for a robot moving amidst fat obstacles. *Comput. Geom.*, 3:353–373, 1993.

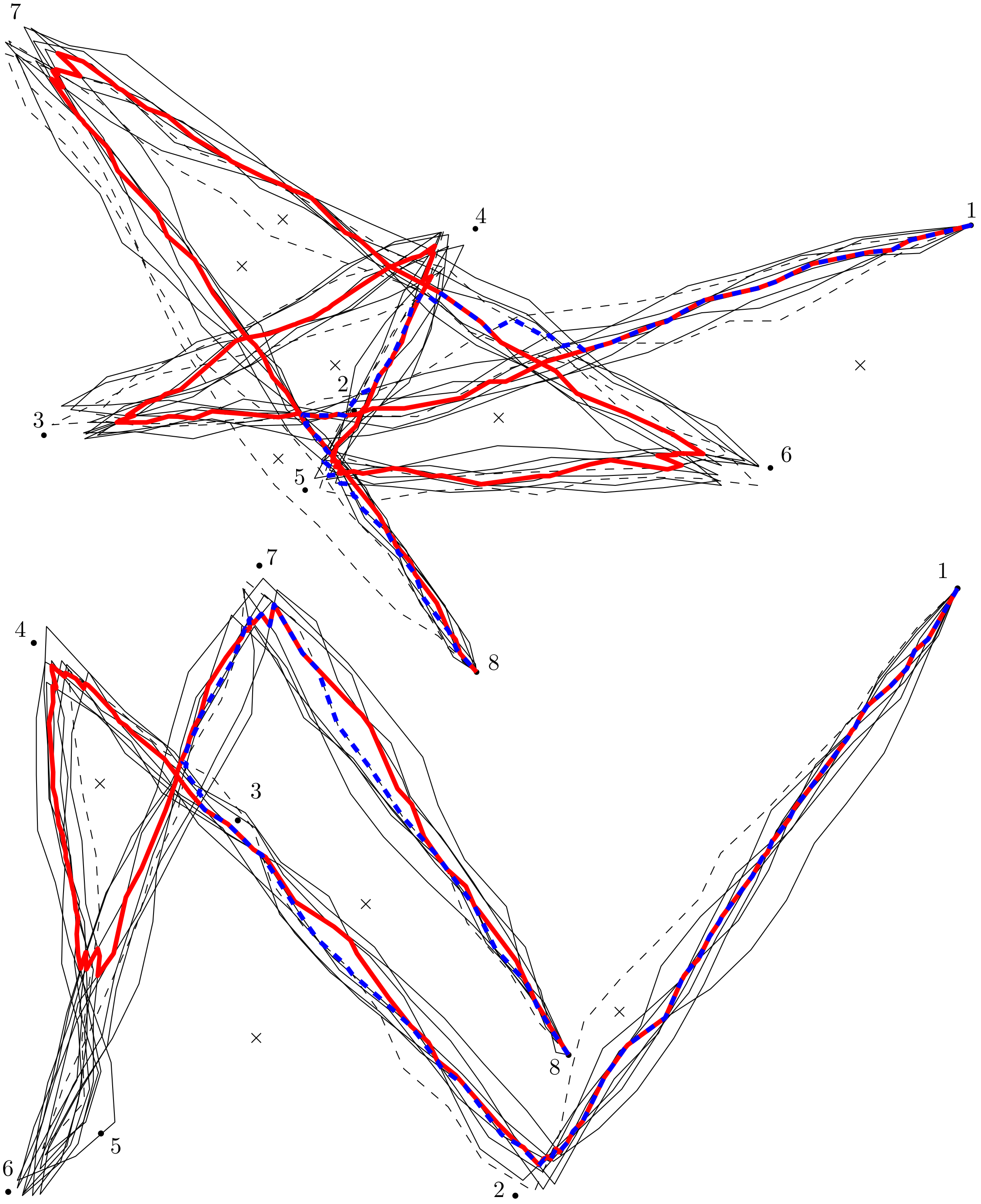

Fig. 6: Two example sets of trajectories generated for testing. Waypoints are shown numbered. The solid bold polyline is the homotopic median, whereas the dotted bold polyline is the simple median. Trajectories not in the largest homotopic subset are shown dashed. Poles are shown as crosses.