Central Trajectories

Marc van Kreveld* Maarten Löffler* Frank Staals*

Abstract

We study the problem of computing a suitable representative of a set of similar trajectories. To this end we define a *central trajectory* $\mathcal{C}$, which consists of pieces of the input trajectories, switches from one entity to another only if they are within a small distance of each other, and such that at any time t, the point $\mathcal{C}(t)$ is as central as possible. We measure centrality in terms of the radius of the smallest disk centered at $\mathcal{C}(t)$ enclosing all entities at time t, and discuss how the techniques can be adapted to other measures of centrality. For entities moving in $\mathbb{R}^1$ we show that an optimal central trajectory $\mathcal{C}$ representing n trajectories, each consisting of τ edges, has complexity $\Theta(\tau n^2)$ and can be computed in $O(\tau n^2 \log n)$ time. For entities moving in $\mathbb{R}^d$ with $d \geq 2$, the complexity of $\mathcal{C}$ is at most $O(\tau n^{5/2})$ and can be computed in $O(\tau n^3)$ time.

1 Introduction

A *trajectory* is a sequence of time-stamped locations in the plane, or more generally in $\mathbb{R}^d$. Trajectory data is obtained by tracking the movements of e.g. animals [1, 4, 6], hurricanes [8], traffic [7], or other moving entities [5] over time. Large amounts of such data have recently been collected in a variety of research fields. As a result, there is a great demand for tools and techniques to analyze trajectory data.

We study representing a set of (similar) trajectories by a single *representative* trajectory that captures the defining features of all trajectories in the set. Representative trajectories are useful for example in clustering. When choosing a representative trajectory for a group of similar trajectories, the first obvious choice would be to pick one of the trajectories in the group. However, one can argue that no single element in a group may be a good representative, e.g. because each individual trajectory has some prominent feature that is not shared by the rest (see Fig. 1(a)), or no trajectory is sufficiently in the middle all the time. On the other hand, it is desirable to output a trajectory that does consist of *pieces* of input trajectories, because otherwise the representative trajectory may display behaviour that is not present in the input, e.g. because of contextual information that is not available to the algorithm (see Fig. 1(b)).

*Department of Information and Computing Sciences, Universiteit Utrecht, The Netherlands, {m.j.vankreveld|m.loffler|f.staals}@uu.nl. M.L. and F.S. are supported by the Netherlands Organisation for Scientific Research (NWO) under grant 639.021.123 and 612.001.022, respectively.

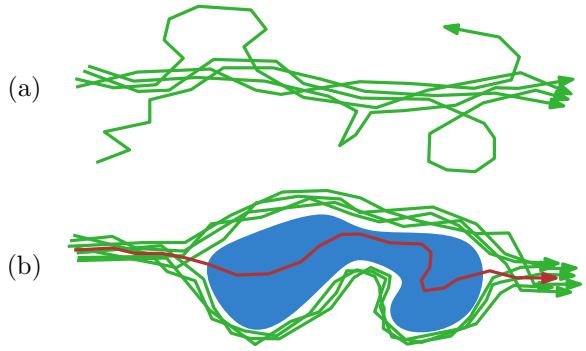

Figure 1: (a) Every trajectory has a peculiarity that is not representative for the set. (b) Taking the pointwise average of a set of trajectories may result in one that ignores context.

Central trajectories. Buchin et al. [2] consider the problem of computing a *median* trajectory for a set of trajectories without time information. Their method considers the trajectories as curves in the plane, and produces a trajectory (curve) that consists of pieces of the input. In this work, we focus on incorporating time into the representative. Ideally, we would output a trajectory $\mathcal{C}$ such that at any time t, $\mathcal{C}(t)$ is the point (entity) that is closest to its farthest entity. Unfortunately, when the entities move in $\mathbb{R}^d$ for $d > 1$, this may cause discontinuities. Such discontinuities are unavoidable: if we insist that the output trajectory consists of pieces of input trajectories *and* is continuous, then in general, there will be no opportunities to switch from one trajectory to another, and we are effectively choosing one of the input trajectories again. At the same time, we do not want to output a trajectory with arbitrarily large discontinuities. An acceptable compromise is to allow discontinuities, or *jumps*, but only over small distances, controlled by a parameter ε. We note that this problem of discontinuities also shows up for representatives without time information and entities moving in $\mathbb{R}^d$, with $d \geq 3$, because the traversed curves generally do not intersect.

Problem description. We are given a set $\mathcal{X}$ of n entities, each moving along a piecewise linear trajectory in $\mathbb{R}^d$ consisting of τ edges. We assume that all trajectories have their vertices at the same times $t_0, .., t_\tau$. For an entity σ, let $\sigma(t)$ denote the position of σ at time t. With slight abuse of notation we will write σ for both entity σ and its trajectory. At a given time t, we denote the distance from σ to the entity farthest away from σ by $D_\sigma(t) = D(\sigma, t) = \max_{\psi \in \mathcal{X}} \|\sigma(t)\psi(t)\|$, where $\|pq\|$ denotes the Euclidean distance between points p and q in $\mathbb{R}^d$.

For ease of exposition, we assume that the trajectories are in general position: that is, no three trajectories intersect in the same point, and no two pairs of entities are at distance ε from each other at the same time.

A *trajectoid* is a function that maps time to the set of entities $\mathcal{X}$, with the restriction that at discontinuities the distance between the entities involved is at most ε. Intuitively, a trajectoid corresponds to a concatenation of pieces of the input trajectories in such a way that two consecutive pieces match up in time, and the end point of the former piece is within distance ε from the start point of the latter piece. More formally, for a trajectoid $\mathcal{T}$ we have that

- at any time t, $\mathcal{T}(t) = \sigma$ for some $\sigma \in \mathcal{X}$, and
- at every time t where $\mathcal{T}$ has a discontinuity, that is, $\mathcal{T}$ *jumps* from entity σ to entity ψ, we have that $\|\sigma(t)\psi(t)\| \leq \varepsilon$.

Note that this definition still allows for a series of jumps within an arbitrarily short time interval $[t, t+\delta]$, essentially simulating a jump over distances larger than ε. To make the formulation cleaner, we slightly weaken the second condition, and allow a trajectoid to have discontinuities with a distance larger than ε, provided that such a large jump can be realized by a sequence of small jumps, each of distance at most ε. When it is clear from the context, we will write $\mathcal{T}(t)$ instead of $\mathcal{T}(t)(t)$ to mean the location of entity $\mathcal{T}(t)$ at time t. We now wish to compute a trajectoid $\mathcal{C}$ that minimizes the function

$$\mathcal{D}(\mathcal{T}) = \int_{t_0}^{t_\tau} D(\mathcal{T}, t)\,\mathrm{d}t.$$

So, at any time t, all entities lie in a disk of radius $D(\mathcal{C}, t)$ centered at $\mathcal{C}(t)$.

Results. Because space restrictions, we present only the situation where entities move in $\mathbb{R}^1$. Our approach can be extended to $\mathbb{R}^d$, as well as other measures of centrality. For these results and all omitted proofs we refer to the full version of this paper [9]. We show that the worst case complexity of a central trajectory in $\mathbb{R}^1$ is $\Theta(\tau n^2)$, and that we can compute one in $O(\tau n^2 \log n)$ time. For entities moving in $\mathbb{R}^d$, for any constant d, the maximal complexity of a central trajectory $\mathcal{C}$ is $O(\tau n^{5/2})$. In this case, computing $\mathcal{C}$ takes $O(\tau n^3)$ time and requires $O(\tau n^2 \log n)$ space.

2 Preliminaries

Let $\mathcal{X}$ be the set of entities moving in $\mathbb{R}^1$. The trajectories of these entities can be seen as polylines in $\mathbb{R}^2$: we associate time with the horizontal axis, and $\mathbb{R}^1$ with the vertical axis (see Fig. 2). We observe that the distance between two points p and q in $\mathbb{R}^1$ is simply their absolute difference, that is, $\|pq\| = |p - q|$.

Let $\mathcal{I}$ be the *ideal* trajectory, that is, the trajectory that minimizes $\mathcal{D}$ but is not restricted to lie on the input trajectories. It follows that at any time t, $\mathcal{I}(t)$ is simply the average of the highest entity $\mathcal{U}(t)$ and the lowest entity $\mathcal{L}(t)$. We further subdivide each time interval $J_i = [t_i, t_{i+1}]$ into *elementary intervals*, such that $\mathcal{I}$ is a single line segment inside each elementary interval.

Lemma 1 *The total number of elementary intervals is* $\tau(n+2)$.

We assume without loss of generality that within each elementary interval $\mathcal{I}$ coincides with the x-axis. To simplify the description of the proofs and algorithms, we also assume that the entities never move parallel to the ideal trajectory, that is, there are no horizontal edges.

Lemma 2 *$\mathcal{C}$ is a central trajectory in $\mathbb{R}^1$ if and only if it minimizes the function*

$$\mathcal{D}'(\mathcal{T}) = \int_{t_0}^{t_\tau} |\mathcal{T}(t)|\,\mathrm{d}t.$$

By Lemma 2 a central trajectory $\mathcal{C}$ is a trajectoid that minimizes the area $\mathcal{D}'(\mathcal{T})$ between $\mathcal{T}$ and the ideal trajectory $\mathcal{I}$. Hence, we can focus on finding a trajectoid that minimizes $\mathcal{D}'$.

3 Complexity of a Central Trajectory

Lemma 3 *For a set of n trajectories in $\mathbb{R}^1$, each with vertices at times $t_0, .., t_\tau$, a central trajectory $\mathcal{C}$ may have worst case complexity $\Omega(\tau n^2)$.*

Two entities σ and ψ are *ε-connected* at time t if there is a sequence $\sigma = \sigma_0, .., \sigma_k = \psi$ of entities such that for all i, σ_i and σ_{i+1} are within distance ε of each other at time t. A subset $\mathcal{X}' \subseteq \mathcal{X}$ of entities is ε-connected at time t if all entities in $\mathcal{X}'$ are pairwise ε-connected at time t. The set $\mathcal{X}'$ is ε-connected during an interval I, if they are ε-connected at any time $t \in I$. We now observe:

Observation 1 *$\mathcal{C}$ can jump from entity σ to ψ at time t if and only if σ and ψ are ε-connected at time t.*

At any time t, we can partition $\mathcal{X}$ into maximal sets of ε-connected entities. The central trajectory $\mathcal{C}$ must be in one of such maximal sets $\mathcal{X}'$: it uses the trajectory of an entity $\sigma \in \mathcal{X}'$ (at time t), if and only if σ is the entity from $\mathcal{X}'$ closest to $\mathcal{I}$. More formally, let $f_\sigma(t) = |\sigma(t)|$, and let $\mathcal{L}(\mathcal{F}) = \min_{f \in \mathcal{F}} f$ denote the lower envelope of a set of functions $\mathcal{F}$.

Observation 2 *Let $\mathcal{X}' \ni \sigma$ be a maximal set of entities that is ε-connected during interval J, and assume that $\mathcal{C} \in \mathcal{X}'$ during J. For any time $t \in J$, we have that $\mathcal{C}(t) = \sigma(t)$ if and only if f_σ is on the lower envelope of the set $\mathcal{F}' = \{f_\psi \mid \psi \in \mathcal{X}'\}$ at time t, that is, $f_\sigma(t) = \mathcal{L}(\mathcal{F})(t)$.*

Let $\mathcal{X}_1, .., \mathcal{X}_m$, denote a collection of maximal sets of entities that are ε-connected during time intervals $J_1, .., J_m$, respectively. Let $\mathcal{F}_i = \{f_\sigma \mid \sigma \in \mathcal{X}_i\}$, and let $\mathcal{L}_i$ be the lower envelope $\mathcal{L}(\mathcal{F}_i)$ of $\mathcal{F}_i$ restricted to interval J_i. A

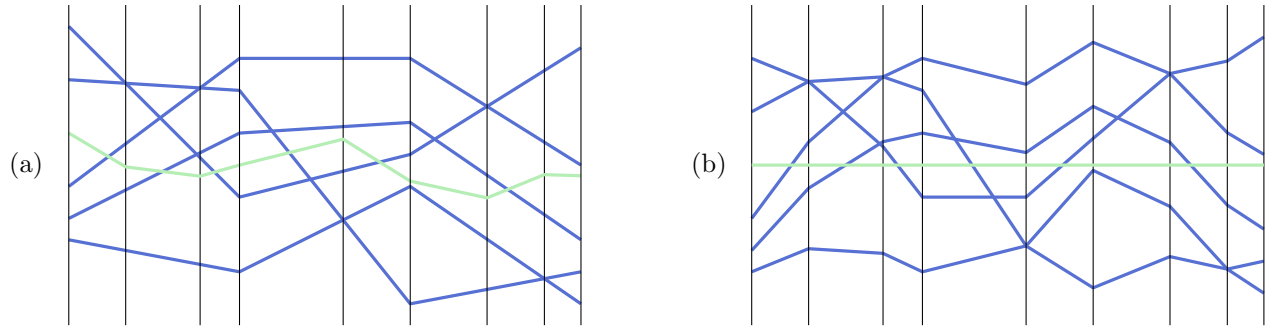

Figure 2: (a) A set of trajectories and the ideal trajectory I. The breakpoints in the ideal trajectory partition time into $O(n\tau)$ intervals. (b) The trajectories after transforming I into a horizontal line.

lower envelope $\mathcal{L}_i$ has a break point at time t if $f_\sigma(t) = f_\psi(t)$, for $\sigma, \psi \in \mathcal{X}_i$. There are two types of break points: (i) $\sigma(t) = \psi(t)$, or (ii) $\sigma(t) = -\psi(t)$. At events of type (i) the modified trajectories of σ and ψ intersect. At events of the type (ii), σ and ψ are equally far from I, but on different sides of I. Let $B = \{(t, \sigma, \psi) \mid \mathcal{L}_i(t) = f_\sigma(t) = f_\psi(t) \wedge i \in \{1, .., m\}\}$ denote the collection of break points from all lower envelopes $\mathcal{L}_1, .., \mathcal{L}_m$.

Lemma 4 *Consider a triplet $(t, \sigma, \psi) \in B$. There is at most one lower envelope $\mathcal{L}_i$ such that t is a break point in $\mathcal{L}_i$.*

Proof. Assume by contradiction that t is a break point in both $\mathcal{L}_i$ and $\mathcal{L}_j$. At any time t, an entity can be in at most one maximal set $\mathcal{X}_\ell$. So if $\mathcal{X}_i$ and $\mathcal{X}_j$ share either entity σ or ψ, then the intervals J_i and J_j are disjoint. It follows t cannot lie in both intervals, and thus cannot be a break point in both $\mathcal{L}_i$ and $\mathcal{L}_j$. Contradiction. □

Lemma 5 *Let $\mathcal{A}$ be an arrangement of n lines, describing the movement of n entities during an elementary interval J. If there is a break point $(t, \sigma, \psi) \in B$, with $t \in J$, of type (ii), then $\sigma(t)$ and $\psi(t)$ lie on the boundary $\partial\mathcal{Z}$ of the zone $\mathcal{Z}$ of I in $\mathcal{A}$.*

Lemma 6 *Let $\mathcal{A}$ be an arrangement of n lines, describing the movement of n entities during an elementary interval J. The total number of break points $(t, \sigma, \phi) \in B$, with $t \in J$, of type (ii) is at most $6.5n$.*

Lemma 7 *The total complexity of all lower envelopes $\mathcal{L}_1, .., \mathcal{L}_m$ on $[t_i, t_{i+1}]$ is $O(n^2)$.*

Theorem 8 *Given a set of n trajectories in $\mathbb{R}^1$, each with vertices at times $t_0, .., t_\tau$, a central trajectory $\mathcal{C}$ has worst case complexity $O(\tau n^2)$.*

Proof. A central trajectory $\mathcal{C}$ is a piecewise function. From Observations 1 and 2 it now follows that $\mathcal{C}$ has a break point at time t only if (a) two subsets of entities become ε-connected or ε-disconnected, or (b) the lower envelope of a set of ε-connected entities has a break point at time t. Within a single time interval $J_i = [t_i, t_{i+1}]$ there are at most $O(n^2)$ times when two entities are at distance exactly ε. Hence, the number of events of type (a) during interval J_i is also $O(n^2)$. By Lemma 7 the total complexity of all lower envelopes of ε-connected sets during J_i is also $O(n^2)$. Hence, the number of break points of type (b) within interval J_i is also $O(n^2)$. The theorem follows. □

4 Computing a Central Trajectory

We now present an algorithm to compute a trajectoid $\mathcal{C}$ minimizing $\mathcal{D}'$. By Lemma 2 such a trajectoid is a central trajectory. The basic idea is to construct a weighted (directed acyclic) graph that represents a set of trajectoids containing an optimal trajectoid. We can then find $\mathcal{C}$ by computing a minimum weight path in this graph.

The graph that we use is a weighted version of the Reeb graph that Buchin et al. [3] use to model the trajectory grouping structure. We review their definition here. The *Reeb graph* $\mathcal{R}$ is a directed acyclic graph. Each edge $e = (u, v)$ of $\mathcal{R}$ corresponds to a maximal subset of entities $C_e \subseteq \mathcal{X}$ that is ε-connected during the time interval $[t_u, t_v]$. The vertices represent times at which the sets of ε-connected entities change, that is, the times at which two entities σ and ψ are at distance ε from each other and the set containing σ merges with or splits from the set containing ψ. See Fig. 3 for an illustration.

By Observation 1 a central trajectory $\mathcal{C}$ can jump from σ to ψ if and only if σ and ψ are ε-connected, that is, if σ and ψ are in the same component C_e of edge e. From Observation 2 it follows that on each edge e, $\mathcal{C}$ uses only the trajectories of entities σ for which f_σ occurs on the lower envelope of the functions $\mathcal{F}_e = \{f_\sigma \mid \sigma \in C_e\}$. Hence, we can then express the cost for $\mathcal{C}$ using edge e by

$$\omega_e = \int_{t_u}^{t_v} \mathcal{L}(\mathcal{F}_e)(t)\,\mathrm{d}t.$$

It now follows that $\mathcal{C}$ follows a path in the Reeb graph $\mathcal{R}$, that is, the set of trajectoids represented by $\mathcal{R}$ contains a trajectoid minimizing $\mathcal{D}'$. So we can compute a central trajectory by finding a minimum weight path in $\mathcal{R}$ from a source to a sink.

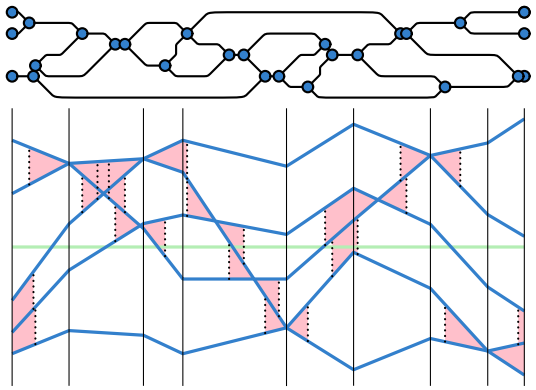

Figure 3: The Reeb graph for a set of moving entities. The dashed lines indicate that two entities are at distance ε.

Analysis. First we compute the Reeb graph as defined by Buchin et al. [3]. This takes $O(\tau n^2 \log n)$ time. Second we compute the weight ω_e for each edge e. The Reeb graph $\mathcal{R}$ is a DAG, so once we have the edge weights, we can use dynamic programming to compute a minimum weight path in $O(|\mathcal{R}|) = O(\tau n^2)$ time. So all that remains is to compute the edge weights ω_e. For this, we need the lower envelope $\mathcal{L}_e$ of each set $\mathcal{F}_e$ on the interval J_e. To compute the lower envelopes, we need the ideal trajectory I, which we can compute I in $O(\tau n \log n)$ time by computing the lower and upper envelope of the trajectories in each time interval $[t_i, t_{i+1}]$.

Lemma 7 implies that the total complexity of all lower envelopes is $O(\tau n^2)$. To compute them we have two options. We can simply compute the lower envelope from scratch for every edge of $\mathcal{R}$. This takes $O(\tau n^2 \cdot n \log n) = O(\tau n^3 \log n)$ time. Instead, for each time interval $J_i = [t_i, t_{i+1}]$, we compute the arrangement $\mathcal{A}$ representing the modified trajectories on the interval J_i, and use it to trace $\mathcal{L}_e$ in $\mathcal{A}$ for every edge e of $\mathcal{R}$.

Using a standard sweep line algorithm, an arrangement of m line segments can be built in $O((m + A) \log m)$ time, where A is the output complexity. We have $O(n^2)$ line segments: $n + 2$ per entity. Since each pair of trajectories intersects at most once during J_i, we have $A = O(n^2)$. Thus, we build $\mathcal{A}$ in $O(n^2 \log n)$ time. The arrangement represents all break points of type (i), of all functions f_σ. Next, we compute all pairs of points in $\mathcal{A}$ corresponding to break points of type (ii). We do this in $O(n^2)$ time by traversing the zone of I in $\mathcal{A}$.

We now trace the lower envelopes through $\mathcal{A}$: for each edge $e = (u, v)$ in the Reeb graph with $J_e \subseteq J_i$, we start at the point $\sigma(t_u)$, $\sigma \in C_e$, that is closest to I, and then follow the edges in $\mathcal{A}$ corresponding to $\mathcal{L}_e$, taking care to jump when we encounter break points of type (ii). Our lower envelopes are all disjoint (except at endpoints), so we traverse each edge in $\mathcal{A}$ at most once. The same holds for the jumps. We can avoid costs for searching for the starting point of each lower envelope by tracing the lower envelopes in the right order: when we are done tracing $\mathcal{L}_e$, with $e = (u, v)$, we continue with the lower envelope of an outgoing edge of vertex v. If v is a split vertex where σ and ψ are at distance ε, then the starting point of the lower envelope of the other edge is either $\sigma(t_v)$ or $\psi(t_v)$, depending on which of the two is farthest from I. It follows that when we have $\mathcal{A}$ and the list of break points of type (ii), we can compute all lower envelopes in $O(n^2)$ time. We conclude:

Theorem 9 *Given a set of n trajectories in $\mathbb{R}^1$, each with vertices at times $t_0, .., t_\tau$, we can compute a central trajectory $\mathcal{C}$ in $O(\tau n^2 \log n)$ time using $O(\tau n^2)$ space.*

5 Entities Moving in $\mathbb{R}^d$

For entities moving in $\mathbb{R}^1$ we used that computing a central trajectory was equivalent to finding a trajectoid that minimizes the distance to the ideal trajectory. In $\mathbb{R}^d$, with $d > 1$, however, this is no longer true. Instead, we directly use the functions D_σ expressing the distance between an entity σ and the entity furthest away from σ. We can then still use Observations 1 and 2 to bound the complexity of $\mathcal{C}$ by $O(\tau n^{5/2})$. An algorithm similar to that of Section 4 that runs in $O(\tau n^3)$ time can then be used to compute a central trajectory. The details can be found in the full version [9].

References

[1] P. Bovet and S. Benhamou. Spatial analysis of animals' movements using a correlated random walk model. *J. Theoretical Biology*, 131(4):419–433, 1988.

[2] K. Buchin, M. Buchin, M. van Kreveld, M. Löffler, R. I. Silveira, C. Wenk, and L. Wiratma. Median trajectories. *Algorithmica*, 66(3):595–614, 2013.

[3] K. Buchin, M. Buchin, M. van Kreveld, B. Speckmann, and F. Staals. Trajectory grouping structure. In *Proc. 2013 WADS Algorithms and Data Structures Symposium*, volume 8037 of *LNCS*, pages 219–230. Springer, 2013.

[4] C. Calenge, S. Dray, and M. Royer-Carenzi. The concept of animals' trajectories from a data analysis perspective. *Ecological Informatics*, 4(1):34 – 41, 2009.

[5] S. Dodge, R. Weibel, and E. Forootan. Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects. *Computers, Environment and Urban Systems*, 33(6):419–434, 2009.

[6] E. Gurarie, R. D. Andrews, and K. L. Laidre. A novel method for identifying behavioural changes in animal movement data. *Ecology Letters*, 12(5):395–408, 2009.

[7] X. Li, X. Li, D. Tang, and X. Xu. Deriving features of traffic flow around an intersection from trajectories of vehicles. In *Proc. 18th Int. Conf. on Geoinf.*, pages 1–5. IEEE, 2010.

[8] A. Stohl. Computation, accuracy and applications of trajectories – a review and bibliography. *Atmospheric Environment*, 32(6):947 – 966, 1998.

[9] M. van Kreveld, M. Löffler, and F. Staals. Central trajectories. *CoRR*, abs/1501.01822, 2015.