A Refined Definition for Groups of Moving Entities and its Computation*

Marc van Kreveld† Maarten Löffler† Frank Staals‡ Lionov Wiratma†§

Abstract

We propose a refined definition of a group of moving entities which corresponds better to human intuition. We also present algorithms to compute all maximal groups from a set of moving entities.

1 Introduction

Nowadays, inexpensive modern devices with advanced tracking technologies make it easy to track movements of an entity. This has led to the availability of movement data for various types of moving entities (human, animals, vehicles, etc.). Since a tracking device typically returns a single location at each time stamp, each moving entity will be represented as a moving point. The data may consist of just one trajectory tracked over a period, or a whole collection of trajectories that are all tracked over a period. It is common to denote the number of trajectories (or moving entities) by n and the number of time stamps used for each by τ. Hence, the input size is $\Theta(\tau n)$.

To analyze moving object data, a number of methods have been developed in recent times. These methods may concern similarity analysis, clustering, outlier detection, segmentation, and various patterns that may emerge from the movement of the entities (for surveys see [3, 14]). These methods are often based on geometric algorithms, because the data is essentially spatial.

One particular type of pattern that has been well-studied is flocking [1, 4, 5]. Intuitively, a flock is a subset of the entities moving together (or simply being together) over a period of time. Other names for this and closely related concepts with slightly different definitions are herds [6], convoys [8], moving clusters [9], mobile groups [7], swarms [11], and groups [2]. The last of these defines a group in a simple and formal way. In [2] a model is introduced called the *trajectory grouping structure* which not only defines groups, but also the splitting of a group into subgroups and its opposite, merging. The algorithmic problem of reporting all maximal groups that occur in the trajectories is solved in $O(\tau n^3 + N)$ time, where $N = O(\tau n^4)$ is the output size (the summed size of all groups reported). The algorithm also considers times in between the τ time stamps where the locations are recorded as relevant. In between these time stamps, locations are inferred by linear interpolation over time.

In this paper we continue the study of such groups, but we propose a refined definition to the one in [2]. We motivate why it captures our intuition better and present algorithms to compute all maximal groups.

2 Problem Description

The definition of a group by Buchin et al. [2] relies on three parameters: one for distance between entities, one for the duration of a group, and one for the size of a group. We review their definitions next.

For a set of moving entities $\mathcal{X}$, two entities x and y are *directly ε-connected* at time t if the Euclidean distance between x and y is at most ε at time t, for some given $\varepsilon \geq 0$. Two entities x and y are *ε-connected in $\mathcal{X}$* if there is a sequence $x = x_0, ..., x_k = y$, with $\{x_0, ..., x_k\} \subseteq \mathcal{X}$ and for all i, x_i and x_{i+1} are directly ε-connected at the same time t.

In [2], a *group* for an entity inter-distance ε, a minimum required duration δ, and a minimum required size m, is defined as a subset $G \subseteq \mathcal{X}$ and corresponding time interval I for which three conditions hold:

(*i*) G contains at least m entities.
(*ii*) I has a duration at least δ.
(*iii*) Every two entities $x, y \in G$ are ε-connected in $\mathcal{X}$ during I.

Furthermore, a group G with time interval I is *maximal* if there is no time interval I' properly containing I for which G is also a group, and there is no proper supergroup G' of G that is also a group during I [2].

One issue with this definition is that it does not correspond fully to our intuition. Two entities x and y may form a maximal group in an interval I even if they are always far apart, as long as there are always entities of $\mathcal{X}$ in between to make x and y ε-connected in $\mathcal{X}$. This can have counter-intuitive effects especially in dense crowds. To avoid such issues, we refine the definition of a group. In particular, we replace condition (*iii*) above by:

(*iii'*) Every two entities $x, y \in G$ are ε-connected in G during I.

*M.L. supported by the Netherlands Organisation for Scientific Research (NWO, grant 639.021.123). F.S. supported by the Danish National Research Foundation (grant nr. DNRF84). L.W. supported by the Directorate General of Resources for Science, Technology and Higher Education of Indonesia.

†Dept. of Information and Computing Sciences, Utrecht University, {m.j.vankreveld,m.loffler,l.wiratma}@uu.nl
‡MADALGO, Aarhus University, f.staals@cs.au.dk
§Dept. of Informatics, Parahyangan Catholic University

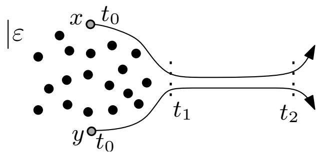

Figure 1: In the definition by [2], x and y are ε-connected during $[t_0, t_2]$

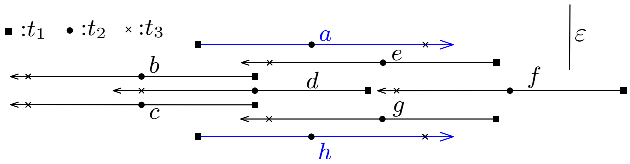

Figure 2: Entities in $G = \{a, h\}$ are ε-connected using entities not in G.

We define maximal groups in the same way as before.

We give two examples that show the difference in these definitions. First, consider a number of stationary entities S and two entities x and y, see Figure 1.

Entity x starts (t_0) to the North of S and moves around its perimeter to the East. Entity y starts (t_0) to the South and also moves around the perimeter to the East. After encountering (t_1) each other at the East side, both continue together eastward, away from the stationary entities in S (ending at t_2).

By the definition in [2], x and y form a maximal group in the interval $[t_0, t_2]$. In our refined definition, they form a maximal group during $[t_1, t_2]$, starting when x and y actually encounter each other.

Second, the previous definition can even see groups of entities that were never close, see Figure 2. Here, $\{a, h\}$ is a maximal group in the interval $I = [t_1, t_3]$ using the definition in [2]. At each time, a and h are ε-connected, but through different subsets of entities. Although a and h move in the same direction with the same speed, intuitively they do not form a group because they are too far apart and separated by other entities that move in the opposite direction. With the new, refined definition, we do not consider $\{a, h\}$ a group in the interval I.

Results and Organization. In this paper, we show that for a set $\mathcal{X}$ of n moving entities in $\mathbb{R}^1$ with τ time stamps each, the number of maximal groups by the refined definition is $O(\tau n^3)$, which is tight in the worst case. We present algorithms to compute all maximal groups, beginning with a basic algorithm that runs in $O(\tau^3 n^6)$ time. Subsequent improvements lead to a running time of $O(\tau^2 n^4)$. For moving entities in $\mathbb{R}^d$ ($d > 1$), we show that all maximal groups can be computed in $O(\tau^2 n^5 \log n)$ time. From now on, we will use the term "group" to denote a group of entities that comply with our refined definition.

3 Preliminaries

Let $\mathcal{X}$ be a set of n entities moving in $\mathbb{R}^1$, given by locations at τ time stamps. A trajectory of an entity in $\mathcal{X}$ can be expressed by a piecewise-linear function which maps time to a point in $\mathbb{R}^1$. If $\mathbb{R}^1$ is associated with the vertical axis and time with the horizontal axis of a 2-dimensional plane, the trajectories of entities in $\mathcal{X}$ are polylines with τ vertices each. We will use the same notation to denote an entity and its trajectory. We assume that all trajectories have their vertices at the same times and that there are no two parallel edges.

Let $d_{ij}(t)$ be the Euclidean distance between $i \in \mathcal{X}$ and $j \in \mathcal{X}$ at time t. When $d_{ij}(t) = \varepsilon$, we say that an *ε-event* occurs. For any ε-event v, we denote by t_v the time when v occurs and $\omega(v)$ the function that returns the two entities that create v. We assume that no two or more ε-events occur at the same time.

Consider an ε-event v; let $\omega(v) = \{i, j\}$. If i and j are further than ε immediately before t_v, then v is a *start ε-event*; if they are further immediately after t_v it is an *end ε-event*. If there is no entity $k \in \mathcal{X}$ located strictly in between i and j at t_v (so $d_{ik}(t_v) + d_{jk}(t_v) = \varepsilon$), then we say that v is a *free ε-event*.

Observation 1 *The number of ε-events is $O(\tau n^2)$.*

Let G be a group of entities in time interval I that is maximal in size. All entities in G are pairwise ε-connected in the interval I, and hence, there are no free ε-events in G during I. In the arrangement of trajectories from G, no face has height greater than ε.

It is also clear that G can begin only at a start ε-event and end only at an end ε-event. Furthermore, we observe that if a start ε-event (or end ε-event) of G is not a free ε-event with respect to the entities in G, then before (or after) the interval I, entities in G are still pairwise ε-connected and we can extend the interval of G. Therefore, G can be a maximal group only if both the start ε-event and end ε-event are free ε-events (but this is not a sufficient condition).

Observation 2 *There can be at most one maximal group that starts and ends at a particular pair of start ε-event and end ε-event.*

Theorem 1 *For a set $\mathcal{X}$ of n entities, each entity moving along a piecewise-linear trajectory of τ edges, the maximum number of maximal group is $\Theta(\tau n^3)$.*

Proof. Any group G that starts at a start ε-event contains at most n entities. When a free end ε-event involving G occurs, only group G ends but a subgroup of G with fewer entites may continue. This can happen at most $n - 1$ times. Therefore, the number of maximal groups is $O(\tau n^3)$. Furthermore, there can be $\Omega(\tau n^3)$ maximal groups because the lower bound for the definition of a group in [2] still applies [13]. ☐

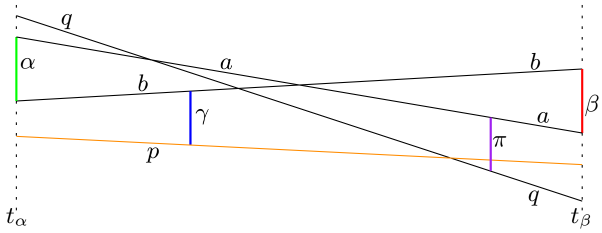

Figure 3: Removing trajectory p (and γ) introduces a new free ε-event: π.

The approach to compute all maximal groups is to work on the arrangement $\mathcal{A}$ of line segments that are the trajectories. For a subset $G \subseteq \mathcal{X}$ and interval I, we can remove entities from G that are separated at a face with height larger than ε in I (corresponding to a free ε-event). Only if there are no such faces, the remaining entities in G can be a group. Note that removing entities in G involves removing the corresponding trajectories from the arrangement $\mathcal{A}$, which can cause new faces that are free ε-events.

4 Basic Algorithm

Next, we describe a simple algorithm to compute all maximal groups. Let ξ_s and ξ_e be the sets of all start ε-events and all end ε-events respectively. Fix $\alpha \in \xi_s$ and $\beta \in \xi_e$. By Observation 2, there is only one maximal group G that starts at α and ends at β. Furthermore, observe that G necessarily contains the entities $\omega(\alpha) = \{a, b\}$ and $\omega(\beta) = \{c, d\}$, and that if G is a maximal group on $I = [t_\alpha, t_\beta]$, then all entities in G are on the same side at time $t_\gamma \in (t_\alpha, t_\beta)$ when a free ε-event γ occurs. We then use the following approach to find G (if it exists):

1. Initialize a set G containing all entities in $\mathcal{X}$.
2. Build an arrangement $\mathcal{A}$ induced by the trajectories of the entities in G in the interval I.
3. A face f in $\mathcal{A}$ contains a free ε-event γ if (and only if) the height of f is more than ε. If f has height larger than ε, test if (the trajectories of) a, b, c, and d, all lie on the same side of f. If not, there is no maximal group G that starts at α and ends at β. If they do pass on the same side, let S denote the set of entities whose trajectories lie on the other side of f. Remove these entities from S, and remove their trajectories from $\mathcal{A}$. Observe that new free ε-events may appear because removal of a trajectory from $\mathcal{A}$ merges two faces of $\mathcal{A}$ into a larger one. See Figure 3. Repeat this step until there is no more free ε-events γ with $t_\gamma \in (t_\alpha, t_\beta)$.
4. Check that α and β are now free. If so, G is a maximal group on I, and hence we can report it. If not, G is actually a group during a time interval $I' \supset I$. Hence, G may be maximal in size, but not in duration. We do not report G in this case.

Theorem 2 *Given a set $\mathcal{X}$ of n entities in which each entity moves in $\mathbb{R}^1$ along a trajectory of τ edges, all maximal groups can be computed in $O(\tau^3 n^6)$ time using the Basic Algorithm.*

Proof. The number of combination of a pair of start and end ε-events is $O(\tau^2 n^4)$. Building an arrangement from trajectories of entities takes $O(\tau n^2)$ time. Removing a trajectory e and checking new faces in $\mathcal{A}$ takes time proportional to the zone complexity of e: $O(\tau n)$. Since there are at most n trajectories to be removed, the whole process to remove entities for each interval I takes $O(\tau n^2)$ time. Therefore, the running time of the algorithm is $O(\tau^3 n^6)$ time. □

5 Improved Algorithm

The previous algorithm checks every pair of possible start and end ε-events α and β to potentially find one maximal group. To improve the running time, we fix a start ε-event α and consider the $O(\tau n^2)$ end ε-events β in increasing order. We show that we can check for a maximal group on $[t_\alpha, t_\beta]$ in amortized $O(1)$ time.

We build the arrangement $\mathcal{A}$ for all trajectories, starting from time t_α, and sort the end ε-events β, with $t_\beta > t_\alpha$ on increasing time. We then consider the end ε-events β in this order, while maintaining a maximal set G that is ε-connected in G throughout the time interval $[t_\alpha, t_\beta]$.

Let $\omega(\alpha) = \{a, b\}$ be the entities defining the start ε-event α, and let $G \supseteq \{a, b\}$ be the largest ε-connected set on $[t_\alpha, t_\beta]$. We compute the largest ε-connected set on $[t_\alpha, t_{\beta'}]$ for the next ending event β' as follows. Note that this set will be a subset of G.

Let S be the set of entities that separate from a and b at β. We remove all trajectories from the entities in S from $\mathcal{A}$. As before, this may introduce faces of height larger than ε. For every such face f, we check if a and b still pass f on the same side. If not, there can be no maximal groups that contain a and b, start at t_α, and end after t_β. If a and b lie on the same side of f, we add all entities that lie on the other side of f to S and remove their trajectories from $\mathcal{A}$. We repeat this until all faces in $\mathcal{A}$ that have non-empty intersection with the vertical strip defined by $[t_\alpha, t_{\beta'}]$ have height at most ε (or until we have found a face that splits a and b). It follows that the set $G' = G \setminus S$ is the largest set containing a and b that is ε-connected throughout $[t_\alpha, t_{\beta'}]$. If α and β' are free with respect to G' then we report G' as a maximal group.

Building the arrangement $\mathcal{A}$ takes $O(\tau n^2)$ time, and sorting the ending-events takes $O(\tau n^2 \log(\tau n))$ time. By the Zone Theorem, we can remove each trajectory in $O(\tau n)$ time. Checking the height of the new faces can be done in the same time bound. It follows that the total running time is $O(\tau n^2(\tau n^2 + \tau n^2 \log(\tau n) + R))$ where R is the total time for removing trajectories from the arrangement. Clearly, R is bounded by the complexity of the arrangement: $O(\tau n^2)$. So, the total running time is $O(\tau^2 n^4 \log(\tau n))$.

Further Improvement We can avoid repeated sorting of end ε-events by pre-sorting them in a list, and for each start ε-event, use this list. The list will contain events that do not concern the entities involved in the start ε-event, but this can be tested easily in constant time. Thus, we conclude:

Theorem 3 *Given a set $\mathcal{X}$ of n entities in which each entity moves in $\mathbb{R}^1$ along a trajectory of τ edges, all maximal groups can be computed in $O(\tau^2 n^4)$ time.*

6 Algorithms for Entities in $\mathbb{R}^d$

In $\mathbb{R}^d$ $(d > 1)$, it is harder to test whether an ε-event really connects or disconnects because the two entities may be ε-connected through other entities in the group. This observation immediately gives the condition for an ε-event to be *free*. We model our moving entities in a graph where vertices represent entities and an edge exists if two entities are directly ε-connected. As in Parsa [12], we can maintain the graph under edge updates, while allowing same component queries, in $O(\log n)$ time per operation.

To compute maximal groups, we start at a start ε-event α ($\omega(\alpha) = \{a, b\}$) and maintain the connected component $\mathcal{C}$ throughout the sequence of sorted ε-events. At each ε-event β, we remove any vertices that are disconnected from $\mathcal{C}$ and start again from α in case we remove anything. We stop if a and b are disconnected. If α is a free ε-event when we reach β again, we report $\mathcal{C}$ as a maximal group and continue.

We start at $O(\tau n^2)$ ε-events, process $O(\tau n^2)$ ε-events for each, and may need to restart up to $n - 1$ times. Hence, we obtain:

Theorem 4 *Given a set $\mathcal{X}$ of n entities move in $\mathbb{R}^d$ along a trajectory of τ edges, all maximal groups can be computed in $O(\tau^2 n^5 \log n)$ time.*

7 Conclusions and Future Work

In this paper we introduced a variation on the grouping structure definition [2] and argued that it corresponds better to our intuition. We have given an algorithm for trajectories moving in $\mathbb{R}^1$ that computes all maximal groups and runs in $O(\tau^2 n^4)$ time. In $\mathbb{R}^d$, our algorithm runs in $O(\tau^2 n^5 \log n)$ time. The number of maximal groups is $\Theta(\tau n^3)$ in the worst case.

The main challenges include reducing the dependency on τ to subquadratic, and the dependency on n. It would also be interesting to develop an output-sensitive algorithm that uses considerably less time if the output is small. Finally, we may be able to develop algorithms that take geodesic distance into account, like was done for the previous definition of a group [10].

References

[1] M. Benkert, J. Gudmundsson, F. Hübner, and T. Wolle. Reporting flock patterns. *Computational Geometry*, 41(3):111 – 125, 2008.

[2] K. Buchin, M. Buchin, M. van Kreveld, B. Speckmann, and F. Staals. Trajectory grouping structure. *Journal of Computational Geometry*, 6(1):75–98, 2015.

[3] J. Gudmundsson, P. Laube, and T. Wolle. Computational movement analysis. In *Handbook of Geographic Information*, pages 423–438. Springer, 2012.

[4] J. Gudmundsson and M. van Kreveld. Computing longest duration flocks in trajectory data. In *Proc. 14th ACM International Symposium on Advances in Geographic Information Systems*, GIS '06, pages 35–42, 2006.

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

[6] Y. Huang, C. Chen, and P. Dong. Modeling herds and their evolvements from trajectory data. In *Geographic Information Science*, volume 5266 of *LNCS*, pages 90–105. Springer, 2008.

[7] S. Hwang, Y. Liu, J. Chiu, and E. Lim. Mining mobile group patterns: A trajectory-based approach. In *Advances in Knowledge Discovery and Data Mining*, volume 3518 of *LNCS*, pages 145–146. Springer, 2005.

[8] H. Jeung, M. Yiu, X. Zhou, C. Jensen, and H. Shen. Discovery of convoys in trajectory databases. *PVLDB*, 1:1068–1080, 2008.

[9] P. Kalnis, N. Mamoulis, and S. Bakiras. On discovering moving clusters in spatio-temporal data. In *Advances in Spatial and Temporal Databases*, volume 3633 of *LNCS*, pages 364–381. Springer, 2005.

[10] I. Kostitsyna, M. van Kreveld, M. Löffler, B. Speckmann, and F. Staals. Trajectory grouping structure under geodesic distance. In *Proc. 31st International Symposium on Computational Geometry, SoCG 2015*, pages 674–688, 2015.

[11] Z. Li, B. Ding, J. Han, and R. Kays. Swarm: Mining relaxed temporal moving object clusters. *PVLDB*, 3(1):723–734, 2010.

[12] S. Parsa. A deterministic $O(m \log m)$ time algorithm for the Reeb graph. In *Proc. 28th Annual Symposium on Computational Geometry*, SoCG '12, pages 269–276, 2012.

[13] A. van Goethem, M. van Kreveld, M. Löffler, B. Speckmann, and F. Staals. Grouping time-varying data for interactive exploration. In *Proc. 32th Annual Symposium on Computational Geometry*, SoCG '16, 2016.

[14] Y. Zheng and X. Zhou. *Computing with Spatial Trajectories*. Springer, 2011.