Erschienen in: IEEE Transactions on Visualization and Computer Graphics ; 25 (2019), 1. - S. 76-86
https://dx.doi.org/10.1109/TVCG.2018.2865049

MotionRugs: Visualizing Collective Trends in Space and Time

Juri Buchmüller, Dominik Jäckle, Eren Cakmak, Ulrik Brandes, Daniel A. Keim

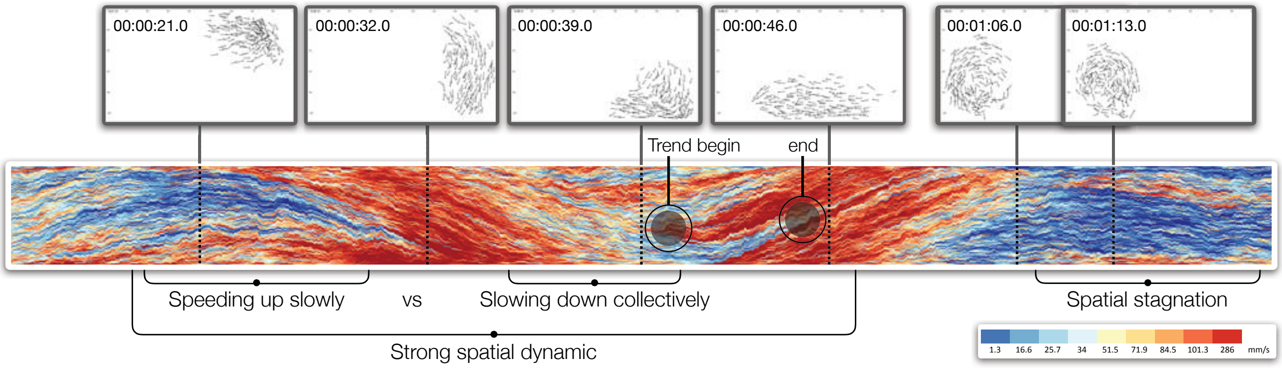


Fig. 1. Example of a MotionRug applied to swarm data comprising 151 golden shiner fish swimming through a water tank. Each frame of the data is represented as one vertical slice, and all slices are aligned sequentially on a time axis. The color encodes the speed of the fish, leading to this visual representation. We selected six frames that indicate the overall states and movement of the fish, as shown above the MotionRug. Below the MotionRug, we highlight key areas of the visualization: Overall, there is a *strong spatial dynamic* at the beginning. First, the fish are slow as they approach the right wall of the tank. Then, the fish *speed up slowly* and *slow down* when hitting the next wall. The fish repeat this behavior when approaching the wall on the left. Arrived, the fish show a *spatial stagnation*, meaning as a whole, the group remains in one place. Trends are typically indicated by tapered feature values.

Abstract—Understanding the movement patterns of collectives, such as flocks of birds or fish swarms, is an interesting open research question. The collectives are driven by mutual objectives or react to individual direction changes and external influence factors and stimuli. The challenge in visualizing collective movement data is to show space and time of hundreds of movements at the same time to enable the detection of spatiotemporal patterns. In this paper, we propose MotionRugs, a novel space efficient technique for visualizing moving groups of entities. Building upon established space-partitioning strategies, our approach reduces the spatial dimensions in each time step to a one-dimensional ordered representation of the individual entities. By design, MotionRugs provides an overlap-free, compact overview of the development of group movements over time and thus, enables analysts to visually identify and explore group-specific temporal patterns. We demonstrate the usefulness of our approach in the field of fish swarm analysis and report on initial feedback of domain experts from the field of collective behavior.

1 Introduction

Rapid advances in image analysis and small-scale GPS tracking devices allow us to capture groups of entities and their movement behavior at large scale: team sports games, bird flocks, fish swarms, or any other collective formed of multiple entities that mutually influence each other's movement. A challenging key task in understanding group behavior is to explore sudden changes in the spatial configuration and other factors that influence decision making, as for example Sumpter describes in his standard reference "Collective Animal Behavior" [43]. Visualization helps to explore the basic movement data evolving over time and track changes to get an understanding of the underlying group dynamics and behavior. In addition, visual analysis approaches can be used to provide an overview of large amounts of data and to reveal singular, preeminent behavior events at the same time.

Several visualization techniques have been developed to provide an overview for tracked entities that move over time. For example, static trajectory or space-time cube [30] visualization techniques show how entities move through space and time. However, occlusion issues increase with both the amount of considered entities and the length of the observed time period. An additional challenge is the concurrent comparison of features that are encoded on the trajectories by color or annotations, due to the problems mentioned before. Andrienko et al. state that static representations are "restricted to relatively simple trajectories and to a small number of moving objects for keeping the representation legible" [3, p.517]. In the field of collective behavior analysis, where the amount of entities to be observed is larger, these issues are critical when trying to find sudden structural changes in the observed behavior and causes for the same. Similar to static approaches, animation-based techniques have been found the "least effective form" for trend analysis [40] as they fail to provide a temporal overview that enables comparison between remote points in time. In contrast, small multiples [47] apply discretization to provide an overview. However, they additionally discretize continuous changes making it difficult to determine the right number of multiples. With *MotionRugs*, we aim to address these issues by providing a visual at-a-glance overview representation for groups of entities that evolve over time.

MotionRugs is a generic approach to compactly display changes in spatiotemporal data of multiple entities based on space partitioning techniques. The general idea is to transform the spatial representation of

- *Juri Buchmüller, Dominik Jäckle, Eren Cakmak, and Daniel A. Keim are with the University of Konstanz, Germany. E-mail: {juri.buchmueller, dominik.jaeckle, eren.cakmak, keim}@uni-konstanz.de*
- *Ulrik Brandes is with the ETH Zurich, Switzerland. E-Mail: ulrik.brandes@gess.ethz.ch*

all entities into a one-dimensional (1D) ordered slice for each moment in time separately. In each slice, the entities are presented as colored, vertically arranged rectangles. Color encodes a user-defined feature value such as speed, acceleration, or direction, for example. To enable comparison between different features, several MotionRugs can be grouped vertically, each showing a different feature (compare Figure 8). While the appearance of a MotionRug naturally is a direct consequence of the requirements we explain in Section 3, the technique also has some resemblance to van Gogh's painting style.

With MotionRugs, we present a static, space-efficient overview visualization for spatiotemporal data that enables the identification of trends and patterns in groups of larger numbers of entities at a glance. Instead of a spatial representation as truthful as possible, MotionRugs rather aims to emphasize changes in the behavior of the visualized entities, yet retains the spatial context in an aggregated form. While conceptually also applicable to three-dimensional movements, for now, we restrict ourselves to movements in two dimensions focusing on the explanation of visual patterns in group movement. In summary, we contribute a novel visualization technique for the visual abstraction of evolving spatial data. We discuss a choice of space partitioning strategies that can be employed to derive the 1D order MotionRugs depends on, and we provide an initial set of methods for the quantification of the visual quality of the results, which the spatial ordering strategies generate. Further, we explain the patterns generated by our technique by the example of a fish swarm. We report on initial feedback provided by domain experts on the applicability of our approach showing that MotionRugs helps to understand collective trends in space and time.

2 Related Work

The initial motivation for MotionRugs has its roots in the research area of Collective Animal Behavior, which aims to decode and explain the mechanisms and coherences of collectives formed by individual entities. A good overview of this field and its methods is provided by Sumpter [43]. Fundamentally, the first step to gain insight into Collective Behavior is the exploration of records of movements of observed entities as expression of their behavior, with the goal to find generalizable rules for modeling the same. Thus, before being able to model the observed behavior, the challenge is to review large amounts of records containing large amounts of moving entities over longer time spans, while looking for general behavior as well as unusual events like unexpected outliers or loss of coherence in a group. Characterizing interactions and extracting behavior from movements is an established field, and an extensive introduction and overview is given by Laube [33]. Still, algorithmic approaches are based on assumptions that have to be established first exploratively - and thus, visually.

Yet, many state-of-the-art approaches in visual spatiotemporal data analysis treat moving entities as individuals without regarding mutual influences. However, taking these coherences between the entities into account is crucial for the analysis of groups, where one has to consider that single members or external events can affect the overall behavior and motion. In the following, we first discuss related approaches from the general field of spatiotemporal data visualization before treating such with a focus on entity interactions.

2.1 Visualization of Change in Spatiotemporal Data

Research on the visualization of spatiotemporal data has a long history. A comprehensive overview and systematization of fundamental static and interactive visualization approaches was carried out by Andrienko et al. [1]. Based on the conventional approaches discussed in the introduction of this work, and their limitations, further techniques aim to alleviate the mentioned shortcomings. For example, Tominski et al. [46] introduce an improvement for the overplotting issues based on stacking in the third dimension. The authors enable the efficient comparison of features between multiple trajectories. Yet, the approach is best suited for spatially coincident movements tied to structures like streets. Moreover, due to occlusion, crossing trajectories are hard to interpret or compare. Another static approach towards the visualization of movement data by Willems et al. [51] is based on density fields. Yet, static techniques are typically lacking the possibility to encode temporal aspects without interaction or other supplemental techniques. An interaction-centered approach to organizing large amounts of trajectories is presented by Hurter et al. [22], who combine brushing and linking interactions with boolean logic operators to reduce the complexity of the data and provide query functionality. However, the approach makes it easy to miss unexpected behavior by filtering it away and also, the temporal aspect is not supported sufficiently for the analysis of *unknown* behavior. Two additional well-known approaches to show changes in moving entities are animation [32] and small multiples [47]. For both techniques, it is difficult to track changes over longer time periods. Small multiples, however, additionally discretize continuous changes, making it difficult to determine the right number of multiples. Thus, small multiples are typically encountered in combination with other techniques or a focus on origin-destination data as proposed by Wood [52]. Also, scalability issues for both small multiples and animation approaches are documented, e.g. by Harrower [19] or Archambault et al. [5]. Due to the sequential spatial ordering introduced by MotionRugs, temporal aspects of the data can be explored without interaction or occlusion. Because MotionRugs is a dense representation, the comparison of different moments in time is independent of animation or the cumbersome selection of the right amount of small multiples.

2.2 Visualization of Spatiotemporal Interactions

The aforementioned techniques focus on presenting single or multiple movements by treating each entity trajectory independently. However, there is also research focusing on interaction spaces and dependencies between the entities. In one of the very few publications on the topic, Andrienko et al. [2] compute a central trajectory for a group, arrange the entities around this center, and visualize the results in a space-time cube like representation. However, their focus lies more on the exploration of individual relative movements within the group. As well, on page 2, the authors state, that they "have found only one visual analytics paper [...] specifically addressing group movement" [2, p.2]. This finding coincides with our expectation to find more specific techniques directly intended for the analysis of group movements. The mentioned paper authored by von Landesberger et al. [49] focuses more on an exploration of the feature space of collective behavior and automatic extraction of interesting events. Contrary to that, MotionRugs is a purely visual technique which can be applied to arbitrary features.

Other techniques are not directly related to group behavior but still take into account coherences between the moving entities. Etienne et al. [13] compute a median trajectory from individual movements and combine it with box-plot representations showing the spatial and temporal spread around the median trajectory. Here, individual behavior gets lost in the aggregation and the technique does not offer any feature (e.g. speed) or feature development visualization. Krüger et al. [31] aggregate movers spatially and extract common event sequences through frequent pattern mining of predefined POIs both in space and time. Haag et al., [18] provide a query language for event sequences in space and time. Yet, information on individual behavior of entities within a group is lost. Konzack et al. [28] compare the interaction between trajectories using a VA approach focusing on reaction delays, but their approach is limited to two moving entities at a time.

Crnovrsanin et al. [10] abstract the movement by proximity to important events in a sparse line chart. Time is mapped to the x-axis and the distance to a single event is mapped to the y-axis. This approach enables the analysis of entity interactions for an event, yet does not allow to track general behavior over time or space. Another way to consider entities as a group is to connect them in a network and then draw conclusions from network changes. But understanding structural changes in dynamic networks still poses a challenge. Burch et al. [9] propose to draw the graphs side by side, but neglect a special ordering and thus mutual dependencies. In contrast, Elzen et al. [48] build on top of the node ordering based on edge length. The visualization reveals highly communicating (connected) node pairs. The approach of Cui et al. [11] is closest to our visual result and depicts the evolution of nodes within a dynamic network. The authors order the nodes based on graph metrics and create a smoothed vector field. The flow indicates the temporal change in ordering. Dynamic network

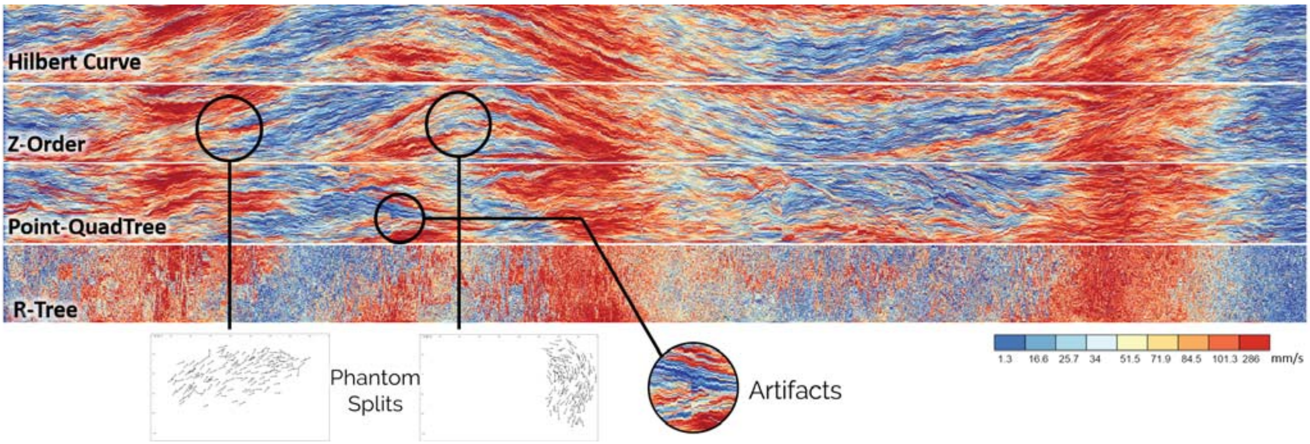


Fig. 2. Comparison of the results of ordering the movement of a group of 151 fish over the course of several minutes time. While the spatial dynamic can only be vaguely perceived for the R-Tree strategy, the Point-QuadTree suffers from artifacts. For the Z-Order Curve we found that it is prone to produce “Phantom-Splits”, visual artifacts creating the impression of a physical split of the entities when there is none. The Hilbert Curve instead displays the spatial dynamic better than the Point-QuadTree and shows fewer artifacts than the Z-Order.

visualizations typically consider connections between nodes. In this paper, entities move independently but are organized in groups. This means that entities share spatial relations but are not directly connected, which represents the main difference to dynamic networks. Instead, we build a topology using spatial indexing and partitioning that we map to a static visualization. This way, we capture spatial changes and dynamics. Another related approach to MotionRugs are the Slit-Tear Visualizations introduced by Tang et al. [44]. A scanline is drawn at a fixed position in a video to create a composite image of the entire video stream. Compared to MotionRugs, the slit-tear approach enables the identification of changes at a fixed location in the video, ignoring changes in the remaining video. MotionRugs explicitly follows and adapts to changes of moving entities.

Approaches in sports analytics, such as by Perin et al. [39] or Janetzko et al. [24] apply aggregation or classification to extract meaningful events and behavior. Still, aggregating multiple trajectories and incomplete event definitions hinder the exploration of unknown collective behavior. With MotionRugs, we intend to overcome the named issues by aggregating the spatial dimensions of the data, resulting in a dense spatial representation. Our technique can thus be considered a *dense pixel display* as defined by Keim et al. [25]; each pixel represents a data point, and the arrangement plays a key role in the technique. For example, dense pixel displays have been applied in various areas including text analysis and financial data visualization [42]. Yet, dense pixel displays for spatial data (e.g. [26]) are rare and typically focus only on one of the mentioned aspects. Visually similar techniques have been presented, for example, by Köthur et al. [29], who unify the cross-correlation between two time series in one plot. In contrast, Luboschik et al. [35] visualize chaotic movements with respect to the development of parameters over time. However, their representation results from a different incentive and is not connected to group movements.

In conclusion, an abundance of related work in the visualization of spatiotemporal data exists [6]. Still, traditional approaches like static visualizations suffer from overplotting and occlusion issues, whereas our approach provides a dense, yet spatially aggregated view over large amounts of entities and longer periods of time. To some extent, visually similar solutions exist in the area of dynamic graphs [11]. However, their construction and intent are inherently different to our technique. These shortcomings have led to the development of *MotionRugs*, which leverages the advantages of dense pixel displays to enable compact overviews over large amounts of data points and features.

3 MotionRugs Technique

MotionRugs provides a space-efficient, at-a-glance overview over large amounts of moving entities over longer periods of time. Building upon state-of-the-art, we encode both the *temporal* and *spatial* dimensions. To enable the comparison of arbitrary moments of time that are not prone to the drawbacks of animation, we aim for a static, two-dimensional representation. The notion of “two-dimensional” suggests encoding two dimensions, in this case on the *x*- and *y*-axes of the Cartesian coordinate system. As usual in mathematics, we encode the independent variable, which here is the time, on the *x*-axis. This leaves us with the *y*-axis to represent the spatial context per time point. Encoding the third dimension (*z*-axis) is possible as well. However, as discussed extensively in Section 2, the result introduces occlusion and perspective issues, thus not meeting our aim. Hence, we reduce the spatial dimensions to a single dimension through transformation.

Our approach follows a three-step pipeline depicted in Figure 3: (1) First, we consider each movement data frame as a single time instant. For each frame, we transform the locations of the moving entities into a spatial data structure. (2) Then, the resulting data structure is traversed based on a deterministic strategy to derive a one-dimensional ordering of the entities' locations in the given frame. The ordering is organized vertically in a one-dimensional *slice*. Each entity is represented as a filled rectangle, which can be scaled down to the size of a pixel, and color-encodes a pre-defined, motion-specific feature such as entity speed or orientation. (3) In the last step, we align all slices side by side on the temporal axis (*x*-axis), creating a rug-like aggregation of time-slices. Key aspects of this approach are:

Space Partitioning & Transformation To transform the spatial dimensions to a one-dimensional slice, we apply space partitioning methods such as spatial tree indices and space-filling curves. Then, we traverse the resulting structure using deterministic methods, whose visiting order determines the order in the slice. Choice and parametrization of the partitioning techniques directly impact the visual outcome.

Strategy Selection The choice of the ordering strategy is key because it not only affects the visual outcome but can also change the visual interpretation of the same. In general, we can distinguish between strategies that take into account the entire available space versus strategies that adapt to the structure of the moving entities. We introduce means to assess the quality of proposed strategies and provide reasoning for making a well-founded choice.

Feature Encoding & Visual Interpretation Each rectangle representing an entity in one time slice is colored according to a feature value that applies to the entity at the given time. Arbitrary features can be chosen, but we focus on features that can potentially express the entities behavior, e.g. speed, acceleration or orientation. The colormap choice (e.g., sequential, diverging or qualitative) depends on the selected feature. The combination of spatial ordering strategy, visualized feature, and color map results in patterns explained in Section 3.4 and 4.1. Following, we give an in-detail overview of these identified key

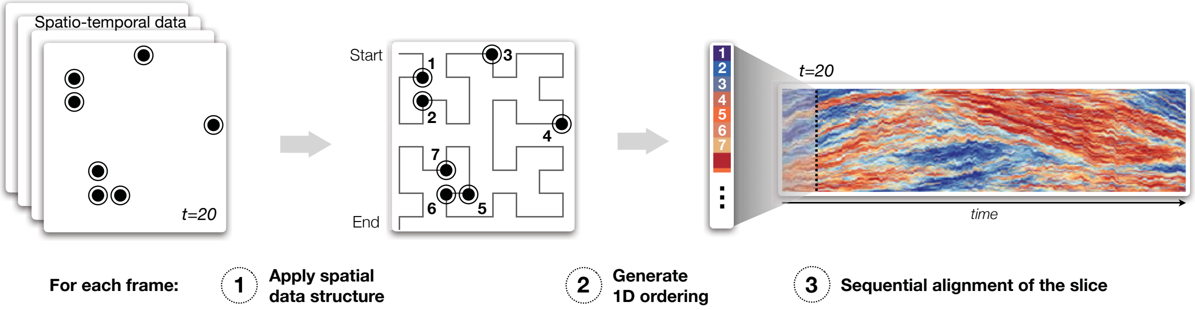


Fig. 3. The MotionRugs technique in principle: Given a large spatiotemporal dataset, our approach follows a three-step pipeline, where we process all time frames of the data successively. For each frame, we (1) first index the location of the data points using a spatial abstraction strategy. The depicted pipeline shows the application of the Hilbert Curve, a space-filling curve approach. (2) By traversing the spatial structure, we then derive a 1D ordering of the data – in this case, a slice for the frame at $t = 20$. Each data entity is represented by a rectangle, which can encode any respective data feature using color. (3) Finally, we align the generated slice with all other ones on the temporal axis.

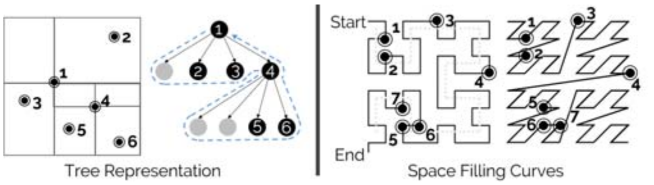


Fig. 4. Left: Spatial partitioning created by a Point-QuadTree and the tree representation. A DFS preorder traversal strategy (blue dotted line) determines the order of the entities. Right: A Hilbert and a Z-order curve and how they can be used to derive a one-dimensional order of entities. In the Hilbert curve, the next curve abstraction level is highlighted in grey.

aspects. We outline two state-of-the-art space partitioning classes in Section 3.1, in particular, tree structures and space-filling curves. Then, we introduce measurements for the ordering quality and provide results for the presented strategies in Section 3.2. Finally, we discuss the encoding of specific features and how to identify and interpret patterns in the visual representation, which we refer to as a *MotionRug*.

3.1 Space Partitioning & Transformation

We make use of spatial indexing data structures to transform the 2D data to a 1D ordered slice for each point in time. Therefore, we build upon the classification of spatial indices introduced by Lu and Ooi [34]. The authors distinguish between two main approaches used to partition a given 2D space: *space-filling curves* and *tree-based structures*, both organizing the spatial objects with respect to their location. Space-filling curves, such as the Hilbert Curve [21], are laid directly on top the 2D canvas ensuring each data point lies on the curve. Traversing the curve provides a natural 1D ordering. This is different for tree-based structures: After the data is mapped to a tree with each leaf representing one entity, an additional traversal step is required to generate a 1D ordering. Lu and Ooi also delineate hash-based approaches, which we omit due to their disregard of location. In another comprehensive publication on spatial orderings and quality measures, Guo and Gahegan [16] compare space-filling curves and clustering-based ordering approaches at the hand of two classes of quality measures comparing the distances between the entities in the original space versus those in the aggregated ordering. The authors state, that the Hilbert curve "avoids long jumps" [16], which, amongst other factors, is important for the continuity of the visual representation of a MotionRug. While the authors find, that the Hilbert curve performs less favorable for SS measures, it performs well in ordering, which is what we aim for. Other compared methods perform better than the Hilbert curve for some measures, but are clustering based. We intentionally omit these methods, since their results are often parameter-dependent (e.g. shared nearest neighbors SNN) and thus prone to generate different orders between two frames - and consequently, discontinuity in the visual result. Following, we discuss different approaches for the generation of a 1D ordering based on space-filling curves and trees. We discuss and showcase their effect regarding the resulting ordering and the ability to uncover visual patterns.

Space-Filling Curves Space-filling curves are a class of continuous, 1D mathematical functions that aim to cover a two- to n-dimensional space while preserving object proximity as accurate as possible [41]. By definition, a point-based space-filling curve is able to pass through each point in space once. Ideally, the proximity between points in space is also reflected by the curve, which can be achieved by adapting the resolution of the curve according to the amount and density of data points. Two well-known representatives of the space-filling curve family are the *Z-Order Curve* [38] and the *Hilbert Curve* [21]. Both curves are well-known and widely used. For example, they play a role in signal processing, load balancing applications or data compression tasks as for example Eavis and Cueva [12] demonstrate. While the Z-Order curve is favorable computation-wise through simple bit interleaving, studies by Jagadish [23] indicate that the Hilbert curve preserves locality best. We can apply both approaches to generate a 1D ordering. For both, we start the traversal of the curve in the upper left corner of the given space and recursively determine the necessary resolution according to the visualized data. The space-filling curves are defined only once using the resolution of the spatial coordinate systems in which they are applied. An example of how we generate the 1D ordering using space-filling curves is depicted in the right half of Figure 4. The left curve outlines space traversal using the Hilbert curve, the right curve using the Z-Order curve. In Section 3.2 we contrast the two approaches and compare their result to tree-based structures.

Spatial Tree Structures Tree-based structures are well-known for storing and retrieving data points in large datasets. Commonly used tree structures for spatial data are QuadTrees [15] (2D) and Octrees [37] (3D), R- and R*- Trees [7, 17], as well as variations of each, such as optimized octrees used for efficient isosurface generation [50]. The commonality between named approaches is that they reflect the spatial proximity between points in their structure. For example, R- and R*-Trees aim to find minimum bounding rectangles, whereas QuadTrees and Octrees recursively divide the space into four (2D) or eight (3D) areas until each area contains exactly one data point. Figure 4 on the left shows the spatial partitioning and the generation of a tree structure based on the QuadTree approach. In the visualization domain, the application of spatial tree structures serves a wide range of purposes, including the calculation of distance fields and reduction of visual clutter [53] or surface reconstruction for 3D models [54].

Compared to space-filling curves, spatial tree structures do not provide an inherent 1D ordering. However, and as mentioned before, such structures aim to preserve the spatial property of the data, which is why we can use the tree representation as means to generate a 1D ordering with a certain degree of positional retention. The special property of trees we can leverage is their universal traversability, expressed as different strategies to visit each node of the tree exactly once, thus

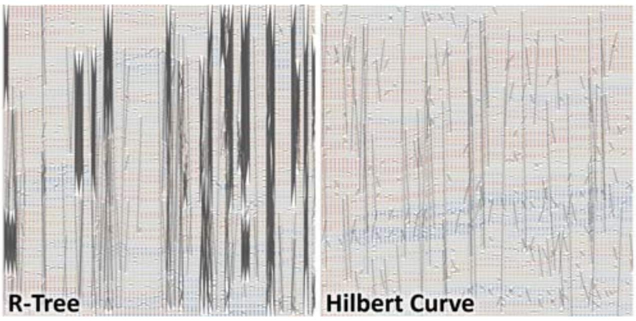


Fig. 5. Excerpt of vertical position changes of entities for the same time interval throughout a MotionRug compared for an R-Tree ordering as well as the Hilbert Curve ordering strategy. We generated this representation by introducing empty space between the ordered entities, then drawing their progress through the visualization. The colored pixels can still be seen in the background. For the same time interval, the R-Tree generates more and larger jumps of the entities back and forth through the visualization as the Hilbert Curve, resulting in a diffuse spatial representation. The impact of these findings can be compared in Figure 2.

creating a 1D ordering. Depth-first traversal strategies (DFS) consider the branch structure of a tree, while breadth-first traversal (BFS) strategies ignore branch separations. Yet, the branch structure encodes the neighborhood information in a spatial tree. For example, jumping back and forth between individual leafs of a QuadTree can result in groups of data points from different quadrants that break neighborhood relations. As a result, we can omit a discussion of strategies that break the spatial relations and adhere to DFS strategies for the rest of this paper. Further detail about graph traversal strategies can be obtained from the book of Even [14]. Generally speaking, we can apply any spatial tree structure for the generation of MotionRugs. However, other approaches, among them the UB-tree by Markl [36] or the k-d-Tree by Bentley [8], are specialized for multi-dimensional data, which is why we compare only the most representative ones, namely the QuadTree and the R-Tree. Since the common implementation QuadTree approach generates the same order as the Z-Order curve, we are concentrating on the Point-Quadtree [15], which splits the space at the location of inserted points. Also, data is not only stored in the leaf nodes, thus potentially generating a different order than the Z-Order at traversal.

3.2 Strategy Selection

Based on an initial ordering strategy, we generate a 1D slice for each point in time. The sequential alignment of all generated 1D slices creates a MotionRug – a visual representation of collective spatiotemporal movement. However, the choice of the ordering strategy affects the overall visual outcome and can influence the way we interpret occurring patterns (we discuss spatial patterns in Section 3.4). As depicted in Figure 2, the ordering strategies produce different visualization results due to the differences of how they abstract space. For trees, this difference lies mainly in the way they *split* the space, while for space-filling curves the question is how they *traverse* it. In the following, we compare the approaches using spatiotemporal fish data. The data was recorded in 25 frames per second, so that between two ordered slices, 40ms pass. For our calculations, we analyze 12 minutes of data, or 18000 frames. For both the Point-QuadTree and the R-Tree approach a preorder depth-first search (DFS) was used to determine the order for each time slice. A fixed insertion order by their sorted unique id of the entities was used for both trees in every frame to ensure a consistent treatment and comparability of results. For the R-tree, the original approach described by Guttmann [17] was used for each time slice.

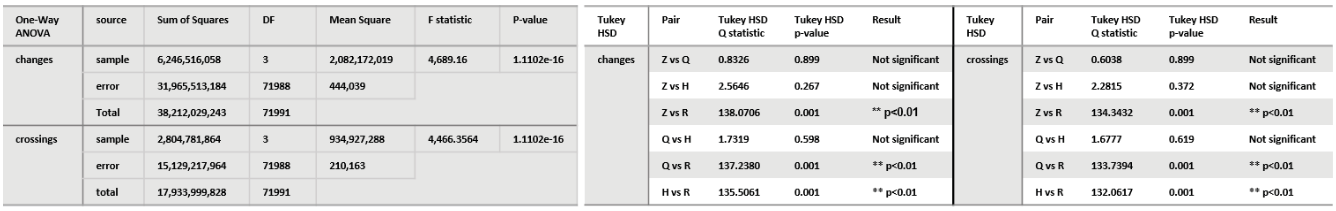
Judging from the visual outcome depicted in Figure 2, it seems that the space-filling curves approximate the spatial dimension more effective than trees, and the R-Tree seems to produce more diffuse results compared to the Hilbert Curve. Below, we provide a quantification of this coherence to provide a step towards an objective degree of quality for an ordering strategy. In a MotionRug, entities are aligned on the *y*-axis, and thus can only move vertically between points in time. Figure 5 visualizes this vertical movement for the R-Tree and Z-Order curve strategy using the same subset of the data used in Figure 2 as well. It is obvious that besides the physical movement of an entity, the vertical movement also depends on the kind of applied ordering strategy. For the R-Tree, the entities change their positions rapidly and over larger distances, thus causing jumps in the visual depiction. In contrast, for the Z-Order curve fewer jumps and a lot of continuity can be observed. Continuity refers to the consistent location of entities in the 1D ordering, meaning that they maintain almost the same position across frames. Since the fish move continuously in the physical space between frames, the observed artifacts stem from the applied ordering strategy and occur mainly when a large tree split is traversed (meaning a large distance between two objects) or at a switch between hierarchy levels of the Hilbert curve.

		Z-Order	PQ-Tree	Hilbert	R-Tree
Kendall's τ	Median	0.9922	0.9922	0.9912	0.9846
	Mean	0.9900	0.9896	0.9886	0.9090
	Max	1	1	1	1
	σ	0.0085	0.0112	0.0095	0.1609
Crossings	Median	44	44	50	87
	Mean	56.63	58.69	64.43	515.7
	Max	533	1677	433	6806
	σ	48.34	63.62	53.77	911.80
Skips	Median	82	82	94	160
	Mean	103.8	107.9	116.5	789.6
	Max	1810	3032	672	9196
	σ	83.32	112.81	91.11	1322.19

Table 1. Statistics generated by the quality metrics we apply for ordering strategies. Values for Kendall's τ can range from -1 (complete dissimilarity) to 1 (identical rank orders). Values for crossings and skips are absolute numbers of occurrence. The R-Tree generally performs worse than the other strategies. The other strategies are closer but distinguished by the maximum values for crossings and skips, which is an indicator for the impact of artifacts generated by the technique. The maximum values for Kendall's τ are uniformly 1, since every strategy created at least once the exact same order in two adjacent frames.

Quality Measurements Based on these observations, we measure the quality of ordering strategies for the generation of MotionRugs by quantifying the amount of vertical change between neighboring frames in three different measures: Kendall's τ [27] is a rank correlation coefficient representing the difference of concordant and disconcordant (order-reversed) pairs between two orderings in relation to the total number of pairs. For a more fine-grained evaluation of the nature of reversals we also give the absolute number of disconcordant pairs, i.e., pairs of fish whose trajectories cross in the MotionRug, and the total number of rows skipped by all fish between consecutive columns. High amounts of changes or crossings result in less visual coherence between two slices. Note that a certain number of changes in the ordering is to be expected as the fish move through space. Changes in relative position lead to changes in the orderings that are not a problem of the linearization strategy but reflect actual movement in the swarm. Other changes, however, are due only to sensitivity of the strategy. Since all strategies are compared on the same data, we are underestimating their stability but their quality ranking is maintained.

Results and Conclusion The results of the comparison are shown in Table 1. The results for the Kendall's τ coefficient show less rank correlation between the orderings for the R-Tree strategy than for the three other strategies, implying an increased visual incoherence between frames. To back this finding, we have conducted two one-way ANOVA to compare the effects of the Z-Order (Z), Point-QuadTree (Q), Hilbert (H) and R-Tree (R) strategies on the spatial ordering results between 18000 continuous frames taken from the dataset described in Section 4.1. We have conducted the ANOVA for both the changes and crossings measure. The pure statistical results can be seen in Figure 6 on the left. The results show that there was a significant effect of strategy choice at the p<0.01 level for both the crossings as well as the changes/skips measures. While ANOVA can only tell the existence of a significant effect, we apply Post-hoc Tukey HSD



One-Way ANOVA	source	Sum of Squares	DF	Mean Square	F statistic	P-value
changes	sample	6,246,516,058	3	2,082,172,019	4,689.16	1.1102e-16
	error	31,965,513,184	71988	444,039		
	Total	38,212,029,243	71991			
crossings	sample	2,804,781,864	3	934,927,288	4,466.3564	1.1102e-16
	error	15,129,217,964	71988	210,163		
	total	17,933,999,828	71991			

Tukey HSD	Pair	Tukey HSD Q statistic	Tukey HSD p-value	Result
changes	Z vs Q	0.8326	0.899	Not significant
	Z vs H	2.5646	0.267	Not significant
	Z vs R	138.0706	0.001	** p<0.01
	Q vs H	1.7319	0.598	Not significant
	Q vs R	137.2380	0.001	** p<0.01
	H vs R	135.5061	0.001	** p<0.01

Tukey HSD	Pair	Tukey HSD Q statistic	Tukey HSD p-value	Result
crossings	Z vs Q	0.6038	0.899	Not significant
	Z vs H	2.2815	0.372	Not significant
	Z vs R	134.3432	0.001	** p<0.01
	Q vs H	1.6777	0.619	Not significant
	Q vs R	133.7394	0.001	** p<0.01
	H vs R	132.0617	0.001	** p<0.01

Fig. 6. One-Way ANOVA (left) and Tukey HSD results (right) for all compared strategies and changes and crossings measures. For ANOVA, we report the sum of squares as the total variance in the observations, degrees of freedom df as number of parameters that may vary independently, total mean square as sample variance, and F-statistic and p-value to estimate the result significance. For Tukey HSD, we compare the studentized range distribution Q and the resulting p-value to judge the significance of differences between the chosen strategies for changes and crossings.

comparisons to determine the effects between the individual strategies. The comparisons to be seen in Figure 6 on the right show that for each case, we can conclude that the R-Tree ordering strategy performs significantly worse (** $p < 0.01$ for each comparison with the R-Tree) than the three others. Hence, these strategies provide a more continuous aggregation of the spatial dimensions, while the R-Tree can be deemed unsuitable as ordering strategy due to its lack in neighborhood retention.

While consequently, we can rule out the R-Tree, the differences between Hilbert curve, Z-Order and Point-QuadTree prove to be statistically insignificant. Still, there are visual differences between them, as Figure 2 shows. While the averages lie closer together, the differences between the maximum values have a visible impact on the visualization, on average and maximum generating larger jumps between the time steps for Z-Order and Point-QuadTree than the Hilbert curve. Consequently, for the compared ordering techniques, we assume the Hilbert Curve as best option from the compared ones to retain a spatial aggregation as continuously as possible.

Visual Artifacts Building upon these findings, we observe visual parameters that require further evaluation. In the Point-QuadTree example in Figure 2, a significant artifact is magnified, which appears to show several entities shifted by a few units. Such artifacts can be caused by branch-splitting and reorders, where entities are re-shifted through the tree branches. Also, the spatial dynamic of the entities is not represented as smoothly as in the other two strategies. Another, more prominent artifact are visual splits in the rug visualization as highlighted in Figure 2. We call such splits "Phantom-Splits", because they give the impression that the entities split up in individual groups. Yet, this artifact can be caused either by the spatial aggregation function, or by differences in the feature distribution amongst the moving entities themselves. In the latter case, the representation is an accurate representation of group composition and trend progression. For the former, the reason for such "Phantom-Splits" lies in the spatial discretization and occurs as soon as root-separated branches in a tree need to be crossed. As for space-filling curves, this phenomenon occurs when hierarchy levels have to be crossed. For example, Figure 4 exemplifies this coherence in the Z-Order curve between entity #4 and entity #5, where the curve jumps to lower left corner. Also note how Z-Order and Hilbert curves may produce locally different orders. We provide further discussion on the issue and possible treatment in Section 5. Judging from the statistics and our observations, the Hilbert Curve ordering generates the most favorable results of the selected strategies with the least amounts of shifting or splitting artifacts.

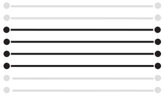
3.3 Visual Encoding of Motion-Features

The MotionRugs technique reduces time-dependent spatial locations of entities to a one-dimensional, ordered pixel display. While the subsequent ordering of entities inherently encodes overall position changes of the group, it remains challenging to draw conclusions regarding the underlying behavior like group coherency or leading entities. For this reason, we encode all entities as pixels with group-specific features using color. Captured data of entity groups typically comprise the entities as well as associated location, time stamp, and orientation. From these basic records, fundamental features can be derived that enable the characterization of the development of the observed behavior. Amongst these basic features, we primarily consider: (1) The distance between each entity and the group centroid gives evidence for the coherency of the group. (2) The speed (distance of the entities' position between frames) of entities states the overall flux of the group. (3) The direction change of entities allows to draw conclusions whether the flux remains static, and (4) the acceleration can point analysts to leading entities. Figure 8 shows examples for the speed, distance to centroid and directional change. Note that the spatial dynamic of the entities is preserved for each feature. To meet the task of comparison [45] and because the data types are diverging, we implement a diverging colormap from red to blue. The colormap highlights structural changes in the overall group, as depicted for example in Figure 7. In this Figure, analysts can detect changes in the underlying group behavior by inspecting both color and orientation of entities. When analyzing behavior, an analyst usually wants to learn about the turning points in the behavior and how these were initiated. Diverging color maps help to find these points by providing distinctions like "near versus far" or "slow versus fast". At the same time, the choice of color map is sensible and should be adjusted to the task.

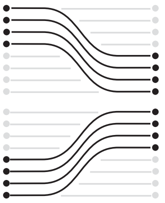
3.4 Visual Interpretation of a MotionRug

Collective or group behavior is characterized through the behavioral commonality between the observed entities rather than the consideration of accurate individual locations. This circumstance also implies, that a sufficient amount of commonality has to be given in general between the entities to be able to determine general behavior as well as outliers in the first place. MotionRugs integrates this property in the sense that the common behavior of entities is reflected in the visual representation by abstracting the spatial dimension. Consequently, visual patterns in a MotionRug can only form and be observed if the entities' behavior shows a certain amount of coherence, meaning that a significantly large subset of entities must demonstrate common behavior.

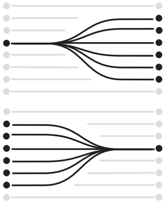
Prominent indicators for collective behavior that can be extracted from movement data alone are common expressions of *speed, acceleration, direction, and distance to the center or central entity of the group*. Further features such as distance and orientation towards a leading entity or amount of neighbors within an entities field of view are conceivable to be applied. Yet, such advanced features require prior knowledge and assumptions, which is why we concentrate on basic movement features derivable from basic movement features only. Visualizing such features can reveal two types of patterns: First, spatial configuration patterns are indicative of a groups movement and spatial activity. Second, feature progression patterns allow a quantitative view on emergence, evolution and gradual disappearance of trends. Following, we introduce the visual patterns that a MotionRug can make visible, and discuss their interpretation regarding their spatial configuration as well as feature value evolution.



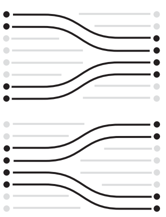
Spatial Stagnation Parallel horizontal lines indicate an overall positional stagnation of the whole group. They form when the entities vaguely stay within their original section of the index structure. If movement occurs just within the group, change in the visualization will only be visible in case this movement is chaotic, generating a more diffusely colored stagnation pattern due to the differing feature values. An example of this case can be seen in Figure 8 in the end of the heading change case. Here, the feature values are distributed in finer granularity since the heading change measurement is more sensitive than the other shown features. Stagnation patterns occur when a group is in a resting state or shows stationary behavior such as "milling", a formation when fish are turning in circles.



Spatial Change A spatial change pattern refers to the mutual direction change of a group and is represented as a ribbon that curves from top to bottom or from bottom to top. Regarding the spatial configuration, direction changes can occur when the group reacts to external stimuli such as impeding obstacles or approach of predators, amongst many other possible reasons. The visual representation, thereby, reveals the evolution of the feature values based on the spatial configuration. Features, such as speed or distance to the group center, reveal the running aground of entities before an obstacle. However, not all features can reveal such behavior, which is why it is key to bring features in relation to the overall spatial configuration.



Trend Progression A common behavior in collectives are subgroups of entities starting or ending a behavioral pattern. For example, a few fish are starting to break out of a resting pattern by getting faster and the group starts to follow. Trend progression patterns make such events visible. Opening shapes distinctive in color from the rest indicate that a trend has been started by a single or few entities and many follow. Vice versa, closing shapes show trends fading out until no entity follows them any more. For beginning trends, the shape of the developing trend at the start indicates the amount of entities involved. Its development (e.g. how quickly and far the new trend stretches over the vertical space) allows to see, how fast how many entities are following a trend and also, whether all or only a portion of the entities are part of the trend. Taken by themselves, trend progression patterns allow no assertion on the spatial progression of the group, but the curvature of trends allows this interpretation as described in spatial stagnation and change before. Visually, trend progression patterns occur when a group begins to change its configuration. These points are of great interest, since an internal or external factor causes the change in behavior. With a MotionRug, these trend events can be characterized at a glance for start and end point, feature distribution and amount of participating entities over time. Figure 1 shows examples of a distinct trend begin and end when few fish start speeding up, leading the group to follow. The spatial stagnation trend in the end consists of the entities slowing down collectively at the same time. Besides the local developments, a MotionRug also gives an idea about the global feature development and distribution. In our example, it is immediately possible to quantify slower versus faster movement phases and durations. Global developments can be perceived at a glance and visually correlated with the development of the other described patterns. In comparison to split patterns, trend progression patterns can, but do not have to start with few entities, resulting in a sharp visual representation. As well, depending on their beginning location, trend progression patters do not necessarily create a split impression.



Splits Splits and merges are another pattern visible in MotionRugs. Being related to trend progression patterns, they can indicate spatial splits and merges of entities with similar feature values. On the other hand, a split does not necessarily indicate a spatial separation of the entity groups. Instead, these patterns can also occur when a trend is starting spatially from within a group (compare the trend progression pattern). Another possibility are what we call "phantom-splits", artifacts introduced by the spatial ordering. We previously discussed this issue in more detail in Section 3.2. Split and merge patterns can be more or less visually prominent depending on the differences in the feature value expressions of the entities. Depending on the use case, a suitable colormap can either emphasize these patterns (binned and diverging color maps) or mitigate them (linear color maps).

4 Use Case and Initial Feedback

To demonstrate the applicability of our technique on a real-world dataset, we present specific examples for the interpretation of patterns as discussed in Section 3.4. The examples have been prepared in collaboration with the same domain experts we have collected feedback from. Since our approach is geared towards facilitating the overview on group movements, it is prudent to collect feedback from potential users directly, which we have documented below in Section 4.2.

4.1 Use Case

Often, the analysis of spatio-temporal data involves the exploration of large amounts of data, which can be a very time-consuming process. Reducing the effort in the spatial dimension (e.g. by zooming out) or in the temporal dimension (e.g. by speeding up an animation) is sometimes possible. Still, such measures cannot be applied to an arbitrary degree due to inevitable suffering spatial/temporal accuracy. As dense overview representation of space and time, MotionRugs supports the fast visual identification of changes in the observed movements, allowing an analyst to skip only to parts with behavior interesting to him. In the following, we show a practical example of MotionRugs applied to a real-world dataset used for the exploration, analysis and explanation of the formation of collective behavior in fish swarms.

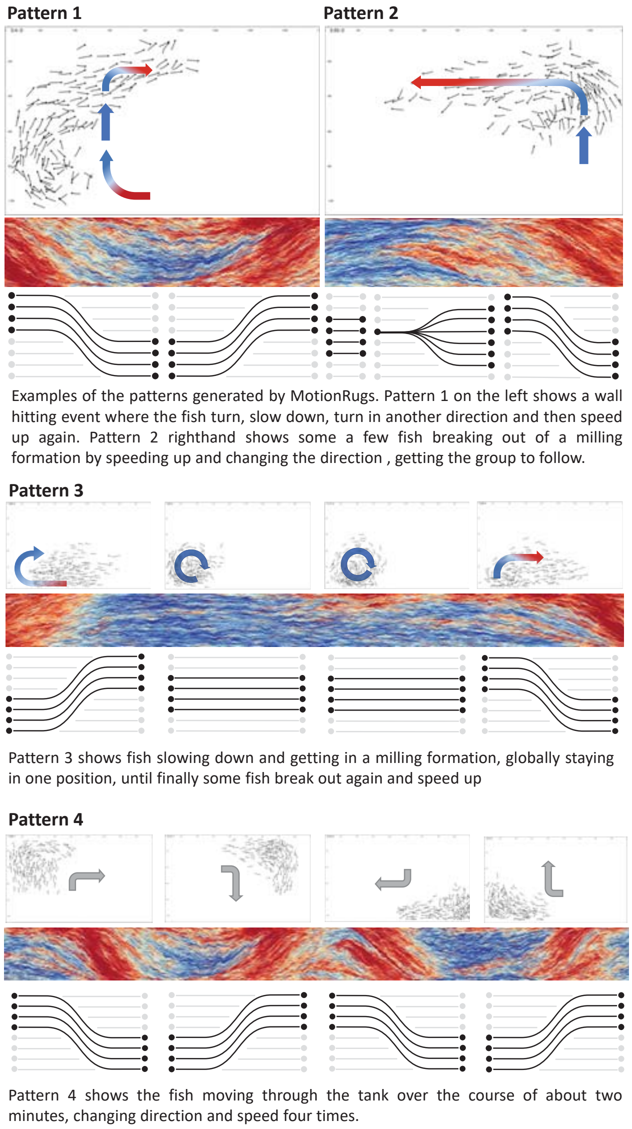
Data In our use case example, we are observing 151 Golden Shiner fish moving through a fish tank. The animals have been tracked swimming through a 2.1m by 1.2m sized tank filled with 5cm of water, so that the fish can only move in two dimensions. The movements of the fish were recorded with 25 frames per second, resulting in 18000 frames with 2.7 million data points over the course of 12 minutes. The fish are moving freely without obstacles throughout the tank. Golden Shiner fish form coherently moving swarms, and it is the goal of the experts to explain how the fish coordinate their movements by finding points in the data where the general behavior changes.

Typical Expert Tasks The analysis of data with a temporal aspect almost always bears the element of sifting through various amounts of data in search of parts in the data that are of interest. These parts can either be punctual or elongated developments in the physical or feature space or in both, respectively, over longer periods of time. To illustrate how such findings can be made, we produce a set of patterns from the fish dataset that are representative for most of the patterns one can read from a MotionRug. The referenced patterns are shown in Figure 7.

Pattern 1: Moving and turning In the first pattern, a curved red dynamic can be observed starting towards the lower middle while turning blue, then bending away to the upper right again, with increasing amounts of red pixels, while blue fades out. The correct interpretation for this pattern is that initially, the fish slow down and turn. When the whole group has caught up, they speed up one by one, turning again. In the data, the fish swarm hit a wall here and turned as reaction.

Pattern 2: Breaking from stagnation Second, we see a horizontal blue pattern that starts to bend down towards the lower right with first few red instances in the lower half of the image. In the last third of the image, the curve turns completely red. In this segment of the visualization, we can observe a slowly swimming, almost stationary group of fish milling around. At one point, few fish start to break out of the milling by speeding up, followed by the majority which has to accelerate quickly to keep up with the ones that swim ahead.

Pattern 3: Start and end of milling In the third pattern, we see an initially mainly red horizontal dynamic turning blue for the majority of the image. At half the image, some red parts appear in the upper regions, starting to turn towards the end of the image, when all parts turn red as well. Here, the whole fish swarm slows down almost coincidentally, then stays milling for a longer period of time at a certain position, until some fish start speeding up and all fish change direction again.



Pattern 1

Pattern 2

Examples of the patterns generated by MotionRugs. Pattern 1 on the left shows a wall hitting event where the fish turn, slow down, turn in another direction and then speed up again. Pattern 2 righthand shows some a few fish breaking out of a milling formation by speeding up and changing the direction , getting the group to follow.

Pattern 3

Pattern 3 shows fish slowing down and getting in a milling formation, globally staying in one position, until finally some fish break out again and speed up

Pattern 4

Pattern 4 shows the fish moving through the tank over the course of about two minutes, changing direction and speed four times.

Fig. 7. Examples of MotionRugs from situations taken from a real-world dataset as described in Section 4.1. Visualized feature is entity speed with the same color map as in Figure 1. The colored arrows indicate how the fish are moving and roughly how the visualized feature distributes at the same time. Below the corresponding MotionRug are the corresponding abstract patterns introduced in Section 3.4.

Pattern 4: Turning around In the fourth pattern, we see several changes both in curvature and color distribution. In this pattern, we can find the fish turning four times, alternating their speed in between.

In conclusion, the use cases have shown, that several properties of the animals' behavior can be extracted using a MotionRug. Amongst them are the change of positional changes of the group, the feature development and the identification of trend characteristics.

4.2 Initial Expert Feedback

Given a small amount of available domain experts for evaluation as well as the interpretative nature of the MotionRugs technique, we are confident that a large-scale qualitative study cannot adequately reflect the strengths of our approach. Instead, we focus on initial feedback given by selected domain experts. Our technique was developed with the intention to alleviate the workflow of analysts who have to explore large amounts of spatio-temporal data regularly to identify "interesting" events and group configurations. To incorporate the user view and opinions, we have asked domain experts to provide initial feedback. Our experts consisted of a group of six academics from the field of Collective Animal Behavior. Amongst them, there was one Master student, two PhD students and three PostDocs. All participants are actively involved in research on Collective Animal Behavior. Analyzing group movement is key for all experts to understand group behavior.

Before collecting the feedback, we first introduced the technique and educated the biologists on how to read and interpret a MotionRug. We presented the visualization generated with the fish dataset since the experts were familiar with it and the analysis needs tied to it. As ordering strategy, the Hilbert curve was applied. After the introduction, we presented a larger time interval to the experts for interpretation and discussion. We collected the feedback on a basic questionnaire and asked for each of the patterns noted in Figure 7 how easy the experts judge the interpretation of the presented pattern on a five-point Likert-Scale from "very easy" to "very hard". Also, we provided free text fields for qualitative feedback to query what and what not the experts liked about the technique and what is missing. Further, we recorded the discussion that developed during the feedback session.

In the discussion and the questionnaires, the experts stated that they liked "to have an overview over a whole dataset" and "to instantly see what the group does over time". They found the technique to be "exciting" and one of them was particularly positive about being able to see the feature development in combination with the feature distribution amongst the entities, comparing it to "the usual averaged line chart" the experts employ. On the downside, some participants stated troubles with interpreting the detailed spatial development, which one of them found "difficult to learn" and another "hard to abstract the movement" from individuals to the whole group from a cognitive point of view.

Wishes for additional functionality the participants had were to be able to relate individual movements and to see the presence of environmental references such as obstacles, as these often play a role in their experiments. During discussion, it was interesting to see that the experts were able to transfer their domain knowledge to interpret the patterns. For example, they were discussing hypotheses why sometimes few and sometimes many fish speed up or slow down and found explanations about wall-following or milling behavior. Also, they were able to identify subgroups of fish speeding up to stay in touch with the swarm. From this observation, we draw the conclusion that the technique can pivotally be interpreted. Altogether, the experts were able to correctly interpret most aspects of the visualization. Two found reading the spatial dynamic in general to be "hard", three "easy" and one "very easy". Considering the short introduction time and unfamiliar way to aggregate space, the results are reasonable, and can potentially be improved with further training. Still, all experts were able to distinguish between areas with high and low spatial dynamic and to draw specific conclusions from that information. Even without animation, they were able to identify the basic nature of the entities' behavior.

5 Discussion and Future Perspectives

The introduction of MotionRugs opens new perspectives on spatio-temporal data by abstracting the spatial component, and the focus is shifted from individual developments towards a profound display of the overall behavior of the observed entities. Yet, proposing a novel and unaccustomed representation of spatial aspects raises new issues and challenges we discuss in the following as further research perspectives.

5.1 Scope, Comparability & Scalability

In general, MotionRugs transforms two-dimensional trajectories into a one-dimensional, space-efficient visual representation. We applied our technique by the example of a fish swarm dataset, and illustrated the main advantage: the at-a-glance saliency of patterns/trends with regard to the movement of interdependent groups of entities. Because our technique is tailored to the movement of organized entity groups, it facilitates the identification of patterns such as changes or stagnations. This is different for incoherent and chaotic movements in, for example, in traffic analysis or logistics systems. Such data is characterized by consistent, but independent movements, which can also be reflected

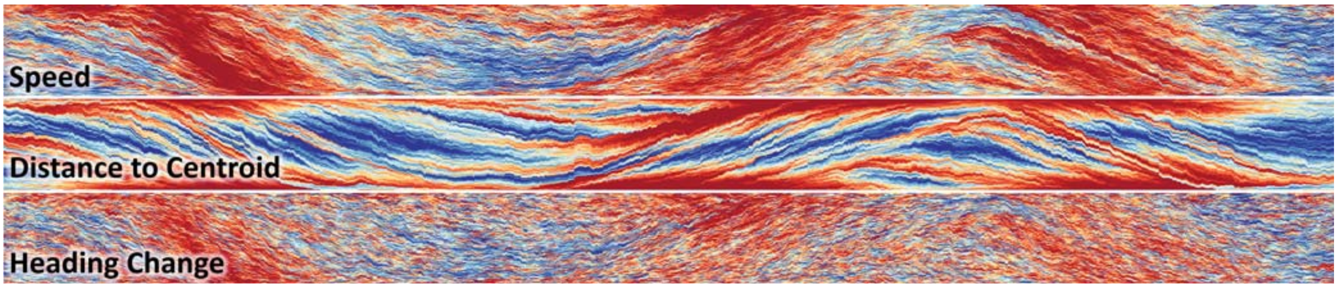


Fig. 8. MotionRugs applied to the same time span encoding three different features. The spatial dynamic reveals a waveform in each feature, which lets conclude that *Speed*, *Distance to Centroid*, and *Heading Change* heavily correlate. For example, the features Speed and Heading Change show the same alternation between the colors blue and red. We reason overall that fast fish are more likely to change their heading than slow fish.

in a respective MotionRug. Although our technique is applicable to such data, we expect no added value compared to approaches as surveyed by Andrienko and Andrienko [4]. Given the initial motivation to understand the behavior of an animal swarm, we have restricted our initial research on a single moving group. The technique works on spatially separate groups as well, but we anticipate, that a visually separating element needs to be introduced for a user to tell the groups apart. Analysts are also interested in comparing different features with each other to get insight in the underlying behavior. We enable such insights by visualizing multiple MotionRugs one below the other, each encoding a different feature (as in Figure 8). Furthermore, we link the representations, so that the analyst can hover over a point in time, and the same point in time is highlighted in all other MotionRugs. Yet, the dimensions of MotionRugs are restricted by the display dimensions. While we offer smooth scrolling possibilities in x- and y-direction, in particular the vertical comparison between feature encodings is affected. So far, we did not encounter any problems with regard to the fish dataset we used in this work. However, we are aware of this limitation with respect to larger datasets and leave a solution to future works with the initial idea to linearly downscale the visualization, which comes at the cost of losing small outliers beyond readability.

5.2 Interpretation & Pattern Correctness

Our MotionRugs technique aggregates the motion of entity groups in two-dimensional space, hence inevitably introducing possible ambiguities regarding positions and directions of the entities. Consequently, entity distances in a MotionRug do not necessarily correspond to the distances in the 2D space at all times. The directional ambiguity can be seen in the first and third turn in Figure 7, pattern 4, where the same pattern is generated for different turns. Still, a user can immediately identify an emerging change, then investigate in detail. For our experts, it was less important to see where the group was turning exactly, but more when and how the change in behavior happened. Also, the technique itself does not specifically encode the density of the entities within a group. Yet, density can be seen as feature, and as such can be visualized using a MotionRug, such as to be seen in Figure 8 with distance to the centroid. In conclusion, aggregation inevitably takes away information, presenting a trade-off between overview and full detail. For this purpose, we apply MotionRugs as overview visualization that enables the efficient identification of trends and changes in behavior and space. Once the analyst identifies an interesting pattern, she can then investigate in full detail using conventional approaches.

We introduced so-called “Phantom Splits” in Section 3.2, which pose a challenge for the interpretation of identified patterns in MotionRugs. So far, this pattern is the only one we identified in terms of a false interpretation. The artifact can occur from the spatial partitioning, in particular when entities rapidly change their position across several time frames. To overcome this issue, we compared various space partitioning techniques, coming to the conclusion that the Hilbert Curve generates the least instances of this effect, yet is not entirely protected against it. We are unaware of the existence of other artifacts and plan to examine this phenomenon further in future work including the development of new strategies tailored to this task at hand. Specifically, we are considering an adaptive frame-by-frame post-processing step taking into account the visual properties of each ordered slice by detecting visual breaks in the order. After reordering the splits to form groups of more homogeneous values, a distance function ensures the retaining of the spatial aspect, and if no solution can be found, a visual uncertainty representation such as blurring an affected area can be applied to raise a user’s awareness. The interpretation of patterns in a MotionRug also largely depends on the color-encoding. For now, we apply a diverging colormap from blue to red, emphasizing the extremes in the overall group motion. For example, if a fish swarm sinks into a stagnating state, it is of major interest to identify leading fish that break out of the formation first and fish that follow only after the group has started to lose coherence. With respect to the visualization of group-specific characteristics such as speed, distance to centroid, or heading change (compare to Figure 8), the diverging colormap highlights drastic differences best. However, we allow the analysts to adapt the applied colors according to the palettes offered by ColorBrewer [20], as well as to change the diverging to a continuous colormap, if necessary.

5.3 Space Partitioning Strategy

We identified the Hilbert Curve as most suitable strategy so far to aggregate the overall motion of entity groups based on the ordering fluctuations caused by each space partitioning strategy. Ordering fluctuations can be seen as a radical change of ordering between two time steps, introducing artifacts into the visualization. Visually, we can identify this phenomenon as line crossings as depicted in Figure 5. To measure these line crossings, we applied basic statistics, showing that we can distinguish better strategies from worse ones. However, the expressiveness of the applied quality measures is limited when it comes to fine-grain aspects like smaller ordering artifacts or the aforementioned “Phantom-Splits”. Hence, we intend to explore further statistics covering these effects. Potential approaches comprise a comparison of entity distances between normal and aggregated spatial representation, or histogram-based continuity functions in the image space.

6 Conclusion

In this paper, we introduced MotionRugs, a novel visualization technique intended to provide an efficient overview of the spatio-temporal behavior of multiple entities by abstracting the spatial dimensions. In comparison to existing approaches, MotionRugs are dense representations of movement allowing to review and compare the behavior of the observed entities over time. We presented several state-of-the-art strategies for spatial aggregation from the domains of tree-based data structures and space-filling curves and investigated quality measures to assess the ordering strategies. Initial expert feedback we collected is promising and outlines viable use cases and application scenarios.

Based on gathered feedback, we see MotionRugs as first step towards the analysis of interdependent entity groups using a space-efficient representation. MotionRugs enables the efficient exploration of spatio-temporal data with respect to the behavior of the observed entities.

Acknowledgments

The authors gratefully acknowledge Prof. Dr. Iain Couzin for providing the dataset and valuable feedback.

References

[1] G. Andrienko, N. Andrienko, P. Bak, D. Keim, and S. Wrobel. *Visual analytics of movement*. Springer Science & Business Media, 2013.

[2] N. Andrienko, G. Andrienko, L. Barrett, M. Dostie, and P. Henzi. Space transformation for understanding group movement. *IEEE transactions on visualization and computer graphics*, 19(12):2169–2178, 2013.

[3] N. Andrienko, G. Andrienko, and P. Gatalsky. Exploratory spatio-temporal visualization: an analytical review. *Journal of Visual Languages & Computing*, 14(6):503–541, 2003.

[4] N. V. Andrienko and G. L. Andrienko. Visual analytics of movement: An overview of methods, tools and procedures. *Information Visualization*, 12(1):3–24, 2013. doi: 10.1177/1473871612457601

[5] D. Archambault, H. Purchase, and B. Pinaud. Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *IEEE Transactions on Visualization and Computer Graphics*, 17(4):539–552, April 2011. doi: 10.1109/TVCG.2010.78

[6] B. Bach, P. Dragicevic, D. Archambault, C. Hurter, and S. Carpendale. A review of temporal data visualizations based on space-time cube operations. In *Eurographics conference on visualization*, 2014.

[7] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger. The r*-tree: An efficient and robust access method for points and rectangles. In *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data, Atlantic City, NJ, May 23-25, 1990.*, pp. 322–331, 1990. doi: 10.1145/93597.98741

[8] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, Sept. 1975. doi: 10.1145/361002.361007

[9] M. Burch, C. Vehlow, F. Beck, S. Diehl, and D. Weiskopf. Parallel edge splatting for scalable dynamic graph visualization. *IEEE Trans. Vis. Comput. Graph.*, 17(12):2344–2353, 2011. doi: 10.1109/TVCG.2011.226

[10] T. Crnovrsanin, C. Muelder, C. Correa, and K.-L. Ma. Proximity-based visualization of movement trace data. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, pp. 11–18. IEEE, 2009.

[11] W. Cui, X. Wang, S. Liu, N. H. Riche, T. M. Madhyastha, K. L. Ma, and B. Guo. Let it flow: a static method for exploring dynamic graphs. In *2014 IEEE Pacific Visualization Symposium*, pp. 121–128. IEEE, 2014.

[12] T. Eavis and D. Cueva. A hilbert space compression architecture for data warehouse environments. In *International Conference on Data Warehousing and Knowledge Discovery*, pp. 1–12. Springer, 2007.

[13] L. Etienne, T. Devogele, M. Buchin, and G. McArdle. Trajectory box plot: a new pattern to summarize movements. *International Journal of Geographical Information Science*, 30(5):835–853, 2016.

[14] S. Even. *Graph algorithms*. Cambridge University Press, 2011.

[15] R. A. Finkel and J. L. Bentley. Quad trees: A data structure for retrieval on composite keys. *Acta Inf.*, 4:1–9, 1974. doi: 10.1007/BF00288933

[16] D. Guo and M. Gahegan. Spatial ordering and encoding for geographic data mining and visualization. *Journal of Intelligent Information Systems*, 27(3):243–266, 2006.

[17] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *SIGMOD'84, Proceedings of Annual Meeting, Boston, Massachusetts, June 18-21, 1984*, pp. 47–57, 1984. doi: 10.1145/602259.602266

[18] F. Haag, R. Krüger, and T. Ertl. Vespa: A pattern-based visual query language for event sequences. In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016) - Volume 2: IVAPP, Rome, Italy, February 27-29, 2016.*, pp. 50–61, 2016. doi: 10.5220/0005716900480059

[19] M. Harrower. The cognitive limits of animated maps. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 42(4):349–357, 2007.

[20] M. Harrower and C. A. Brewer. Colorbrewer.org: An online tool for selecting colour schemes for maps. *The Cartographic Journal*, 40(1):27–37, 2003. doi: 10.1179/000870403235002042

[21] D. Hilbert. Über die stetige abbildung einer line auf ein flächenstück. *Mathematische Annalen*, 38(3):459–460, 1891.

[22] C. Hurter, B. Tissoires, and S. Conversy. Fromdady: Spreading aircraft trajectories across views to support iterative queries. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1017–1024, Nov 2009. doi: 10.1109/TVCG.2009.145

[23] H. V. Jagadish. Linear clustering of objects with multiple attributes. In *ACM SIGMOD Record*, vol. 19, pp. 332–342. ACM, 1990.

[24] H. Janetzko, D. Sacha, M. Stein, T. Schreck, D. A. Keim, and O. Deussen. Feature-driven visual analytics of soccer data. In *2014 IEEE Conference on Visual Analytics Science and Technology, VAST 2014, Paris, France, October 25-31, 2014*, pp. 13–22, 2014. doi: 10.1109/VAST.2014.7042477

[25] D. A. Keim, P. Bak, and M. Schäfer. *Dense Pixel Displays*, pp. 789–795. Springer US, Boston, MA, 2009. doi: 10.1007/978-0-387-39940-9_1131

[26] D. A. Keim, C. Panse, M. Sips, and S. C. North. Pixelmaps: A new visual data mining approach for analyzing large spatial data sets. In *Third IEEE International Conference on Data Mining*, pp. 565–568, 2003.

[27] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[28] M. Konzack, T. McKetterick, T. Ophelders, M. Buchin, L. Giuggioli, J. Long, T. Nelson, M. A. Westenberg, and K. Buchin. Visual analytics of delays and interaction in movement data. *International Journal of Geographical Information Science*, 31(2):320–345, 2017.

[29] P. Köthur, C. Witt, M. Sips, N. Marwan, S. Schinkel, and D. Dransch. Visual analytics for correlation-based comparison of time series ensembles. In *Computer Graphics Forum*, vol. 34, pp. 411–420. Wiley Online Library, 2015.

[30] M.-J. Kraak. The space-time cube revisited from a geovisualization perspective. In *Proc. 21st International Cartographic Conference*, pp. 1988–1996, 2003.

[31] R. Krueger, D. Thom, and T. Ertl. Visual analysis of movement behavior using web data for context enrichment. In *Visualization Symposium (PacificVis), 2014 IEEE Pacific*, pp. 193–200. IEEE, 2014.

[32] G. Kumar and M. Garland. Visual exploration of complex time-varying graphs. *IEEE Trans. Vis. Comput. Graph.*, 12(5):805–812, 2006. doi: 10.1109/TVCG.2006.193

[33] P. Laube. *Computational movement analysis*. Springer, 2014.

[34] H. Lu and B. C. Ooi. Spatial indexing: Past and future. *IEEE Data Eng. Bull.*, 16(3):16–21, 1993.

[35] M. Luboschik, M. Röhlig, A. T. Bittig, N. Andrienko, H. Schumann, and C. Tominski. Feature-driven visual analytics of chaotic parameter-dependent movement. In *Computer Graphics Forum*, vol. 34, pp. 421–430. Wiley Online Library, 2015.

[36] V. Markl. *MISTRAL: Processing Relational Queries using a Multidimensional Access Technique*, vol. 59 of *DISDBIS*. Infix Verlag, St. Augustin, Germany, 1999.

[37] D. Meagher. Geometric modeling using octree encoding. *Computer Graphics and Image Processing*, 19(1):85, 1982. doi: 10.1016/0146-664X(82)90128-9

[38] G. M. Morton. *A computer oriented geodetic data base and a new technique in file sequencing*. International Business Machines Company New York, 1966.

[39] C. Perin, R. Vuillemot, and J. Fekete. Soccerstories: A kick-off for visual soccer analysis. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2506–2515, 2013. doi: 10.1109/TVCG.2013.192

[40] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko. Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1325–1332, 2008.

[41] H. Sagan. *Space-filling curves*. Springer Science & Business Media, 2012.

[42] B. Shneiderman. Extreme visualization: squeezing a billion records into a million pixels. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 3–12. ACM, 2008.

[43] D. J. Sumpter. *Collective animal behavior*. Princeton University Press, 2010.

[44] A. Tang, S. Greenberg, and S. Fels. Exploring video streams using slit-tear visualizations. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, pp. 3509–3510. ACM, 2009.

[45] C. Tominski, G. Fuchs, and H. Schumann. Task-driven color coding. In *12th International Conference on Information Visualisation, IV 2008, 8-11 July 2008, London, UK*, pp. 373–380. IEEE Computer Society, 2008. doi: 10.1109/IV.2008.24

[46] C. Tominski, H. Schumann, G. Andrienko, and N. Andrienko. Stacking-based visualization of trajectory attribute data. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2565–2574, Dec 2012. doi: 10.1109/TVCG.2012.265

[47] E. Tufte. *Envisioning Information*. Graphics Press, Cheshire, CT, USA, 1990.

[48] S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk. Dynamic network visualization withextended massive sequence views. *IEEE Trans. Vis. Comput. Graph.*, 20(8):1087–1099, 2014. doi: 10.1109/TVCG.2013.263

[49] T. Von Landesberger, S. Bremm, T. Schreck, and D. W. Fellner. Feature-

based automatic identification of interesting data segments in group movement data. *Information Visualization*, 13(3):190–212, 2014.

[50] J. Wilhelms and A. Van Gelder. Octrees for faster isosurface generation. *ACM Transactions on Graphics (TOG)*, 11(3):201–227, 1992.

[51] N. Willems, H. van de Wetering, and J. J. van Wijk. Visualization of vessel movements. In *Proceedings of the 11th Eurographics / IEEE - VGTC Conference on Visualization*, EuroVis'09, pp. 959–966. The Eurographs Association &#38; John Wiley &#38; Sons, Ltd., Chichester, UK, 2009. doi: 10.1111/j.1467-8659.2009.01440.x

[52] J. Wood, J. Dykes, and A. Slingsby. Visualisation of origins, destinations and flows with od maps. *The Cartographic Journal*, 47(2):117–129, 2010.

[53] H. Yu, J. Xie, K.-L. Ma, H. Kolla, and J. H. Chen. Scalable parallel distance field construction for large-scale applications. *IEEE transactions on visualization and computer graphics*, 21(10):1187–1200, 2015.

[54] K. Zhou, M. Gong, X. Huang, and B. Guo. Data-parallel octrees for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 17(5):669–681, 2011.