Computers & Graphics 101 (2021) 23–34

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Section on EuroVA 2020

SpatialRugs: A compact visualization of space and time for analyzing collective movement data

Juri F. Buchmüller [a,*], Udo Schlegel [a,b], Eren Cakmak [a], Daniel A. Keim [a], Evanthia Dimara [a,c]

[a] University of Konstanz, Universitätsstrasse 10, POB 78, 78457 Konstanz, Germany
[b] Untangle AI, Singapore
[c] Utrecht University, The Netherlands

ARTICLE INFO

Article history:
Received 20 November 2020
Received in revised form 8 August 2021
Accepted 11 August 2021
Available online 14 August 2021

Keywords:
Computers and graphics
Information visualization
Collective behavior visualization
Spatiotemporal data

ABSTRACT

Compact visualization techniques such as dense pixel displays find application in displaying spatio-temporal datasets in a space-efficient way. While mostly focusing on feature development, the depiction of spatial distributions of the movers in these techniques is often traded against better scalability towards the number of moving objects. We propose SpatialRugs, a technique that can be applied to reintroduce spatial positions in such approaches by applying 2D colormaps to determine object locations and which enables users to follow spatio-temporal developments even in non-spatial representations. Geared towards collective movement datasets, we evaluate the applicability of several color maps and discuss limitations. To mitigate perceptional artifacts, we also present and evaluate a custom, time-aware color smoothing method.

1. Introduction

The visual exploration of spatio-temporal data can be tedious due to the need simultaneously regard space and time. Besides established techniques such as animation or space–time-cubes, some recent approaches aim to employ abstract, static, and dense representations to enable an efficient overview of spatio-temporal datasets (see Section 2). Such techniques order data points seamlessly in the visualization space to create a space-efficient representation, coming at the cost of reducing or even completely giving up a user's ability to relate the displayed objects to their actual spatial positions.

We propose *SpatialRugs*, an approach to encode spatial positions using mappings of real space to 2D color maps. Our technique is intended for the visualization of collective movement data, leveraging common behavior to create visually salient patterns that also allow the identification of outliers. We expect that our technique can be used for further movement datasets, but the saliency of the resulting patterns will degrade with fewer or less coherent movers.

Bellman's *Curse of Dimensionality* [1] does not only affect computational problems, but also the visualization of high-dimensional data on the two-dimensional display surface of a computer screen. Spatio-temporal data, in particular, contains two or three dimensions to represent the position of data points and one additional time dimension if one wants to oversee temporal developments. In the specific case of collective animal movement data [2], e.g., in schools of fish or flocks of birds, uncovering these spatio-temporal patterns is challenging due to large numbers of entities moving simultaneously over longer periods of time, close to each other in a similar fashion.

Most state-of-the-art techniques do not scale well to large amounts of movers and elongated datasets and traditionally resort to complex linked views in this case (refer to Andrienko et al. [3] for a comprehensive survey). Especially coordinated behavior as to be found in collective movement poses another challenge, as similar behavior cannot so easily be discriminated compared to random, unrelated behaviors of movers.

Nevertheless, recently several visualization techniques that abstract spatial relations have been proposed to facilitate the analysis of complex and large-scale spatio-temporal structures, such as collective movement or dynamic graph data (discussed in Section 2). For collective movement, in particular, the *MotionRugs* technique displays all movers in a static, compact fashion [4]. The MotionRugs approach provides the ideal canvas for exploring our spatial color feature encoding. Thus, we employ it and the used dataset to generate the base representations to which we apply the spatial coloring. In short, the principle of the MotionRugs ordering technique is based on the idea to linearize the positions of movers from 2D positions to a one-dimensional order in each time step. These 1D orderings, generated by space-filling curves, are then aligned sequentially along the *x*-axis and colored according to feature values. For example, in the MotionRug representation in Fig. 1C, each pixel represents one

[*] Corresponding author.
E-mail address: juri.buchmueller@uni-konstanz.de (J.F. Buchmüller).

https://doi.org/10.1016/j.cag.2021.08.003
0097-8493/© 2021 Elsevier Ltd. All rights reserved.

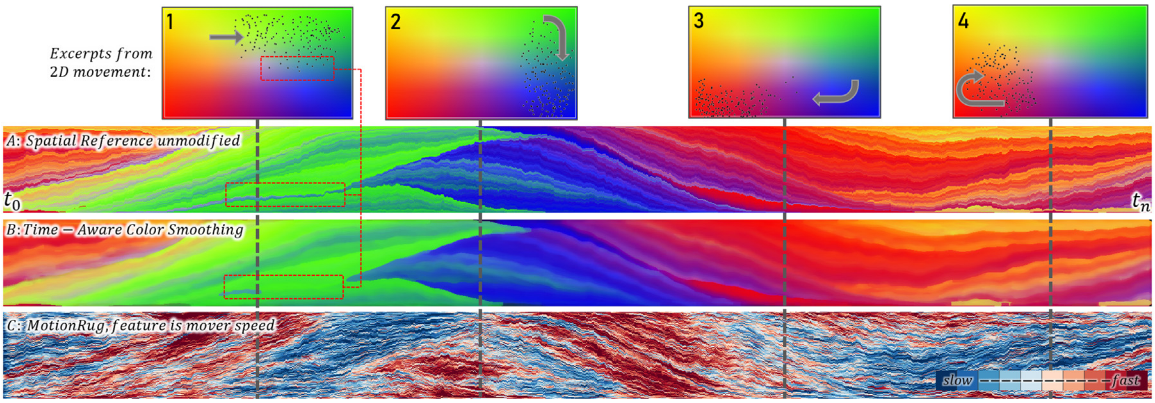


Fig. 1. *SpatialRugs* A+B and *MotionRugs* C, all with the same underlying dataset of 151 fish moving in a tank for about 90 s. Excerpts 1–4 show static snippets of the fish turning from the upper right over the lower right to the lower left. Part A shows unmodified *SpatialRugs*, where colors can be related to spatial positions (compare colors to Parts 1–4). Part B shows color-smoothed *SpatialRugs* that mitigate distorted patterns (outlined in red boxes). Part C shows mover speed encoded in the colors instead of the position. In conjunction, *SpatialRugs* and *MotionRugs* can be applied to relate space to features (e.g., in which areas of A movers are fast or slow as indicated in C.). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

mover, while the X-axis denotes time and the Y-axis represents the 1D order of all movers derived by the spatial linearization. Several numeric features of interest, such as the speed of the entities, can then be encoded by color, evolving over time. In our example in Fig. 1C, mover speed is encoded from blue to red. Several trends of slowing down (red) and speeding up (blue) of the movers are visible at a glance, while the curvature reveals spatial dynamics of the collective behavior (e.g., changes in group orientation and position). The dataset used is taken from the MotionRugs approach to be able to compare and evaluate the retention of the underlying visual structures generated by the 1D ordering. It encompasses 151 fish moving in a tank, as shown in the excerpts at the top of Fig. 1 over the course of about 90 s. For the remainder of this work and all generated visualizations, we employ the MotionRugs approach with the Hilbert Curve spatial linearization to generate the 1D orderings. We color the ordered pixels by relating the real positions of a mover in 2D space with a color from the 2D color map. Note that the order of the pixels within the visualization is not changed, and thus, MotionRugs and SpatialRugs can be directly compared.

Such “dense pixel displays” as introduced by Keim [5] typically sacrifice the representation of certain spatial data properties, like the precise location or the distance between moving entities. That way, the visualization enables the detection of patterns otherwise hidden in sparse representations or animations and provides better scalability towards larger datasets. Yet, with spatial properties fully or partially distorted, relating data points to their original position in space and time can be difficult, as the *MotionRugs* visual results prove, where the spatial aspect only shows spatial dynamic, but not position or direction as is possible with other techniques like simple static plotting or animation [6]. This is a drawback since retaining the spatial context is necessary in many use cases. To explain mover behavior, it is often essential to identify spatial positions to relate them to areas with semantic meaning like foraging grounds.

To enhance spatial awareness while preserving a compact display for collective movement analysis, we combine the space-efficiency of *MotionRugs* with the space-awareness advantages of other advanced techniques for trajectory visualization [7–9]. We introduce *SpatialRugs* (Fig. 1A and B), a technique that applies 2D color maps to dense pixel visualizations by encoding the spatial locations of movers as colors within the chosen color map. To design *SpatialRugs*, we conduct a systematic analysis and comparison of various state-of-the-art color spaces. Note that our work focuses on representing spatial relations of the movers themselves and their progress through space. We do not regard contextual spatial features such as regions or borders with semantic context (e.g., foraging grounds or territorial boundary), since there is no more room for further features in the design space. We discuss and exemplify an alternative approach for encoding such features in Section 7. We observe that the use of color can introduce perceptional complications, which may lead users to misinterpret salient color differences. These issues are caused by color map intrinsics in combination with individual color perception. As an approach to mitigate these perceptual issues arising from color space transformations (see Fig. 1B), we refine *SpatialRugs* with a time-aware color smoothing. Our proposed color correction process focuses on preserving the visual saliency of patterns in the generated visualizations by enabling users to parameterize the smoothing process according to their individual needs. We provide heuristics and an approach using edge detectors for estimation of the very use-case-dependent parameter settings. To validate the time-aware color smoothing, we evaluate the corrected result using descriptive statistics. Throughout this work, we use the same real-world dataset used in [4] to illustrate results and to enable comparison and contextualization with the MotionRugs feature encoding. As illustrated in Fig. 1, the dataset contains 151 golden shiner fish, which were tracked moving through a shallow water tank for about 90s. The examples of moving clusters in Section 6.3 are generated using a collective behavior generation model [10,11].

2. Spatio-temporal visualizations

This publication is an extension of a previously published work [12]. We have extended different aspects of our work: In Section 5, we have extended the explanations of the smoothing parameterization and provide examples for sensible parameter choices. In the same Section, we also introduce a new approach for estimating parameters for the time-aware color smoothing. We updated and extended Section 2 with further related approaches and elaborated on the construction process in Section 3. Finally, in Section 7, we applied our technique to a new dataset with more movers moving in several clusters as opposed to only one before, together with a discussion of implications.

The visual analysis of movement capitalizes on human perception to reveal patterns in space and time [6]. Andrienko et al. [7] provide an approach using spatial abstraction for collective movement, transforming mover trajectories to group-based reference points in time. However, such trajectory-based visualizations do not scale to large-scale collective movement due to the visual clutter caused by potential overlaps in space and time.

Space-efficient visualizations are proposed to produce a compact visual summary of long sequences of movement data. *MotionRugs* [4] reduce the space of the moving entities from a 2D to a dense 1D representation while still reflecting physical distances between the movers as accurately as possible. To create the 1D order from a set of 2D positions, spatial linearization strategies such as space-filling curves or spatial index structures [13] are used to retain neighborhoods as close to the original neighborhoods as possible within the limitations of a 1D order. In a *MotionRug*, every mover in one frame is represented by a single pixel that is colored according to a feature (e.g., speed in Fig. 1C). The process is repeated for each time frame ordering the slices on the *x*-axis by time. This method creates wave-like patterns, which allow the identification of spatial dynamics. The result is a static dense pixel display [5], showing the feature development of the movers over time.

ParaGlide [14] is another example of a dense representation for spatiotemporal relations that helps to understand biological aggregations in the field of collective behavior, such as the zigzagging of flocks of birds. ParaGlide allows experts to explore multi-parameter spaces of simulation models and displaying 1D projections of marginal densities in the form of a histogram (space and time). Likewise, Luboschik et al. [15] show features in a dense visualization to provide an overview of simulated movement data. The authors propose an overview visualization that presents the relationships between simulation model parameters and the resulting movement characteristics, visualized as color-coded cells sorted by time. In this paper, we apply the spatial linearization approach of *MotionRugs* to create the 1D spatiotemporal order of movers, as in contrast to other visually related approaches, *MotionRugs* are primarily used to provide an overview of feature distributions. The core concept of *SpatialRugs* is to employ colors mapped to spatial positions using a 2D color map. Essentially, this enables us to use color as visual variable to encode position, making our approach ideal for spatial feature encoding for the continuous, dense MotionRugs visualizations.

Dense representations have also been proposed in the context of dynamic graphs. Burch et al. [16] introduced parallel edge splatting, a technique that displays a sequence of graphs as a series of narrow stripes. The parallel edge splatting technique visualizes a weighted dynamic graph in a single static view, providing a scalable overview of the temporal dimension. Van den Elzen et al. [17] extend massive sequence views for the analysis of dynamic graphs. The authors propose multiple reordering strategies for 1D graph layouts to highlight and interpret temporal patterns, such as trends and anomalies. Another pixel-based visualization for dynamic graphs is GraphFlow [18], which visualizes evolving graph metrics to provide an overview of structural changes in the temporal data. Contrary to these techniques for dynamic graphs, *SpatialRugs* aims to present the evolving spatial distributions in collective movement and leverages the space-preserving properties of *MotionRugs*, which retains spatial distances to large degrees in a 1D linearization, allowing an overview of evolving characteristics (e.g., speed or acceleration) in a dense pixel-based representation.

Conclusively, most techniques for trajectory visualization [7–9] lack scalability for larger amounts of conformingly behaving movers. On the other hand, dense pixel methods like *MotionRugs* lack spatial awareness by omitting to display accurate spatial locations of the movers; they capture changes in space and mover orientation over time but do not expose *whereto* entities are moving exactly. This limitation is critical for many use cases where analysts need to know the regions in which the entities are moving. To enhance spatial awareness while preserving the scalability of *MotionRugs*, we propose *SpatialRugs*, a technique that reintroduces spatial positions into dense spatio-temporal visualizations, eliminating the necessity for tedious analyses with, for example, clutter-prone static trajectory plots or time-consuming animations.

3. *SpatialRugs* main design: Retaining spatial readability

SpatialRugs is a compact movement visualization technique that enhances spatial awareness by projecting the positions of movers in a 2D-color space to assign each position a color in a continuous space. Fig. 2 at the top illustrates our approach:

(I) We transform a given color space from its original dimensions to a 2D cubic representation as a base for the second step.

(II) We transform the 2D color space to cover the maximum extent of the spatial dimensions used by the mover dataset.

(III) We assign the 2D position of a mover to the corresponding color of the transformed color map. The assigned color is then applied to the respective data point in the dense pixel display.

Spatial positions are now represented by color, which can be used in conjunction with pixel-based visualizations of movement, such as *MotionRugs*, to encode mover locations. With the colormap reference, users are able to identify the spatial distribution of entities at a given time. Fig. 1 shows that the movers come from the upper right corner (green, first excerpt), take a right turn towards the lower right (blue, second excerpt), move through the lower middle of the represented space in purple to the lower (red, third excerpt) and finally the middle left in orange color tones (fourth excerpt). The resulting patterns allow perceiving the movers' spatial distribution, while viewers can also estimate how the movers progress within the color zones. For example, between excerpts 1 and 2, just a few movers start to move towards the blue until everyone follows. This behavior is shown as a cone-shaped transition from green to blue. Consequently, the color mapping enables to see patterns over long periods of time compactly, also relating the spatial development to the feature development by comparing the excerpts (e.g., by relating Fig. 1A and C). In this example, we use SpatialRugs to encode spatial relations, whereas another feature, speed, is encoded using a blue-to-red colormap as initially described in [4]. It is possible to stack even more views on the same data with other colormaps encoding further features, for example, acceleration or heading. If these views are aligned, users can compare different features and put them in context, with the spatial relations being one of them.

We have implemented a Java-based prototype that takes CSV-based movement data and applies a selected 2D color map. The input data has to provide ids and positions of all movers in regularly sampled intervals and needs to be free of gaps. For transforming the chosen color space to a raster image with cubic dimensions, we refer to the individual and widely differing approaches described for each color map as referenced in Section 4. The resolution of the resulting image needs to cover the full coordinate space of the 2D movements to ensure that each location can (potentially) be encoded with a different color. We use this base representation of a color space and apply bicubic interpolation for transformation to reflect the minimum and maximum coordinate extents the input data shows. Finally, coordinates in the movement space and in the color space are matched and can then be applied as spatial colormap.

4. *SpatialRugs* color space selection

With color perception being a very individual property differing from person to person [20], selecting the appropriate color map is a critical design choice for the *SpatialRugs* approach. Several previous works apply color space mappings to represent spatial or temporal relations: Northern Lights Maps [21] by Janetzko et al. map spatio-temporal properties of movers to a continuous RGB-color scale. PhenoVis [22] presents color-coded normalized stacked bar charts to allow comparative analysis over longer time spans. MotionExplorer by Bernard et al. [23] employs a projection-based view displaying human motions in a

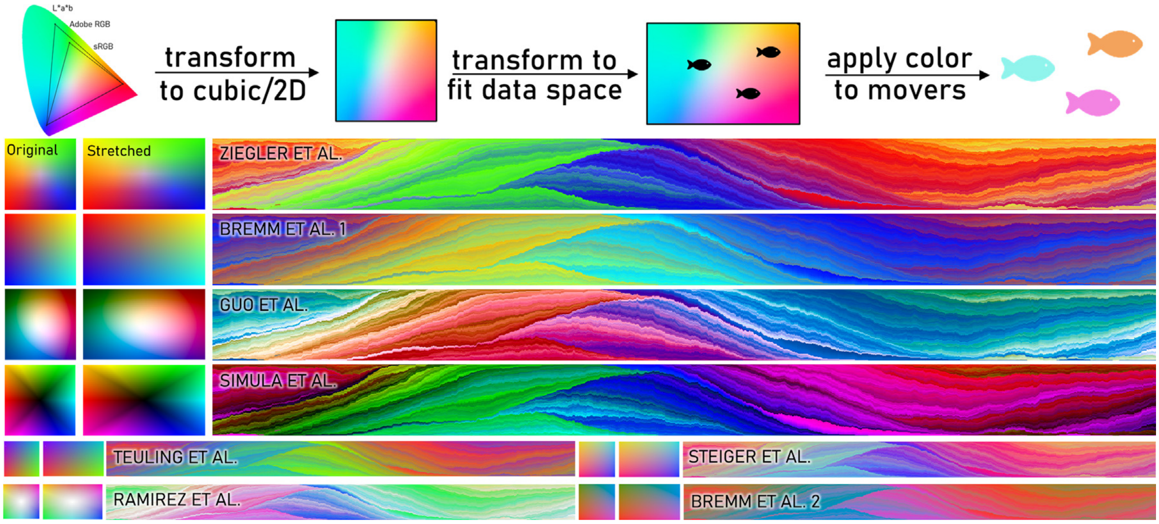


Fig. 2. Top: In *SpatialRugs*, a color space is transformed into a 2D cubic form, then adapted to the extent of the moving area. A position is then encoded using the corresponding color from the color space. Below: Application examples of different colormaps [19] applied to a real-world dataset containing 151 golden shiner fish expressing collective behavior. Left of each visualization, we see the underlying transformed 2D color space. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

discretized 2D color-coding to highlight temporal patterns. *SpatialRugs* employ color spaces, which are continuously and linearly transformed in each dimension to accommodate the available 2D-space to the fullest. Yet, it is important to consider that, given the individuality of color perception, different individuals will judge the same colors to be at slightly different positions. Still, Emery and Webster [20] state that at least the color perception within an individual person remains quite stable under varying conditions.

A thorough quality assessment of two-dimensional color spaces has been conducted by Bernard et al. [19] with respect to multivariate data. We consider our use case to be within a specific subset of their study and consequently apply their findings to identify suitable colormaps for *SpatialRugs*. However, it is important to keep in mind the specific nature of the dense pixel representations we intend to enrich with spatial information, where every pixel encodes a spatially annotated data point. In contrast, multivariate datasets of general purpose with two or more features, as regarded by Bernard et al. are usually sparse to varying degrees. Thus, the visual representations of space-efficient techniques are continuous in nature, opposed to the gaps which can be observed in scatterplots for example.

In a widely recognized article, Peuquet introduced a conceptual framework for geospatial dynamics with the fundamental concepts of time (*When*), space (*Where*) and context (*What*), and how these concepts are connected to each other [24]. In concordance with this approach, we derived three requirements for *SpatialRugs*: (1) identify individual spatial positions of movers or mover groups (*Where*) from the color space, (2) track the temporal evolution (*When*) of movers or mover groups continuously through the color space, and (3) judge the relative distances between movers or mover groups over time by comparing two given colors (*When+Where*). The listed requirements include the spatial–temporal (*When+Where*)-aspects of Peuquet's model in a dense representation. As a result of this, a user can explore *What* happens between the movers, as shown in a recent extension of Peuquet's framework by Andrienko et al. [3].

We can translate these requirements (1–3) to the elementary tasks (ER 1–3) of color map assessment by Bernard et al. [19]. The first elementary task states that a viewer should be able to locate and identify a single object in color space accurately **(I)**. For instance, if we have n movers at distinct positions in space, then the color space ideally also provides n visually separable colors to encode the movers. In the second elementary task, a viewer must maintain and link equally salient colors with spatial positions **(II)**. For example, the utilized color space should not highlight particular movers due to perceptual color differences, such as bright colors on a rather dark color map. The third elementary task describes the need for accurate comparison of two or more locations to identify similar or dissimilar objects **(III)**. For instance, the distance between movers or mover groups in space should be perceptually similar to the distance in the color space. Overall, possible color space candidates need to enable users to accomplish these three tasks **(I–III)**.

These tasks and requirements constitute ideal conditions for a color map, which in reality cannot be fulfilled completely. For example, the amount of visually distinguishable colors is limited, and thus, the amount of encoded movements is limited, too. Yet, our use case concerns collective movement, where accurate movement representations stand back against the analysis of the similar movements of many movers and possible outliers.

In addition, standard color spaces, e.g., CIELAB, HSV, or sRGB, are mostly organized in three dimensions and usually do not form a symmetrical shape. Yet, *SpatialRugs* needs to represent the 2D positions of the observed movers. Consequently, a chosen color space should be mappable to 2D space without compromising so much uniformity of color distribution that the requirements cannot be kept anymore. As well, color perception is individually different in viewers [25], resulting in different abilities to identify fine-grained differences. Thus, a sensible color space choice is critical for the effectiveness of *SpatialRugs*.

In their survey, Bernard et al. [19] investigate the capabilities of 22 state-of-the-art 2-D color maps with respect to these analytical tasks and perceptual properties. They compare spatial distributions of color space properties and then evaluate several quality assessment measures for each color map. Finally, they judge how well an approach can fulfill their defined requirements using a basic grading system. Importantly, they judge independently between having a black and white background for data points represented by the compared color spaces. This is the main difference to our approach, which, due to the density of the representation, does not feature any background within its canvas. Below, we discuss the criteria we consider for adequate color map choice at the hand of a selection of candidate color maps in three categories. In Section 7, we argue for a suitable

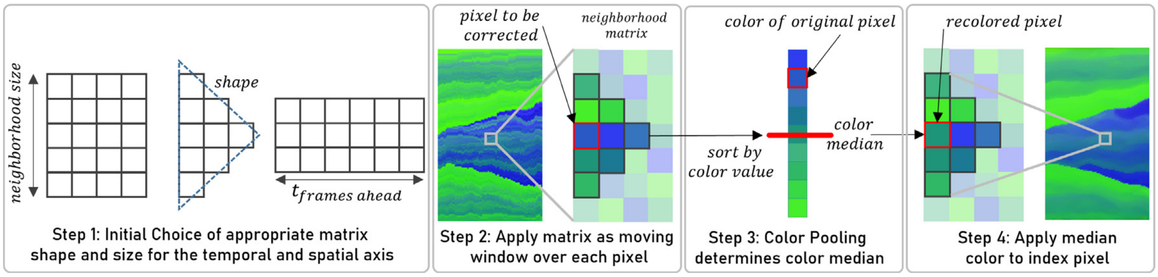

neighborhood size
shape
$t_{frames\ ahead}$
Step 1: Initial Choice of appropriate matrix shape and size for the temporal and spatial axis
pixel to be corrected
neighborhood matrix
Step 2: Apply matrix as moving window over each pixel
color of original pixel
sort by color value
color median
Step 3: Color Pooling determines color median
recolored pixel
Step 4: Apply median color to index pixel

Fig. 3. Pooling-based color correction. One matrix dimension determines the size of the regarded neighborhood, the other the time ahead to be considered for the correction. After selecting a use-case appropriate shape and size (step 1), the matrix is shifted over each pixel in every time step (step 2). In step 3, the colors of the matrix cells are sorted by Euclidean distance in the RGB space. The median color is then applied to the original pixel in step 4. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

color map that fulfills the requirements in the context of the applied datasets.

Task assessment: Fig. 2 shows a comparison of color maps taken from Bernard et al. [19] generated with the data described in Fig. 1. According to the task assessment table of Bernard et al. colormaps taken from Bremm et al. [26] (labeled as “Cube Diagonal Cut B-C-Y-R” in [19]), Ramirez et al. [27], Steiger et al. [28] (labeled “Mittelstädt et al.”) and Teuling et al. [29] (labeled “TeulingFig4a”) receive high ratings for the tasks ER1-3 and either or both background conditions, and thus, would be best suitable given our defined tasks **I–III**.

Yet, the task-based recommendations [19] are made for sparse distributions of the colors with gaps in between. They are not designed to regard the perceptibility of visual structures *within* the visualization space — in other words, visual structures which are entirely created by the seamless order of the colored data points themselves without background interference. As retaining these structures is important to our approach, we consider further color maps and turn to the quality assessment measures provided by Bernard et al. to do so.

Quality assessment: The JND measure describes the “Just Noticeably Different Colors” [19], indicating how well a colormap exploits a color space. Here, the colormaps by Simula and Alhoniemi [30] and Guo et al. [31] perform best but iterate over black or white. Such color maps with a low black- or white distance score work well only in conjunction with backgrounds of the opposite color. As dense pixel technique, *SpatialRugs* does not feature intermediate spaces between the data points. Hence, using color maps with black or white color ranges could interfere with the perceived brightness and saturation of the surrounding colors due to contrast effects difficult to measure [32], rendering the color map not applicable for our case. The next best color maps according to the JND feature are the Cube Diagonal Cut B-C-Y-R [26] (labeled “Bremm et al. 1 in Fig. 2”) and the Four Corners R-B-G-Y color map (“Ziegler et al.”) [33].

Transformation assessment: The visual outcome of *SpatialRugs* is also determined by the amount of applied transformation to the color space. Changing the ratio of an original color space in one axis affects the color discriminability along the same axis. This holds even if the ratio is changed on both axes. In both directions (either shrinking or enlarging the color space), color discriminability suffers since either there will be less space to represent all colors a color space can provide, or the same colors are stretched over a larger space. Yet, since color perception is not necessarily linear, such effects can only be measured in perceptual studies.

While we acknowledge these effects, we expect our technique to be applicable to uniformly distributed spatio-temporal data in space. However, movement data that is not evenly distributed in space is more challenging to interpret, for instance, in datasets with a few spatial outliers that expand the size of the coordinate system. In such cases, we recommend preprocessing the data appropriately by removing outliers and using spatial regularization methods.

As the visual outcome is more dependent on movement distribution instead of the physical size of the movement space, we propose to estimate the amount of regions users can visually distinguish by applying the JND metric as discussed by Bernard et al. [19]: They count the “Just Noticeable Different Colors” for a colormap, which consequently also denotes, how many regions a user can distinguish within a color map. By dividing the available space by the JND metric, we receive the average size of regions that users can visually distinguish (JND-region-size). Due to the non-uniformity of color perception, this region size can vary locally, which Bernard et al. provide a standard deviation measure $\sigma_{JND-region-size}$ for. In our case, we also have to factor in the possible distortion caused by a changing aspect ratio (e.g., 16:9 or 1.78:1). As the amount of JND colors and thus also regions does not change with the aspect ratio, the following formula details how users can calculate the size of these areas:

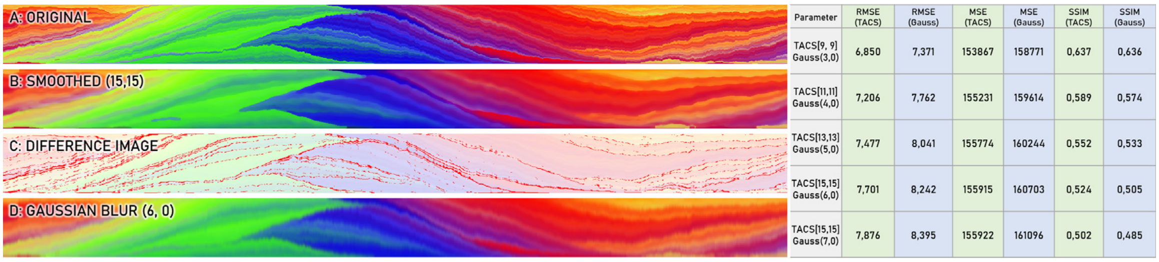
$$size = \sigma_{JND-region-size} * f_{ar}$$

With f_{ar} being the change of aspect ratio (e.g., 1:1 to 1.78:1/16:9, so 1.78). As we only distort the color space, the scaling incorporates such a distortion factor into the standard deviation. The result is the maximum area in which a user is not able to further distinguish the colors in and can be considered the worst case for parts of the 2D color map.

5. *SpatialRugs* color design: Pooling-based time aware color smoothing

We observed adverse perceptual distortions for certain use cases, especially in the transition areas between primary color tones. Our use case of collective movement analysis has a strong focus on group coherence. However, perceptual artifacts can occur in *SpatialRugs*, when a part of an otherwise homogeneous group of movers partially protrudes into another color area. Fig. 1 shows a case of perceptual distortion in excerpt 1, where most movers are in the green quadrant, with a few extending into the transition area to the blue quadrant, resulting in a salient blue line (outlined in the red box). The same effect can be observed in Figs. 4 and 6. Here, the perceived color distances appear larger than the actual distances of the blueish movers to the rest of the green group, possibly creating the false impression of two independent groups moving around. Such perceptual distortions are artifacts of the color map showing movers already crossing color borders, perturbing real-world situations by presenting these movers as outliers.

To mitigate such perceptual distortions, we propose a time-aware color smoothing technique. Our method regards the mover distribution of the current and subsequent steps to determine a



Parameter	RMSE (TACS)	RMSE (Gauss)	MSE (TACS)	MSE (Gauss)	SSIM (TACS)	SSIM (Gauss)
TACS[9, 9] Gauss(3,0)	6,850	7,371	153867	158771	0,637	0,636
TACS[11,11] Gauss(4,0)	7,206	7,762	155231	159614	0,589	0,574
TACS[13,13] Gauss(5,0)	7,477	8,041	155774	160244	0,552	0,533
TACS[15,15] Gauss(6,0)	7,701	8,242	155915	160703	0,524	0,505
TACS[15,15] Gauss(7,0)	7,876	8,395	155922	161096	0,502	0,485

Fig. 4. Comparing an unmodified SpatialRug A to a smoothed one (matrix size 15 × 15) B and Gaussian blur (sigma y, x) D. C provides a difference image between A and B and highlights the areas our smoothing focuses on in red. The table shows quantitative assessment results for time-aware color smoothing (TACS) versus standard Gaussian blur (Gauss). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

color correction factor. If entities close to each other are located in different color areas, their respective color is corrected towards the majority. Such a correction enables smoothing artificial borders introduced by a selected color map to focus on the movers' general behavior.

After applying the smoothing, we do not intend to reflect locations as good as in the original, but to enable a focus on the movement of the group by removing color map artifacts and artificial outliers. Such artificial outliers can lead non-expert analysts to incorrect hypotheses based on issues arising from a distorted color schema, e.g., wrong leaders.

Fig. 3 shows our method consisting of three steps: color collection, pooling, and adaption, which are repeated for every pixel. During initialization (Fig. 3, Step 1), users adjust the pooling matrix, selecting three parameters: *neighborhood size*, *time frames* ahead, and *matrix shape*. Step 2 applies the user-defined pooling matrix around the target pixel and collects the colors of included pixels. In step 3, the collected pixels are ordered with a stable sorting algorithm (e.g., mergesort) on the RGB values individually. At first, the blue values of the RGB will be sorted, then the green and finally the red values. Through the stable sorting algorithm, the ordering includes a hierarchy for the RGB values and enables a better comparison in the color schema. Outlier pixel colors will be sorted to both ends of the list, while more similar colors move to the middle. In Step 4, after the sorting, the median of the array yields the most prominent color value of the collected pixels, and the index pixel is corrected using the color generated by taking the median values of each sorted color channel. In comparison to calculating an RGB distance value from the combined color channels, this approach minimizes unwanted color channel effects in which the ordering neglects the possibilities of similar colors belonging to each other. Note that in this process, no pixels are reordered in the visualization. The color ordering process in step 3 is used to determine the color median to apply to the index pixel to correct, but it has no impact on the order of pixels in the result.

Next, we discuss the implementation of the parameters and provide initial guidelines on how to select them. Yet, we expect that optimal parameterizations depend on the specific movement behavior expressed by the movers. In the following discussion, we outline the relation of the parameters and different kinds of mover behavior. It is important to note that our proposed approaches for determining parameters should be regarded as initial recommendations. Given the large range of applicable use cases as outlined in Section 7, optimal parameters need to be further tuned according to the specific dataset.

5.1. Neighborhood size

The neighborhood size parameter determines the number of its direct neighbors affecting the resulting color correction area for each data point at hand. The neighborhood size is the most use-case-dependent parameter. It relates to both the visually apparent neighbors in the image space and to the original spatial domain. By setting the neighborhoods, the algorithm is steered to include the specific characteristics of a dataset that an analyst weighs heavier than others. Knowledge about the typical size of neighborhoods is necessary to select an appropriate size. Thus, an analyst can include her domain knowledge about typical neighborhood size in the smoothing process for specific types of collective movements.

To illustrate, when the behavior between the movers tends to be more coordinated, the color smoothing should reward common behavior by grouping the movers together. In contrast, for more individualistic movers, less neighbors should be considered to prevent smoothing differing behavior away. Due to the case-dependent nature of collective movement behavior, finding a general heuristic might not be feasible, and a user-driven exploration of the parameter space is essential. To initially determine default values, we recommend a simple initial heuristic towards the max neighborhood size (n_mover) by taking a fixed percentage of ten percent as our initial neighborhood size.

$$n\_nb = n\_mover * 0.10$$

5.2. Time frames ahead

Together with the neighborhood, the movement and frames looked ahead parameters hold information necessary to understand collective movement patterns (e.g., how neighborhoods change over time). Looking ahead facilitates to further smooth color artifacts and incorporates long-term neighborhoods. However, detecting such long-term neighborhoods is rather difficult and needs domain knowledge to include the often hidden underlying information into the algorithm. For example, domain knowledge about observed movers could tell about the foreseen neighborhood coherence over time. If it is known that the movers only form loose groups which do not stay together for long, not many time frames ahead should be regarded and vice-versa.

We propose a basic heuristic that considers the length of the sequence with an incorporation of the neighborhood size. E.g., assuming that only one-third of the neighbors of a mover stay with it within a given period, we first take one-third of neighborhood size (n_nb). Next, we take one percent of the number of frames (n_fr). If one-third of the neighborhood size is smaller than the short one percent length, we subtract this value from the one percent of the sequence to incorporate a weighting of the neighborhood size into the frame ahead. Such a weighting enables to balance the frame ahead with the neighborhood parameter.

$$n\_tfa = \begin{cases} n\_fr * 0.01 - \frac{n\_nb}{3}, & \text{if } \frac{n\_nb}{3} < n\_fr * 0.01 \\ n\_fr * 0.01, & \text{otherwise} \end{cases}$$

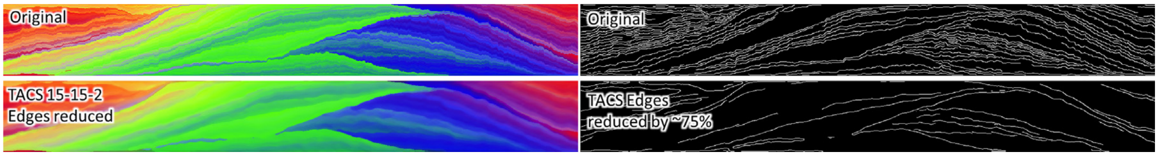


Fig. 5. Results of the TACS smoothing with parameters determined by the heuristics described in Section 5.4. On the right, edges detected by the edge detector for the original and TACS smoothed version with a decrease of about 25% in edges compared to the original.

5.3. Matrix shape

Developing neighborhoods can be challenging for a general smoothing as a rectangular matrix assumes static importance of neighborhoods over the time frames ahead. Thus, we argue for another important parameter, the matrix shape. Especially a matrix in the form of a triangle pointing into the future with steps like the ones shown in Fig. 6 enables a focus on changing neighborhoods, since the further we look ahead, the fewer original neighbors exert influence on the data point to correct.

Yet, the matrix shape can also be applied for other scenarios: A rectangular matrix ignores changes over time in the neighborhoods and uses all possible neighbors over the frames with the same importance, which is useful for movers with strong and ongoing group coherence. A less steep, triangular matrix incorporates the neighborhood another time frame ahead to the smoothing. Such a matrix helps to include slowly developing neighborhoods and enforces a smoothing with an influence of the neighbors over time, which is useful for detecting initial group formation. Finally, a more steep matrix only incorporates the current neighborhood and the developing movement of the focused mover over the time frames ahead. Such a matrix enables to focus only on specific movers and their direct neighbors, which is useful for more individualistic movers (e.g., car traffic).

5.4. Assessing quality using edge detectors

Edge detectors enable another way to further investigate parameter choices for TACS. As described before, especially for grouped behavior, visual artifacts in the color space can occur when parts of a group leap into spatially closer, but perceptually more distant color areas, resulting in misleading edges in the result image. Edge detectors like the Canny edge detector [34] are able to detect and quantify these edges in color transitions, a circumstance we can exploit to find suitable parameters: Under the assumption that a smoothing algorithm applied does not create additional edges, the general idea is to compare the number of edges found in the original image to the amount in the smoothed output image. Depending on the number of edge pixels we found through an edge detector, we then calculate a score. For instance, we can find around 13% edge pixels in Fig. 4A of the overall pixels while the smoothed version of Fig. 4B reduces the edge pixels to around 4%. A reduction of edge pixels is in general favorable as it smooths the image at a global scale. However, our comparison against a Gaussian blur in Fig. 4 leads to around only 1% of retained edge pixels. Thus, only decreasing edge pixels as much as possible is not a suitable heuristic to identify appropriate default parameters since prominent visual structures would be destroyed in the process. A balance between edge pixels and smoothing needs to incorporate the percentage of edge pixels towards the overall image and a minimum of edge pixels. Preliminary results examining the results of various parameter options suggest a minimum of one-fourth of the initial percentage of edge pixels. For instance, with the 13% edge pixels in Fig. 4A, we can smooth to 3.25% edge pixels, both reducing unwanted artifacts while keeping visual structures intact. For further analysis and investigations, a complete user study is necessary to investigate the limits of the smoothing in order to determine which parameterizations constitute sensible choices for the best outcome.

Our preliminary results using the edge quantification show promising results for most smoothing parameters, which use the one-fourth rule of thumb. In our experiments, smoothing parameters which retain major edges from the original image are favorable towards others. Especially, smoothing parameters that reduce edges near each other lead to promising results with a focus on the goal of the smoothing itself in mind; reducing color transitions of spatial artifacts. Figs. 5 and 7 show the two use cases with the original *SpatialRugs* and the smoothed versions edges. Both use cases benefit from the smoothing, showing clearer transitions between different areas, while clutter and fuzzy areas are notably reduced. In our use cases, we find promising recommended parameters based on such an edge detection metric to be rather small. For instance, 15 and 17 are both promising default parameters for the smoothing in Fig. 5 for neighbors and frames with the matrix shape as a triangle.

5.5. Discussion

Visual artifacts can mislead viewers due to the non-linearity of applied color maps, individual perception, or both. With the TACS, we propose a mitigation strategy for such artifacts, which can be parameterized to a user's specific needs: Based on the analyst's domain knowledge on the dataset and task specifications, the analyst can modify the parameters in Step 1 to her needs. The neighborhood size parameter describes the spatial region around the focused pixel in the vertical axis. For analyzing movers of coherent behavior (e.g., fish as opposed to monkeys), the analyst can adjust the neighborhood size so that stronger (or weaker for monkeys) relationships are incorporated.

The time frames ahead incorporate the spatial movement into the future to smooth in the horizontal direction. For observations that include fast-changing movements (e.g., in insect movements), the analyst chooses to capture fewer steps in time ahead to cover fast changes of color as opposed to slower changes in movement (e.g., for larger animals). Lastly, the matrix shape offers a way to reduce the amount of neighborhood lookahead in space and time, limiting the importance of the neighboring movements. For movers tending to behave more coherently over time, such as schools of fish, a rectangular matrix shape is recommended, while less coherent behavior requires a triangular matrix shape, focusing more on temporal than exact spatial and orientational coherence (e.g., cow herds). The comparison in Fig. 6 shows the impacts of different parameter settings. The parameter choice recommendations should be seen as initial suggestions. Their optimal choice relies on mover properties such as size or average group density and even behavioral features like the number of neighbors influenced by actions of a single moving entity. These factors cannot be generalized in a single parameter choice formula. Thus, we recommend extracting prototypical use cases with experts using real-world data to identify parameter boundaries individually.

The code for the color smoothing is publicly available as Python notebook [35]. Each of the notebooks shows an implementation of TACS with the parameters chosen according to

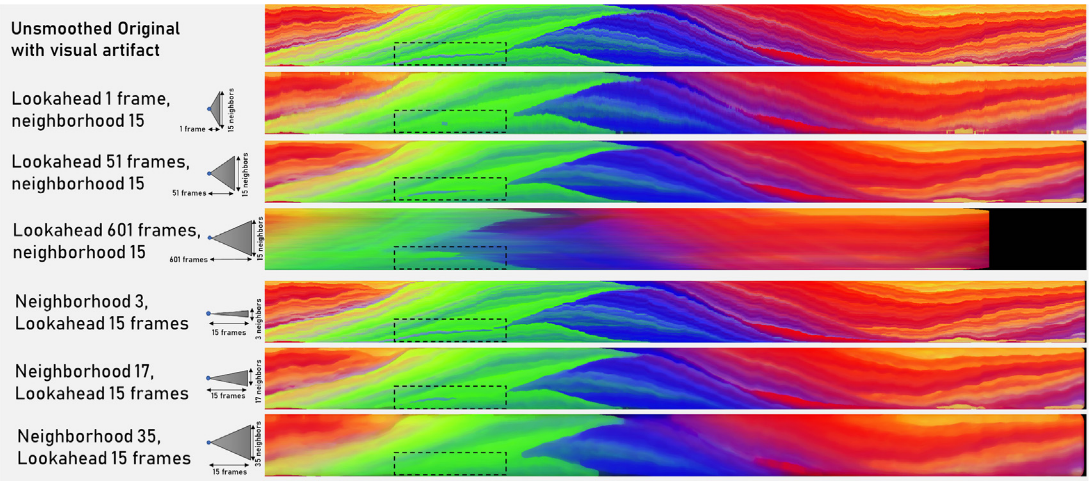


Fig. 6. Comparison of parameter effects to the original. Second to fourth result: Low, medium, and large lookahead. Fuzzy features are smoothed and sharpened, but large values shift and distort patterns. Also, larger values shrink the result in size since the lookahead cannot be larger than the remaining time frames. Fifth to seventh result: Increasing neighborhood size smoothes artifacts as discussed in Section 5 but introduces more blur for higher values. The marked area corresponds to the perceptual artifact also highlighted in Fig. 1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

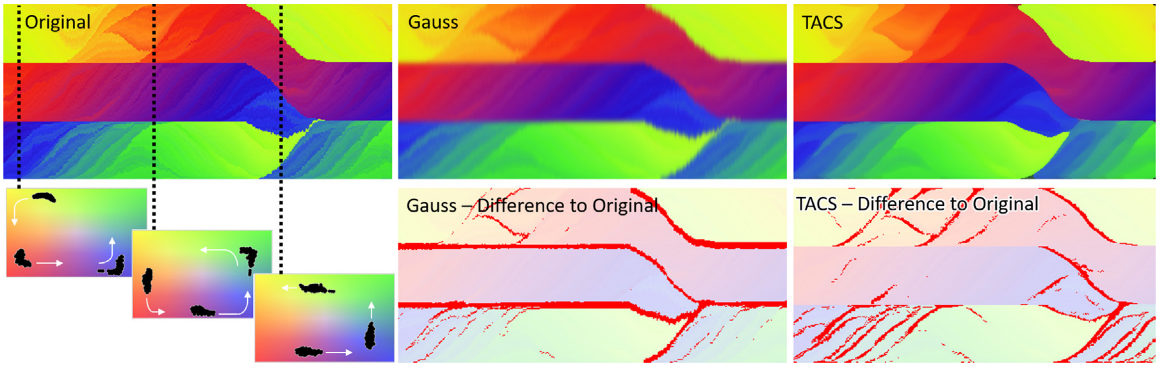


Fig. 7. *SpatialRugs* generated with a dataset of three groups of about 65 movers each, moving counter-clockwise (compare excerpts in the lower left). While the Gaussian smoothed version in the middle blurs the clearly visible borders between the three groups, the TACS smoothed version corrects mostly within each group and sharpens the visible edges instead of fuzzing them. This becomes especially apparent when comparing the difference images in the lower row, outlining large differences in the areas affected by the two smoothing approaches. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the discussed heuristics. The runtimes for the creation of one SpatialRug using the dataset described in Section 1 average at 27 s. They were generated on a desktop PC with Intel Core i7-8550U CPU and 32 GByte RAM. The first notebook contains the image as a matrix of RGB values, the second as the neighborhood size, the third as the frames ahead, the fourth as the steps (in this case, the matrix shape), and lastly, a switch where the steps should start. The initial matrix (image) gets enlarged by ones in height by half of the size of the neighbors at the top and the bottom (one-padding in y-direction). We employ this padding to be able to start at the first pixel at the top left with the whole sliding matrix. Afterward, we slide over the x and y directions and apply our matrix to get all possible colors, sort them, and take the median to derive the new color value. To aid with parameterization, we suggest several basic strategies, including a result estimation using edge detectors. We also provide all results and base images in the code repository to reproduce our own results.

6. Results: Assessing visual outcomes

We next elaborate on the choice of appropriate 2D color maps for *SpatialRugs* and provide statistics on the results of our TACS smoothing method illustrated by examples generated using our technique. As well, we showcase *SpatialRugs* with another dataset with multiple moving groups.

6.1. Color map choice

In Section 4, we proposed an initial set of color maps using the work of Bernard et al. [19] and defined the tasks **I–III**. We further narrow down the selection of well-applicable color maps by visually investigating color space properties (see Fig. 2). First, derived colors should be well distinguishable to relate them to an accurate spatial location, satisfying task **I**. The color maps of Bremm et al. 2 [26], Steiger et al. [28] and Teuling et al. [29] are clearly inferior to their competitors for this property, which is also expressed in their JND value as stated by Bernard et al. Second, task **II** states that the viewer has to maintain a mental map to associate particular colors with spatial positions. Here, the color map provided by Simula et al. [30] introduces a black/dark area between neighboring colors in the corners, impacting the perceptual continuity and potentially introducing false perceptions of brightness and contrast in the dense pixel visualizations we employ here. The color regions by Ramirez et al. [27] and Bremm et al. 1 [26] are also not linearly distributed, thus distorting the distance perception if used as intended by *SpatialRugs*. This leaves the colormaps by Ziegler et al. [33] and Guo et al. [31] as candidates. Ziegler et al. anchor four distinctive colors, amongst them three primary colors, to the corners of the color space, creating a semantic notion of spatial orientation resembling the natural division of four cardinal directions. Guo et al. extend the color space radially around a white center. Both approaches scale well to different aspect ratios, satisfying task **III**. Guo et al. enable to encode the center area in white, as well. Yet, this could interfere perceptually if an additional feature should be encoded as modification of the color brightness or, again, cause issues with brightness and contrast perception, and it would only work if no black or white components are present.

In conclusion, we expect the color maps by Ziegler et al. and Guo et al. to fulfill our tasks, while we expect that Ziegler et al. works better for most cases. Consequently, we use this color map in this publication to illustrate our approach. In addition, as described above, the four-sided anchoring of the color space by Ziegler et al. is a unique feature that can be easily related to four cardinal areas or directions. This is very intuitive for representing spatial areas, and transitions between the clearly distinguishable color areas can also be identified easily. This recommendation is based on the most fundamental tasks for encoding spatial relations with colors in dense pixel visualizations. Yet, more specific use cases could possibly profit from using other color maps. The key decision factor are the user's specific information needs. The suggested color maps enable users to distinguish spatial positions of movers in a 2D cartesian coordinate system. If a polar coordinate system would be applied, encoding the pole region in addition to the cardinal directions can gain importance, favoring color maps with a central reference area such as the one provided by Guo et al. [31]. In another potential use case, not the absolute spatial positions of the movers are regarded, but the change in spatial arrangement over time between the movers as observed from a given reference point. Here, a circular monochrome color map starting from the reference point could enable users to estimate distances, without having to compare different color hues. A tradeoff between the readability of the spatial position versus the encoding of distances becomes apparent. If the user intends to encode spatial context, linear color maps as presented could not be applied anymore, and distance information would be lost. Still, we discuss this possibility briefly in Section 7 and show an initial example in Fig. 8.

6.2. Color smoothing

The time-aware smoothing tries to mitigate the effects of neighboring colors (outlined in red Fig. 4A) by including the temporal color distribution. In Fig. 4A and B, we see that the methods reduce visible outliers while retaining the temporal structures. The difference image between (A) and (B) (see Fig. 4C) provides preliminary evidence for the value of the applied smoothing method as it only affects the color transition areas, leaving the visual patterns still crisp and visible. In contrast, the Gaussian blur (D) creates a fuzzy impression, aggravating the accurate interpretation of colors at a given point by blurring visual structures.

A quantitative assessment of our color-smoothing (table in Fig. 4) shows results of applied quality measures by measuring the distance to the original, unsmoothed image. The measures include the root mean squared error (RMSE) [36], the mean squared error (MSE) [36] and the structural similarity index [37] (SSIM). We compare our time-aware color smoothing (TACS) to a standard Gaussian smoothing (Gauss). Similar reference area parameters are chosen to allow the comparison of the smoothing methods. Lower RMSE and MSE values indicate better results, and a higher value for SSIM indicates better similarity between original and smoothed images. The results indicate that our pooling method outperforms the Gaussian blur even for small sigmas and large window sizes.

6.3. Applicability to other collective movement datasets

The dataset used for the SpatialRugs in Figs. 1, 2, 4 and 6 employs only one group of 151 movers expressing coherent movement behavior. As the application scope of *SpatialRugs* is not limited to single groups of movers, we also evaluate the applicability of our technique to datasets containing more than one group of movers. For our demonstration purposes, we use a synthetic dataset generated using a collective movement data generator [11] which relies on established behavioral models such as the Reynolds model [38] in combination with path following and obstacle avoidance features. The visual representation is created using the spatial linearization provided by MotionRugs [4], the spatial colormap refers is the one argued for in Section 6.1 and referenced in the excerpts on the lower left in the background.

The dataset we generated, shown in Fig. 7, displays the movers moving in three independent clusters (see the excerpts on the lower left of the Figure), following a counter-clockwise movement pattern. In the rug representation, the three groups are clearly distinguishable as three stripes moving in different areas of the dataset. The transitions of the groups between the regions of the 2D color map can be observed very well, generating visible transition patterns between yellow, orange and red, red, purple and blue, and blue and green. Due to the intrinsics of the applied spatial linearization, the clusters switch positions vertically in the last quarter of the visualization. This is an artifact of the spatial linearization technique as provided by MotionRugs [4] and not related to our spatial coloring approach. Techniques to alleviate such artifacts have been proposed in [39].

Since the movers are moving continuously, the observed stripe-like transition patterns originate from two factors: First, the movers do not necessarily move in a uniform distribution through the color space. Second and more important, the transformation of the original square-shaped colormap to adapt to the mover's space lead to a horizontal distortion. This distortion increases the distances between two arbitrary points in the red to purple, purple to blue, and green to yellow areas, meaning more space for more continuously perceived color interpolation. Between yellow and red and between blue and green, on the other hand, the distances between the colors remain short, and visually well distinguishable colors lie closer to each other.

Again, we compare Gaussian smoothing and our TACS and observe difference images in Fig. 8. The clear original representation allows to easily distinguish the three groups of movers and their transitions between the color regions. Gaussian smoothing again blurs the visual result and decreases the saliency of the encoded patterns. The TACS version instead smoothes color transitions and some coarseness in areas such as the blue area in the middle group at the end of the rug. By looking at the difference images comparing Gauss and TACS to the original, a remarkable effect becomes apparent: While the gaussian blurring mostly affects the borders *between* the moving groups, thus worsening their visual delimitation against each other, TACS ignores these areas and instead turns to correct the color space transitions *within* individual groups, which is exactly the expected and desired behavior. This way, the applied TACS parameterization ensures that the correction only applies with respect to close neighbors, ignoring further off entities of other groups, while the lookahead eases the sharpest transitions created by a single group transiting perceptively distinguishable color areas.

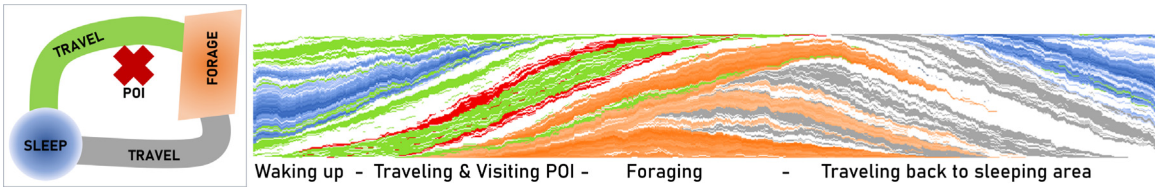


Fig. 8. The principle to encode spatial positions using color can also be applied to define semantic regions. This simulated image shows an example where the different colors encode several predefined areas with semantic meaning. Using the reference map on the left, one can read from the SpatialRug on the right how the movers moved from the blue sleeping area via the green travel area to the foraging area (orange) and back on another route (gray), while some visit a certain POI (red). Unencoded positions appear in white. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In summary, our approach also works for datasets with multiple moving groups. We were able to demonstrate that the parameterization of TACS is suitable to specifically define which areas should be affected by the smoothing and to which degree while retaining the visual saliency of patterns.

7. Discussion and initial expert feedback

We collected more feedback by informal interviews from four domain experts (two on PhD level, two PostDocs) from the area of behavioral ecology to further understand their specific needs when it comes to the representation of spatial features. All are involved with research on collective animal behavior, with a focus on different aspects. The main aspect of their work is the analysis of tracked animal movements yet in largely differing scales ranging from observing the behavior of a rather small group of monkeys in the African desert to large swarms of locusts. Still, the common analysis tasks are very similar: The experts try to understand how the animals coordinate between themselves and how they interact with their environment.

The experts state that two principle approaches are applied in their research: Lab experiments and tracking animals "in the wild". For the former, the animals are observed in a controlled environment to determine how they move or react to precisely specified stimuli. These experiments focus on analyzing how reactions propagate spatially through mover groups, e.g., in schools of golden shiner fish [40]. The latter kind of experiments involve tracked animals in their original, natural environments and thus, draws more attention to the interaction between movers and their surrounding spatial surroundings to learn about behavior specific to certain areas. For example, it is of interest where animals sleep, forage, or roam.

The focus of SpatialRugs is to support the exploration and analysis of collective movement, helping users to retain spatial context and identify areas of interest. Thus, on the one hand, it serves use cases where the semantic spatial context can be disregarded. On the other hand, especially for unexplored datasets, *SpatialRugs* can be applied to identify areas of interest previously unrecognized by presenting users a static representation of spatial mover distribution over time.

The SpatialRugs approach could even be applied to identify movements with respect to semantic spatial context by encoding areas using a color mapping that directly reflects these semantically important locations. Fig. 8 shows an illustrative example: By coloring by semantic contexts such as sleeping area, travel paths, POIs, and foraging areas, the resulting map can be used to identify when movers have been at which position for how long and how they transited between these locations. While we acknowledge the fundamental applicability of SpatialRugs also in semantic spatial contexts such as the described ones or others like administrative areas and boundaries, the resulting design space is complex and requires its own elaboration: The approach would shift the analysis focus from an explorative perspective (i.e., discovering spatial developments and patterns) to a process more oriented towards hypotheses testing, as one has to define points and areas of interest beforehand and assign specific colors. Both approaches could be combined, but the perceptual implications for choosing visually distinguishable color spaces for both semantic areas and non-labeled space are complex and lie beyond the scope of this work.

Initial feedback on the SpatialRugs principle we demonstrated using the data and visualizations shown in Fig. 1 was largely positive, and the approach considered a useful extension of the MotionRugs principle, alleviating the shortcomings of the spatial linearization. According to their statements, the experts were generally able to match the colors to a general region. One expert stated that he thinks that the colormap by Ziegler et al. [33] could possibly be memorized due to the four corner-anchored, leaving it interpretable even without reference to a 2D image. One expert raised concerns about the number of features that can be put into context meaningfully. Another comment was to introduce interactive quantification aids to enable users to measure the distribution of movers in different areas at the same time. With this initial brief feedback, it becomes apparent that the range of possible use cases is broad and covers different aspects of spatial information, varying group sizes, and different grades of expected behavior. Given this degree of complexity, we focus on introducing the *SpatialRugs* approach for the most basic and universal aspect these use cases share, which is the elementary movement exhibited by the observed moving objects.

8. Conclusion and future work

SpatialRugs uses 2D color mapping to allow users to perceive spatial relations in space-efficient visualization designs. The intended use of *SpatialRugs* is as an overview in conjunction with other pixel-based movement visualizations that display further features of interest, enabling to relate space and feature developments. In the MotionRugs context, *SpatialRugs* can be considered a spatial feature encoding (compare SpatialRug and MotionRug in Fig. 1).

We compared several color spaces and discussed perceptual issues following color artifacts, where movements appear to be more distant to each other than their physical distance actually accounts for. To mitigate such distortion effects, we proposed a color smoothing approach (TACS), which we illustrated in examples with different parameterizations and we evaluated TACS using several quality metrics. To find suitable parameter values, we also propose employing edge detectors to find a compromise between excessive smoothing and potential visual artifacts. Our results can be reproduced using our code [35] and base images provided there. We expect that our approach can be applied to non-spatial 2D point distributions as well, e.g., to projections of dynamic datasets. Yet, due to possible contrast effects with the background, a re-evaluation of 2D color spaces would be necessary if such point distributions would be sparse.

The *SpatialRugs* color-coding comes at the cost of several limiting factors. Foremost, the visual interpretability of *SpatialRugs* depends on the ordering technique applied to create the pixel visualization in the first place. For example, the visual outcome deteriorates with increasingly independent movement behavior, which does not create salient visual patterns [39]. As well, we expect that large amounts of individual clusters are harder to interpret due to the (individual) amount of colors an observer can meaningfully distinguish. Since *SpatialRugs* encodes spatial positions in dense pixel displays using the full range of a color map, further properties can hardly be encoded on top of the visualization. To do so, we suggest using MotionRugs encoded with features of interest in conjunction with a SpatialRug of the same data. The same perceptual limitations and the color smoothing process also introduce spatial errors when trying to read precise positions, and balancing the parameterization of the color smoothing for specific use cases can be difficult. As well, in cases where the spatial context of the observed movement plays an important role, we discourage the application of the TACS smoothing due to possible loss of information.

Our approach is not suited for users suffering from limited color perception, who would be severely limited by the amount of perceivable space. We also expect contrast effects as described by Mittelstädt et al. [32], which cannot be measured so far. These aspects need to be evaluated, while guidelines for the correct parameterization have to be explored. In future work, we intend to quantify the viewer's perception of our technique and choice of color spaces. Also, the perceptual implications of our color correction process have to be tested thoroughly. Instead of using a single color map, we anticipate that *SpatialRugs* can benefit from an adaptive color map approach adjusted to the specific movement distributions, user task, covered area and aspect ratio.

We expect *SpatialRugs* to be applied as an overview visualization for users to identify interesting developments. Here, it seems natural to introduce interactions for the user to link areas in the SpatialRug with detail views in more traditional representations, enabling an overview-to-detail workflow. This selection could show the current situation at a point in time on the SpatialRug the user points to, e.g., in a classical 2D plot. More sophisticated selections could be applied, such as a spatio-temporal clustering around the selected position to be displayed in more detail to only focus on spatially close moving entities at a given time. Finally, we would like to investigate visualizing spatial context features as described in Section 7.

Credit authorship contribution statement

Juri F. Buchmüller: Concept, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing, Supervision. **Udo Schlegel:** Concept, Methodology, Software, Visualization, Writing – review & editing. **Eren Cakmak:** Concept, Investigation, Writing – original draft, Writing – review & editing. **Daniel A. Keim:** Validation, Project Administration, Funding acquisition, Resources. **Evanthia Dimara:** Supervision, Investigation, Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partly funded by the German Research Foundation DFG under Germany's Excellence Strategy – EXC 2117 – 422037984 and the EU Horizon 2020 research and innovation programme under grant agreement No 826494.

References

[1] Bellman R, Corporation R, Collection KMR. Dynamic programming. In: Rand Corporation research study, Princeton University Press; 1957.

[2] Sumpter DJ. Collective animal behavior. Princeton University Press; 2010.

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

[4] Buchmüller J, Jäckle D, Cakmak E, Brandes U, Keim DA. Motionrugs: Visualizing collective trends in space and time. IEEE Trans Vis Comput Graphics 2018;25(1):76–86.

[5] Keim DA. Visual exploration of large data sets. Commun ACM 2001;44(8):38–44.

[6] Andrienko N, Andrienko G. Visual analytics of movement: An overview of methods, tools and procedures. Inf Vis 2013;12(1):3–24.

[7] Andrienko N, Andrienko G, Barrett L, Dostie M, Henzi P. Space transformation for understanding group movement. IEEE Trans Vis Comput Graphics 2013;19(12):2169–78.

[8] Hurter C, Tissoires B, Conversy S. Fromdady: Spreading aircraft trajectories across views to support iterative queries. IEEE Trans Vis Comput Graphics 2009;15(6):1017–24.

[9] Tominski C, Schumann H, Andrienko G, Andrienko N. Stacking-based visualization of trajectory attribute data. IEEE Trans Vis Comput Graphics 2012;18(12):2565–74.

[10] Couzin ID, Krause J, James R, Ruxton GD, Franks NR. Collective memory and spatial sorting in animal groups. J Theoret Biol 2002;218(1):1–11.

[11] Piljek I. VisSwarmR: Visual analytics tool for the generation of collective movement datasets. Github; 2020, https://github.com/piljek/VisSwarmR.

[12] Buchmüller J, Schlegel U, Cakmak E, Dimara E, Keim DA. SpatialRugs: Enhancing spatial awareness of movement in dense pixel visualizations. In: EuroVis workshop on visual analytics; 2020.

[13] Lu H, Ooi BC. Spatial indexing: Past and future. IEEE Data Eng Bull 1993;16(3):16–21.

[14] Bergner S, Sedlmair M, Moller T, Abdolyousefi SN, Saad A. Paraglide: Interactive parameter space partitioning for computer simulations. IEEE Trans Vis Comput Graphics 2013;19(9):1499–512.

[15] Luboschik M, Rohlig M, Bittig AT, Andrienko NV, Schumann H, Tominski C. Feature-driven visual analytics of chaotic parameter-dependent movement. Comput Graph Forum 2015;34(3):421–30. http://dx.doi.org/10.1111/cgf.12654.

[16] Burch M, Vehlow C, Beck F, Diehl S, Weiskopf D. Parallel edge splatting for scalable dynamic graph visualization. IEEE Trans Vis Comput Graphics 2011;17(12):2344–53.

[17] van den Elzen S, Holten D, Blaas J, van Wijk JJ. Dynamic network visualization with extendedmassive sequence views. IEEE Trans Vis Comput Graphics 2013;20(8):1087–99.

[18] Cui W, Wang X, Liu S, Riche NH, Madhyastha TM, Ma KL, et al. Let it flow: a static method for exploring dynamic graphs. In: Proceedings of IEEE Pacific visualization symposium. IEEE; 2014, p. 121–8.

[19] Bernard J, Steiger M, Mittelstädt S, Thum S, Keim D, Kohlhammer J. A survey and task-based quality assessment of static 2D colormaps. In: Visualization and data analysis, vol. 9397. International Society for Optics and Photonics; 2015, p. 93970M.

[20] Emery KJ, Webster MA. Individual differences and their implications for color perception. Curr Opin Behav Sci 2019;30:28–33.

[21] Janetzko H, Mansmann F, Bak P, Keim DA. Northern lights maps: Spatiotemporal exploration of mice movement. In: Proceedings of EuroVis 2009 : Eurographics / IEEE-VGTC symposium on visualization; Berlin, Germany, 10 - 12 June 2009; 2009.

[22] Leite RA, Schnorr LM, Almeida J, Alberton B, Morellato LPC, Torres RdS, et al. PhenoVis–A tool for visual phenological analysis of digital camera images using chronological percentage maps. Inform Sci 2016;181–95.

[23] Bernard J, Dobermann E, Vögele A, Krüger B, Kohlhammer J, Fellner D. Visual-interactive semi-supervised labeling of human motion capture data. Electron Imaging 2017;2017(1):34–45.

[24] Peuquet DJ. It's about time: A conceptual framework for the representation of temporal dynamics in geographic information systems. Ann Assoc Am Geogr 1994;84(3):441–61.

[25] Dasgupta A, Poco J, Rogowitz B, Han K, Bertini E, Silva CT. The effect of color scales on climate scientists' objective and subjective performance in spatial data analysis tasks. IEEE Trans Vis Comput Graphics 2018.

[26] Bremm S, von Landesberger T, Bernard J, Schreck T. Assisted descriptor selection based on visual comparative data analysis. Comput Graph Forum 2011;30(3):891–900.

[27] Ramirez C, Argaez M, Guillen P, Gonzalez G. Self-organizing maps in seismic image segmentation. Comput Technol Appl 2012;3(9).

[28] Steiger M, Bernard J, Mittelstädt S, Lücke-Tieke H, Keim D, May T, et al. Visual analysis of time-series similarities for anomaly detection in sensor networks. Comput Graph Forum 2014;33(3):401–10.

[29] Teuling A, Stöckli R, Seneviratne SI. Bivariate colour maps for visualizing climate data. Int J Climatol 2011;31(9):1408–12.

[30] Simula O, Alhoniemi E. SOM Based analysis of pulping process data. In: Engineering applications of bio-inspired artificial neural networks. Berlin, Heidelberg: Springer Berlin Heidelberg; 1999, p. 567–77.

[31] Guo D, Gahegan M, MacEachren AM, Zhou B. Multivariate analysis and geovisualization with an integrated geographic knowledge discovery approach. Cartogr Geogr Inf Sci 2005;32(2):113–32.

[32] Mittelstädt S, Stoffel A, Keim DA. Methods for compensating contrast effects in information visualization. Comput Graph Forum 2014;33(3):231–40.

[33] Ziegler H, Nietzschmann T, Keim DA. Visual exploration and discovery of atypical behavior in financial time series data using two-dimensional colormaps. In: Proceedings of the 11th International conference information visualization. IEEE; 2007, p. 308–15.

[34] Canny J. A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell 1986;(6):679–98.

[35] Schlegel U. Time-aware color smoothing. GitHub; 2020.

[36] Wajid R, Mansoor AB, Pedersen M. A human perception based performance evaluation of image quality metrics. In: Advances in visual computing. Springer International Publishing; 2014, p. 303–12.

[37] Bovik AC. Automatic prediction of perceptual image and video quality. Proc IEEE 2013;101(9):2008–24.

[38] Reynolds CW. Flocks, herds and schools: A distributed behavioral model. In: Proceedings of the 14th annual conference on computer graphics and interactive techniques; 1987. p. 25–34.

[39] Wulms J, Buchmüller J, Meulemans W, Verbeek K, Speckmann B. Spatially and temporally coherent visual summaries. 2019, arXiv preprint arXiv:1912.00719.

[40] Rosenthal SB, Twomey CR, Hartnett AT, Wu HS, Couzin ID. Revealing the hidden networks of interaction in mobile animal groups allows prediction of complex behavioral contagion. Proc Natl Acad Sci 2015;112(15):4690–5. http://dx.doi.org/10.1073/pnas.1420068112.