

# Interactive Visualization of Multivariate Trajectory Data with Density Maps

Roeland Scheepens\*

Niels Willems†

Huib van de Wetering‡

Jarke J. van Wijk§

Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands

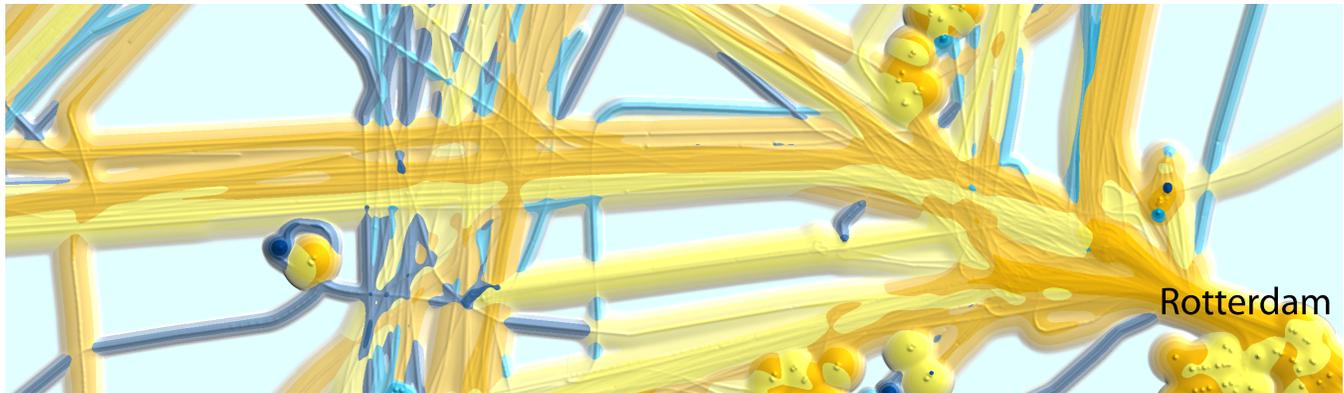


Figure 1: A density map of vessel traffic in front of Rotterdam during a single day. The density map is a combination of four density fields each covering a quarter of the day. The following manually defined color map is used: night is dark blue, morning is bright yellow, afternoon is dark yellow, and evening is bright blue. Furthermore, the saturation of the color represents the density field contribution and the hue is given by the period with the highest density. To discriminate daylight patterns from nighttime, the night and evening use half the kernel radius of the other periods. This figure shows that the main routes are the most used during daylight, while in the night deviations from these routes occur.

## ABSTRACT

We present a method to interactively explore multiple attributes in trajectory data using density maps, i.e., images that show an aggregate overview of massive amounts of data. So far, density maps have mainly been used to visualize single attributes. Density maps are created in a two-way procedure; first smoothed trajectories are aggregated in a density field, and then the density field is visualized. In our approach, the user can explore attributes along trajectories by calculating a density field for multiple subsets of the data. These density fields are then either combined into a new density field or first visualized and then combined. Using a widget, called a distribution map, the user can interactively define subsets in an effective and intuitive way, and, supported by high-end graphics hardware the user gets fast feedback for these computationally expensive density field calculations. We show the versatility of our method with use cases in the maritime domain: to distinguish between periods in the temporal aggregation, to find anomalously behaving vessels, to solve ambiguities in density maps via drill down in the data, and for risk assessments. Given the generic framework and the lack of domain-specific assumptions, we expect our concept to be applicable for trajectories in other domains as well.

**Index Terms:** I.3.3 [Computing Methodologies]: Computer Graphics—Picture/Image Generation

\*e-mail: R.J.Scheepens@tue.nl

†e-mail: C.M.E.Willems@tue.nl

‡e-mail: H.v.d.Wetering@tue.nl

§e-mail: J.J.v.Wijk@tue.nl

## 1 INTRODUCTION

In the analysis of moving objects we search for patterns in trajectory data obtained by sensors. Many movement patterns are classified by Dodge *et al.* [6], and we can also find them in interaction between objects [21], clustering of trajectories [3], and aggregated patterns [2]. In this paper, we focus on aggregated spatial patterns, in particular those from trajectories with various attributes.

With current technology, we are able to track massive amounts of objects with other attributes than just *time* and *position*, such as forces, object type, or object size. Additional knowledge about objects can be added by attributes obtained by reasoning or using web sources [29]. Current state-of-the-art aggregation methods for moving objects often do not take these additional attributes into account. Kernel Density Estimation (KDE) [24] is an aggregation method suitable for showing an overview of massive amounts of data and has been adapted for trajectory data, but only a few of these methods take attributes into account. For instance, Willems *et al.* [28] only encode the velocity of ships and Peters *et al.* [22] only encode the direction of airplanes in a density field. Since a single density field has limited capabilities for further extensions towards including more attributes, we focus on combining multiple density fields [28] for subsets defined with multiple attributes.

In this paper we present a new approach for the interactive exploration of density maps of trajectories with multiple attributes. Interactive exploration requires efficient and effective methods to select subsets of interest as well as to generate density fields. As a result, multiple density fields need to be shown simultaneously in a density map to allow the user to determine similarities and differences. With our generic model, we are able to generate a wide range of different maps, which support story telling to explain what happened with the objects in the data set.

The paper is organized as follows: In Section 2 we discuss work related to density maps. Section 3 gives a model of data and density of trajectories. We describe the main features of our density

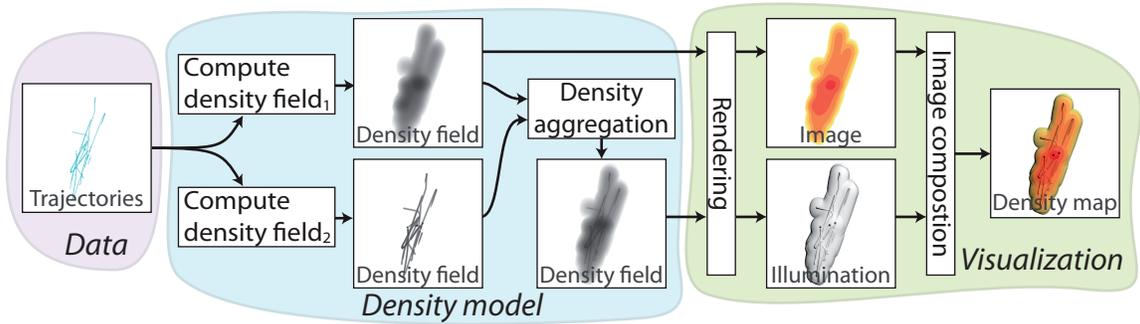


Figure 2: The rendering pipeline of vessel density maps by Willems *et al.* [28]. From left to right, trajectories are smoothed with a large and a small kernel resulting in two density fields. In the rendering, the large kernel density field is used for color mapping and the aggregated density fields are used for the illumination. In the final density map, the color image and the gray-scale image with the illumination are multiplied.

maps in Section 4 and elaborate on the implementation on graphics hardware in Section 5. Section 6 shows the versatility of density maps with use cases. In the final section, we conclude the paper and suggestions for future work.

## 2 RELATED WORK

Visualization is one of the many methodologies that can be used to analyze moving objects. In Section 2.1 we discuss a selection of visualizations related to moving objects. For our visualization we have used techniques related computer graphics, such as convolution as discussed in Section 2.2 and volume rendering in Section 2.3. We conclude with related topics in scientific visualization and cartography in Section 2.4.

### 2.1 Moving Object Analysis

Additional attributes are taken into account in the analysis of moving objects by Dykes *et al.* [7] using multiple views. A visual analytics tool to analyze vessel data is demonstrated by Riveiro *et al.* [23]. Bak *et al.* [4] show by means of glyphs the spatio-temporal characteristics of mouse trajectories in multiple areas. Hurter *et al.* [11] pioneer with high-end graphics hardware for interactive visualizations of massive amounts of airplane trajectories. All these trajectory visualizations have in common that they do not smooth the data, while using multiple attributes.

### 2.2 Convolution

KDEs are generated by convolution of the original data and smooth data to give an overview at various levels of detail. Trajectories can also be convolved by moving a smoothing kernel along a trajectory. To convolve trajectories, a line needs to be convolved, which happens in continuous parallel coordinate plots [10] as well. Jin *et al.* [14] investigate methods to analytically solve the convolution equations with polynomial kernels, resulting in efficient computation. Willems *et al.* [28] visualized vessel traffic at two levels of detail simultaneously revealing both global patterns, such as traffic lanes, and local patterns, such as anchoring zones. Hurter *et al.* [12] extended their own hardware-accelerated visualization with accumulation, where only the sample points of the trajectory are convolved, in contrast to line segments in [28] or a variant in 3D by Demšar [5]. Apart from trajectories other spatio-temporal data can be convolved as well, such as syndromic hotspots by Maciejewski *et al.* [18].

In our approach, we focus on specific subsets by assigning various kernel radii to them. This has a similar effect as a semantic depth of field as proposed by Kosara *et al.* [16], where unimportant parts of an image are convolved with relatively large kernels, to disable pre-attentive vision of sharp features in these areas.

### 2.3 Volume Rendering

In classic volume rendering [17], an iso-surface is constructed based on a transfer function, which is a mapping of a scalar value to a color and an opacity, to visually divide the data set in recognizable surfaces, such as skin and bones. In our approach, we have a similar goal, since we want to divide the data in semantically different subsets, but instead of mapping a density value to a visualization parameter, such as color, we take other attributes into account at the position to be displayed. To keep the pictures interpretable by the user, we use simple step functions, instead of continuously defined higher-dimensional transfer functions as Kniss *et al.* [15] proposed to incorporate multiple attributes in volume data.

### 2.4 Scientific Visualization and Cartography

A convenient representation of a density field is as a regular grid of cells with values, called a raster map. Raster maps are one of the various techniques in multivariate scientific visualization [8]. In cartography, map algebra [26] is a well-known method to mathematically combine these raster maps to derive features. Mennis *et al.* [19] extend map algebra to spatio-temporal data for raster maps that change over time. Raster maps can also be combined visually with a grid of small glyphs as shown by Miller [20].

## 3 MODELS FOR TRAJECTORIES

This section gives a brief overview of the models for trajectory data and a trajectory density field as proposed by Willems *et al.* [28].

### 3.1 Data Model

The movement of an object  $o$  is modeled by a trajectory, which is a sequence  $\tau_o$  of tuples  $\alpha_i^o$ . For an object  $o$ , we abbreviate a tuple as  $\alpha_i$  containing the following elements: a time stamp  $t_i$ , a position  $\mathbf{p}_i$ , and other derived or measured attributes, such as speed  $v_i$ , type, width, and volume. The tuples are ordered by time and most attributes can be interpolated between consecutive tuples.

For tuples  $\alpha_0$  and  $\alpha_1$ , we reconstruct the continuous path  $\mathbf{p}(t)$  with  $t \in [t_0, t_1]$  using the displacement  $x(t)$ , which is derived from the measured positions  $\mathbf{p}_i$ , time stamps  $t_i$ , and speed  $v_i$ :

$$\begin{aligned}
 x(t) &= \frac{1}{2}a(t-t_0)^2 + v(t-t_0) \\
 \text{with } a &= \frac{v_1-v_0}{t_1-t_0} \quad \text{and} \quad v = \dot{x}(t_0) = \frac{\|\mathbf{p}_1-\mathbf{p}_0\|}{t_1-t_0} - \frac{v_1-v_0}{2}, \quad (1) \\
 \mathbf{p}(t) &= \mathbf{p}_0 + x(t) \frac{\mathbf{p}_1-\mathbf{p}_0}{\|\mathbf{p}_1-\mathbf{p}_0\|}. \quad (2)
 \end{aligned}$$

By taking the speed measures into account as well, we obtain a more accurate approximation of the path of the actual movement from the sensor data. With this movement model we are able to compute an accurate density field of trajectories, by moving a smoothing kernel along the actual path of the movement.

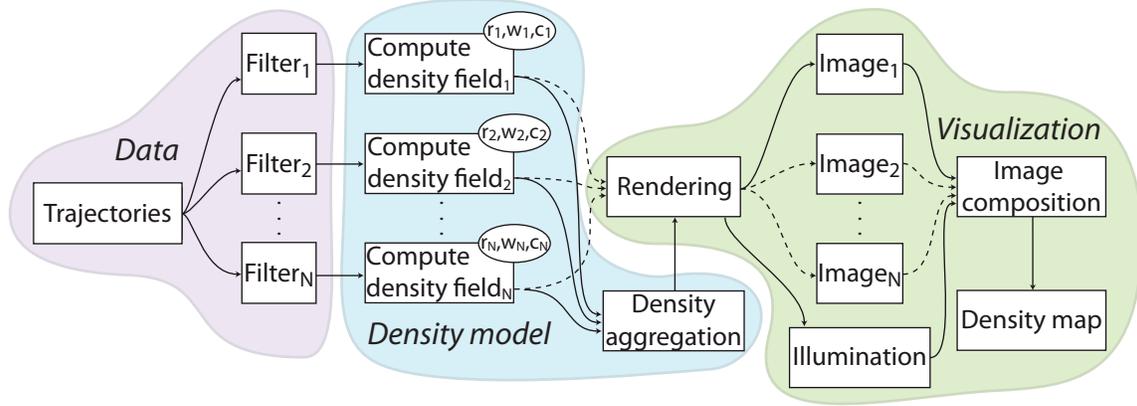


Figure 3: The architecture of our framework to generate density maps. From left to right, the trajectory data is split with various filters resulting in a number of subsets for which we compute a density field with the given weight  $w_i$  and kernel radius  $r_i$ . From here there are two routes. We may aggregate the density fields in a single density field via the solid lines. Then the single density field is rendered to an image with a color map  $c_i$  and visualized together with its illumination in a density map. The other route, also via the dashed lines, first renders the density fields using a color map  $c_i$  and then composes them and applies the illumination of the aggregate density field resulting in a density map.

### 3.2 Density Field

In [28] trajectories are smoothed using a radially symmetric kernel function  $k_r$  with radius  $r$  to obtain a density field with both local and global movement features. A continuous density field is a function  $C: \mathbb{R}^2 \rightarrow \mathbb{R}$ , defined in [28] for a single object  $o$  with path  $\mathbf{p}(t)$  as

$$C_o(\mathbf{q}) = \frac{1}{T} \int_0^T k_r(\mathbf{q} - \mathbf{p}(t)) dt. \quad (3)$$

$C_o(\mathbf{q})$  is the contribution of object  $o$  to the density in point  $\mathbf{q}$ . It is normalized in time to enable comparison of density fields with varying durations. The density field  $C(\mathbf{q})$  of a set of trajectories is the summation of the density fields  $C_o(\mathbf{q})$  of all objects. We obtain a (discrete) density field  $D(Q)$  on a raster of cells by sampling  $C$  in the center of each cell  $Q$  and multiplying by a weight  $w$ .

Figure 2 shows how density fields have been visualized as so called vessel density maps [28]. These maps show variations in speed as different density contributions highlighting significant maritime areas, such as groups of high contributions for anchoring zones where vessels wait to enter a harbor. However, in these maps only a single attribute is visible. For instance, with vessel density maps we can not see the variations of area usage over time. Since trajectories are essentially multivariate data, the generalization of vessel density maps introduced in this paper can take more attributes into account.

## 4 DENSITY MAPS

We present a framework to enhance the vessel density fields as described in the previous section with additional attributes. Figure 3 shows an overview of the new architecture for density maps, in which the architecture for creating vessel density maps recurs. The main principle we apply is that the data is filtered resulting in various subsets, which are aggregated into a single representation in either the density model or the visualization phase. In the remainder of this section we discuss the various aggregations in more detail. Examples of how these features can be used in real-world scenarios are shown in Section 6.

### 4.1 Subsets and Parameters

A density field as defined in Section 3.2 is computed based on a subset of the data defined by a filter. Typically, a filter selects all data elements where the value of a specified attribute is within a

given range. For example, filtering on the time attribute may result in a subset representing all objects in the night. In Section 4.5 we explain techniques for creating filters using multiple attributes.

For each subset a density field is computed with its own user-defined parameters: weight  $w$  and kernel radius  $r$ . The values of these parameters can be set for each subset with an interactive bar chart, with each bar shown with the hue of the color map  $c$  of the density field. Figure 4 shows such a bar chart for the kernel radius of five subsets used in Figure 12 where each subset represents a ship type. The density fields are updated after changing the bar chart.

Once we have a set of density fields we can follow two routes: Either we forward the density fields to the rendering stage and compose the images as explained in Section 4.4, or we can aggregate the density fields into a single density field.

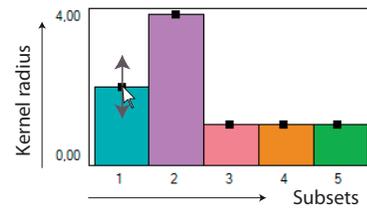


Figure 4: Interactive bar chart to set a density parameter for subsets.

### 4.2 Density Aggregation

For density aggregation we choose the following variations: addition, absolute difference, and anomaly detection (see Figure 5). For a cell  $Q$  density aggregations  $\mathcal{D}$  are defined in terms of densities  $D_i$ :

$$\mathcal{D}_{\text{addition}}(Q) = \sum_{i=1}^N D_i(Q), \quad (4)$$

$$\mathcal{D}_{\text{difference}}(Q) = |D_2(Q) - D_1(Q)|, \text{ and} \quad (5)$$

$$\mathcal{D}_{\text{anomaly}}(Q) = \max(0, D_2(Q) - D_1(Q)). \quad (6)$$

In Equation (4) we define a simple addition for multiple density fields. Note that by choosing the density field weights  $w_i = 1/N$ , we obtain an average density value. With  $\mathcal{D}_{\text{addition}}$  we obtain the vessel density map of [28] if we use two density fields containing both all data and convolve one with a large and one with a small kernel radius. The absolute difference in Equation (5) and the anomaly detection in Equation (6) are defined for two densities. In aggregation

$\mathcal{D}_{\text{difference}}$ , we simply search for a symmetrical difference between the two density fields. The aggregation  $\mathcal{D}_{\text{anomaly}}$  is typically used to find deviations of a sparse density field  $D_2$  containing instances of behavior with respect to a dense density field  $D_1$  representing normal movements. In Figure 5 the gray trajectory is only gray, or is anomalous, at the places where the other trajectory is not located. Section 6.2 shows anomalies with real-world data.

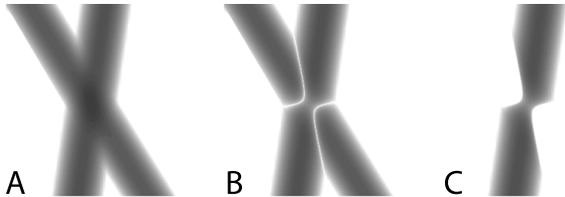


Figure 5: Density aggregation: (A) Weighted addition, (B) absolute difference, and (C) anomaly detection.

### 4.3 Rendering

In the visualization part of the architecture in Figure 3 the density fields are rendered to images by mapping density values to colors via a color map  $c$ . For a small number of density fields a single hue color map is used for each density field. The user is supported by a preset of clearly distinguishable hues to obtain a perceptually balanced map, as used in Figure 12. The preset is a pastel rainbow color map generated with PaletteView [27] and sampled at equal intervals for a given number of colors. The chosen color is associated to the highest density value of the field, and a complete color map is created by interpolating the saturation towards white, see the lower right corner of Figure 11.

If only one density field is visualized a multi-hue color map can be used, as hue is not needed to distinguish between density fields. We can choose, for instance, to use a yellow-to-red color map with more contrast between density levels, as used in Figure 9 and 11A.

### 4.4 Image Composition

Not only density fields but also the color mapped images of density fields can be aggregated. To obtain a density map, we aggregate the color mapped images with an operation called image composition and, finally, apply Phong shading on a height field given by the density aggregation field (see Figure 3).

We distinguish five types of image compositions: single-field  $\mathcal{J}_{\text{single}}$ , aggregate-field  $\mathcal{J}_{\text{aggregate}}$ , opacity-blend  $\mathcal{J}_{\text{opacity}}$ , max-blend

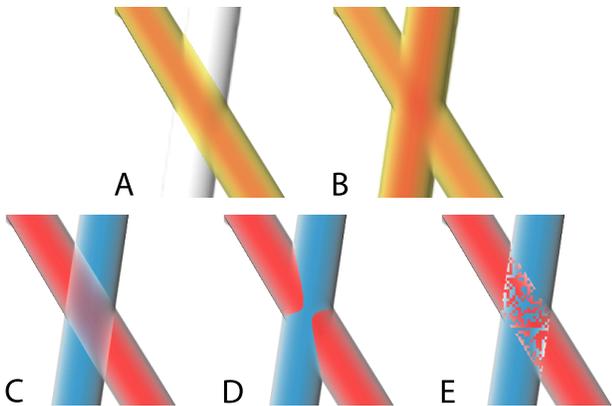


Figure 6: Image composition: (A) Single-field, (B) multi-field, (C) opacity-blend, (D) max-blend, and (E) block.

$\mathcal{J}_{\text{max}}$ , and block  $\mathcal{J}_{\text{block}}$  (see Figure 6). The output of  $\mathcal{J}_{\text{single}}$  is simply one of the color mapped density fields and for  $\mathcal{J}_{\text{aggregate}}$  it is the color mapped aggregated density field obtained by  $\mathcal{D}$ . Since these two image compositions have a single density field as input they can use a multi-hue color map. In the vessel density shown in Figure 11A  $\mathcal{J}_{\text{single}}$  is used, in Figure 9  $\mathcal{J}_{\text{aggregate}}$  is used.

The  $\mathcal{J}_{\text{opacity}}$  composition is a weighted sum of the colors of the rendered density fields in RGB-space with the opacity as weight. With  $\mathcal{J}_{\text{max}}$  we show the color of the density field with the highest value, as used in Figure 1. From the latter two image compositions it is hard to see which colors are used in a certain neighborhood, therefore we introduce  $\mathcal{J}_{\text{block}}$ . In this image composition, a new image is created covering the same area as the input images, but with a much lower resolution to avoid dithering effects. The color of the pixels in this new image are set as follows: If a pixel overlaps with one or more density values larger than zero, one of these density fields is randomly chosen and the corresponding color is used. Hagh *et al.* [9] have evaluated this composition, called color weaving, and show that it is more effective than color blending ( $\mathcal{J}_{\text{opacity}}$ ) for two to four colors, while for upto six colors the error rates significantly increase. In Section 6.3,  $\mathcal{J}_{\text{block}}$  is used to reveal various ship type in a neighborhood.

### 4.5 Multivariate Filters

So far, filters for single attributes have been considered. We can expand the usability of our method greatly by defining interactive filters based on multiple attributes. To this end, we have developed an interactive widget, called a Distribution Map (DM), as shown in Figure 7 with various pairs of attributes used in vessel tracking.

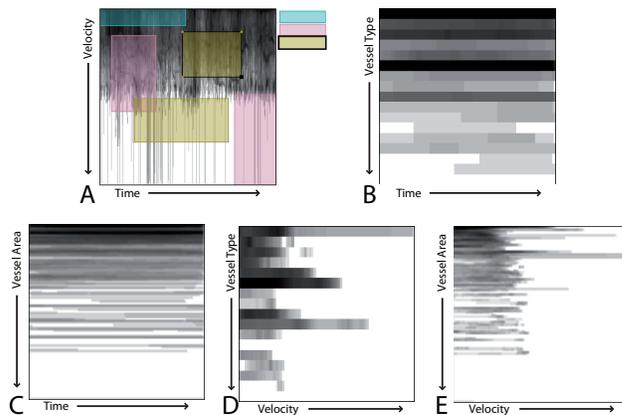


Figure 7: Distribution maps from pairs of vessel attributes.

A DM consists of a 2D-plane with two axes corresponding to two chosen attributes and shows in gray-scale the distribution of the values of these attributes as they appear in the trajectories. Guided by this distribution, the user, while defining a filter, selects combinations of attribute intervals by drawing a set of rectangles, colored with the same hue as the color map  $c$  for rendering the density field. For example the pink rectangles in Figure 7A represent medium-speed moving vessels in the morning together with fast moving vessels in the evening. Adaptations of the rectangles result in an update of the density field of the corresponding subset.

The distributions are generated by drawing, with additive blending, bright gray polylines as trajectories for the two attributes given on the axes. With this technique we mimic counting how long an attribute pair occurs in the trajectories.

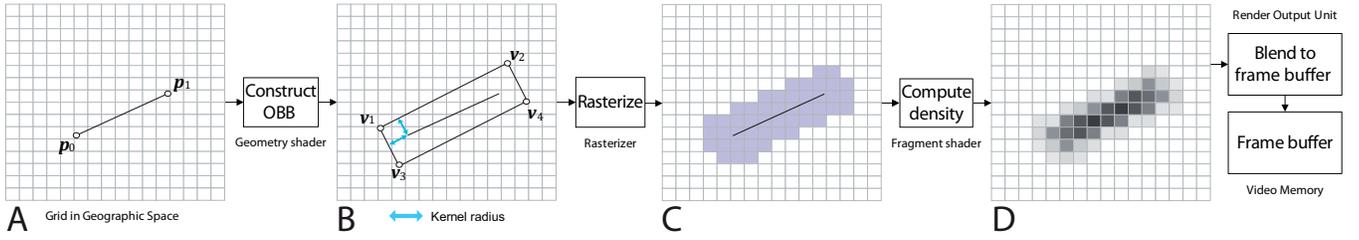


Figure 8: The pipeline for computing density fields on a GPU: (A) The trajectory segment  $\mathbf{p}_0\mathbf{p}_1$  given in geographic coordinates. (B) The segment mapped to map coordinates using the cylindrical equal-area projection and its OBB  $\mathbf{v}_1\mathbf{v}_2\mathbf{v}_3\mathbf{v}_4$  at distance  $r$ . (C) The fragments inside OBB. (D) The density field computed for each fragment in OBB. This density field is then added to already computed densities of other segments.

## 5 IMPLEMENTATION

In this section we give a method for parallel computation of density fields on the GPU based on the non-parallel version in [28]. As in that algorithm we assume that the trajectory points are given in geographic coordinates and a cylindrical equal-area map-projection  $G$  [25] is used to transform these in map coordinates. Given the density fields the visualization of the final density map is straightforward, also realized on the GPU, and omitted here.

### 5.1 Parallelization

A method for parallelization of density field computations either has an image-space or an object-space approach. In an image-space approach the parallelization is done over computations per pixel, where in an object-space approach it is done over computations per object. Here we chose an object-space approach where we simply traverse the objects, in this case the trajectory segments. Additional advantages are that adding and removing trajectory segments is trivial and that the regular graphics pipeline elements, such as the rasterizer and the render output unit, can be used without explicit synchronization to obtain high throughput.

---

#### Algorithm 1 ParallelComputeDensityField()

---

```

CopyTrajectoriesToVideoMemory()
 $D \leftarrow \mathbf{0}_{H,B}$  {D becomes zero matrix}
for  $o \in \text{Objects do}$ 
  for tuples  $\alpha_i$  and  $\alpha_{i+1}$  in trajectory  $\tau_o$  do in parallel
     $\mathbf{p}_0 \leftarrow \mathbf{p}_o(t_i); \mathbf{p}_1 \leftarrow \mathbf{p}_o(t_{i+1})$ 
     $OBB \leftarrow$  oriented bounding box at distance  $K$  to  $\widehat{\mathbf{p}_0\mathbf{p}_1}$ 
    for  $(u, v) \in OBB$  do in parallel
      {Get geographical coordinate of the cell center}
       $\mathbf{q} \leftarrow G^{-1}(u, v)$ 
      {Assign density based on Equation (3) in Section 3.2}
       $D(u, v) \leftarrow D(u, v) + C_{\widehat{\mathbf{p}_0\mathbf{p}_1}}(\mathbf{q})$ 
return  $D$ ;

```

---

Our approach is given in pseudo code in Algorithm 1 and illustrated in Figure 8. We first construct a texture according to the subdivision of our geographic space into uniform cells with equal area. For each vessel  $o$  we render its trajectories to this texture that ultimately represents the density field. For each trajectory we handle all line segments between two consecutive tuples  $\alpha_i$  and  $\alpha_{i+1}$ . Line segments are processed in parallel by geometry shaders that construct, given a line segment  $\mathbf{p}_0\mathbf{p}_1$ , an oriented bounding box (OBB) for the points at most distance  $r$ , the kernel radius, from the rhumb line  $\widehat{\mathbf{p}_0\mathbf{p}_1}$  [1] and return four vertices defining an OBB (see Figure 8B). Using the rasterizer the OBB is then subdivided into fragments (see Figure 8C), representing the cells to whose density the object  $o$  contributes. For each fragment  $(u, v)$  a fragment shader is run that computes the geographical coordinates  $\mathbf{q}$  of the fragment (see Figure 8D) using the inverse of the cylindrical equal-area

map-projection  $G$ . Finally, the density contribution  $C_{\widehat{\mathbf{p}_0\mathbf{p}_1}}(\mathbf{q})$  is additively blended into density field  $D$  by the render output unit.

Some issues are not handled in the pseudo code of Algorithm 1. First, for reasons of accuracy almost stationary objects are handled different in the sense that their density is computed by drawing a texture mapped fat point with radius  $r$ . Second, the standard formula for computing the great circle distance, which is inaccurate for points at small distances, is replaced by an approximation that is stable for such points. Third, numerical integration of Equation (3) uses Simpson's rule. And finally, kernel evaluation is a texture look up, which allows to us to use arbitrary finite-support kernels.

### 5.2 Performance

The pictures and performance tests in this paper have been generated on an Intel Core i7 CPU at 2.8 GHz with 6 GB of RAM memory and an NVidia GeForce GTX 285 with 1 GB of video memory. The computation time of visualizing a single density field depends on the size of the density field, the kernel radius, the number of steps in the integral approximations, and the size of the data set. For a typical data set, containing data from a single day in the North Sea with 1390 objects and 306,521 data points, a density field of 1250x1020 cells, representing an area of 125 km by 102 km, is computed and displayed on a screen of 900x900 pixels in, on average, 0.77 seconds for a kernel radius of 2 km. This does not give full interactive rate, but this can be accomplished by diminishing the accuracy of the density map *during interaction*. There are several ways to do so. First of all reducing in the above example the density map to 125x102 cells and using Catmull-Rom spline interpolation for intermediate values reduces the computation time to 0.13 seconds. The cell size is then still half the kernel radius and the overall appearance of the image remains intact. Alternatively, the number of numerical integration steps can be reduced from the default 10 steps to 5 steps to gain up to a factor of two in performance while still obtaining acceptable results.

The interactivity could suffer when computing multiple density fields for a single density map. However, in practice, this is not problematic. If for the density map the data has been split in several subsets, one for each density field, the total computation time is approximately the computation time of a density field using the complete data set. If the density map requires multiple density fields using the full data set, the interaction is mostly such that only one density field needs recomputation and in this case only the initial computation of the density map is relatively slow.

## 6 MARITIME USE CASES

The density map framework is intended for expert users, who explore distributions of attributes defined along trajectories. In the exploration, the user mainly interacts with the distribution maps, though the framework is extensive in fine tuning for optimizing details in the density maps. One of the possible outcomes of the exploration is a density map, which can be used as a static overlay

in addition to the regular operational maps containing live traffic for supporting monitoring tasks. To show the expressiveness of our framework, we have defined various use cases towards open problems in the analysis of moving objects with a density approach. The use cases are taken from the maritime domain and concern trajectory data from vessels.

Professional vessels with a gross tonnage of three hundred tons and up are obliged to broadcast their current status using the Automatic Identification System (AIS) [13]. The trajectory data consists of many attributes, such as time, location, ship type, dimensions, destination, and so forth. AIS is used for safety and security by either captains sailing on board of a ship, or by surveillance operators guarding coastal areas. Historical AIS data can be used for planning of the spatial usage of coastal areas, for instance by validating whether or not all ships have followed the rules. The data set contains both route-bounded vessels (e.g., tankers, cargo ships) and non-route-bounded vessels (e.g., pleasure craft, tugs), hinting that our method works for both constrained trajectories (e.g., cars, trains) and unconstrained trajectories (e.g., animals, pedestrians).

### 6.1 Temporal Aggregation

In a vessel density map [28], the order in which movements take place is lost, since all trajectories are convolved with the same smoothing kernel. With the new density maps, we can vary the kernel radius and kernel weight over time. Figure 9 shows a single day of vessel traffic in front of Rotterdam harbor with four subsets of six hours. During the day, the kernel radius is decreasing, while the weights are increasing, and by using density aggregation  $\mathcal{D}_{addition}$  various moments in time are distinguishable. The resulting density map consist of a rendering in both color and illumination of a single aggregated density field using  $\mathcal{J}_{aggregate}$  image composition.

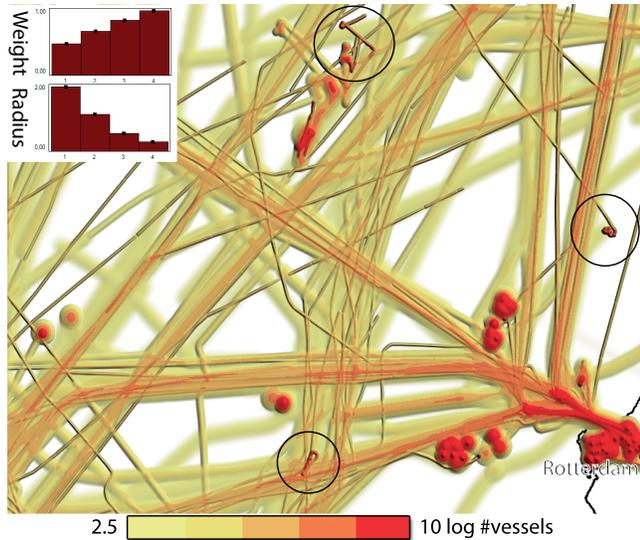


Figure 9: An aggregation of four density maps each covering six hours of a day using addition as density aggregation.

In Figure 9 we see that the subset in the evening, shown with small and dark trajectories are highlighted, while the others serve as a context. Noticeable movement patterns in these trajectories in the evening that were not visible before are encircled. In the circles we clearly distinguish relatively high density along narrow tracks indicating slow moving night ships. Notice that we can also show variations over time using image composition  $\mathcal{J}_{max}$  as displayed in Figure 1.

### 6.2 Anomaly Detection

A density field of a reasonable amount of trajectories represents the nautical history in an area, indicating, for instance, which movements are usual. By comparing other trajectories with this density field, we can determine abnormal behavior in areas where the density field values are low. Density maps can be used to show these anomalously behaving vessels. Figure 10 shows the result of the density aggregation  $\mathcal{D}_{anomaly}$  of two density fields: one with six days of data between Amsterdam and Scheveningen and a density field with the traffic from the last two hours, which can be animated by varying the current time. For the latter one, the kernel radius is decreasing backwards in time, i.e., the head is the most current position. The resulting density field is shown with  $\mathcal{J}_{single}$  image composition and displays potential anomalies in color from white (none) and green (low) to red (high) in the context of all data shown in the shading. This example shows how density based anomaly detection can be used in a real-time system.

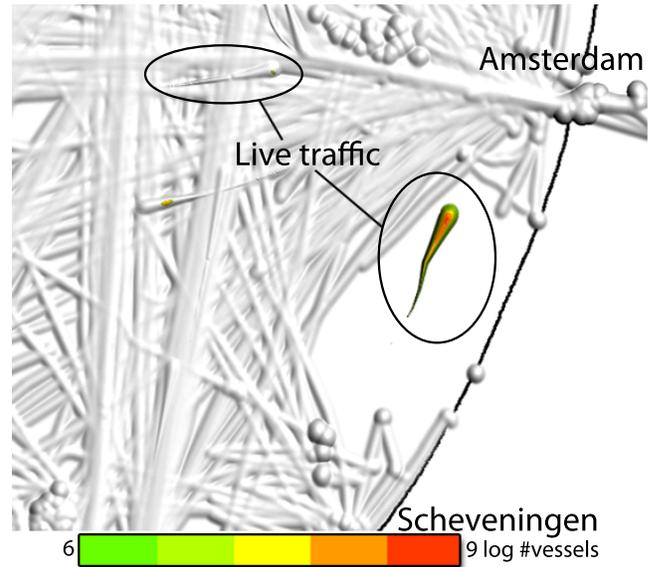


Figure 10: A vessel sailing between Amsterdam and Scheveningen is marked as anomalous, since normally no vessels sail in this area.

### 6.3 Stopping Areas

A vessel density map [28] highlights anchoring zones as groups of dark dot-like parts of trajectories, as shown in the rectangles in Figure 11A. However, similar dark features popup for other slow movements, which are not in anchoring zones. In this use case, we try to find explanations for what happens in these areas.

First, we isolate the areas with slow movements by defining two subsets using the DM shown in Figure 11B; a red subset for stationary vessels and a blue one for moving vessels. Using image composition  $\mathcal{J}_{max}$  we show the colors assigned to the subsets in the density map of Figure 11C. The official anchoring zones are marked with an anchor. Area number 7 is Rotterdam harbor. The other areas with slow movement need additional inspection.

The type of a vessel and the way it moves often explains what happens. Therefore, we change the time axis of the DM to velocity and change the velocity axis to vessel type (see Figure 11D) and define six subsets with slow moving vessels of different types. The colors of the subsets in the DM correspond with the legend on the bottom right of Figure 11. To find the various subsets available in these small areas we use  $\mathcal{J}_{block}$  for image composition. In Figure 11E we see a cargo ship and a nearby tanker. When decreasing the

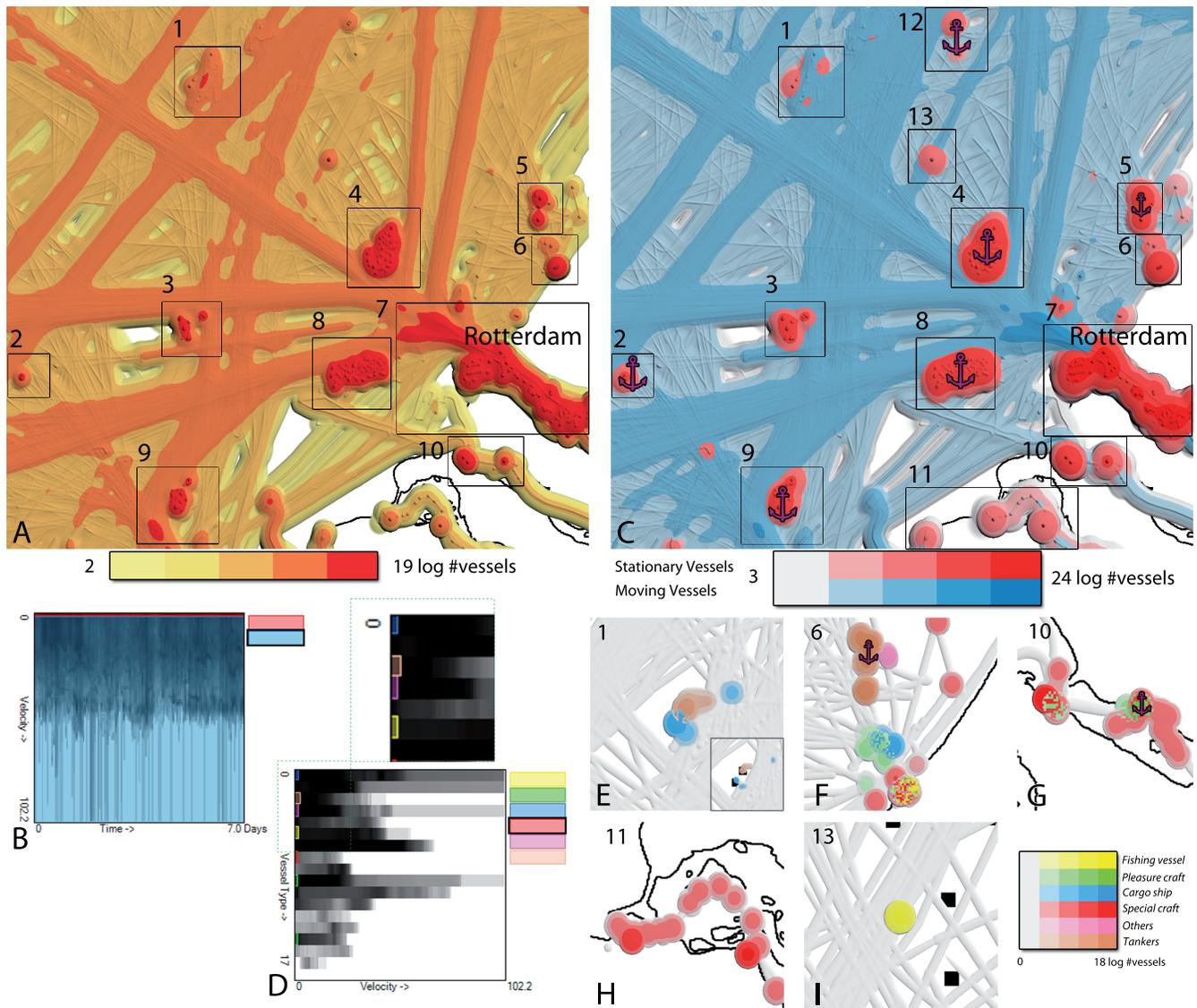


Figure 11: Classifying behavior of slow moving vessels in front of Rotterdam harbor during one week. In (A) a vessel density map [28] is shown and in (C) slow moving areas are isolated using the DM in (B) and anchor zones are marked with an anchor. After defining different types of slow moving vessels with the DM in (D) we can figure out what happened in the zoomed in areas (E...I), which correspond with the numbers in (C).

kernel radius black squares indicating oil platforms become visible (see inset) and actually explain the slow moving vessels. The surroundings of the harbor of Scheveningen with its popular beach are shown in 11F. Some typical vessels close to the coast are visible via a manually defined color map: fishing vessels (yellow), pleasure craft (green), special craft, e.g., a rescue vessel, in red, and others, often small crafts, in pink. Noticeable in this area is a cargo ship (blue) relatively close to the beach; this is suspicious as such ships are not expected close to the coast. In Figure 11G, there are two hotspots with small vessel types; the one on the right is the harbor of the city Hellevoetsluis, while the left one is near the Haringvlietdam. The existence of a lock explains the concentration of waiting ships on the bottom left-hand side of the Haringvlietdam and also explains why only small vessels sail in this neighborhood. By clicking on vessels we find in Figure 11H that the special crafts are a dredger at work (left) and a law enforcement vessel (right). In the course of its duty the latter stops multiple times. Finally, Figure 11I shows a potential threat since a fisher stops in a sea lane.

#### 6.4 Risk Assessment

Some vessels are more dangerous than others. In this use case, we create a density map showing the possible risks of various vessel types. In the DM we put the vessel type and the area on the axes and define the following subsets, with 'large' being a larger area than 9000 m<sup>2</sup> and 'small' the rest: Large cargo vessels (blue), large tankers (purple), small passenger ships (pink), small high speed craft (orange) and other type of small ships (green). The color mapped densities are composed with  $\mathcal{J}_{max}$  to show the most dangerous types in a certain area. To illustrate possible risks of dangerous vessels, we have identified three classes of danger and we have assigned a large kernel with high weight for more dangerous vessel types. In this risk density map we can see that the shoreline is not put in danger, and that dangerous tankers take the left route, while less dangerous cargo vessels take the route closer to the shore.

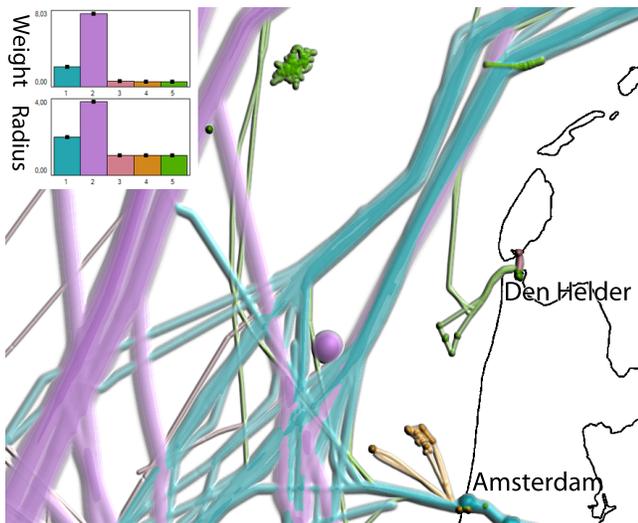


Figure 12: A risk map with various vessel types: large cargo vessels (blue), large tankers (purple), small passenger ships (pink), small high speed craft (orange), and other type of small ships (green).

## 7 CONCLUSIONS AND FUTURE WORK

We have presented a method to explore multivariate trajectories with density maps. To support massive real-world trajectory data sets, we have significantly improved the density field computation time with respect to previous implementations by means of high-end graphics hardware. As a result, we are able to combine multiple density fields quickly, which allows us to enrich density maps. The combination of density fields takes place in either density aggregation or the combination of images of density fields, which can be interactively defined by the user. Density aggregation is typically useful for quantitative analysis, while image composition enables to make a distinction between subsets. The set of aggregations tends to be fairly complete, though the framework can be extended easily with new aggregations. We have applied our method to vessel trajectories allowing us to reveal *what* takes place *when*, find anomalously behaving vessels, drill down on attributes to solve ambiguities in density fields, and conduct risk assessments.

In future research, we will further investigate the parameters in our method, the interaction to set them, and possibilities for new visual cues to render these parameters. A user study should answer how effective density maps are for analysis tasks. The framework may be extended with a multi-pass technique that takes a density field as a parameter for another density field. For instance, we can vary the kernel radius based on another density field. For vessels, this could lead, for instance, to more features in busy harbors, if the kernel radius is decreased for higher densities. Furthermore, the direction is an important feature of trajectory data, which has not been addressed yet in density maps in an appropriate way. Finally, we are interested if the method can support analysis of moving objects in other domains, by exploring different data sets.

## ACKNOWLEDGEMENTS

We thank the reviewers and Hans Hiemstra for their feedback. This work has been carried out as a part of the Poseidon project at Thales Nederland under the responsibilities of the Embedded Systems Institute (ESI). This project is partially supported by the Dutch Ministry of Economic Affairs under the BSIK program.

## REFERENCES

[1] J. Alexander. Loxodromes: A rhumb way to go. *Mathematics magazine*, 77(5):349–356, Dec. 2004.

[2] G. Andrienko and N. Andrienko. Spatio-temporal aggregation for visual analysis of movements. *IEEE VAST*, pages 51–58, October 2008.

[3] G. Andrienko *et al.* Interactive visual clustering of large collections of trajectories. *IEEE VAST*, pages 3–10, October 2009.

[4] P. Bak *et al.* Spatiotemporal analysis of sensor logs using growth ring maps. *IEEE TVCG*, 15(6):913–920, 2009.

[5] U. Demšar and K. Verrantaus. Space-time density of trajectories: Exploring spatio-temporal patterns in movement data. *International Journal of Geographical Information Science*, 24(10):1527–1542, October 2010.

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

[7] J. A. Dykes and D. M. Mountain. Seeking structure in records of spatio-temporal behaviour. *Computational Statistics & Data Analysis*, 43(4):581–603, 2003. Data Visualization.

[8] R. Fuchs and H. Hauser. Visualization of multi-variate scientific data. *Computer Graphics Forum*, 28(6):1670–1690, 2009.

[9] H. Hagh-Shenas, S. Kim, V. Interrante, and C. Healey. Weaving versus blending. *IEEE TVCG*, 13:1270–1277, 2007.

[10] J. Heinrich and D. Weiskopf. Continuous parallel coordinates. *IEEE TVCG*, 15:1531–1538, 2009.

[11] C. Hurter, B. Tissoires, and S. Conversy. FromDaDy: Spreading aircraft trajectories across views to support iterative queries. *IEEE TVCG*, 15(6):1017–1024, 2009.

[12] C. Hurter, B. Tissoires, and S. Conversy. Accumulation as a tool for efficient visualization of geographical and temporal data. *AGILE Workshop Geospatial Visual Analytics: Focus on Time*, May 2010.

[13] ITU. Technical characteristics for an automatic identification system using time division multiple access in the VHF maritime mobile band. *Recommendation ITU-R M.1371-1*, 2001.

[14] X. Jin and C.-L. Tai. Analytical methods for polynomial weighted convolution surfaces with various kernels. *Computers & Graphics*, 26(3):437–447, June 2002.

[15] J. Kniss *et al.* Multidimensional transfer functions for interactive volume rendering. *IEEE TVCG*, 8:270–285, 2002.

[16] R. Kosara, S. Miksch, and H. Hauser. Focus+context taken literally. *IEEE CG&A*, 22:22–29, 2002.

[17] M. Levoy. Display of surfaces from volume data. *Computer Graphics and Applications*, 8(3):29–37, May 1988.

[18] R. Maciejewski *et al.* A visual analytics approach to understanding spatiotemporal hotspots. *IEEE TVCG*, 99(RP):205–220, 2009.

[19] J. Mennis, R. Viger, and C. D. Tomlin. Cubic map algebra functions for spatio-temporal analysis. *Cartography and GIS*, 32:17–32, 2005.

[20] J. R. Miller. Attribute blocks: Visualizing multiple continuously defined attributes. *IEEE CG&A*, 27:57–69, 2007.

[21] D. Orellana *et al.* Uncovering interaction patterns in mobile outdoor gaming. *International Conference on Advanced Geographic Information Systems & Web Services*, pages 177–182, 2009.

[22] S. Peters and J. M. Krisp. Density calculation for moving points. *AGILE Int. Conference on Geographic Information Science*, 2010.

[23] M. Riveiro, G. Falkman, and T. Ziemke. Visual analytics for the detection of anomalous maritime behavior. In *International Conference Information Visualization*, pages 273–279, 2008.

[24] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Number 26 in Monographs on Statistics and Applied Probability. Chapman & Hall, 1992.

[25] J. P. Snyder. *Flattening the Earth: Two Thousand Years of Map Projections*. University of Chicago Press, 1993.

[26] C. D. Tomlin and J. K. Berry. A mathematical structure for cartographic modeling in environmental analysis. *Symposium of the American Congress on Surveying and Mapping*, pages 269–283, 1979.

[27] M. Wijffelaars, R. Vliegen, J. J. van Wijk, and E.-J. van der Linden. Generating color palettes using intuitive parameters. *Computer Graphics Forum*, 27(8):743–750, May 2008.

[28] N. Willems, H. van de Wetering, and J. J. van Wijk. Visualization of vessel movements. *Computer Graphics Forum*, 28(3):959–966, 2009.

[29] N. Willems, W. R. van Hage, G. de Vries, J. H. Janssens, and V. Malaisé. An integrated approach for visual analysis of a multi-source moving objects knowledge base. *International Journal of Geographical Information Science*, 24(10):1543–1558, October 2010.