

Interactive Density Maps for Moving Objects

Roeland Scheepens, Niels Willems, Huub van de Wetering, and Jarke J. van Wijk
■ Eindhoven University of Technology

Objects move in trajectories. Analysts search for patterns in trajectories to understand why objects move in a certain way (see the “Trajectory Data Model” sidebar). In many domains, such analysis contributes to the design and maintenance of the space in which objects move. This includes traffic-monitoring systems, urban design, and wildlife migration.

There are basically two approaches to trajectory analysis. In the first one, analysts look at a small number of trajectories and search for specific patterns.¹ For humans, these patterns can be people walking next to each other, waiting for a bus, meeting someone, or pursuing something. In the second approach, analysts investigate all trajectories at once and search for locations with common patterns. For ships, these patterns can be dropping anchor before entering a harbor or sailing on mandatory routes. Here, we focus on the second approach, using visualization tools.

Current technology lets us track massive numbers of objects. For instance, we can use GPS to obtain time and position information. Analysts can enrich this data with other attributes by using more sensors, using questionnaires, including data about the objects from the Web, or reasoning about existing attributes.² However, with current visualization methods (see the “Related Work” sidebar), analyzing all these attributes for all trajectories is difficult.

One way to visualize large numbers of trajectories is by smoothing them—similar to blurring

an image. This technique is similar to kernel density estimation (see the “Trajectory Smoothing” sidebar).³ That method moves a kernel along the trajectory, smearing the trajectory’s contribution to the density over the neighborhood.^{4,5} This produces a *density field*, a regular grid of cells with density values. By manipulating the kernel, you can encode speed⁶ and direction⁷ in the density, but there has been no way to encode arbitrary attributes.

To alleviate this situation, we’ve taken a different approach. In it, analysts create several filters, and our method computes a density field for each. We call the visualization of these combined fields a *density map*. The remaining questions are, how do we find the right filters, and how do we display these density fields together?

Creating Density Maps

Figure 1 shows the architecture for creating density maps. In the data phase, analysts split the trajectories using several filters. In the density model phase, we compute a density field for each filter, using a model that moves a kernel along the trajectory with the same speed as the object.⁶ Each density field has several visualization parameters, such as kernel radius r , weight w , and color c . Besides the basic density fields, we compute an aggregated density field.

In the visualization phase, we render each density field as a separate image. The basic density fields are color-mapped or rendered with icons. We render the aggregated density field as an illuminated height field. During image composition, we combine all the rendered images to get a density map. Analysts follow the architecture by setting the filter, aggregation, and composition parameters, using widgets and standard user-interface components.

Density maps show trends in objects’ trajectories. Density map creation involves aggregating smoothed trajectories in a density field and visualizing the field. Using an interactive distribution map, users can define subsets and, supported by graphics hardware, get fast feedback for these computationally expensive density field calculations.

Trajectory Data Model

An object moves in continuous time and space. However, a computer can store only a finite amount of data. So, movement is sampled in time. An approximating interpolation between the samples can reconstruct the original continuous movement. We used the following model in the main article.¹

Suppose we have an object $o \in O$. Its sampled trajectory is a list of tuples α_i^o , or α_i for a single object, containing a time stamp t_i , a position \mathbf{p}_i , and other derived or measured attributes, such as velocity v_i , type, width, and volume. Between two consecutive tuples—for instance, the first two tuples α_0 and α_1 —we reconstruct $\mathbf{p}(t)$, with $t \in [t_0, t_1]$, by placing it between the measured points \mathbf{p}_0 and \mathbf{p}_1 at displacement $x(t)$, modeled by

$$x(t) = \frac{1}{2}a(t - t_0)^2 + v(t - t_0),$$

with the reconstructed acceleration a ,

$$a = \frac{v_1 - v_0}{t_1 - t_0},$$

and reconstructed velocity v ,

$$v = \dot{x}(t_0) = \frac{\|\mathbf{p}_1 - \mathbf{p}_0\|}{t_1 - t_0} - \frac{v_1 - v_0}{2},$$

$$\mathbf{p}(t) = \mathbf{p}_0 + x(t) \frac{\mathbf{p}_1 - \mathbf{p}_0}{\|\mathbf{p}_1 - \mathbf{p}_0\|}.$$

By taking v_i into account, we obtain a better approximation of the original movement's path. All other continuously defined attributes can be interpolated in time or in space using path \mathbf{p} .

Reference

1. N. Willems, H. van de Wetering, and J. J. van Wijk, "Visualization of Vessel Movements," *Computer Graphics Forum*, vol. 28, no. 3, 2009, pp. 959–966.

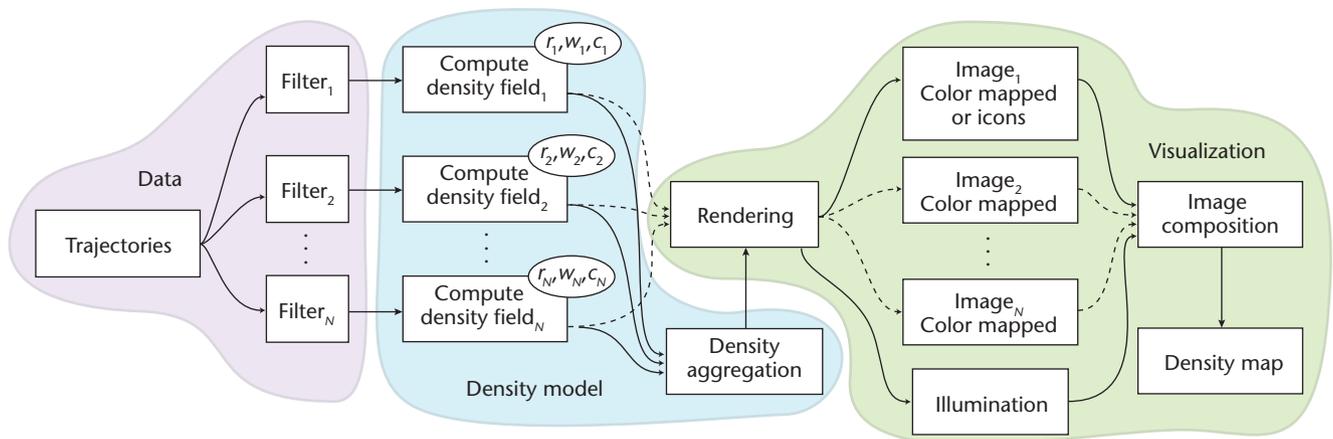


Figure 1. Our architecture for generating density maps. First, various filters split the trajectory data. For each filtered subset, we compute a density field with a given weight w and kernel radius r . Next, we aggregate all density fields and send the result to the rendering component. In that component, we can visualize the fields separately in images obtained by using a single-hue color map with color c . Or, we can combine the density fields into a single image, which adds the option to use icons and a multihue color map. Additionally, we render an illuminated height field from a single density field as a grayscale image. Finally, we compose all the images into a density map.

Interactively Specifying Density Fields

To define filters, analysts use the distribution map (DM) widget. A DM consists of a 2D plane with two axes corresponding to two chosen attributes. Figure 2a shows DMs with various combinations of attributes. The grayscale distribution shows the durations of attribute-value pairs along the trajectories.

To define a filter, analysts draw a set of rectangles, simultaneously defining the two attributes' ranges. A rectangle has the same color as its related density field. For example, the two pink rectangles

in Figure 2a represent medium-speed ships in the morning and fast ships in the evening. When analysts change a rectangle, the system updates the corresponding density field.

Analysts can interactively change the remaining parameters for each density field. Analysts change color and weight using an interactive bar chart for each parameter (see Figure 2b). They can raise or lower the bars and color them with the density field's color. The system updates the fields after the bar chart changes.

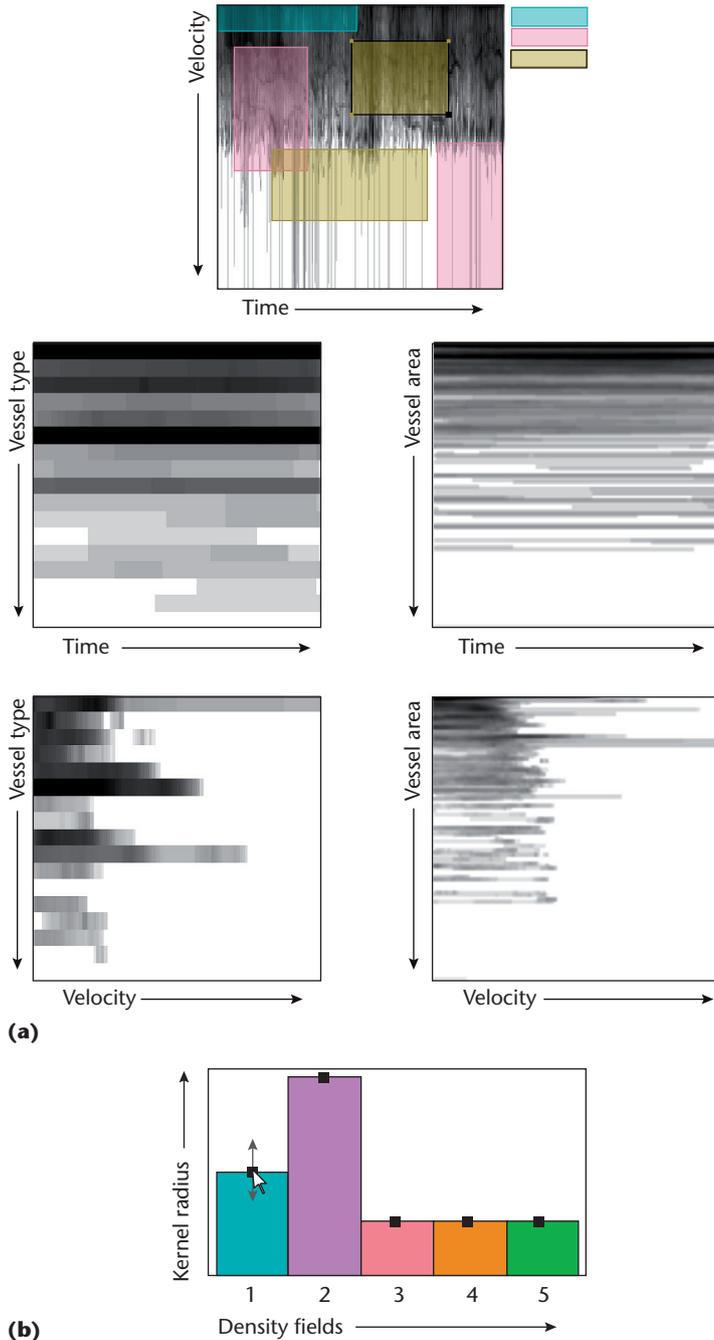


Figure 2. Widgets for creating density maps. (a) The distribution map widget lets analysts define filters using various pairs of attributes. (b) This interactive bar chart lets analysts set one parameter for five density fields.

Density Aggregation

During density aggregation, we combine, per cell, the scalar values of the basic density fields into a new aggregated density field. In geographical information systems (GISs), this is similar to map algebra for combining raster maps.⁸ We demonstrate three variations: addition, absolute difference, and anomaly detection. However, others can easily be included.

We define these density aggregations \mathcal{D} in terms of densities D_i for each cell Q :

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

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

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

$\mathcal{D}_{\text{addition}}$ simply adds all density fields (see Figure 3a). By choosing $w_i = 1/N$ as weights, we can also get a field with density value averages. We’ve used this aggregation before in a vessel density map (see the “Trajectory Smoothing” sidebar) with two density fields with different kernel radii.

$\mathcal{D}_{\text{difference}}$ and $\mathcal{D}_{\text{anomaly}}$ are only defined for combining two density fields. $\mathcal{D}_{\text{difference}}$ computes the symmetrical difference between two density fields (see Figure 3b). We conduct basic anomaly detection with $\mathcal{D}_{\text{anomaly}}$ (as we demonstrate later). The idea is that we compare a density field D_1 , containing many long trajectories representing normal movements, with a density field D_2 , containing short trajectories with current movements. Anomalies occur where hardly any normal movement is taking place. Figure 3c shows an anomaly; the trajectory is gray where the other trajectory didn’t occur.

Rendering

As we mentioned before, we render the density fields using color mapping, icons, and illuminated height fields.

Color mapping. We use hue to differentiate density fields. The density values map proportionally to the saturation; low values are white, and high values are their given color c . The default setting for the density field colors is a perceptually balanced set of distinguishable colors, which has a pleasant appearance because no color overpowers another. Our method samples a pastel rainbow color map (obtained with PaletteView⁹) and automatically generates colors, with equal steps between selected colors.

If we render only one density field, we can use a multihue color map, such as yellow-to-red, which better highlights details (see Figures 4a and 4b).⁹ To depict multiple density fields, we use single-hue color maps or icons (see Figures 4c–4f).

Iconification. Instead of generating an image for each basic density field, we can combine the fields into one image. Using icons, we show the fields’ distribution for a certain area (see Figure 4c). We chose pie charts because of their symmetry, which results in a balanced picture. The charts are proportional in area to the combined total density, and

the wedges are proportional to the ratio between the individual and total density field values in the area of the pie chart. The charts are iteratively placed at nonoverlapping locations with the highest total density values. Analysts can interactively set the number of charts, the spacing, and the scale.

We could also use pie charts to quantify the proportions between density fields. However, this would show only a coarse spatial sampling and therefore wouldn't be suitable for detailed analysis.

Illumination. This technique renders a density field as an illuminated height field in 3D, viewed from above. This results in a grayscale image in which each pixel contains the intensity of the light reflected on the surface of the height field while illuminating it with a white light source. We use Phong shading¹⁰ to compute the light intensity on the basis of the light's ambient, diffuse, and specular reflections on the surface according to the material properties. Typically, we render the aggregated density field in this way and use it as a context for the composed density fields. All the images in Figure 4 illustrate the use of this technique.

Image Composition

After obtaining a set of colored images, we combine them into one image. We distinguish between three types of image composition. *Opacity-blend* is a weighted average of the colors per pixel in RGB color space (see Figure 4d). *Max-blend* shows the density field's color with the highest density value per pixel (see Figure 4e). The *block* composition uses a coarse grid of blocks to determine which colors occur in a given area (see Figure 4f). For each block, we randomly choose a density field and assign it a corresponding color with a probability mass function generated by the proportions of the density values per field in the block.

Implementation

To compute the density fields, we use the programmable graphics pipeline in modern consumer

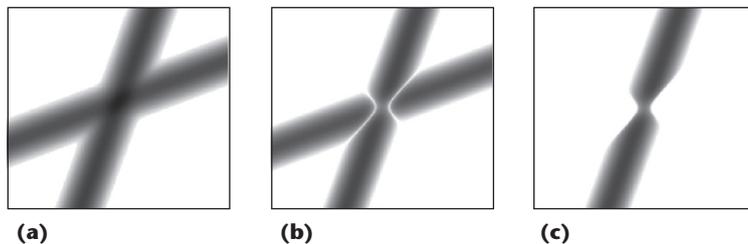


Figure 3. Density aggregations combine density fields for computing relations between them. We have defined the following variations: (a) addition, (b) absolute difference, and (c) anomaly detection.

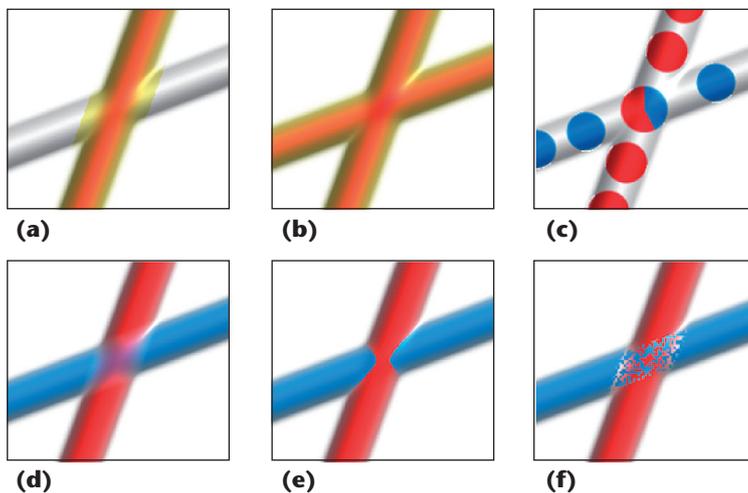


Figure 4. Image coloring and composition with illuminated height field rendering: multihue color mappings of (a) one basic density field, (b) the aggregated density fields, and (c) icons; and single-hue color mappings with (d) opacity-blend, (e) max-blend, and (f) block composition.

graphics hardware (see Figure 5). First, we construct a texture that represents a uniform grid in geographic space with cells of equal area to take into account the earth's curvature. Then, we render a polyline with geographic coordinates to this texture for each trajectory. For each line segment, we use a geometry shader to construct a bounding box that tightly fits the kernel radius. For each pixel in the bounding box, we use a fragment shader to evaluate the density equation for the corresponding line segment. The hardware computes the density per line segment in parallel. We obtain

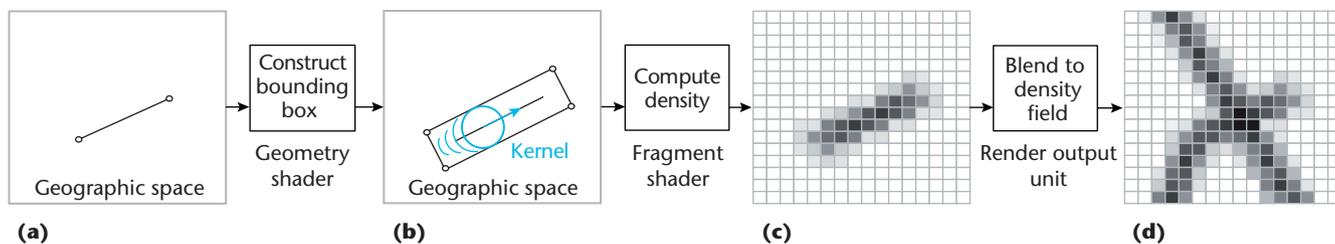


Figure 5. The pipeline for computing density fields on a GPU. (a) The trajectory segment in geographic coordinates. (b) A bounding box around the segment at distance r . (c) The density computed for each cell in the bounding box. (d) The density values added to other segments' densities.

Related Work in Trajectory Visualizations

Scientists have used different visualizations to analyze objects' movements. Jason Dykes and David Mountain analyzed attributes along trajectories using a multiple-view approach;¹ each view showed a different perspective on the data. Maria Riveiro and her colleagues analyzed ship movements using a visual analytics tool that employed multiple views to find anomalously behaving ships.² Using mouse trajectories, Peter Bak and his colleagues demonstrated how to use icons to find spatiotemporal patterns.³ Christophe Hurter and his colleagues were pioneers with graphics hardware for interactive visualizations of large numbers of airplane trajectories.⁴ None of these trajectory visualizations smoothed data when using multiple attributes.

We use a smoothing technique for analyzing movements to show trends. Another alternative might be a graph-based approach.⁵ We smooth trajectories by moving a kernel along a trajectory (see the "Trajectory Smoothing" sidebar), which is similar to the smoothed polylines in the continuous parallel coordinate plots that Julian Heinrich and Daniel Weiskopf developed.⁶ We can efficiently compute such a smoothed line by using kernels defined by a polynomial, as investigated by Xiaogang Jin and Chiew-Lan Tai.⁷ Hurter and his colleagues implemented a simplified trajectory density by convolving the samples instead of the lines, using graphics hardware. Urška Demšar and Kirsu Virrantaus⁸ showed temporal information as well by computing the density in a space-time cube⁹ in which an extra axis makes time visible.

References

1. J.A. Dykes and D.M. Mountain, "Seeking Structure in Records of Spatio-temporal Behaviour," *Computational Statistics & Data Analysis*, vol. 43, no. 4, 2003, pp. 581–603.
2. M. Riveiro, G. Falkman, and T. Ziemke, "Visual Analytics for the Detection of Anomalous Maritime Behavior," *Proc. 12th Int'l Conf. Information Visualization (IV 08)*, IEEE CS Press, 2008, pp. 273–279.
3. P. Bak et al., "Spatiotemporal Analysis of Sensor Logs Using Growth Ring Maps," *IEEE Trans. Visualization and Computer Graphics*, vol. 15, no. 6, 2009, pp. 913–920.
4. C. Hurter, B. Tissoires, and S. Conversy, "FromDaDy: Spreading Aircraft Trajectories across Views to Support Iterative Queries," *IEEE Trans. Visualization and Computer Graphics*, vol. 15, no. 6, 2009, pp. 1017–1024.
5. N. Andrienko and G. Andrienko, "Spatial Generalization and Aggregation of Massive Movement Data," *IEEE Trans. Visualization and Computer Graphics*, vol. 17, no. 2, 2011, pp. 205–219.
6. J. Heinrich and D. Weiskopf, "Continuous Parallel Coordinates," *IEEE Trans. Visualization and Computer Graphics*, vol. 15, no. 6, 2009, pp. 1531–1538.
7. X. Jin and C.-L. Tai, "Analytical Methods for Polynomial Weighted Convolution Surfaces with Various Kernels," *Computers & Graphics*, vol. 26, no. 3, 2002, pp. 437–447.
8. U. Demšar and K. Virrantaus, "Space-Time Density of Trajectories: Exploring Spatio-temporal Patterns in Movement Data," *Int'l J. Geographical Information Science*, vol. 24, no. 10, 2010, pp. 1527–1542.
9. T. Hägerstrand, "What about People in Regional Science?" *Papers in Regional Science*, vol. 24, no. 1, 1970, pp. 6–21.

a density field by additively blending the density values into the grid texture. Finally, we project the density fields into screen space, where we can perform further operations, such as color mapping and image composition, using additional shaders to obtain a density map. With this implementation, we can render density maps in a split second, and interactions can take place in real time.

Use Cases

While exploring object trajectories, analysts interact mainly with the widgets. Once they find an interesting density map, they can fine-tune the details and export it as a static raster map for a GIS. They can use these in addition to the regular operational maps containing live traffic to support monitoring tasks.

The following use cases show density solutions for open problems in the analysis of moving objects. The cases examined ship traffic in the North Sea and pedestrians in the city center of Delft.

Ship Traffic Analysis

Professional ships with a gross tonnage of 300 tons or more must broadcast their current status using the Automatic Identification System (AIS).¹¹ The trajectory data consists of many attributes, such as time, location, ship type, dimensions, and destination. Captains sailing a ship and surveillance operators guarding coastal areas use AIS for safety and security. They can also use past AIS data to plan the spatial usage of coastal areas—for instance, by validating whether ships have followed maritime traffic rules. The dataset contains both route-bound ships (such as tankers and cargo ships) and non-route-bound ships (such as pleasure craft and tugs).

If our method works with this data, it will probably work both for constrained trajectories (such as those of cars and trains) and unconstrained trajectories (such as those of animals and pedestrians).

Temporal aggregation. Normally, vessel density maps don't identify the order in which movements occur. With our density maps, we can vary the kernel radius and weight over time to obtain a semantic depth of field that highlights, for example, the most recent movements.¹²

Figure 6a shows a single day of ship traffic outside Rotterdam's harbor with four density fields of six hours each. Throughout the day, the kernel radii decrease while the weights increase. The four density fields combined with additional density aggregation let us distinguish different moments in time in a single density field. The evening movements appear as small, dark trajectories, with

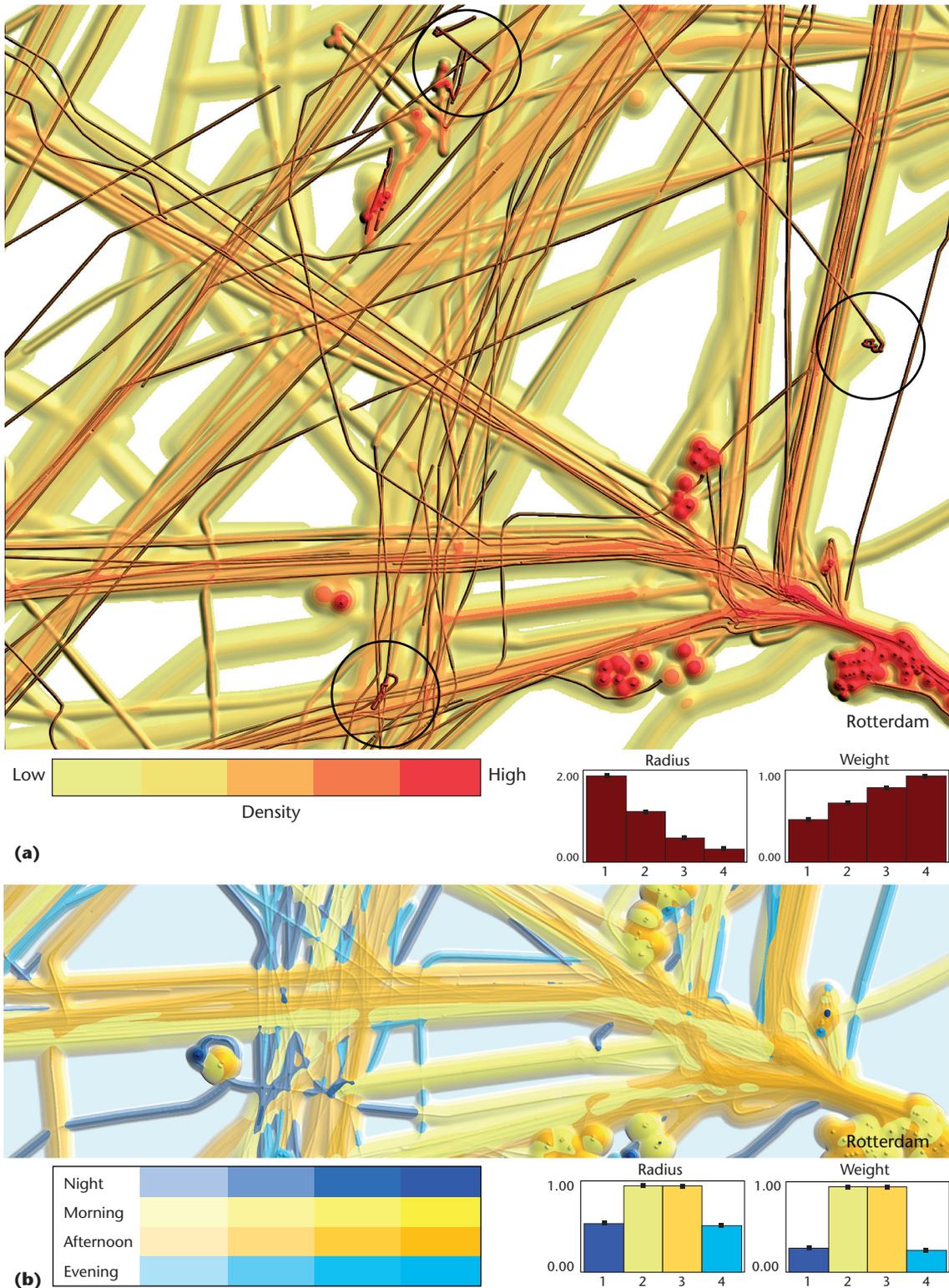


Figure 6. Four density fields with different weights and kernel radii, each covering six hours of a day outside Rotterdam's harbor. (a) Using density aggregation lets us distinguish different moments in time in a single density field. (b) Using image composition, we color the different periods of the day with yellow and blue tints so that we can see that ships follow the sea lanes more strictly during daylight than during the night.

the others giving context. Circles show noticeable evening movement patterns that weren't visible in the original vessel density maps.

Figure 6b shows the night hours in blue and the daylight hours in yellow and combines them using max-blend. Many blue deviations occur, revealing

Trajectory Smoothing

Kernel density estimation (KDE) smoothes data points to get an overview of data.¹ It first replaces each data point by a distribution. This distribution is given by a unit-size kernel function k_r , with radius r defining the size of the neighborhood, centered at the point. KDE eventually sums all the distributions to obtain a density field C . For trajectories, we can obtain C by smoothing the positions $\mathbf{p}^o(t)$ of all objects in $o \in O$. That is, in a point \mathbf{q} at time t , the density $C(\mathbf{q}, t)$ is

$$C(\mathbf{q}, t) = \sum_{o \in O} k_r(\|\mathbf{q} - \mathbf{p}^o(t)\|).$$

This density field gives the smoothed number of objects at time t . $C(\mathbf{q}, t)$ divided by the total number of objects is a probability density function. To smooth the complete trajectories, we integrate the smoothed number of objects over time. This effectively moves the kernel along the trajectories of all objects at the speed of these objects:²

$$C(\mathbf{q}) = \frac{1}{T} \int_0^T C(\mathbf{q}, t) dt.$$

$C(\mathbf{q})$ in point \mathbf{q} represents the number of objects averaged over time. We obtain a discrete density $D(Q)$ per area unit on a raster of cells by sampling C in the center of each cell

Q . In our approach, multiple density fields combine with two visualization parameters for each field: a weight parameter w_i and a kernel radius r_i to differentiate between the fields (as shown in the architecture of Figure A). Quantitative density analysis with weighted density fields is difficult; however, hot spots remain visible.

Figure A shows how we visualized density fields as vessel density maps.² These maps show variations in speed as different density contributions highlighting significant maritime areas, such as groups of high contributions for anchor zones, where ships wait to enter a harbor. In these maps, only one attribute (velocity) is visible. So, we can't see the variations of area usage over time. Because trajectories are essentially multivariate data, we designed the vessel density maps' generalization to handle more attributes.

References

1. B.W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman & Hall, 1992.
2. N. Willems, H. van de Wetering, and J. J. van Wijk, "Visualization of Vessel Movements," *Computer Graphics Forum*, vol. 28, no. 3, 2009, pp. 959–966.

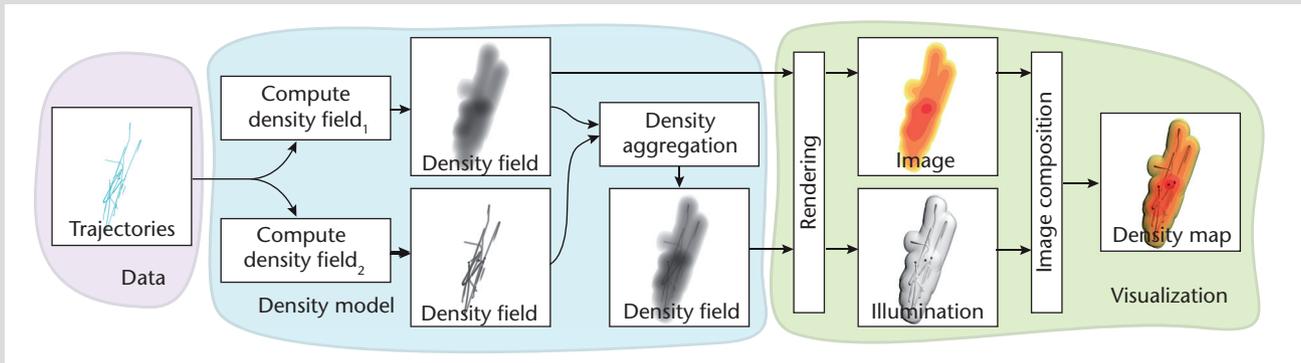


Figure A. Our method's architecture for creating vessel density maps.² First, we use a large and a small kernel to smooth trajectories, resulting in two density fields. The rendering uses the large-kernel density field for color mapping and the aggregated density fields for illumination. The final density map multiplies the color image and the grayscale image with the illumination.

that ships follow the sea lanes more strictly during daylight than during the night.

Anomaly detection. A density field generated from many past trajectories indicates which movements are common in an area. By comparing a density field of other trajectories with this historical density field, we can find abnormal movements. Such movements occur where the current density is high but the historical density is low.

For example, we generated a density field of trajectories between Amsterdam and Scheveningen over a period of six days. We then generated a

density field of trajectories over two hours, which we animated to mimic a live stream of traffic data. The kernel size for live traffic was large for the most recent movements and decreased over time. We applied anomaly density aggregation for these two density fields and rendered the resulting density field with a multihue color map together with all data in the shaded height field (see Figure 7a).

In another example, we compared six days of passenger ship movements to a single day's movements. The results indicated anomalously behaving passenger ships (see Figure 7b).

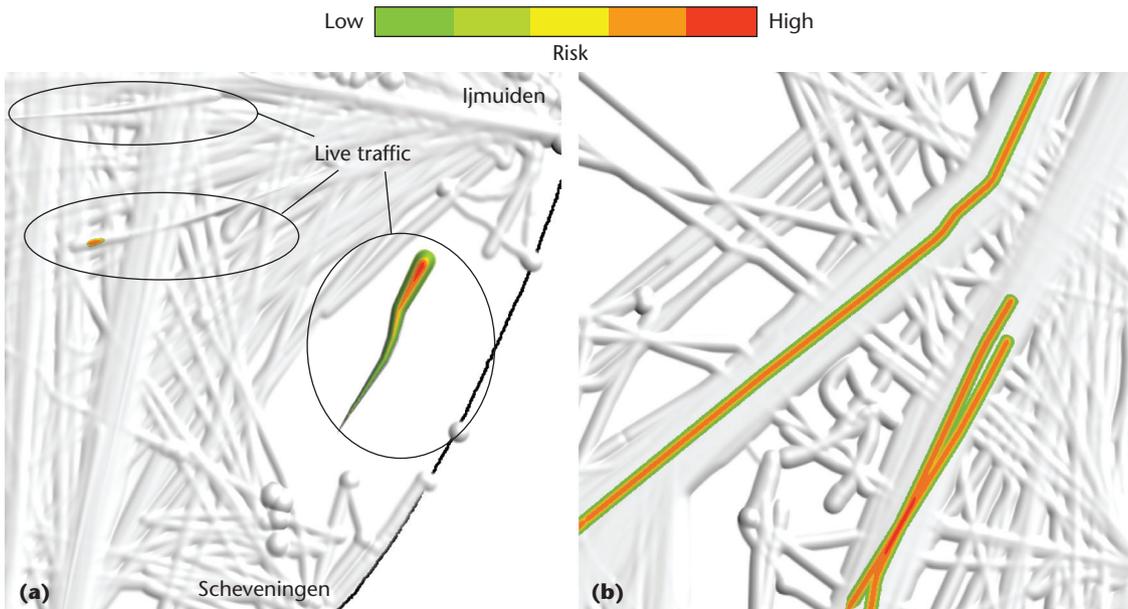


Figure 7. Anomalous ships highlighted in the context of all movements. (a) An anomalous ship sailing between Amsterdam and Scheveningen. (b) Anomalous passenger ships sail where no passenger ships normally sail.

Stopping areas. A vessel density map highlights intensively sailed areas—areas with many movements or with few slow movements. Figure 8a shows these areas in the rectangles. Some of them appear as a group of dots, indicating that ships are stopped at a prescribed anchor zone (marked with an anchor). However, similar patterns occur at other places. We aimed to figure out what was going on in those places by exploring the trajectories’ attributes.

To focus on the stopping areas, we first isolated the areas with slow movements by defining a red density field for slow movements and a blue density field for others, using the DM in Figure 8b. Figure 8c shows the results using max-blend. By comparing it with Figure 8a, we can see that area 7 (containing Rotterdam’s harbor) shrank; the harbor mouth contained many movements; and areas 12 and 13, which each have a single stop, appeared. We discovered stopping areas that weren’t marked as anchor zones or weren’t in Rotterdam’s harbor.

The ship’s type and how it moves can often explain what’s happening. So, we changed the DM’s time axis to velocity and changed its velocity axis to ship type (see Figure 8d). Then, we defined six density fields for the most important ship types containing only slow ships, each with a different color as shown in the color map on the bottom right of Figure 8. To see which ship types sailed in these small areas, we used block composition.

Figures 8e through 8i focus on the investigated areas. Figure 8e shows a cargo ship and a nearby tanker. When the kernel radius decreased, black squares indicating oil platforms became visible (see the inset). These ships were probably exchanging goods with these platforms.

Figure 8f shows the surroundings and popular

beach by Scheveningen’s harbor. Some typical ships close to the coast are visible, such as fishing boats (yellow), pleasure craft (green), and special craft such as rescue ships (red). A cargo ship (blue) is relatively close to the coast, which is suspicious because such ships generally don’t come that close.

Figure 8g shows two hot spots with small ships. The one on the right is the harbor of the city Hellevoetsluis; the one on the left is near the Haringvlietdam, a structure comprising a dam and a lock. The lock explains the concentration of waiting ships at the bottom left of the Haringvlietdam and explains why only small ships were sailing in this neighborhood.

By clicking on the ships in Figure 8h, we learned that the special craft were a dredger at work (left) and a law enforcement ship (right). In the latter’s course of its duty, it stopped multiple times.

Finally, Figure 8i shows a potential threat: a fishing boat stopped in a sea lane.

Risk assessment. Some ships are more dangerous than others. We created a density map showing the various ship types’ possible risks (see Figure 9). In the DM, we put the ship’s type and deck area on the axes. For the density fields, we defined “large” as an area greater than 9,000 m²; this included cargo ships and tankers. We defined the other ship types as “small”: passenger ships, high-speed craft, and “other.” Default settings automatically generated the colors.

We composed the color-mapped densities with max-blend to show the most dangerous types in each area. To illustrate the possible risks, we identified three classes of danger and assigned large kernels with heavy weights for more dangerous ship types. In this risk density map, we can see

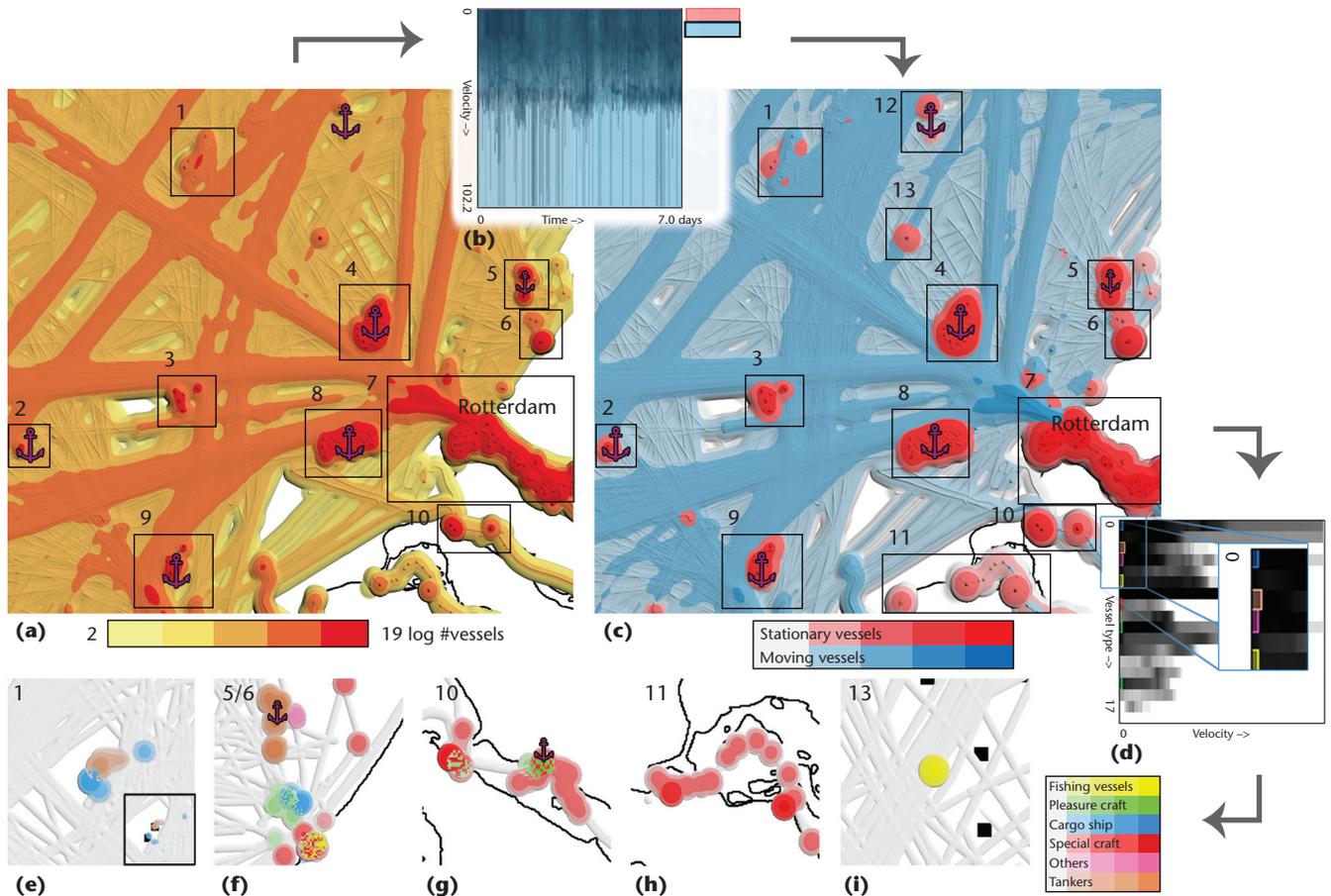


Figure 8. Classifying the behavior of slow ships outside Rotterdam’s harbor over one week. (a) A vessel density map. Anchor icons indicate anchor zones. (b) A distribution map (DM) that helped isolate (c) slow-moving areas. (d) A DM that we used to define the different types of slow ships, which helped us figure out what happened in the zoomed-in areas: (e) a cargo ship and a nearby tanker, (f) several ships near Scheveningen’s harbor, (g) two hot spots with small ships, (h) a dredger and law enforcement ship at work, and (i) a fishing boat stopped in sea lane. These areas correspond with the numbers in Figure 8c.

that the shoreline wasn’t in danger and that dangerous tankers took the west route, whereas less dangerous cargo ships took the route closer to the shore.

Pedestrians in Urban Planning

From an urban-planning perspective,¹³ we examined the dataset of pedestrian movements in the city center of Delft over four days. This data’s temporal resolution is similar to the ships—a sample every couple of seconds. However, the paths are entangled, owing to small GPS devices’ spatial noise. So, an aggregated view tends to provide more insight than a detailed view.

The pedestrians filled out questionnaires explaining the purposes of their visits: shopping, tourism, leisure, or something else. We wanted to find areas of interest that attract visitors with a certain purpose and see whether the visitors appeared in the expected areas. To do so, we divided our dataset into four density fields, one for each purpose, with equal weight and kernel radius. (As

in the previous use case, the default settings automatically generated the colors.)

We combined the density fields using the image compositions for single-hue color maps. However, the resulting images were unreadable because many density fields had nonzero values. With opacity-blend, only areas with a single density field defined are readable, whereas the other places give indistinguishable colors (see Figure 10a). With max-blend, the picture is clearer, and the most dominant purpose is visible (see Figure 10b). However, shopping (blue) is too dominant to make the analysis useful.

Because the weak distribution of cones in the human eye prevents us from seeing the colors of small patches, we can better distinguish the colors in the large blocks by using block composition. The difference between a uniform distribution (see Figure 10c) and a distribution proportional to the density values (see Figure 10d) for the randomization becomes clear. In the latter, the more dominant density fields are more visible. The propor-

tions between density fields are visible, but with icons (see Figure 10e), quantifying the proportions is easier. (However, the orange trend in the middle is less visible than in Figures 10a through 10d.)

Figure 10f shows the density map with icons integrated into Google Earth, a popular GIS. We see several areas of interest where the distribution of purposes is biased toward one or more purposes. Using Google Earth, we can find explanations for these biases. For example, at the market square in area 1, tourism increases near a church and an art exhibition center (the Vermeer Centrum), whereas the “other” category becomes dominant around the city hall (the Stadhuis). Owing to the presence of several diners and stores, the shopping and leisure categories also appear. In areas 2 and 3, the leisure category dominates, explained by the presence of a gallery and a number of bars and cafes, respectively.

Overall, using our method in Google Earth, we found a relationship between an area and pedestrians’ visit purposes by using the additional data sources available in such a tool. This could aid urban planners in their decision-making.

A limitation of our method is that a density field filter can’t always express a feature of interest, such as a relation between attributes. However, the use cases show that the method is already useful for various domains, each with its own requirements. We can make the method even stronger by adding new density aggregations, image compositions, and improved filters. ■■

Acknowledgments

We thank Hans Hiemstra (Thales) and Stefan van der Spek (Delft Univ. of Technology) for providing data and feedback. This research was part of the Poseidon project at Thales Nederland under the responsibilities of the Embedded Systems Institute. The Dutch Ministry of Economic Affairs partly supported this project under the Besluit Subsidies Investerings Kennisinfrastuctuur program. This article is an extended version of “Interactive Visualization of Multivariate Trajectory Data with Density Maps.”¹⁴

References

1. S. Dodge, R. Weibel, and A.-K. Lautenschütz, “Towards a Taxonomy of Movement Patterns,” *Information Visualization*, vol. 7, nos. 3–4, 2008, pp. 240–252.
2. N. Willems et al., “An Integrated Approach for Visual Analysis of a Multisource Moving Objects Knowledge Base,” *Int’l J. Geographical Information Science*, vol. 24, no. 10, 2010, pp. 1543–1558.

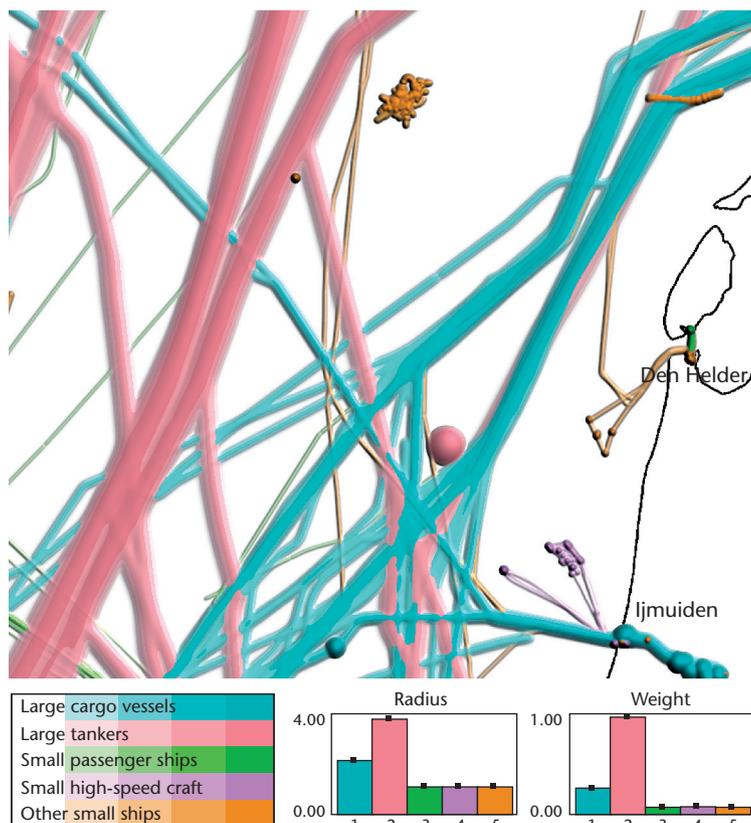


Figure 9. A risk density map showing various ship types over a week. Dangerous tankers took the west route, whereas less dangerous cargo ships took the route closer to the shore.

3. B.W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman & Hall, 1992.
4. C. Hurter, B. Tissoires, and S. Conversy, “Accumulation as a Tool for Efficient Visualization of Geographical and Temporal Data,” Presentation at AGILE Workshop Geospatial Visual Analytics: Focus on Time, 2010; www.lii-enac.fr/~conversy/research/papers/geovat2010-accumulation.pdf.
5. O.D. Lampe and H. Hauser, “Interactive Visualization of Streaming Data with Kernel Density Estimation,” *Proc. 2011 IEEE Pacific Visualization Symp. (PacificVis 11)*, IEEE CS Press, 2011, pp. 171–178.
6. N. Willems, H. van de Wetering, and J.J. van Wijk, “Visualization of Vessel Movements,” *Computer Graphics Forum*, vol. 28, no. 3, 2009, pp. 959–966.
7. S. Peters and J.M. Krisp, “Density Calculation for Moving Points,” *Proc. 13th AGILE Int’l Conf. Geographic Information Science*, 2010; http://agile2010.dsi.uminho.pt/pen/ShortPapers_PDF%5C135_DOC.pdf.
8. C.D. Tomlin and J.K. Berry, “A Mathematical Structure for Cartographic Modeling in Environmental Analysis,” *Proc. 39th Symp. Am. Congress on Surveying and Mapping*, ACSM, 1979, pp. 269–283.
9. M. Wijffelaars et al., “Generating Color Palettes Using Intuitive Parameters,” *Computer Graphics Forum*, vol. 27, no. 8, 2008, pp. 743–750.
10. B.T. Phong, “Illumination for Computer Generated

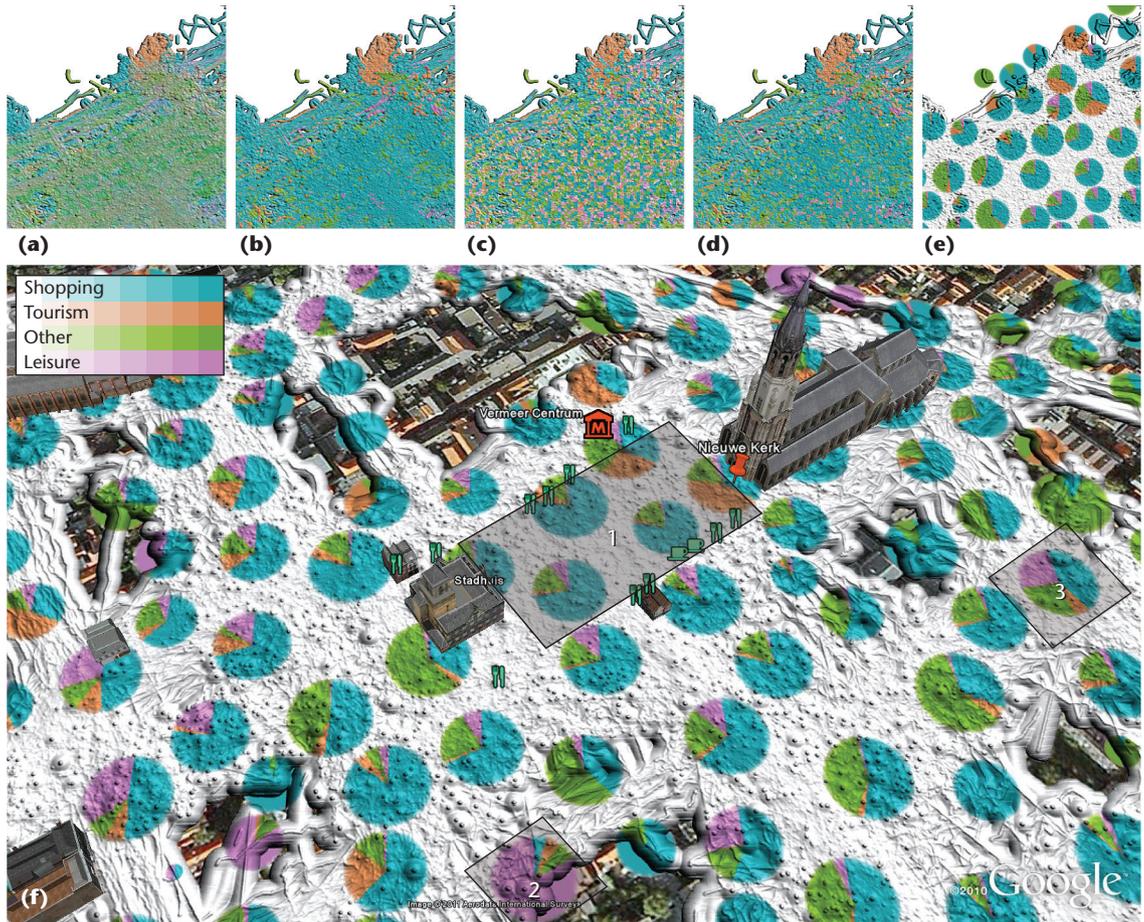


Figure 10. An overview of pedestrian movements around the market square in Delft, divided into four purposes: shopping, tourism, leisure, and “other.” We combined the four density fields using (a) opacity-blend, (b) max-blend, (c) block distribution with uniform probability, (d) block distribution with a probability proportional to density value, and (e) icons. (f) A density map with icons in Google Earth shows several marked areas of interest (base image © 2010 Google; © 2011 Aerodata International Surveys).

Pictures,” *Comm. ACM*, vol. 18, no. 6, 1975, pp. 311–317.

11. Recommendation ITU-R M.1371-1, *Technical Characteristics for an Automatic Identification System Using Time Division Multiple Access in the VHF Maritime Mobile Band*, Int’l Telecommunications Union, 2001.
12. R. Kosara, S. Miksch, and H. Hauser, “Focus+Context Taken Literally,” *IEEE Computer Graphics and Applications*, vol. 22, no. 1, 2002, pp. 22–29.
13. S. van der Spek et al., “Sensing Human Activity: GPS Tracking,” *Sensors*, vol. 9, no. 4, 2009, pp. 3033–3055.
14. R. Scheepens et al., “Interactive Visualization of Multivariate Trajectory Data with Density Maps,” *Proc. 2011 IEEE Pacific Visualization Symp. (PacificVis 11)*, IEEE CS Press, 2011, pp. 147–154.

Roeland Scheepens is a PhD candidate at Eindhoven University of Technology. His research interests include GPU-accelerated information, uncertainty, and geographical visualization. Scheepens has an MSc in computer science from Eindhoven University of Technology. He’s a member of IEEE. Contact him at r.j.scheepens@tue.nl.

Niels Willems is a software architect at Synerscope. His research interests include the visual analysis of moving objects. Willems has a PhD in information visualization from Eindhoven University of Technology. He’s a member of IEEE. Contact him at niels.willems@synerscope.com.

Huib van de Wetering is an assistant professor in the visualization group of Eindhoven University of Technology’s Department of Mathematics and Computer Science. His research interests include information visualization and computer graphics. Van de Wetering has a PhD in computer science from Eindhoven University of Technology. Contact him at h.v.d.wetering@tue.nl.

Jarke J. van Wijk is a full professor of visualization at Eindhoven University of Technology. His research interests include information visualization, visual analytics, and interaction. Van Wijk has a PhD in computer graphics from Delft University of Technology. He’s a member of IEEE, ACM, and Eurographics. Contact him at j.j.v.wijk@tue.nl.