D4.1 (Web) Service and Configuration Tool

Project Number:	
Classification **Deliverable No.:**	Public D4.1
Work Package(s): **Document Version:** **Issue Date:** **Document Timescale:**	WP4 Vs. 1 28.02.2020 Project Start: June/July 1st (Poland), 2018
Compiled by:	OFF, IN, PUEB
Authors:	Matthias Steidel, Krzysztof Węcel, Dominik Filipiak, Milena Stróżyna, Witold Abramowicz, Reinhard Zimmermann, Thomas Hahnel
Issue Authorisation:	Sebastian Feuerstack, OFF

© All rights reserved by HANSA consortium

This document is supplied by the specific HANSA work package quoted above on the express condition that it is treated as confidential to those specifically mentioned on the distribution list. No use may be made thereof other than expressly authorised by the HANA consortium.

HANSA is funded by the MarTERA partners German Federal Ministry of Economic Affairs and Energy (BMWi), Polish National Centre for Research and Development (NCBR) and Research Council of Norway (RCN) and co-funded by the European Union.

DISTRIBUTION LIST		
Copy type[1]	Company and Location	Recipient
T	HANSA Consortium	all HANSA Partners

[1] Copy types: E=Email, C=Controlled copy (paper), D=electronic copy on Disk or other medium, T=Team site (OwnCloud)

RECORD OF REVISION		
Date	Status Description	Author
2020-04-06	Initial version (draft)	OFFIS

Table of Contents

1 Introduction

Subject of this Deliverable is the description of the developed algorithms in order to detect anomalous vessel behavior automatically. The automatic detection of anomalous vessel behavior is a challenge that arises from the expected further increasing traffic numbers at sea in the future, which will be accompanied by an increased collision risk. Automated sea area surveillance system that detects potentially dangerous and anomalous behavior and generates an appropriate alarm for the VTS officers is a promising pre-emptive action in this context.

Besides voyage planning, one aim of the HANSA project is to investigate on methods for automatic anomaly detection. This Deliverable describes the findings of the appropriate work package.

2 Anomaly Detection

To enable the development of a system that supports VTS officers during sea area surveillance, it is necessary to extract typical traffic patterns that represent normal vessel movements. In the previous deliverables of WP3 the corresponding algorithms for the extraction of these traffic patterns were presented. As a result, a mesh is available which models the extracted traffic patterns. From this mesh, it is possible to generate Recommended Corridors (RC). Based on these concepts, the following anomalies were defined in HANSA during the work in WP4:

Lateral Scope

Each RC has a center line, which has an equal distance to both RC edges. Here, a Lateral Scope can be defined, which describes the maximum allowed lateral distance which a vessel can have to the center line while sailing inside this corridor. This concept is depicted in Figure 1. As soon as the distance is exceeded, an anomaly alarm is generated to warn the responsible VTS officer.

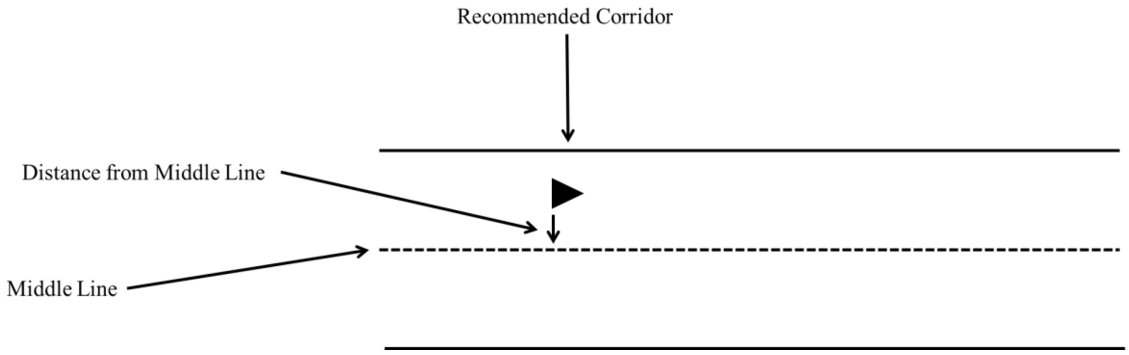


Figure 1: Concept for detecting lateral anomalies

Directional Scope

For detecting anomalies in the Course over Ground (COG) of a vessel, the orientation of the center line of the appropriate RC is considered, which is illustrated by Figure 2. To accomplish this, the difference between this orientation and the COG of the vessel is calculated. If a predefined value is exceeded, an appropriate alarm is triggered.

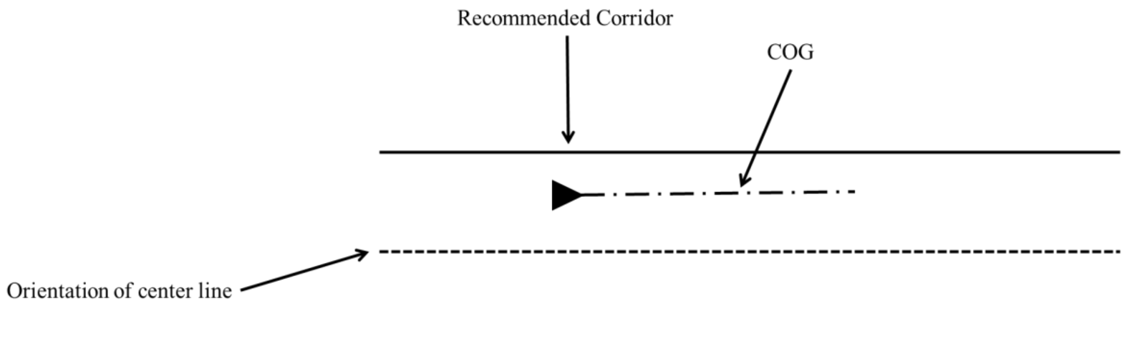


Figure 2: Concept for detecting directional anomalies

Minimum and Maximum Speed

For each RC, the minimum and maximum observed speed of vessel are given. These value are constantly compared with the current vessel's speed. As soon as the vessel's speed exceeds or is below the minimum speed, a speed related anomaly alarm is triggered.

Directional Anomaly

As defined before, a RC has a travel direction which defines whether it is allowed to sail in only one direction or either direction. If just one direction is allowed and a vessel is travelling in the opposite direction inside the RC, an alarm is trigged. Following this logic, an alarm is also triggered if a vessel e.g. crosses a RC from north to south, but the allowed sailing direction of this corridor is east to west and vice versa.

3 Mesh Visualization

The visualization of the meshes are explained in order to provide a better understanding of the results. Interpretation of the visualizations is also provided.

The following variables are visualized on the maps:

- waypoints
    - location of the waypoint (geographical coordinates)
    - size of the waypoint – no special meaning, size is fixed in pixels, not scaled with a map scale
    - color – average direction of the traffic passing through the waypoint segment
    - tooltips: id of the waypoint, latitude, longitude, strength, direction
- edges
    - location – connecting two waypoints
    - color – number of ships passing the edge (encoded variable cnt), or direction north/south
    - width – in some visualizations to show width of the corridor, scaled with a map

    - tooltips: ids of connected waypoints, their coordinates, distance in kilometers, etc.

- corridors
    - location – connecting two waypoints
    - color – different colors to show left/right, northward/southward edges
    - width – represents real width of the corridor (value provided in meters)
    - tooltip: additionally, width of the corridor

3.1 Waypoints

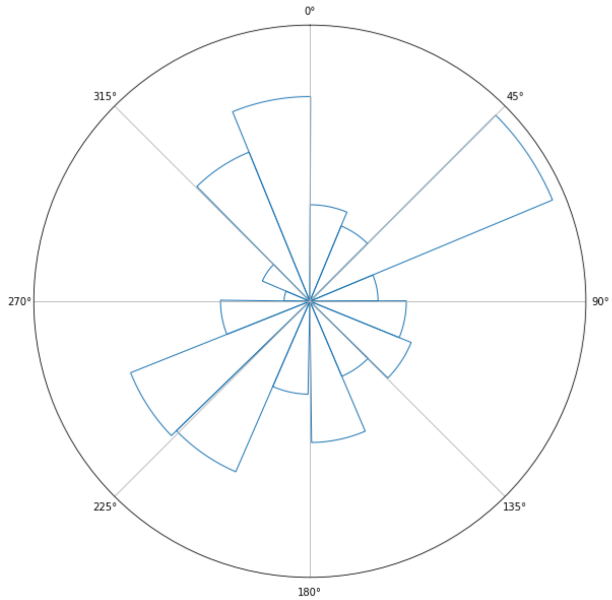
Calculation of the average direction

We analyze all AIS points assigned to a given waypoint. There are two outcomes of this analysis:

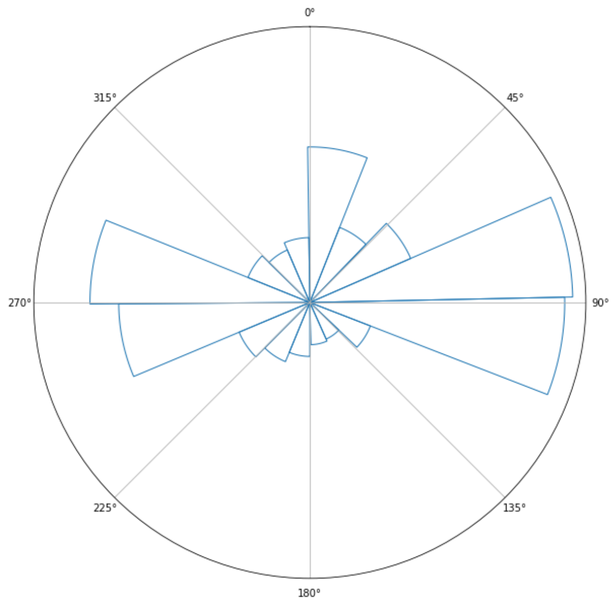
- distribution of the directions presented in the radial histogram (rose plot)
- average direction and strength.

The radial histogram nicely presents directions whether there is one direction preferred or ships go in various directions.

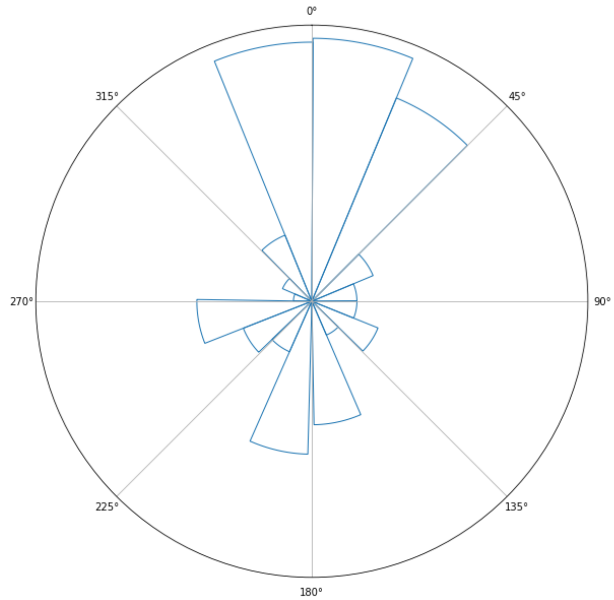
The fist chart presents distribution without any preference. In such a case the value of the variable “strength” will be low. It is a waypoint somewhere in open sea.



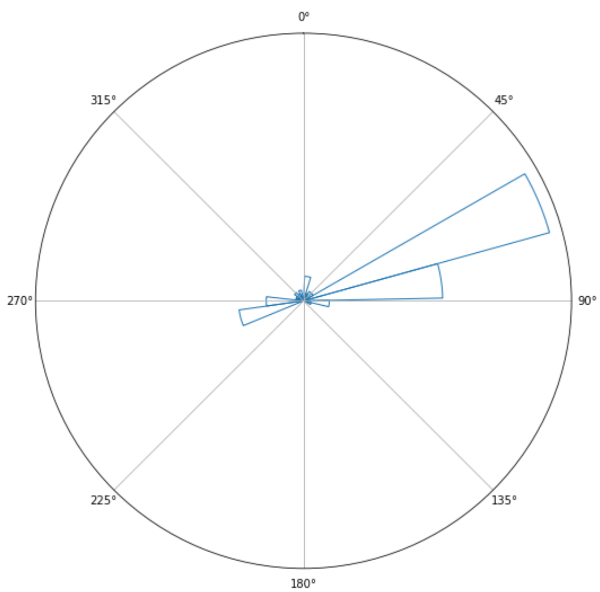
The next chart presents a waypoint taken on Elbe river on the route to Hamburg. We see that the dominating traffic is east-west (as we could expect on the river). The small fraction represents also entrance to the Nord-Ostsee-Kanal.

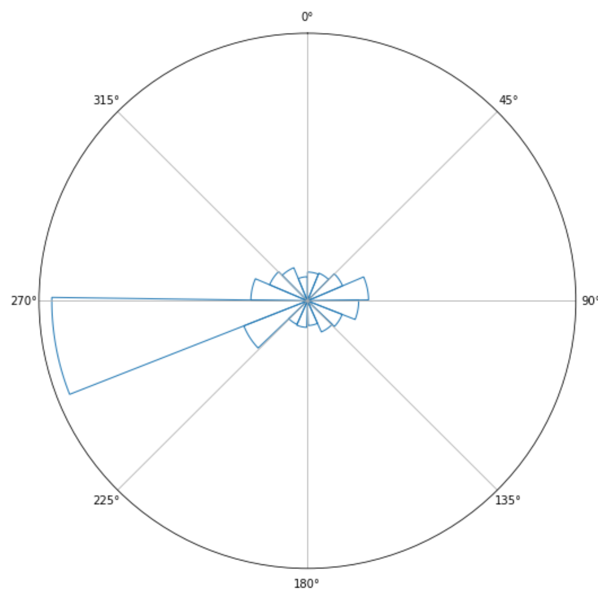


Near Bremerhaven the dominating traffic is north-south.



The final chart presents the visualization of waypoints taken from the traffic separation zones. As can be seen, they unambiguously show in which direction ships should move. The corridor here is very narrow and the variable “strength” is very close to 1.0.





Average direction

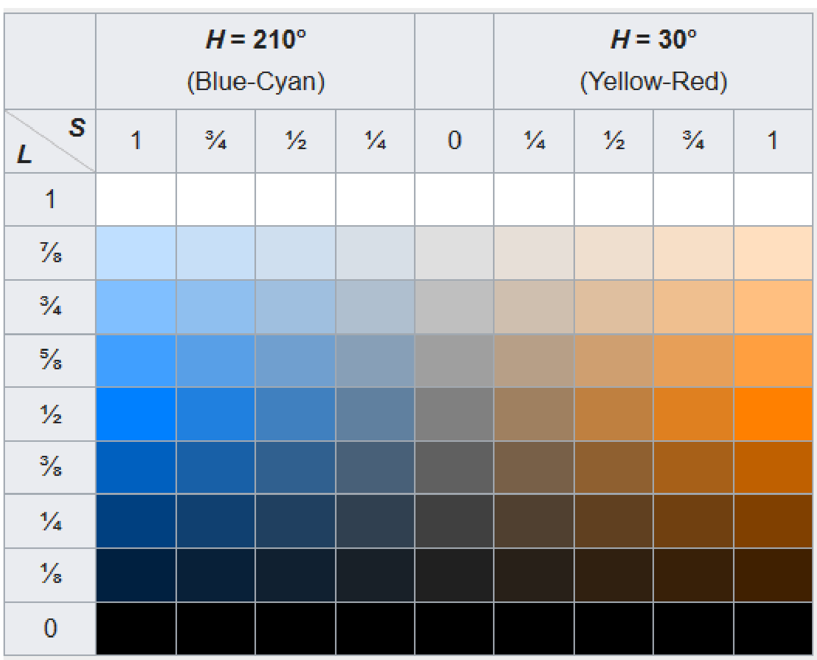
The average direction is calculated in order to represent dominating traffic using a single number instead of the histogram. It can then be used for further reasoning about typical traffic. There are actually two variables returned from the calculation:

- theta – the actual angle (0 pointing to the North)
- strength – how strong is the preference; 0 for no preference, 1 for unambiguous direction.

Both variables need to be encoded in a single color. We therefore used HSL as follows:

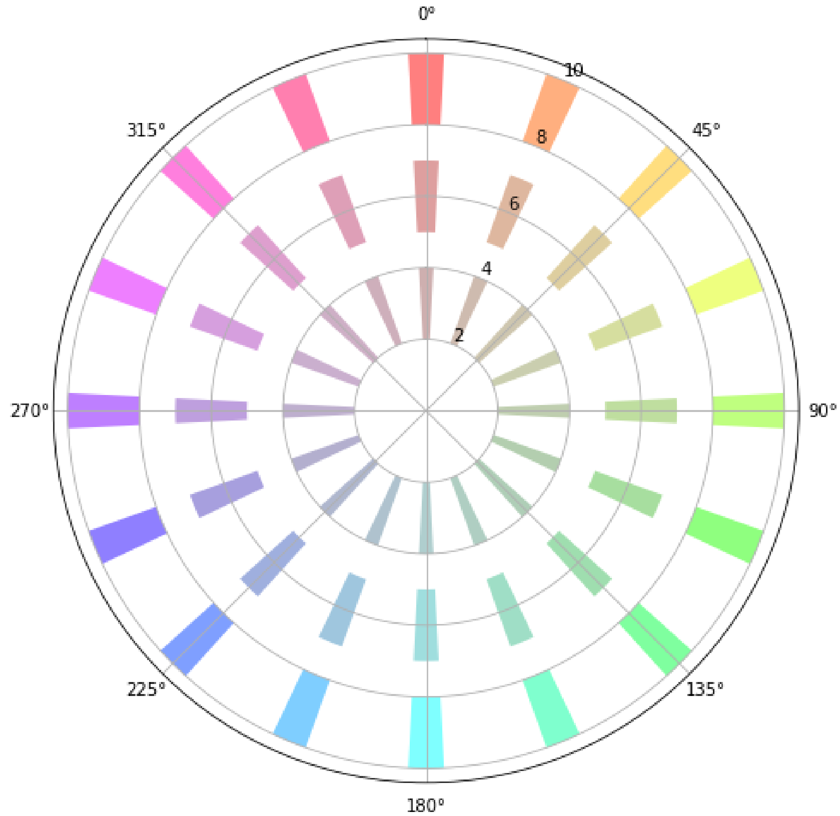
- H (hue) - represents direction 0 - 360 deg, different basic colors
- S (saturation) - represents the intensity of direction, the closer to one the higher directionality of a waypoint, ranges from grey (no direction) to pure saturated color
- L (lightness) - will be fixed at 0.5, making sure that either pure colors will be visible or grey

The chart below presents how the resulting colors can look like.



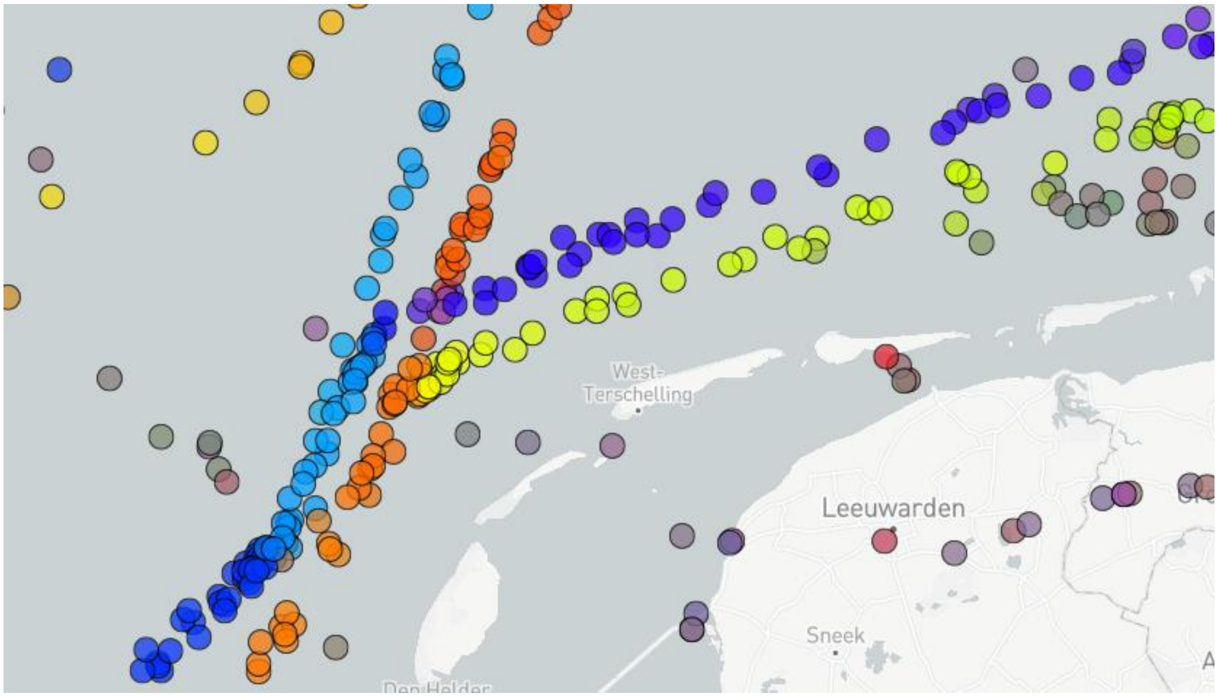
Color meaning

Coding of the direction is presented in the following figure.



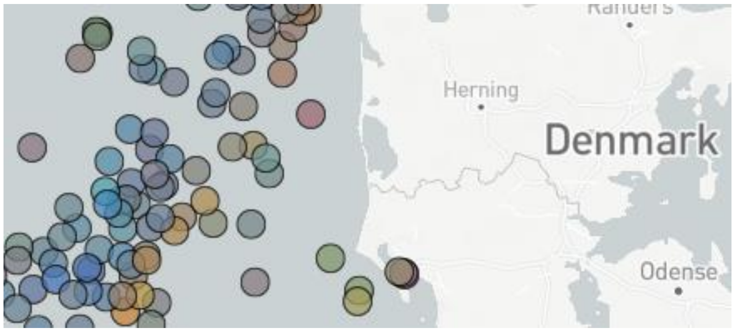
Sample chart

The following figure nicely presents how the colors allow better understanding of the ship movements based purely on AIS. The traffic separation schemes are clearly visible.



The interpretation is as follows. The value of yellow is ca. 60º while dark blue is ca. 240º. This is TSS going more East-West. The other YTSS is enforced more North-South and represented as orange (20º) and light blue (200º).

Saturated colors mean that the domination of one direction is clear. When the number of ships going in different directions is balanced, the color becomes grey.



Tooltips

For each node we also provide tooltips which presents additional detailed information about a waypoint.

3.2 Edges

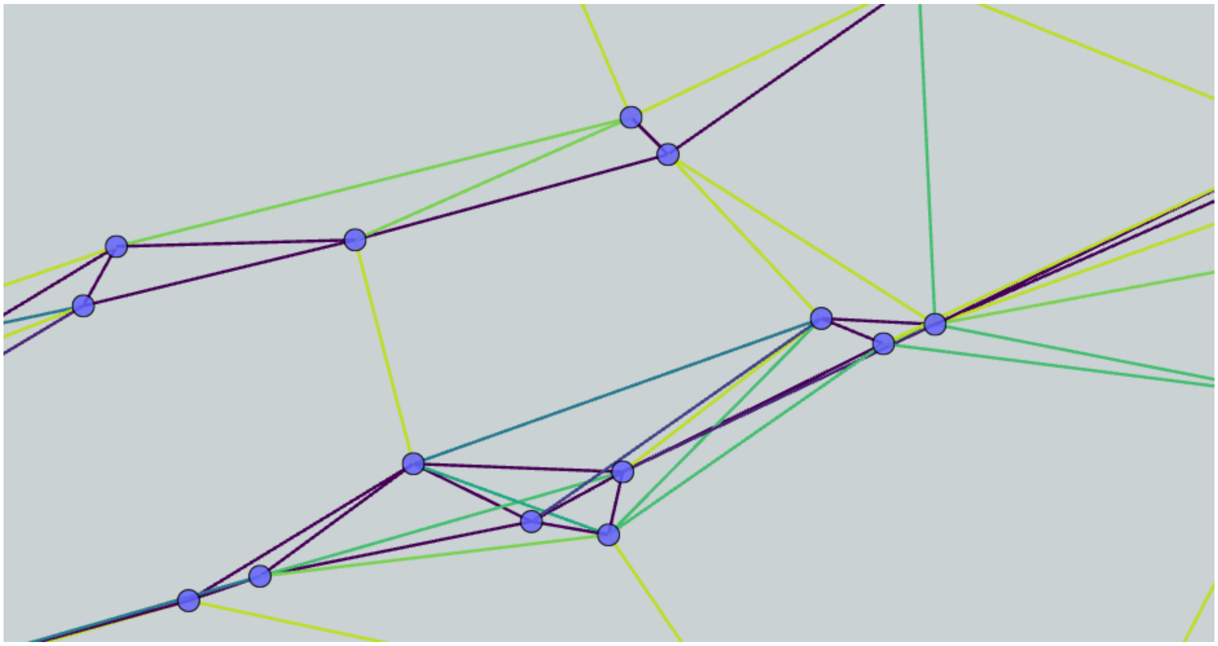
Edges connect waypoints. During determination of edges in a graph, we prepare several aggregate variables. One of them is how often the edge was followed by different vessels, referred to as cnt – count.

Color meaning

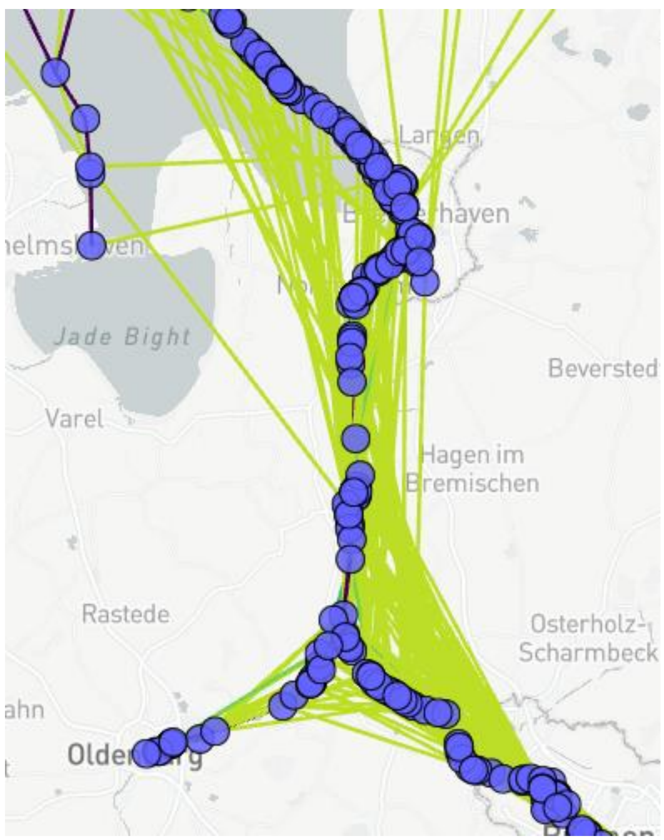
From available visual feature we only use color. We were mostly interested in rarely followed edges, therefore we coded colors in a way to expose such routes. The chart below presents the coding.

Each value higher than 10 is encoded as a dark violet.

In the sample below we can see popular connections (darker) and also few rare connections (lighter).



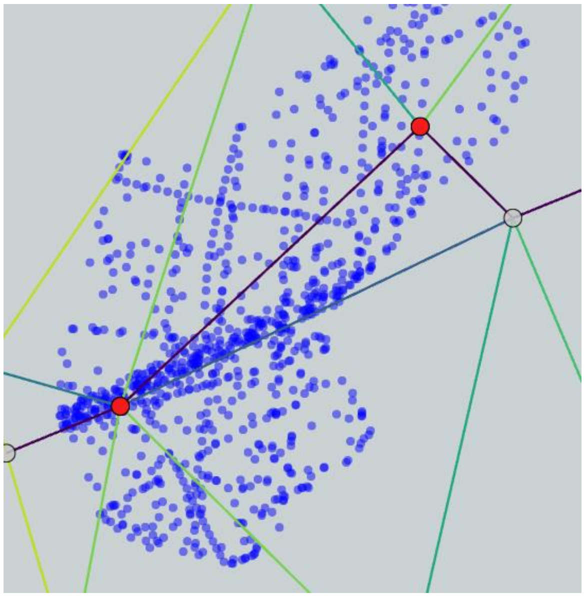
Knowing the cnt value, decision can also be made as which edges could be eliminated from the mesh. The good example is a route to the ports in midland, as can be seen in the figure below.



Such a phenomenon is the result of not satisfactory quality of AIS data and to some extent also the frequency at which data is probed. Again, these are only edges with a single connection that are removed later.

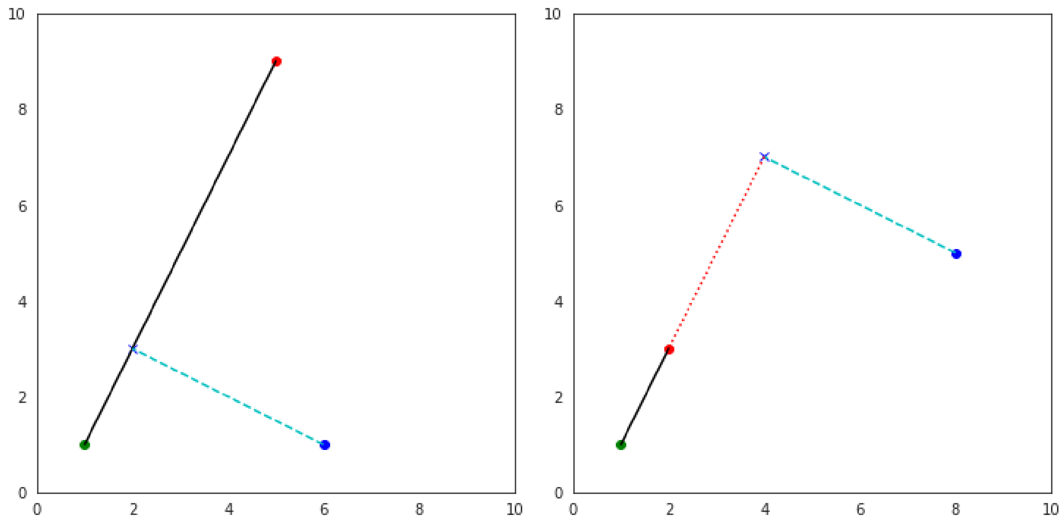
3.3 Corridors

In order to determine the width of the corridor, we visualize AIS points that are found around the edge connecting two waypoints. More specifically, there are the AIS points that were left by vessels travelling between these two waypoints. A sample cloud of AIS points is visualized below.

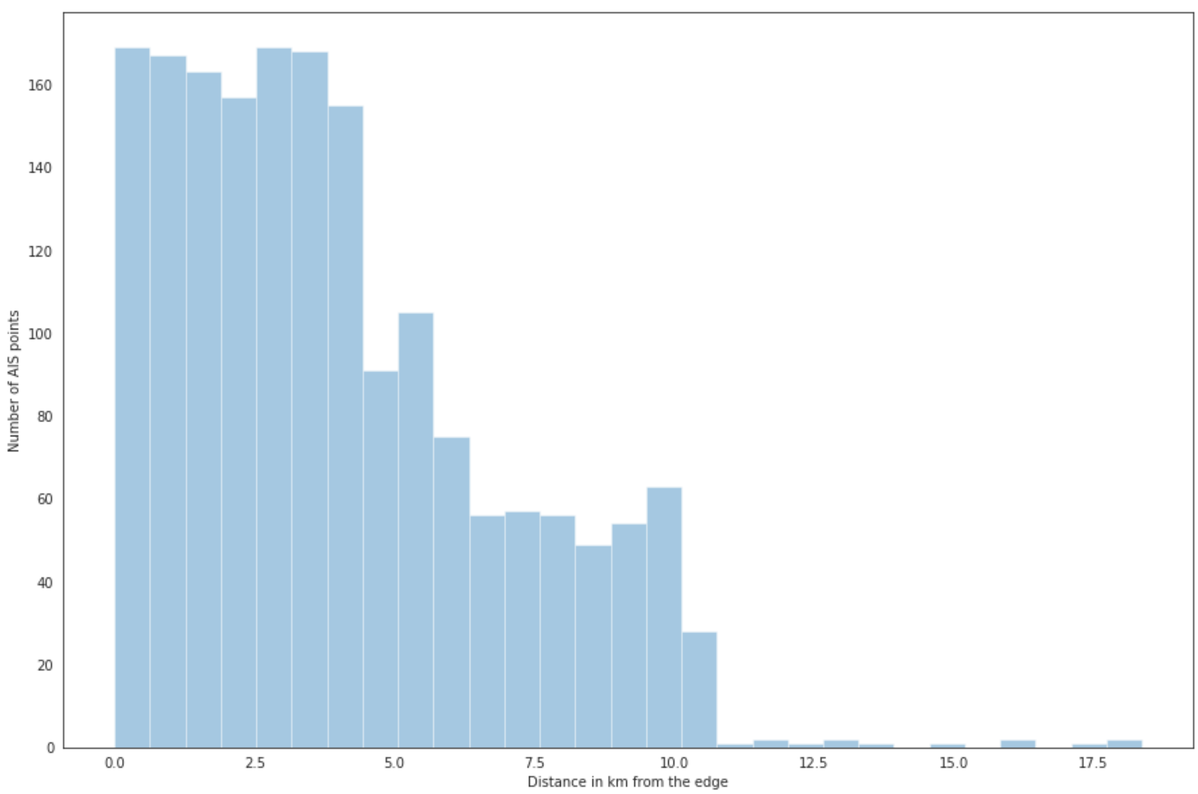


We project all points to the segment connecting two waypoints (red dots). We only consider these points exactly between waypoints, so the projection should be exactly on the segment and not its extension.

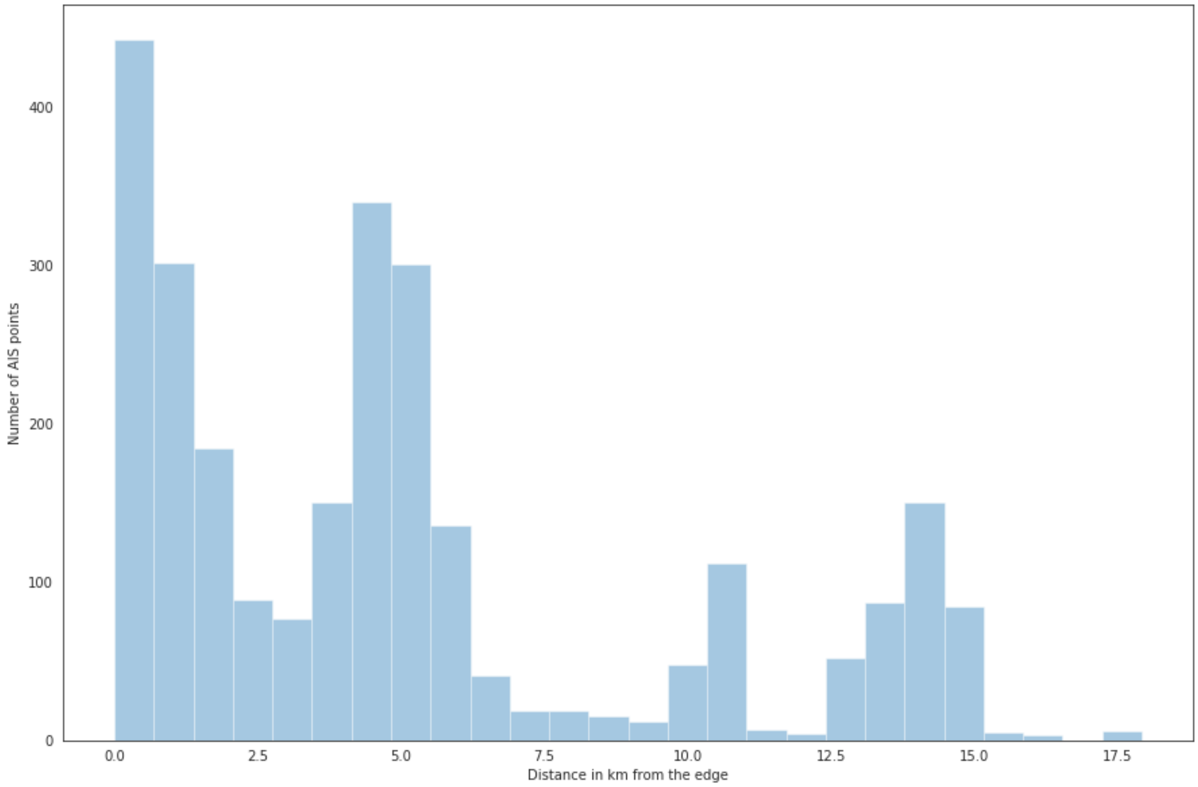
In the following figure the first case is counted in (left), the second is not considered (right).



After the projection, we are able to collect all distances (dashed blue line in figure above). Below we present some distributions. Here we see that AIS points are distributed more or less evenly.



There are also cases where AIS points seem to form groups where for examples traffic separation schemes are in effect.



There was a requirement that the corridor should encompass certain percent of ships. Having such distributions, it is trivial to calculate the requested percentile.

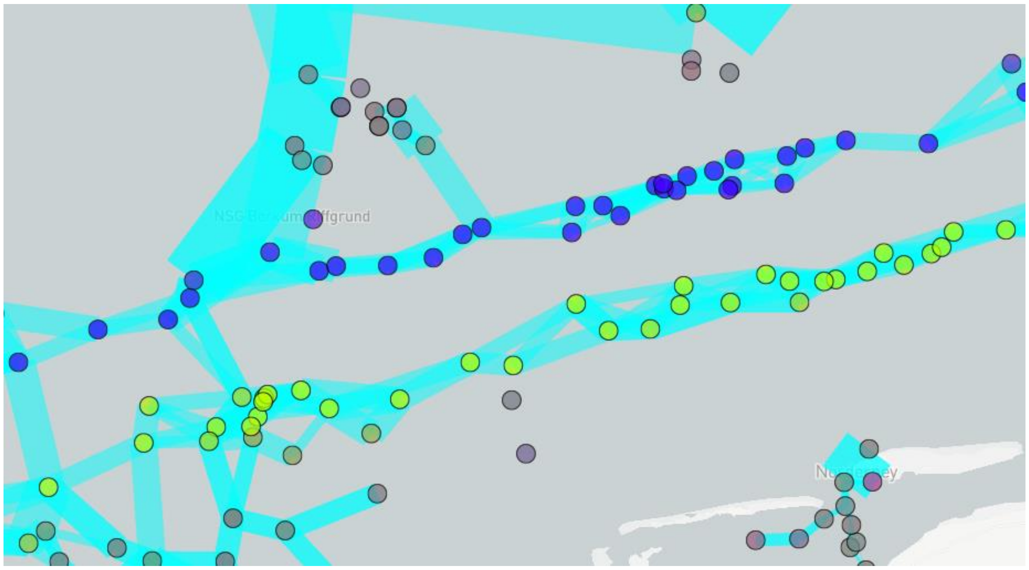
In the first case, the corridor width encompassing 90% vessels between waypoints 401 and 1990 is 8.59 km.

In the second case, the corridor width between waypoints 600 and 325 is 13.65 km.

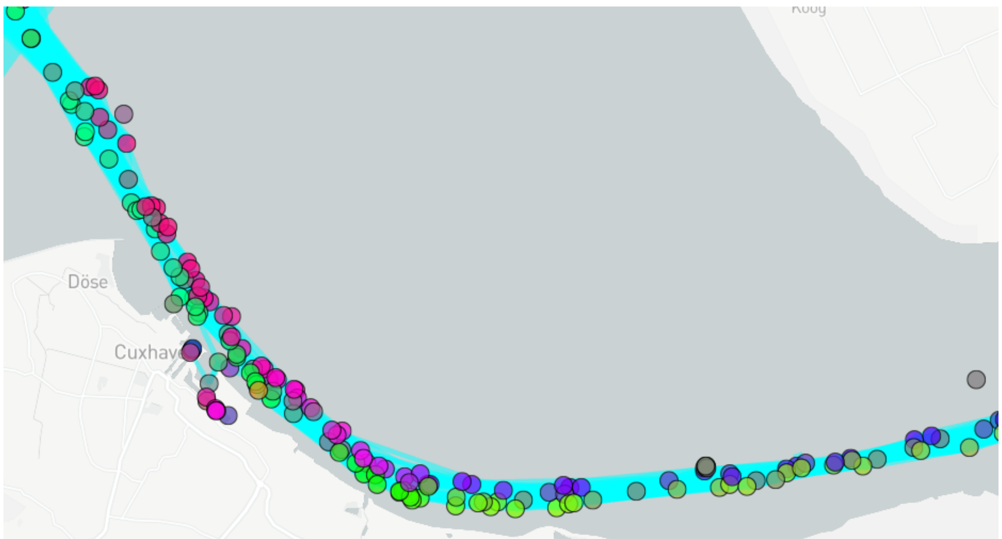
Corridors on the map

Corridors are visualized on the maps accordingly. Here the color is fixed (cyan, 50% transparency, can be parametrized). We only use real width of the corridor on the map (expressed in data in meters).

Visualization of the width of the corridors is convincing. They behave very well in traffic separation areas.



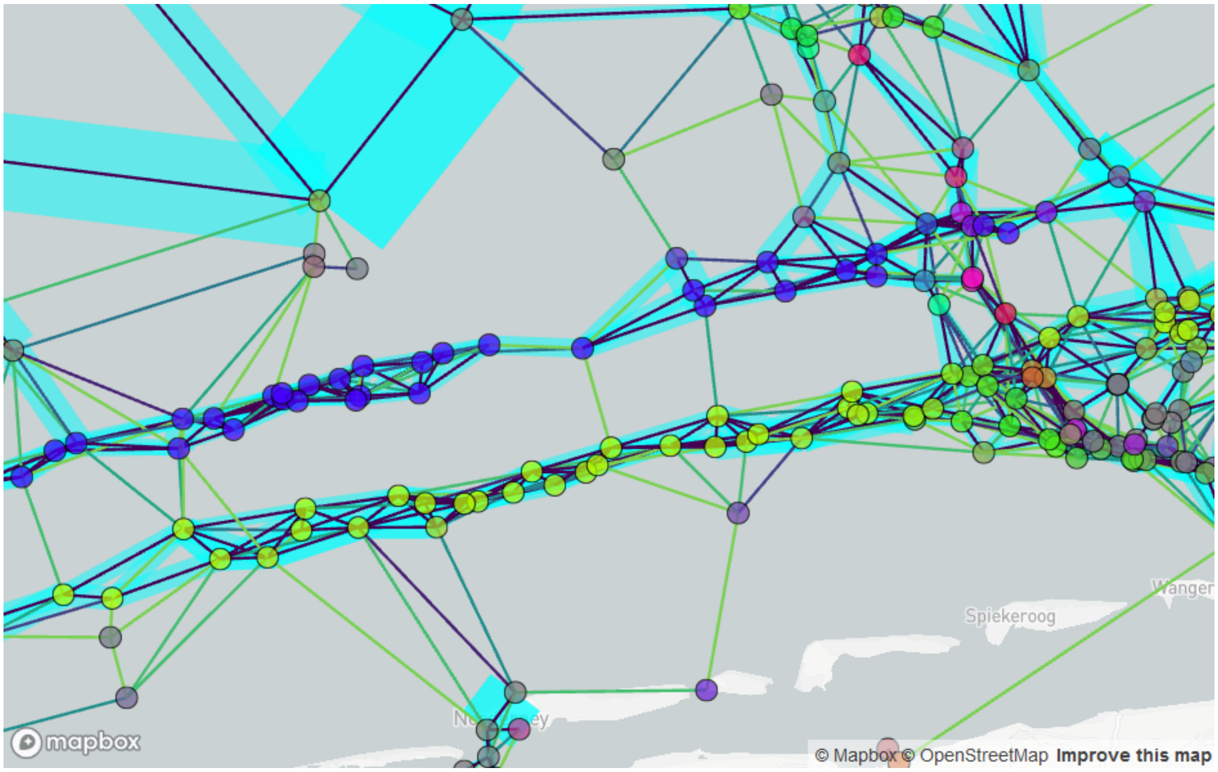
They also realistically represent entrance to the Elbe river (or any other port).



One of the interesting cases is presented in the figure below. As we can see, the corridor is very wide. This is because vessels need to go around the island (to the West).

3.4 Sample Charts

The all-inclusive chart was prepared to show all variables on a single map. It is called **mesh-german-cargo-navtor11.html** and stored in ownCloud (the project repository).

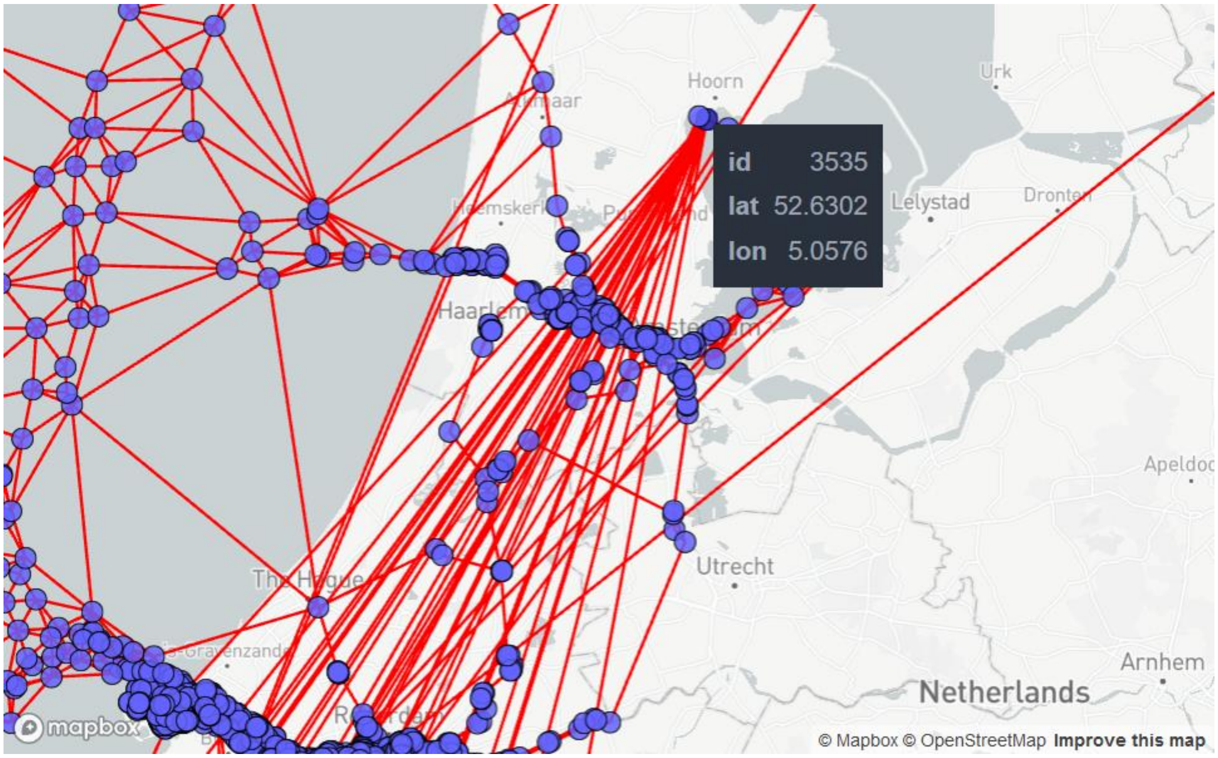


3.5 Identification of data quality problems

The analysis has been conducted for the following data scope:

- data provided by Navtor, for Oct and Nov 2019 ('../data/navtor_1011_2019')
- filtered by area: Baltic
- filtered by shiptype: cargo
- other parameters reflected in filename: navtor_1011_2019-baltic-c-kdb-part128e300cl10p10r2.0mf0.1

When we visualize the mesh on the map, we see a lot of problematic edges, i.e. going through the land. Probably the worst is the one near Amsterdam, presented in the figure below. The chart was prepared after filtering count, where count should be at least 5. It means that the filtering based on popularity is not a sufficient discriminator.



There are seven "transitions" contributing to the reconstructed edge (3535 to 2218):

	mmsi	timestamp_ais	status	lon	lat	speed	dist_to_wp_km	to_waypoint	from_waypoint	timestamp_delta
0	244645890	1570213175	0	4.467357	51.033417	0.0	7.651172	2218	3535	232
1	244645890	1570214222	0	4.468798	51.032883	0.1	7.702938	2218	3535	206
2	244645890	1570215123	0	4.468825	51.032982	0.0	7.710512	2218	3535	25
3	244645890	1570433061	0	4.469543	51.032917	0.0	7.748798	2218	3535	164
4	244645890	1570434022	0	4.469477	51.032887	0.0	7.743031	2218	3535	224
5	244645890	1570435580	0	4.469487	51.032902	0.0	7.744707	2218	3535	757
6	244645890	1570437981	0	4.469490	51.032909	0.0	7.745469	2218	3535	459

In fact, there are 173 edge outgoing from waypoint 3535 (also 176 incoming):

	from_waypoint	to_waypoint	cnt	lon	lat	avg_speed	min_speed	max_speed	stddev_speed	distance_km
0	3535	734	3	3.677196	51.091174	2.933333	1.1	4.7	1.800926	195.319778
1	3535	2704	2	4.371428	51.082785	7.750000	5.5	10.0	3.181981	178.224075
2	3535	2665	1	3.840953	51.346245	7.200000	7.2	7.2	NaN	165.466507
3	3535	1334	1	3.840148	51.274872	7.500000	7.5	7.5	NaN	172.576706
4	3535	2538	8	4.014550	51.447828	0.000000	0.0	0.0	0.000000	149.733276
5	3535	2398	1	3.708233	51.103279	5.700000	5.7	5.7	NaN	193.734894
6	3535	2430	2	4.094887	51.381771	7.100000	6.4	7.8	0.989950	153.461105
7	3535	1907	1	4.343748	51.094360	7.900000	7.9	7.9	NaN	177.806763
8	3535	3082	1	4.300108	51.342342	8.800000	8.8	8.8	NaN	152.580582
9	3535	1095	1	4.311977	51.116405	9.800000	9.8	9.8	NaN	176.150192
...	...	...	...	...	...	...	...	...	...	...
163	3535	2677	2	4.420406	51.884596	0.800000	0.0	1.6	1.131371	93.680542
164	3535	3077	1	4.127760	51.374813	6.200000	6.2	6.2	NaN	153.913895
165	3535	2798	1	4.441225	51.891071	7.700000	7.7	7.7	NaN	92.355705
166	3535	164	1	4.234208	51.428074	8.600000	8.6	8.6	NaN	145.144440
167	3535	1753	6	4.407070	51.886183	0.000000	0.0	0.0	0.000000	93.978310
168	3535	911	7	4.450561	51.880790	0.000000	0.0	0.0	0.000000	93.093262
169	3535	629	1	3.967123	51.415932	9.200000	9.2	9.2	NaN	154.846344
170	3535	2054	1	4.333903	51.098045	10.900000	10.9	10.9	NaN	177.961578
171	3535	615	1	3.669253	51.089340	0.000000	0.0	0.0	NaN	196.878311
172	3535	2988	4	3.598557	51.390090	1.900000	0.0	5.4	2.412468	170.521423

173 rows × 11 columns

We then studied details of AIS data (with assigned waypoints) and we found out that there are possibly two different ships with the same MMSI number:

mmsi	status	lon	lat	shiptype	course	heading	speed	draught	waypoint	dist_to_wp_km	datetime_ais
244645890	0	4.467357	51.033417	79	178.399994	NaN	0.0	0.1	2218	7.651172	2019-10-04 18:19:35
244645890	5	5.087640	52.642174	79	23.600000	71.0	0.0	0.1	3535	2.425236	2019-10-04 18:33:33
244645890	5	5.087640	52.642174	79	23.600000	71.0	0.0	0.1	3535	2.425236	2019-10-04 18:33:36
244645890	0	4.468798	51.032883	79	29.299999	NaN	0.1	0.1	2218	7.702938	2019-10-04 18:37:02
244645890	5	5.087637	52.642174	79	121.400002	70.0	0.0	0.1	3535	2.424942	2019-10-04 18:51:38
244645890	0	4.468825	51.032982	79	0.000000	NaN	0.0	0.1	2218	7.710512	2019-10-04 18:52:03

By querying the full AIS database we found the following:

timestamp_ais	mmsi	status	turn	speed	lon	lat	course	heading	imo	callsign	shipname	shiptype	to_b
1570215123	244645890	0	NaN	0.0	4.468825	51.032982	0.000000	NaN	NaN	PE3999	DELIVERY	79	70.0
1570213175	244645890	0	NaN	0.0	4.467357	51.033417	178.399994	NaN	NaN	PE3999	DELIVERY	79	70.0
1570214013	244645890	5	0.0	0.0	5.087640	52.642174	23.600000	71.0	NaN	PE3999	DELIVERY	79	70.0
1570214016	244645890	5	0.0	0.0	5.087640	52.642174	23.600000	71.0	NaN	PE3999	DELIVERY	79	70.0
1570214222	244645890	0	NaN	0.1	4.468798	51.032883	29.299999	NaN	NaN	PE3999	DELIVERY	79	70.0
1570215098	244645890	5	0.0	0.0	5.087637	52.642174	121.400002	70.0	NaN	PE3999	DELIVERY	79	70.0

The ships cannot be even distinguished by all other ID attributes. The only difference is navigational status and course/heading.

Possible explanations:

- timestamp is not synchronized and in fact we observe the same ship but at different moments
- incorrectly propagated callsign and shipname based on the same MMSI.

The latter is not very probable because when we look at all data and look for combinations of identifiers, we have the following data about the studied MMSI:

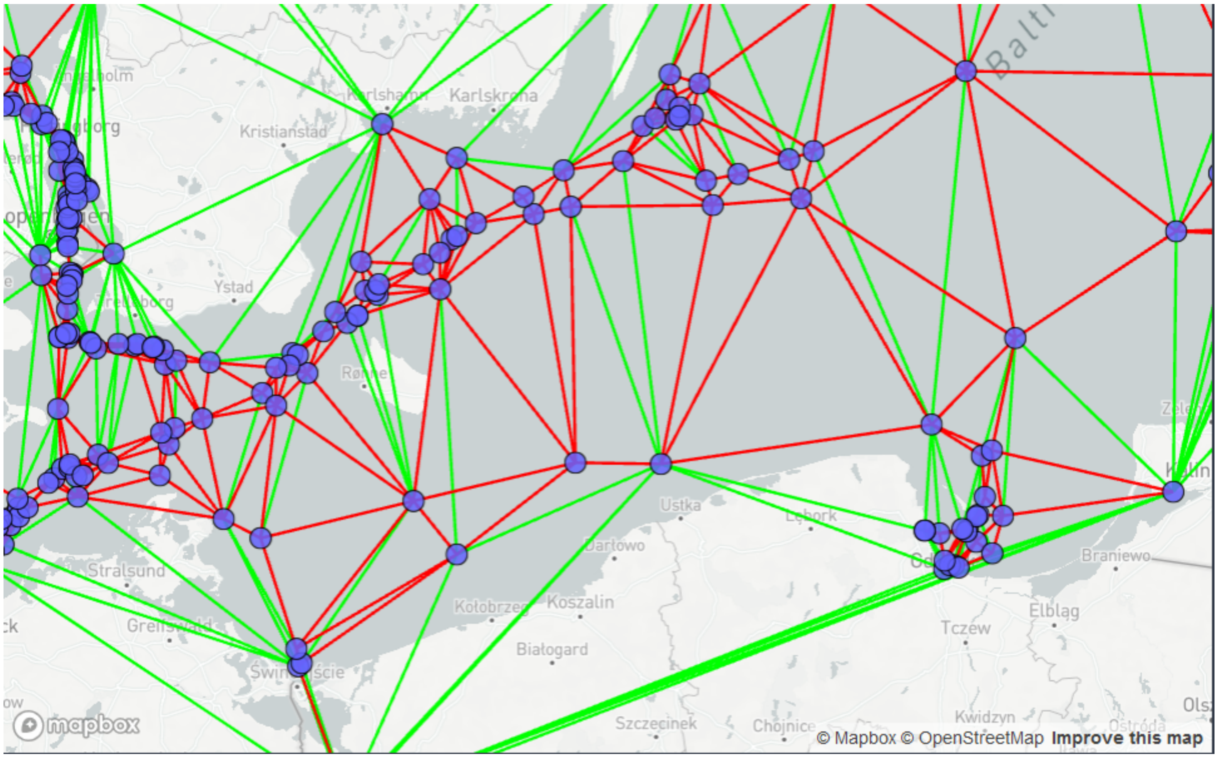
mmsi	imo	callsign	shipname
244645890	NaN	PG8622	PROGRESS
244645890	NaN	PB8040	HYDROVAC 11
244645890	NaN	PB8046	HYDROVAC 11
244645890	NaN	PB8046	HYDROP
244645890	NaN	PE3999	DELIVERY
244645890	NaN	PG5078	BAS ANNE

There is just one ship using the callsign PE3999 but two ships with the same name:

mmsi	imo	callsign	shipname
244051287	NaN	PG3198	DELIVERY
244645890	NaN	PE3999	DELIVERY

Delaunay triangulation

It shows what are the relations between the closest neighbors. The green line represents all possible connections. The red lines show what connections were actually identified by us.

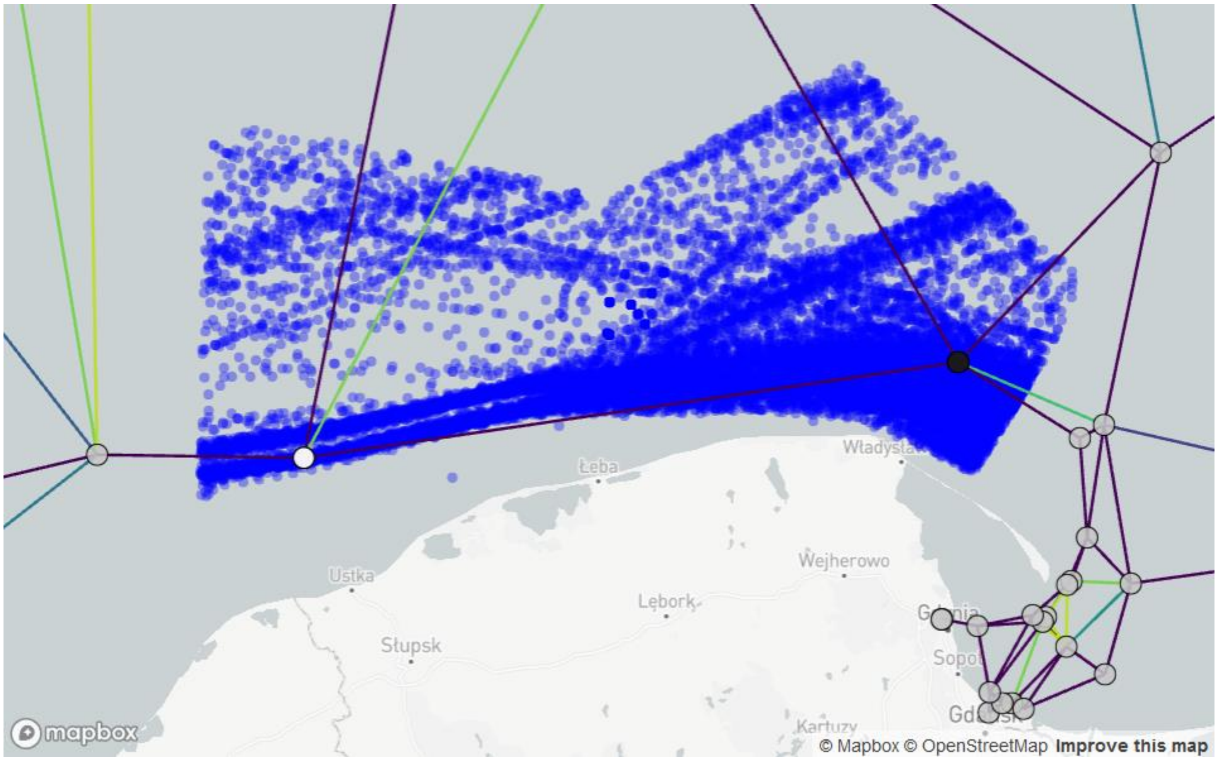


3.6 Corridors

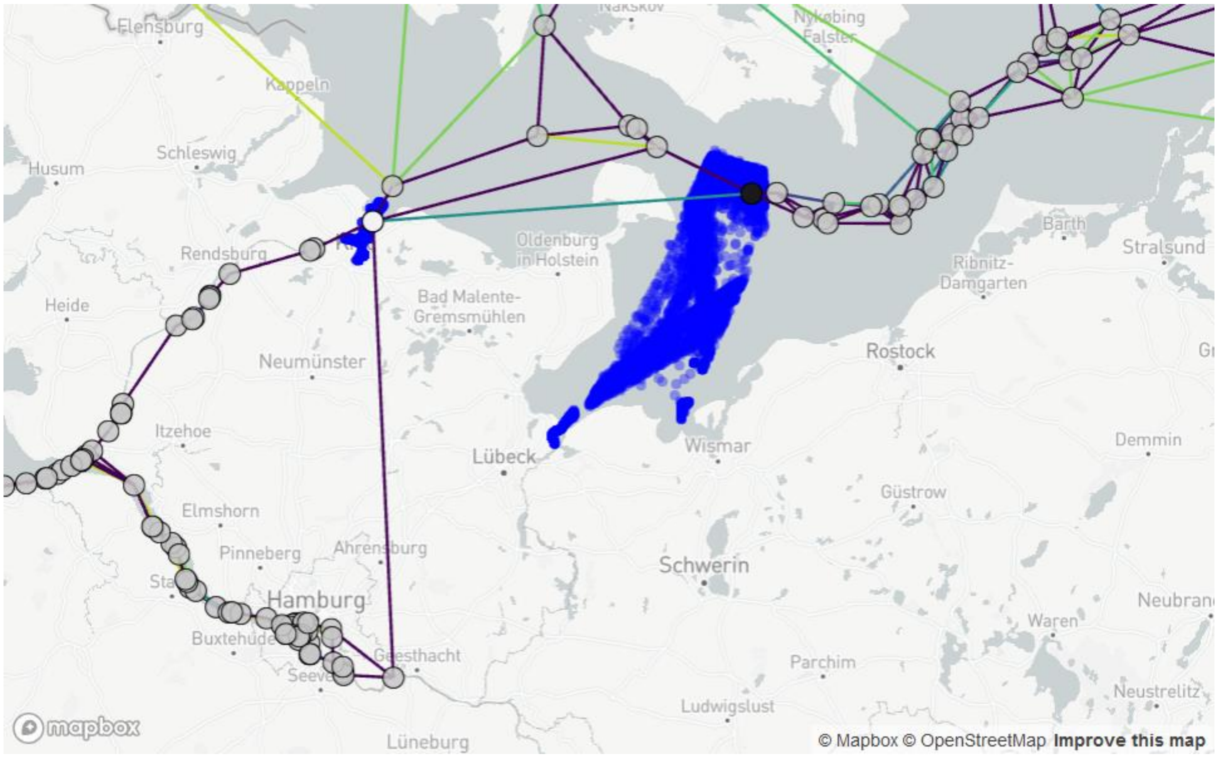
3.6.1 AIS-Cloud

Calculation of the width of the corridor depends on points assigned to the edge connecting two waypoints.

Therefore we need to collect all AIS messages.

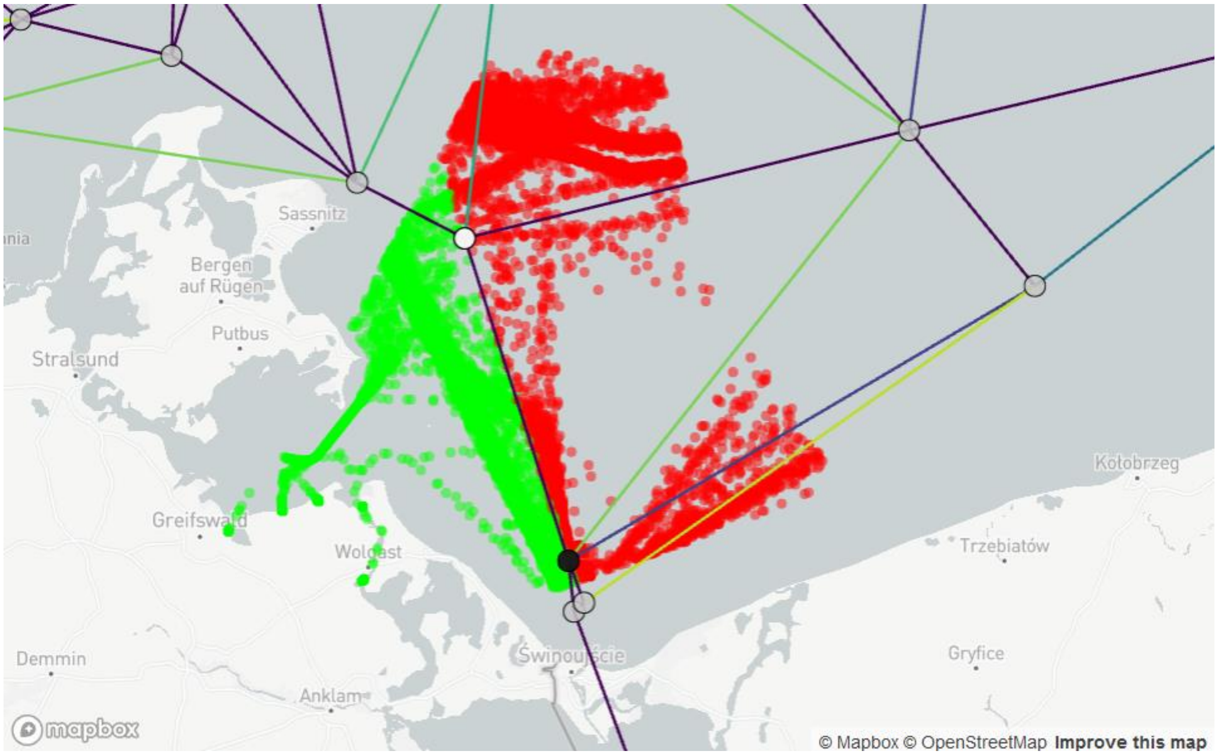


For identification of the issues, it is possible to visualize all points in a form of a cloud. Here we see why there is an incorrect connection over the land – the missing waypoint in Lübeck.

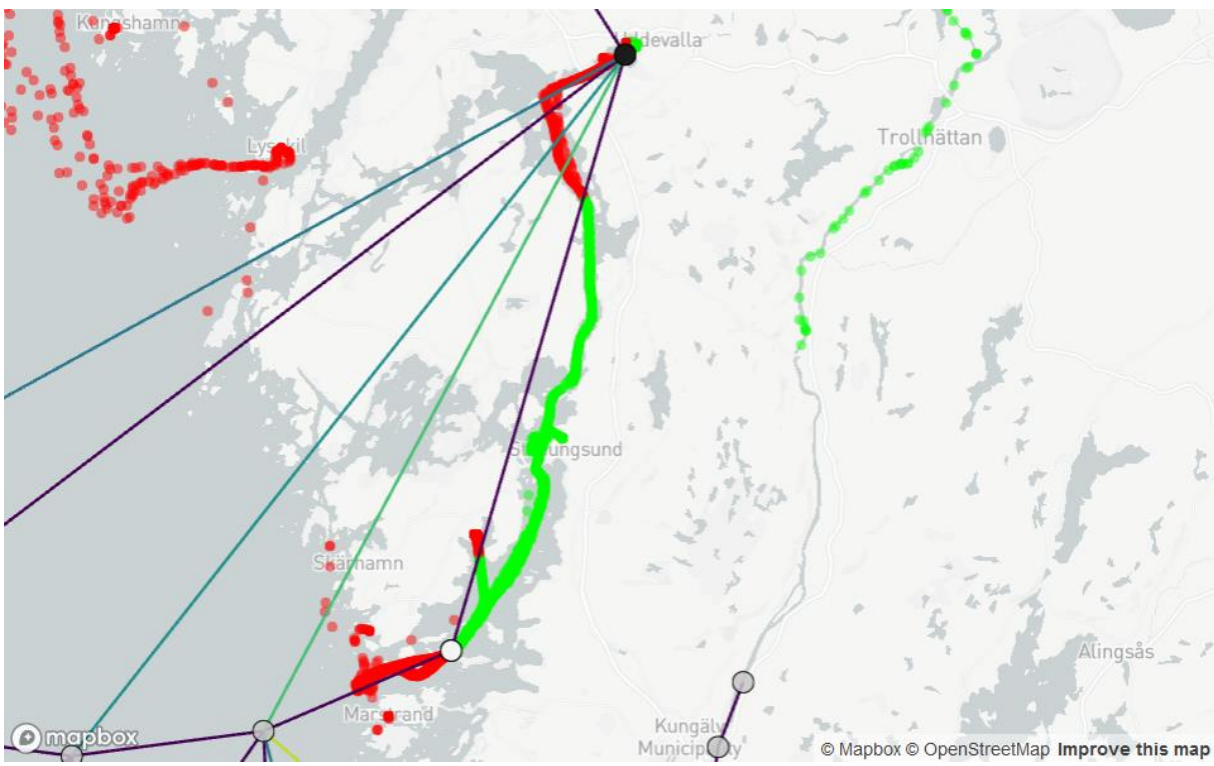


3.6.2 Cloud with Separation

The cloud of points can also be visualized by splitting into left (red) and right (green), to allow calculation of the port and star board corridors, respectively. The starting waypoint is white, the destination is black.



Distinction of the distance can be useful for analysis of the corridors in narrow entries.



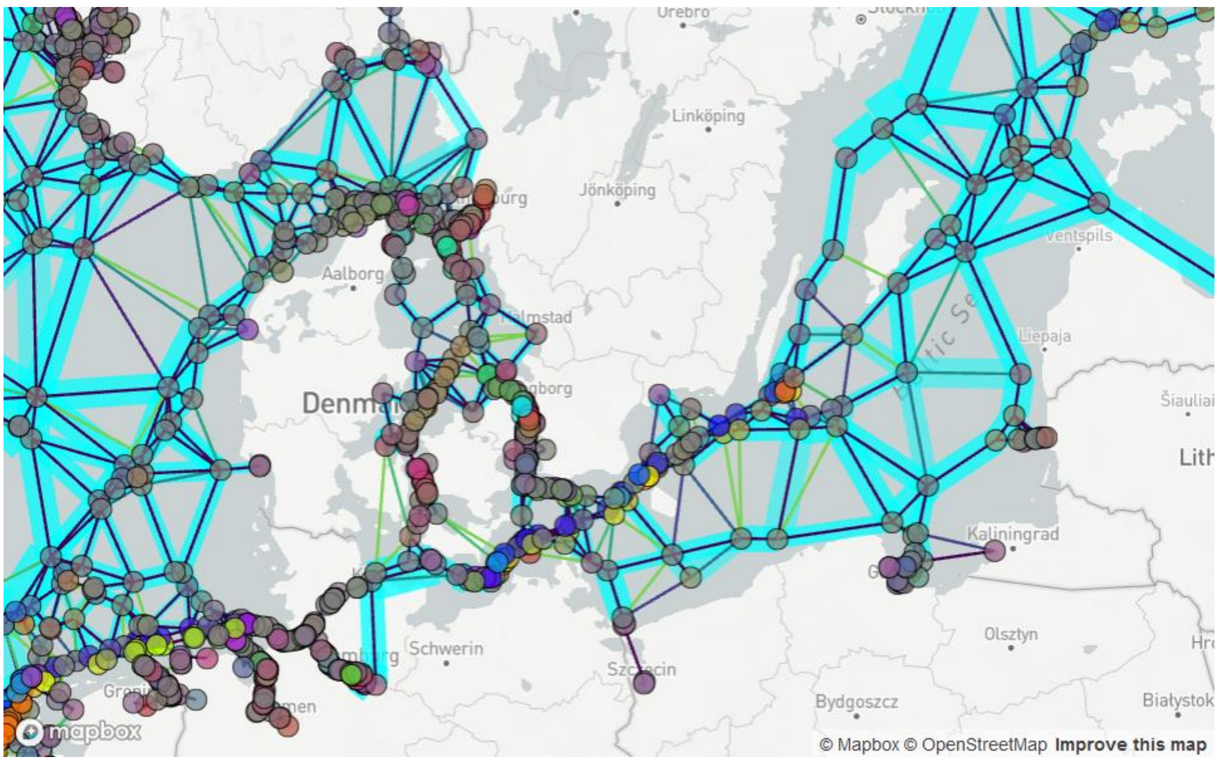
3.6.3 Calculation of the corridors on improved data

It is important to note that calculation of the corridors for ca. 23.000 edges took almost 14 hours.

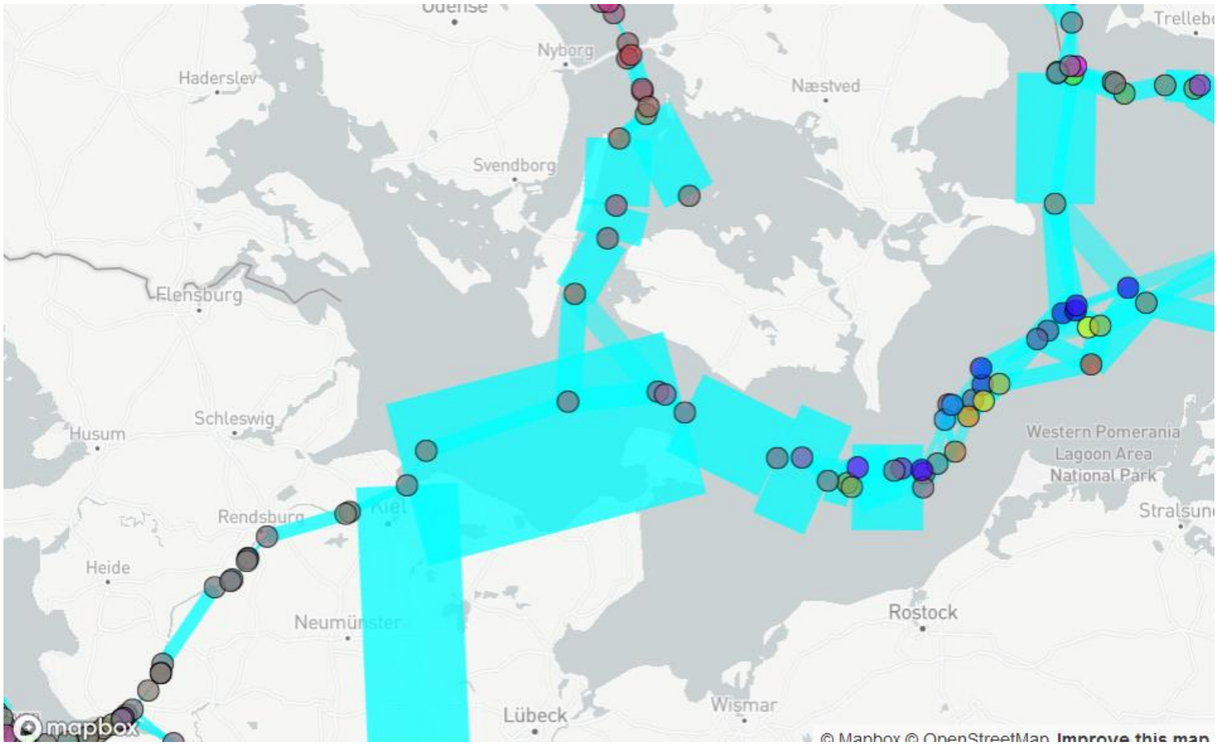
```
%%time

deck_edges_corr2_df = deck_edges_df.loc[deck_edges_df.cnt>=10]
deck_edges_corr2_df['corr_width_m'] = deck_edges_corr2_df.apply(get_corridor_width_from_rc
```

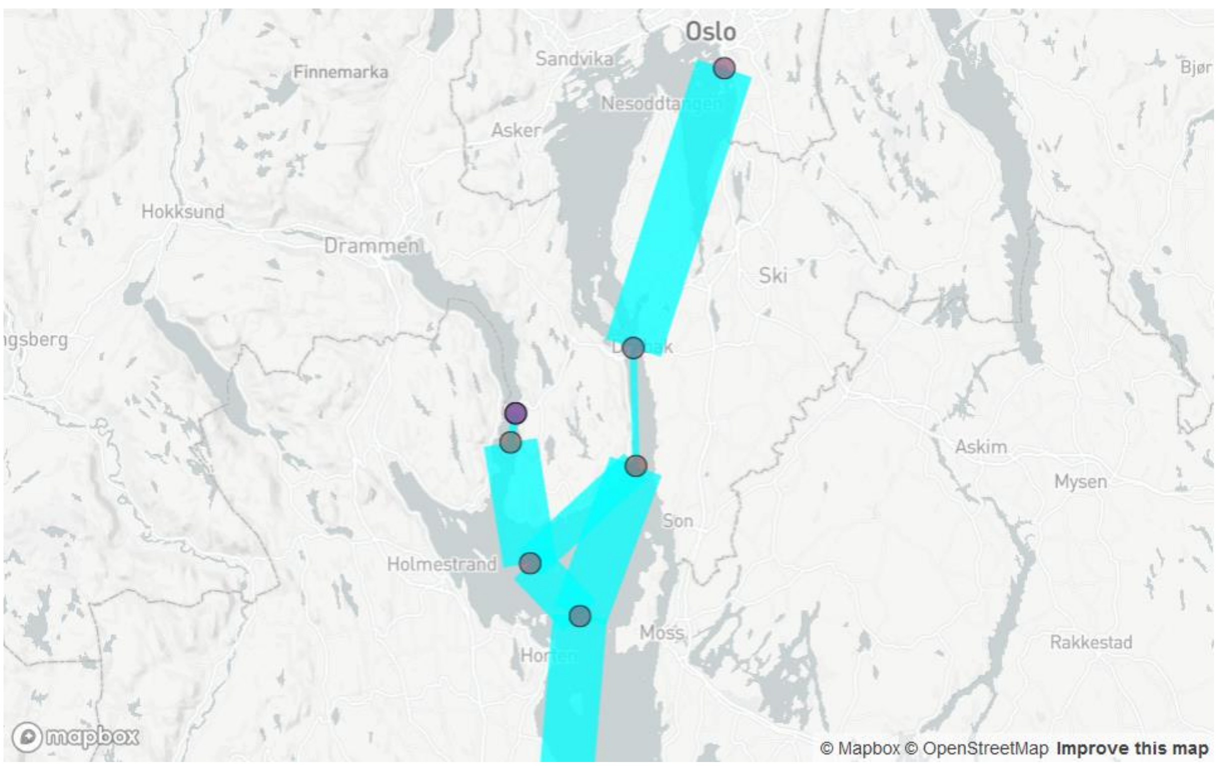
```
CPU times: user 3h 23min 45s, sys: 18h 30min 46s, total: 21h 54min 31s
Wall time: 13h 51min 27s
```

High-level overview of corridors



Nevertheless, the more waypoints, the more details can be reflected. If the waypoints are close to each other, and the connection is straight, the corridor is narrow. When the waypoints are far from each other and there are many ways to travel (e.g. open sea or going around land in a complex terrain), then the corridor is wide.


Nyborg
Haderslev
Næstved
Trelleb
Svendborg
Flensburg
Schleswig
Husum
Western Pomerania
Lagoon Area
National Park
Rendsburg
Kiel
Stralsun
Heide
Neumünster
Rostock
Lübeck
Wismar
mapbox
© Mapbox © OpenStreetMap Improve this map


Oslo
Sandvika
Finnemarka
Nesoddtangen
Bjør
Asker
Hokksund
Drammen
Ski
Askim
Mysen
Son
Holmestrand
Moss
Rakkestad
mapbox
© Mapbox © OpenStreetMap Improve this map

4 HANSA Meshes – Labels and Analysis

In this section the meshes are specified that have been generated. Furthermore, the different visualization methods for the meshes are described as well.

4.1 Data Sources and Areas

The meshes were generated based on two data sources:

1. Navtor – data from October and November 2019 was used.
2. Sprint – data from October and November 2019 was used.

The following geographical areas with respective bounding boxes were used:

- 'german': (53N, 2.5E), (57N, 9.5E),
- 'baltic': (50.65N, 4.65W), (71.50N, 35.5E),
- 'sbaltic': (53.1N, 13E), (60.94N, 30.73E).

Please note that the label `baltic’ is not strictly appropriate as this label was used to mean “includes also Baltic Sea”. In fact it encompasses the biggest and most heterogeneous area including: the German Bight, the North Sea and the Baltic Sea.

Navtor data was partitioned using ‘german’ and ‘baltic’ areas, whereas Sprint data was filtered using ‘sbaltic’ area.

4.2 Classification Categories

4.2.1 Vessel Type

There are three vessels types for which separate recommended corridors might be calculated. These are:

- Cargo vessels (AIS type 70-79),
- Tankers (AIS type 80-89),

- Passenger vessels (AIS type 60-69).

4.2.2 Draught

The following table presents the class names and ranges for draught depending on the vessel type. They were defined in the previous project works and presented in Deliverable 2.3. These defined classes were used to prepare separate meshes. It was then tested if the movement patters of vessels are changing with regard to the class. Conclusions are summarized in a common visualization. Indeed, there are observed significant differences between movement patterns of vessels.

Table 1: Classification of vessel types using draught

Draught in meters	Very small	Small	Medium	Large	Very large
All	(0 – 1>	(1 – 6>	(6 – 10>	(10 – 15>	>15
Tankers	(0 – 1>	(1 – 8>	(8 – 13>	(13 – 17>	>17
Cargo	(0 – 1>	(1 – 4>	(4 – 7>	(7 – 11.5>	>11.5
Passenger	(0 – 1>	(1 – 5>	(5 – 7>	(7 – 9 >	>9

4.2.3 Length and Width

The following table presents the class names and ranges for length and width depending on the vessel type (defined previously in Deliverable D2.3). These classes were used to prepare separate meshes. It was then tested if the movement patters of vessels are changing with regard to the length or width of the vessel. Conclusions are summarized in common visualizations. Indeed, there are observed significant differences between movement patterns of vessels.

Table 2: Vessel classification based on length and width

Length in meters	Very small	Small	Medium	Large	Very large
All	(0 – 25>	(25 – 75>	(75 – 125>	(125 – 200	>200
Tankers	(0 – 75>	(75 – 170>	(170 – 210>	(210 – 310>	>310
Cargo	(0 – 25>	(25 – 75>	(75 – 125>	(125 – 200>	>200
Passenger	(0 – 75>	(75 – 125>	(125 – 200>	>200	
Width in meters					
All	(0 – 15>	(15 – 25>	(25 – 35>	>35	
Tankers	(0 – 13>	(13 – 25>	(25 – 39>	(39 – 50>	>50
Cargo	(0 – 13>	(13 – 25>	(25 – 39>	(39 – 50>	>50
Passenger	(0 – 25>	(25 – 50>	(50 – 75>	(75 – 100>	>100

4.2.4 Wind Speed and Wave Height

Out of ten weather variables two were chosen for testing differences in pattern movements: wind speed and wave height. Both were quantized using Beaufort scale as presented in the table below. These variables were integrated with AIS data but no visualizations were prepared.

Table 3: Beaufort scale

Beaufort number	Description	Wind speed (wind_speed)	Wave height (sea_surface_height_above_sea_level / sla)
0	Calm	<0.5 m/s	0 m
1	Light air	0.5–1.5 m/s	0–0.3 m
2	Light breeze	1.6–3.3 m/s	0.3–0.6 m
3	Gentle breeze	3.4–5.5 m/s	0.6–1.2 m
4	Moderate breeze	5.5–7.9 m/s	1–2 m
5	Fresh breeze	8–10.7 m/s	2–3 m
6	Strong breeze	10.8–13.8 m/s	3–4 m
7	High wind, moderate gale, near gale	13.9–17.1 m/s	4–5.5 m
8	Gale, fresh gale	17.2–20.7 m/s	5.5–7 m
9	Strong/severe gale	20.8–24.4 m/s	7–9 m
10	Storm, whole gale	24.5–28.4 m/s	9–11.5 m
11	Violent storm	28.5–32.6 m/s	11.5–16 m
12	Hurricane force	>36.6 m/s	> 14 m

4.3 Combinations and Variants

Combinations encompass different parameters that were used for generation of waypoints. They refer to the area and also the parameters of the genetic algorithm. Once the waypoints were ready, they were used to prepare different variants, i.e. meshes were generated after filtering data meeting specific subcriteria.

The following combinations were used for Navtor data:

- ga_area = ['baltic', 'german'] – refers to the geographical area filtered as specified in the first section;
- ga_type = ['c', 't', 'p'] – refers to the vessel type, i.e. cargo, tanker, and passenger respectively;
- ga_part = ['512', '256', '128'] – refers to the number of partitions; more partitions will yield more waypoints and thus more complex meshes;
- ga_cl_p = ['cl20p20', 'cl10p10', 'cl2p10'] – refers to the number of solutions in a partition; it is combined from other parameters, i.e. chromosome length (number of genes) and population size (number of chromosomes); the higher the product of these numbers, the more complex the mesh; for simplification we refer to these parameter as L – larger, M – medium, and S – small.

Thus there are 54 combinations. The folder name is constructed from the following pattern:

navtor_1011_2019-{area}-{vtype}-kdb-part{partition}e300{cl_p}r0.3mf0.1

The following combinations were used for Sprint data:

- ga_area = ['sbaltic'];
- ga_type = ['c', 't', 'p'];
- ga_part = ['256', '128', 64'] – refers to the number of partitions; more partitions will yield more waypoints and thus more complex meshes:
- ga_cl_p = ['cl20p20', 'cl10p10', 'cl2p10'] – refers to the number of solutions in a partition; it is combined from other parameters, i.e. chromosome length (number of genes) and population size (number of chromosomes); the higher the product of these numbers, the more complex the mesh; for simplification we refer to these parameter as L – larger, M – medium, and S – small.

Please note that 'sbaltic' area is smaller than 'baltic' therefore we reduced the number of partitions in the iteration (i.e. no 512).

As a result, there are 27 combinations. The folder name is constructed from the following pattern:

sprint_1011_2019-{vtype}-kdb-part{partition}e300{cl_p}r0.3mf0.1

Variants refer only to draught, length of the vessel and width of the vessel.

The following variants are considered, for each combination:

- all
- draught: very_small, small, medium, large, very_large,
- length: very_small, small, medium, large, very_large,
- width: very_small, small, medium, large, very_large.

Label 'all' means that no subcriteria are used for filtering and this is the representative mesh for a combination. Altogether, there are 16 variants.

Taking into account 81 combinations and 16 variants we have a theoretical number of meshes: 1296. However, not all variants produced meshes (due to lack of data after filtering using a defined set of criteria), therefore the actual number of meshes is smaller.

4.3.1 Naming convention

The naming convention for folder is valid on the HANSA server. For API we have simplified the labels in the following way (path in unix to label):

sprint_1011_2019-c-kdb-part128e300cl10p10r3.0mf0.1-graphweather -> sprint-midbaltic-cargo-part128M

navtor_1011_2019-baltic-t-kdb-part256e300cl20p20r3.0mf0.1-graphweather -> navtor-all-tanker-part256L

navtor_1011_2019-german-p-kdb-part512e300cl2p10r3.0mf0.1-graphweather -> navtor-german-passenger-part512S

Labels are further refined when we want to access certain variant. See section Appendix 1 List of mapping between combinations and labels.

4.4 Visualization of Meshes

4.4.1 Popularity of the Edges (cnt)

In the popularity visualization we emphasize which legs are the most often travelled by vessels. We take into account the attribute 'cnt' which is taken from the aggregation step. It shows the count of vessels travelling from one specified waypoint to the other (the graph is directional).

For coding of colors we use the following scale. The variable 'cnt' is scaled linearly until median and then the color stays as dark violet (all above median). Edge in yellow in most cases means just cnt=1.

In the series of visualizations, we show the impact of number of waypoints, which in turn depends on the parameters of the genetic algorithms, referred to as S, M, L.

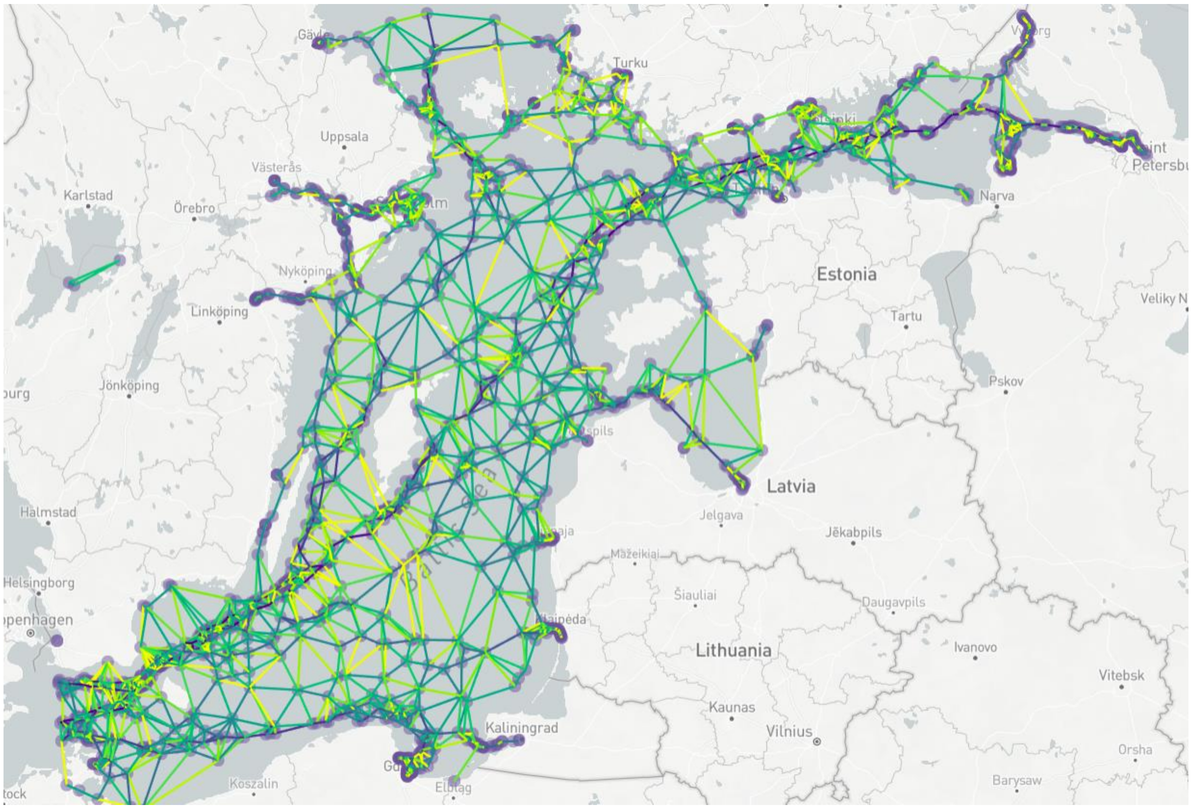


Figure 3: Sprint, South Baltic, type-cargo, partition 128, size S

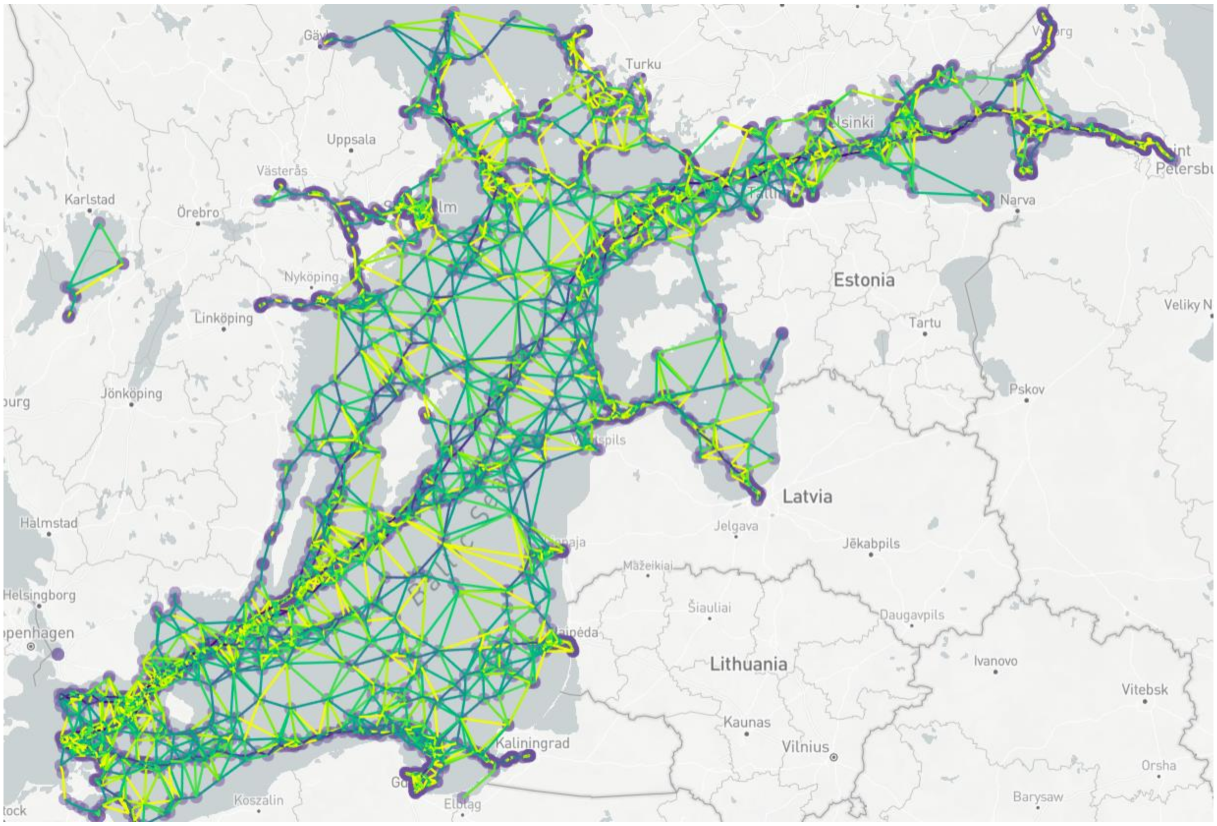


Figure 4: Sprint, South Baltic, type cargo, partition 128, size M

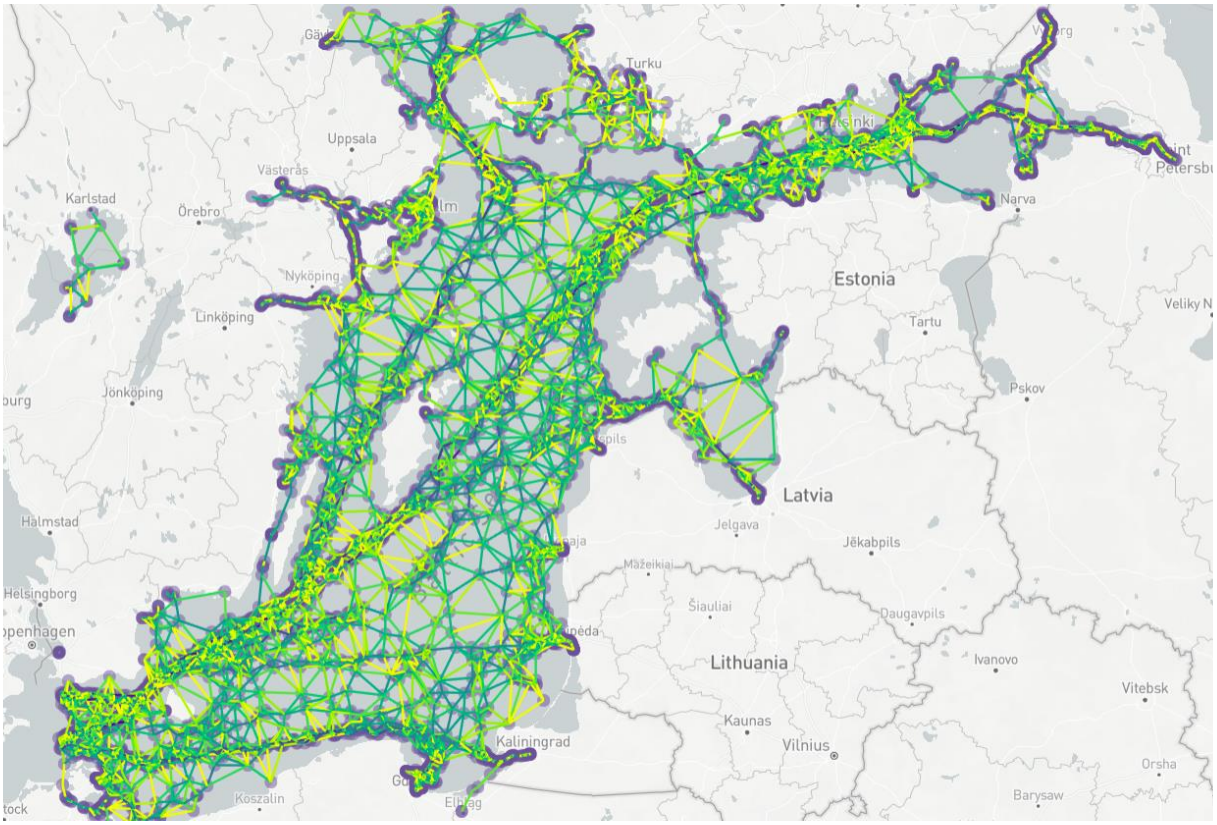


Figure 5: South Baltic, type cargo, partition 128, size L

Differences can be studied in a big as well as in a small scale. The following figures show differences for the Gulf of Gdańsk.

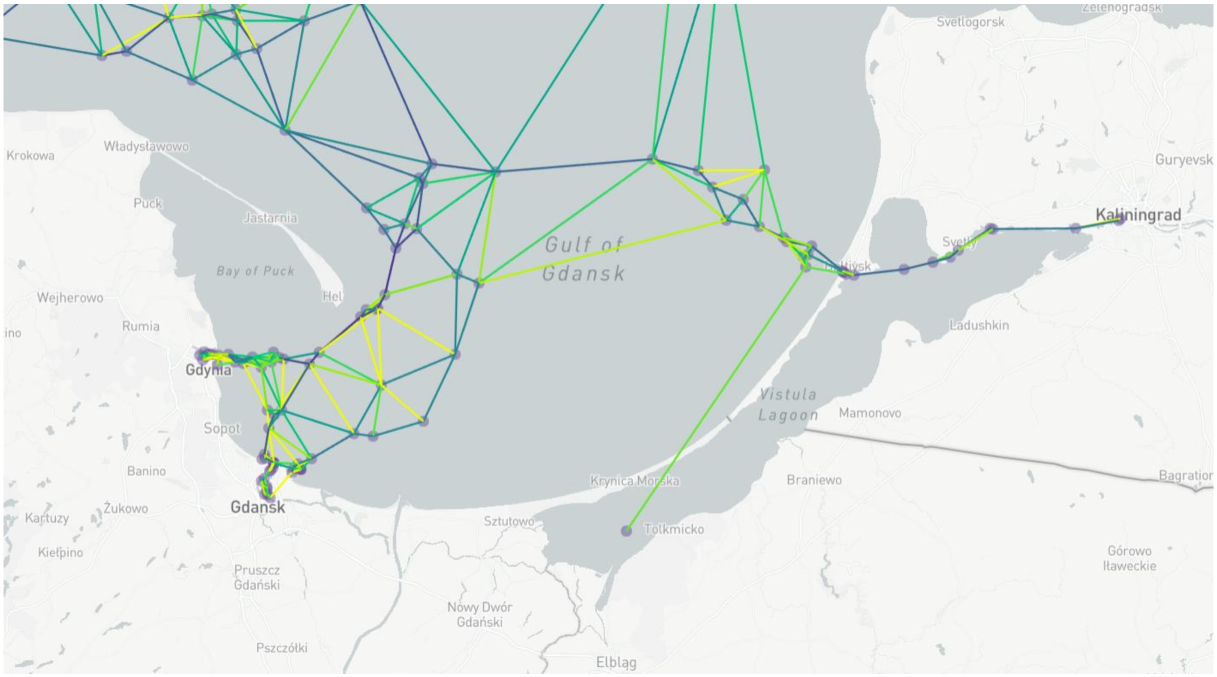


Figure 6: Sprint, Gulf of Gdańsk, type cargo, partition 128, size S

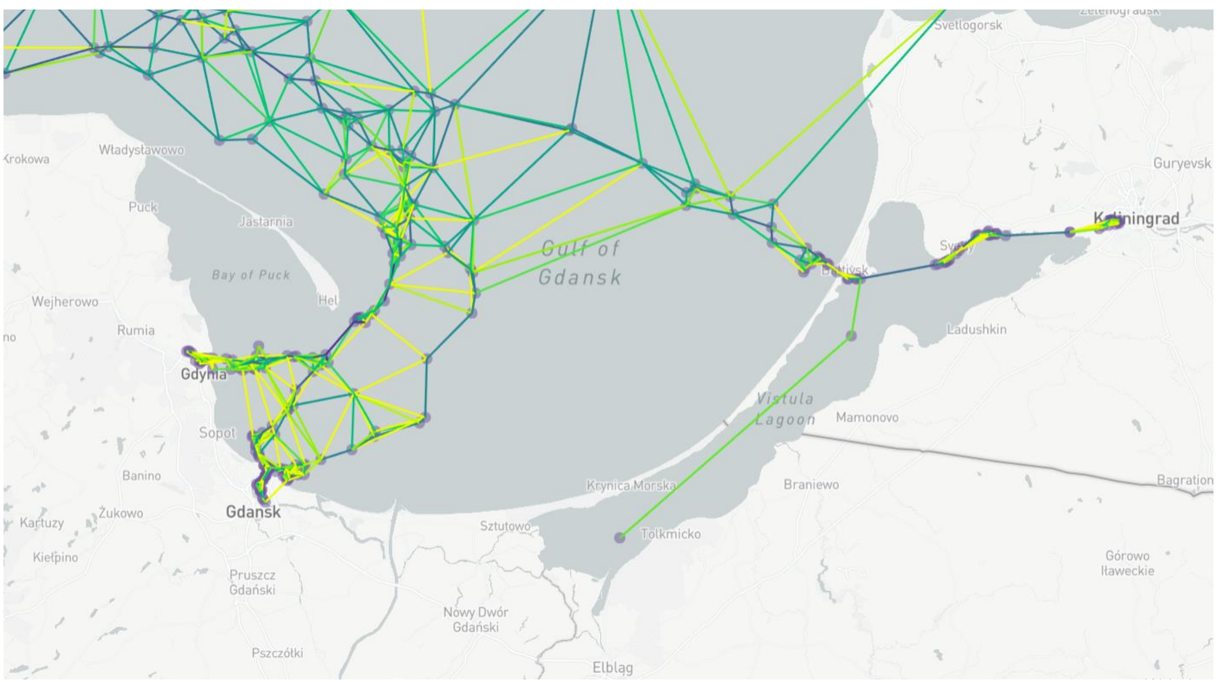


Figure 7. Sprint, Gulf of Gdańsk, type cargo, partition 128, size M

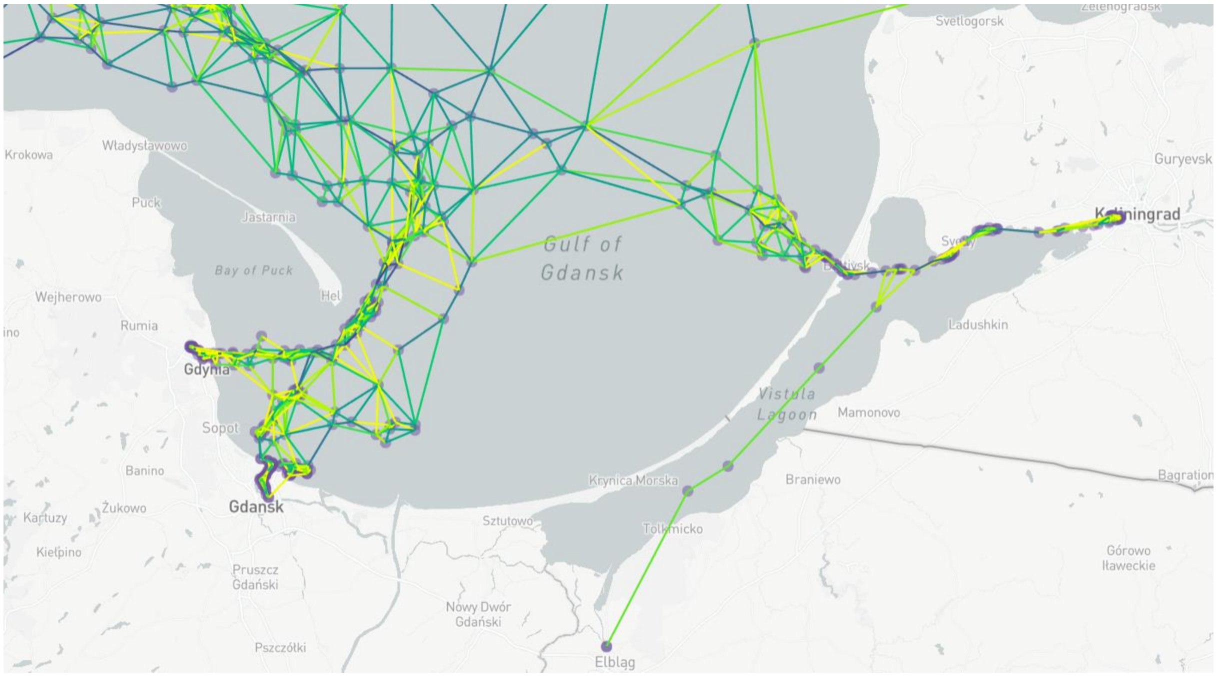


Figure 8. Sprint, Gulf of Gdańsk, type cargo, partition 128, size L

The differences in the number of waypoints can also be caused by the number of partitions. The bigger number of partitions, the smaller the single partition, and the more detailed is the mesh. Examples are presented below for the South Baltic.

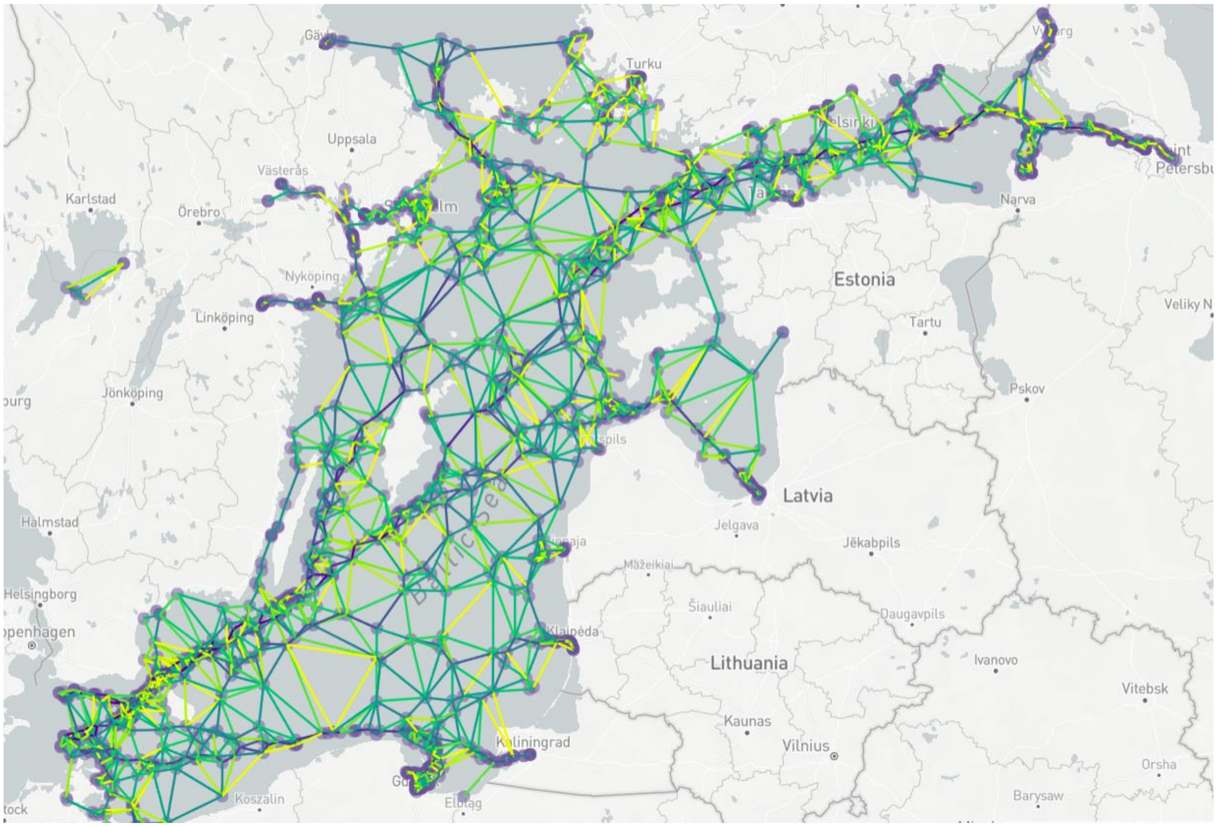


Figure 9. Sprint, South Baltic, type cargo, size M, 64 partitions

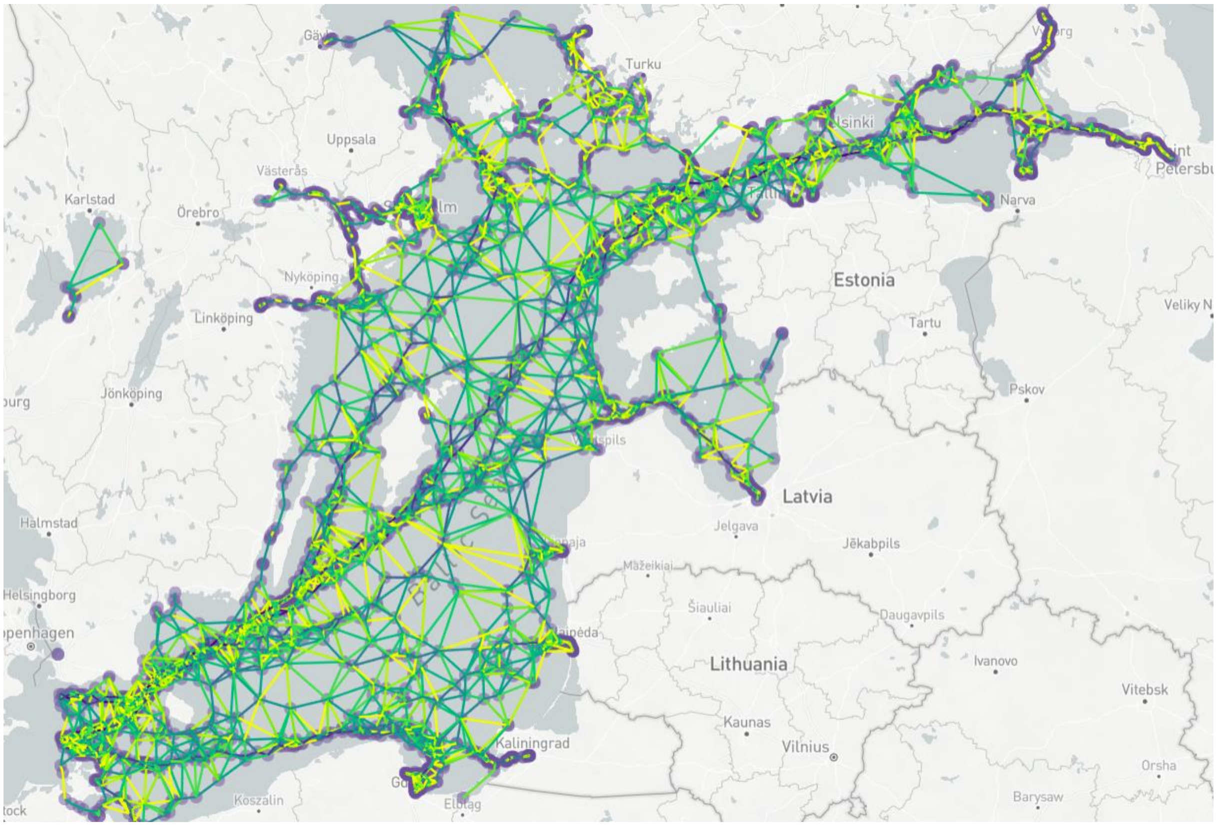


Figure 10. Sprint, South Baltic, type cargo, size M, 128 partitions

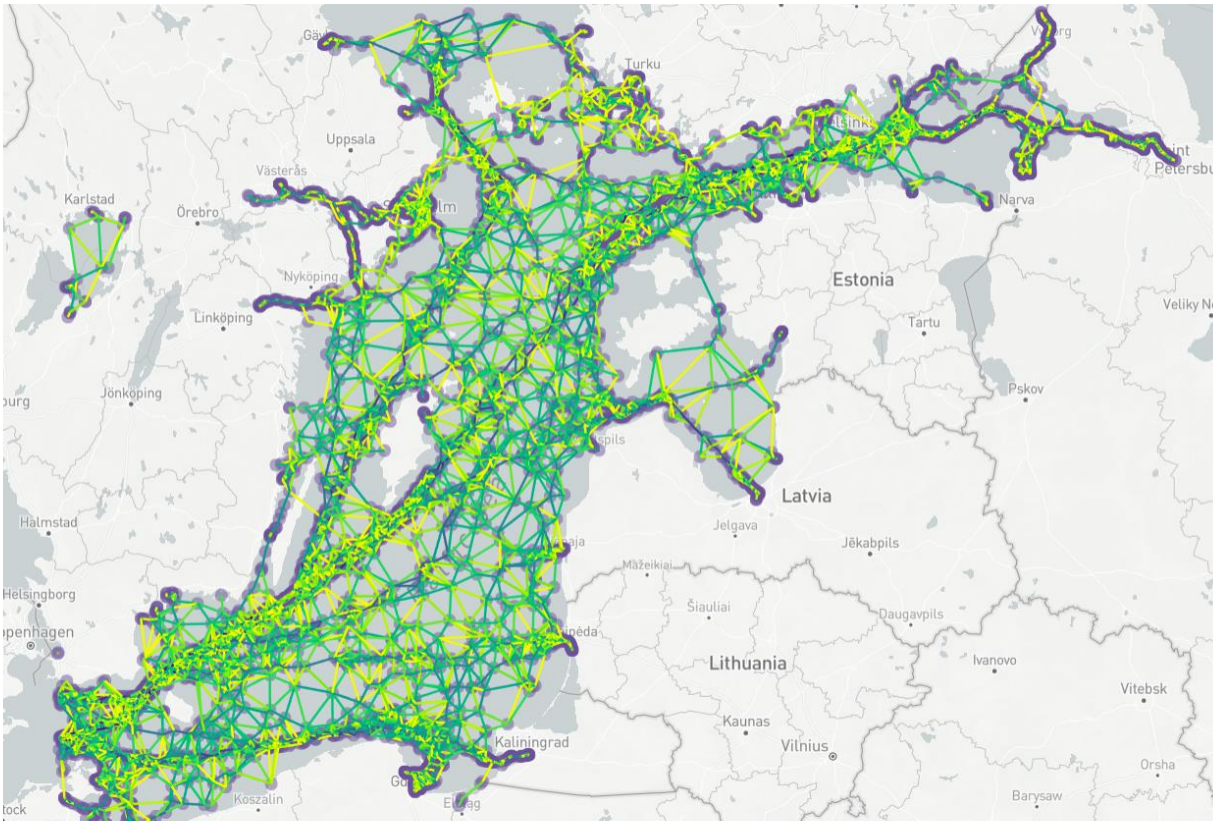


Figure 11. Sprint, South Baltic, type cargo, size M, 256 partitions

Different meshes are also obtained for different vessel types. So far, we have presented meshes for cargo vessels. Below are also meshes for tankers and passenger vessels. While tankers resemble the mesh for cargo, the patterns for passenger vessels are totally different. It was also harder to find edges in some area as there are not many passenger vessels going.

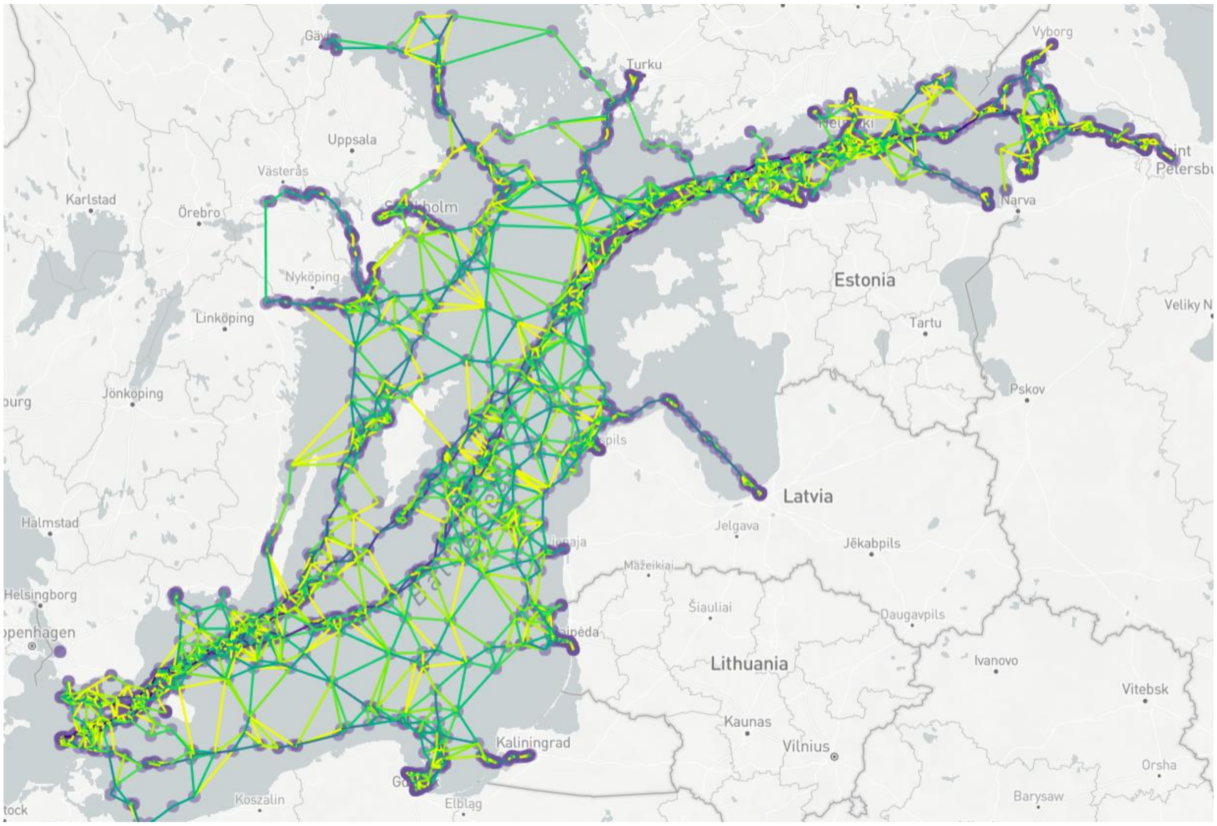


Figure 12. Sprint, South Baltic, 128 partitions, size M, vessel type tanker

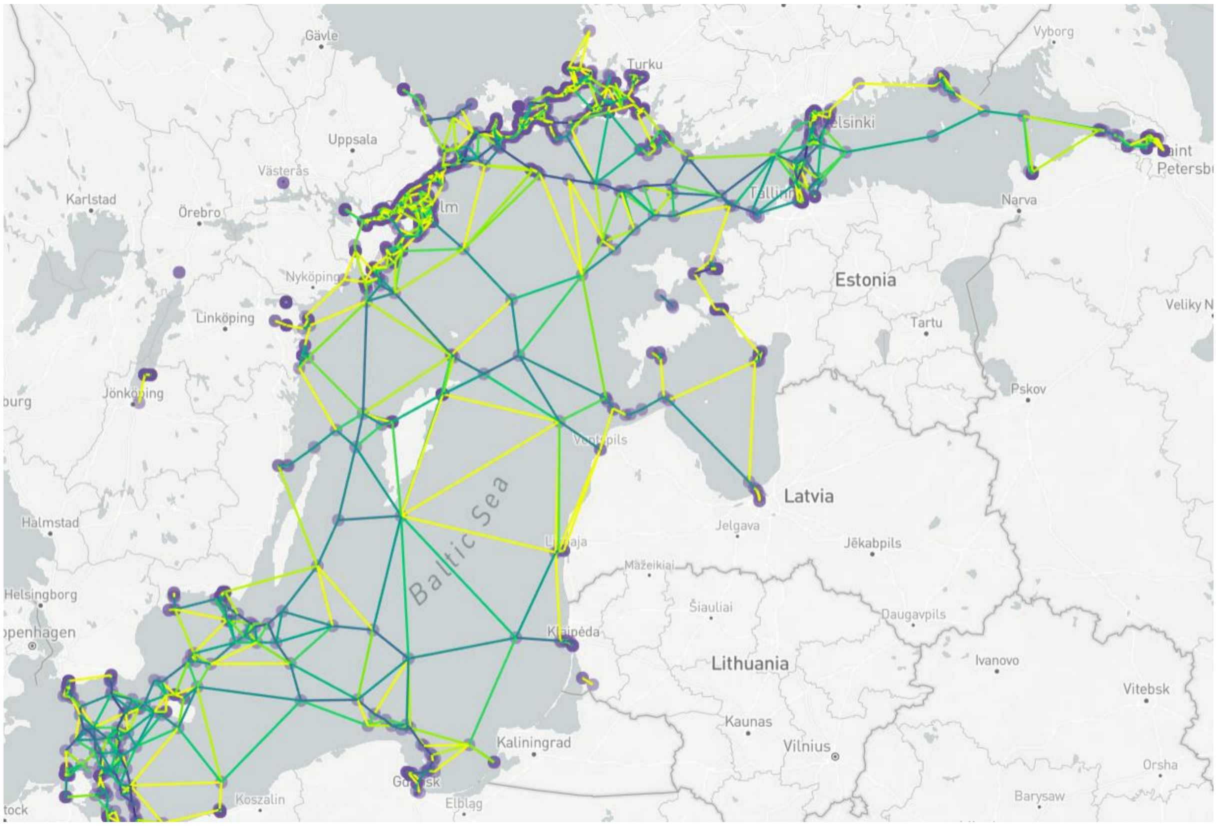
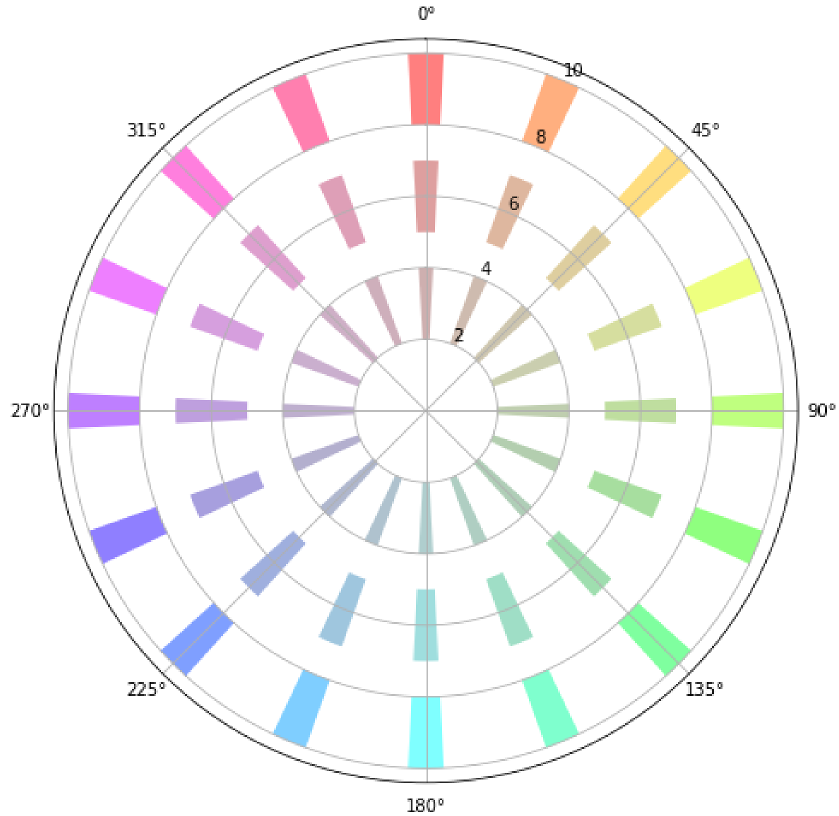


Figure 13. Sprint, South Baltic, 128 partitions, size M, vessel type passenger

4.4.2 Dominating Course and Directions of the Edges

The goal of this kind of visualization is to present the dominating course in waypoints (along with its strengths) and also direction of edges. This allows to observe certain patterns like traffic separation schemes. Edges are coded as follows: red – only northward, blue – only southward, violet – vessels travelling in both directions.

Coding of the direction of waypoints is presented in the following figure.



The following figures present meshes with the increasing number of waypoints. It is achieved by both number of partitions and number of genes in a partition. The difference between 64L and 256S is not big but it seems that bigger number of partitions is a little bit more precise in showing directions.

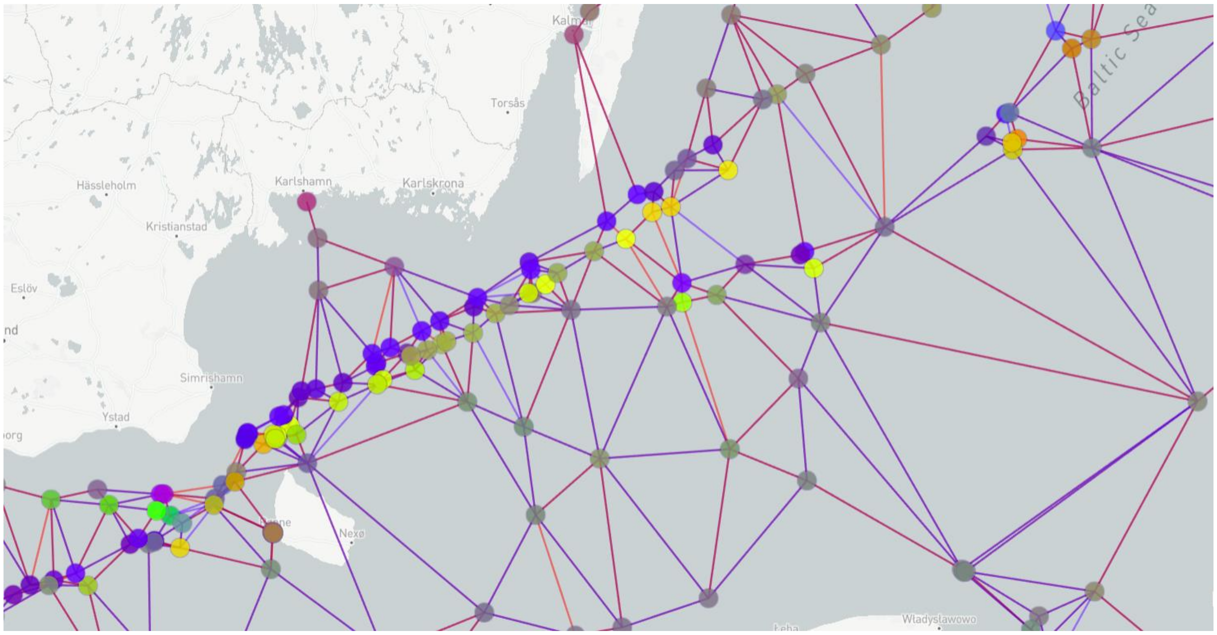


Figure 14. Directional mesh for Sprint, West Baltic, tankers, 64 partitions, size S

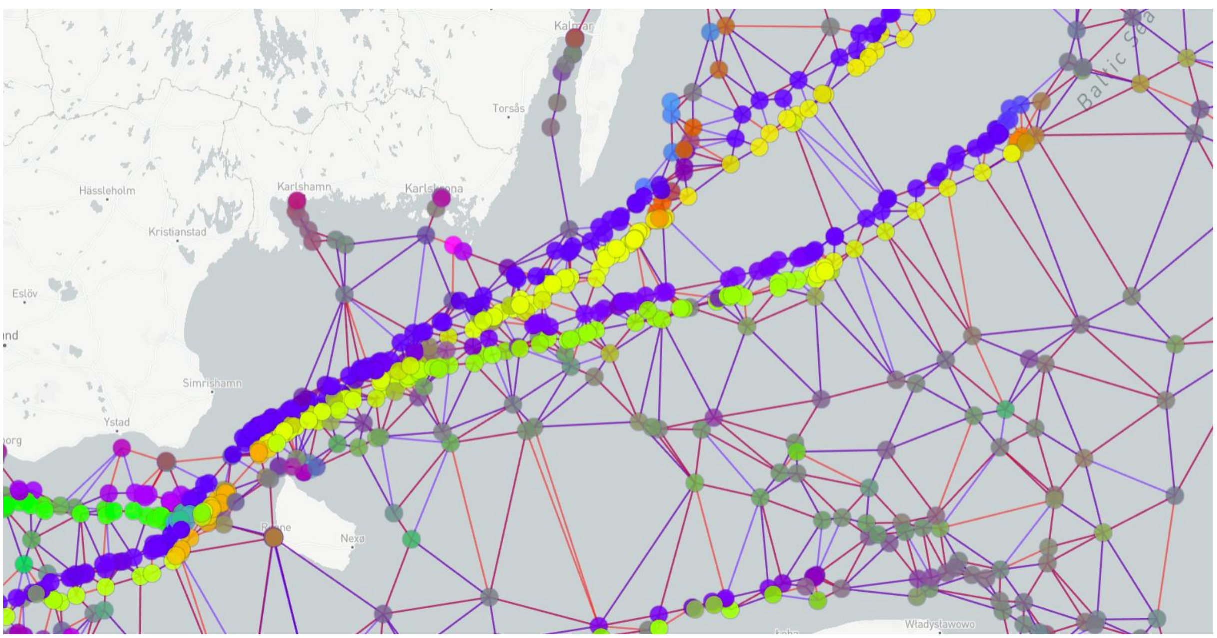


Figure 15. Directional mesh for Sprint, West Baltic, tankers, 64 partitions, size L

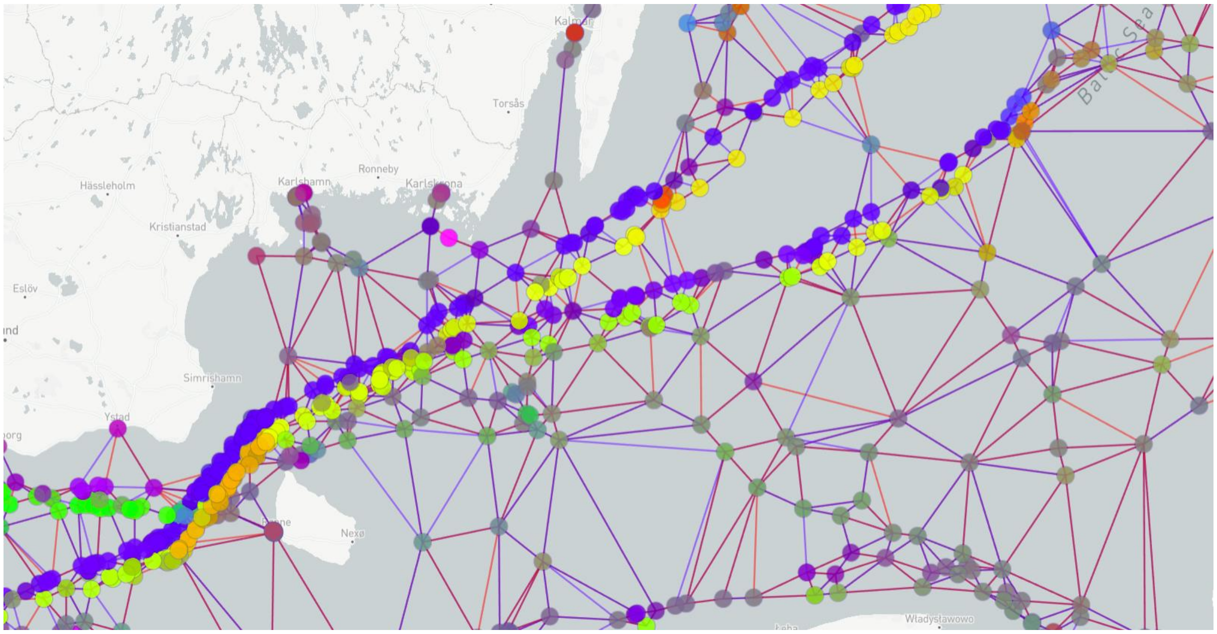


Figure 16. Directional mesh for Sprint, West Baltic, tankers, 256 partitions, size S

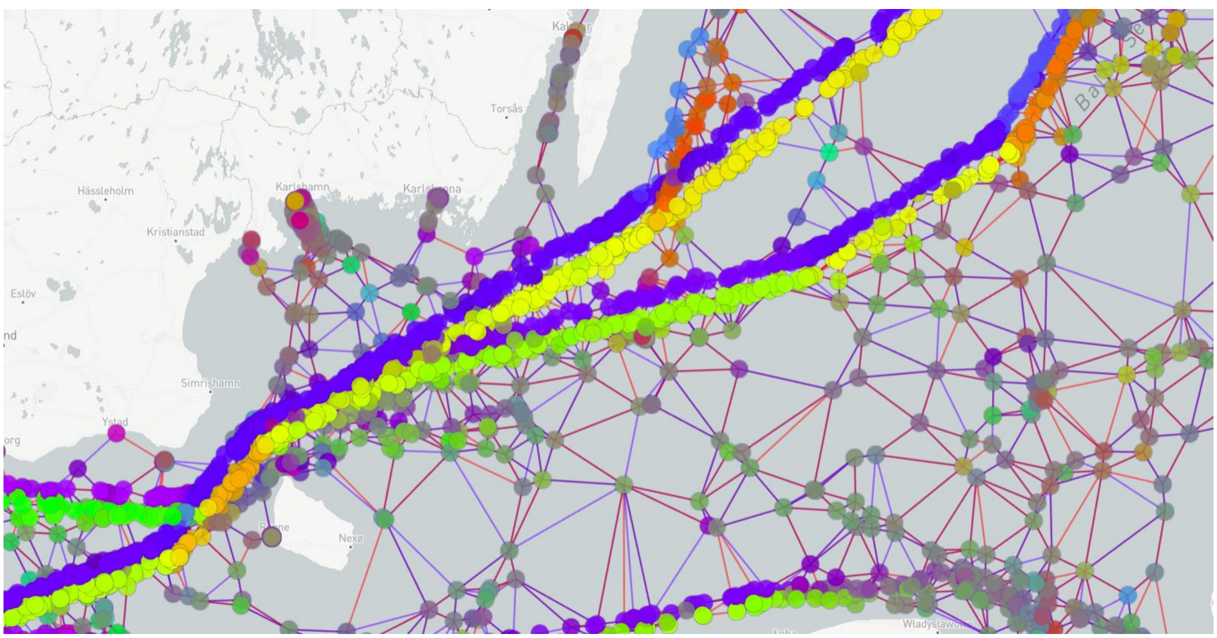


Figure 17. Directional mesh for Sprint, West Baltic, tankers, 256 partitions, size L

4.4.3 Width of Corridor

We have also prepared maps showing the width of corridors. The challenge for visualization is to include information about both left and right corridor and also the direction. This means that two waypoints can be connected with 4 lines. Different colors were assigned to these situations, but the overall picture is not very informative. The additional issue is that visualization tool does not allow to draw separately "left line" and "right line" in relation to the starting point.

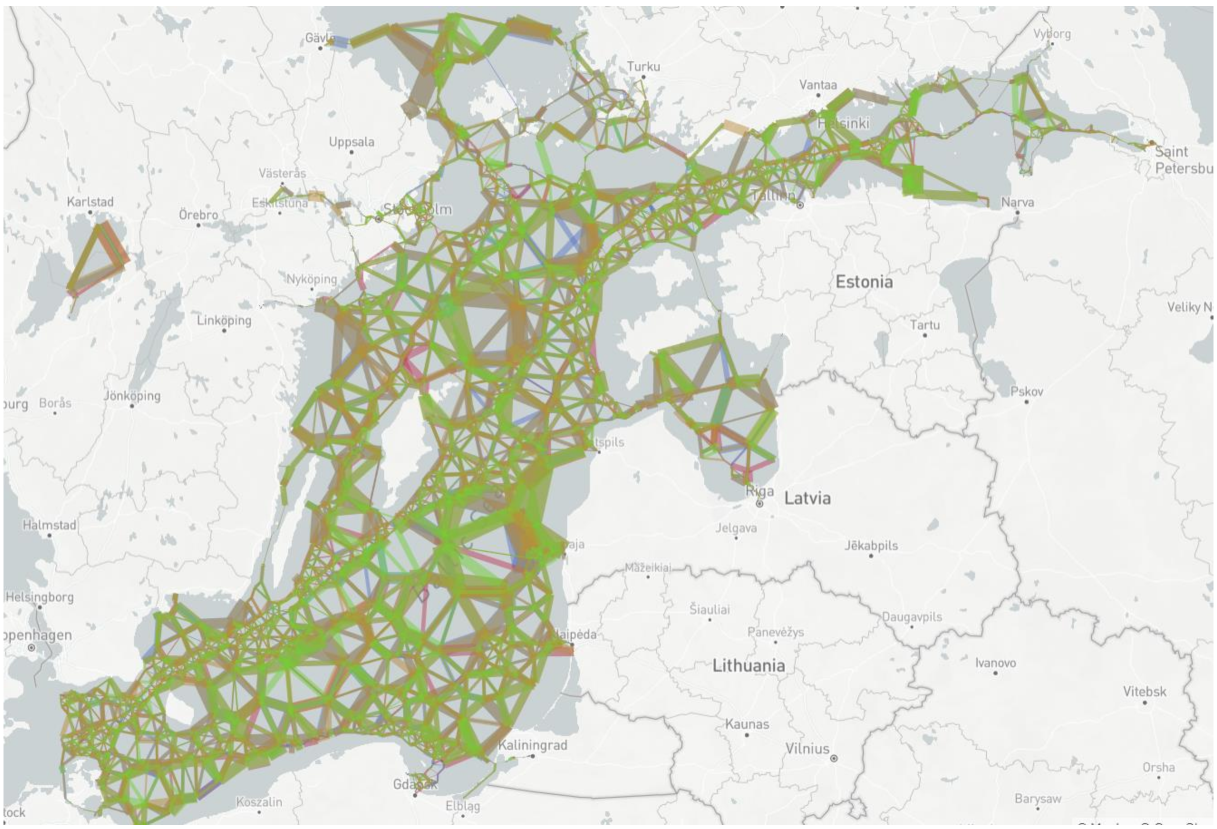


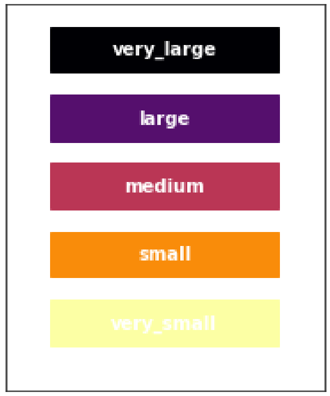
Figure 18. Corridors widths, Sprint, South Baltic, cargo, 128 partitions, size M

The corridor width is primarily used for anomaly detection and we give the visualization a supportive purpose. More information can be obtained in smaller areas.

Figure 19. Corridors widths, Sprint, Gulf of Gdańsk, cargo, 128 partitions, size M

4.5 Draught

Draught constitutes one of the variants for mesh generation. We have divided draught into 5 classes as described in section **Fehler! Verweisquelle konnte nicht gefunden werden.**. Each edge is annotated with i nformation what was the maximum draught of the vessel that travelled along given edge. The coding of colors is as follows.



The figure below shows the mesh colored with the draft value. The main routes for the biggest ships are visible in black.

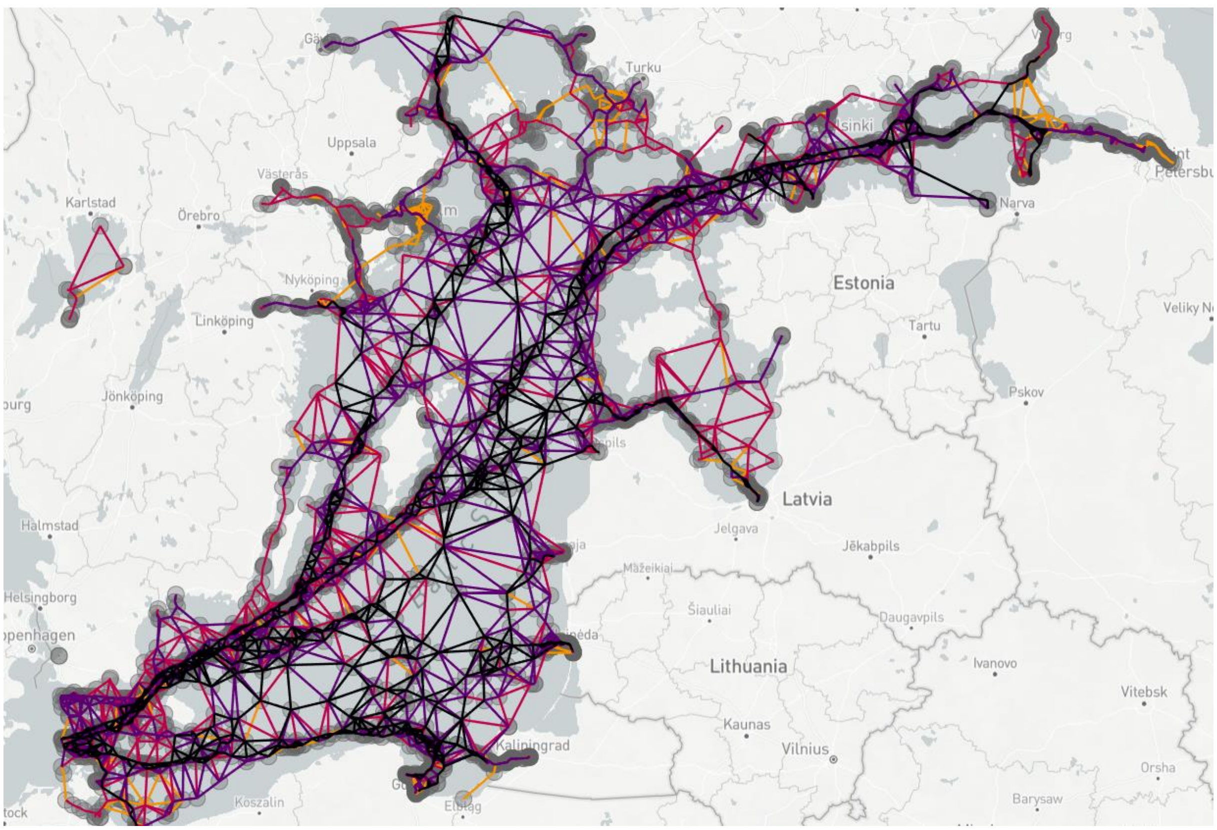


Figure 20. Mesh with draft classes, Sprint, South Baltic, cargo, 128 partitions, size M

Below we also show other visualization from a more complex area – the connection between the German Bight and the Baltic Sea. This time the mesh was generated from Navtor data.

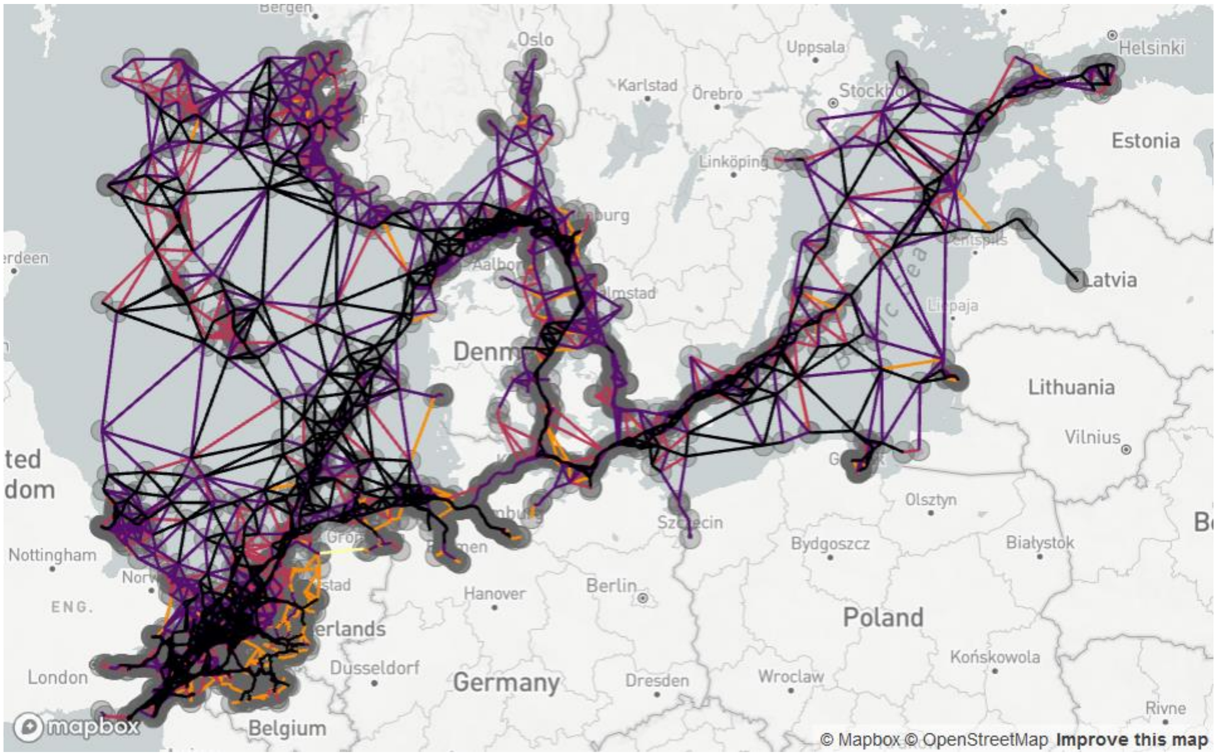


Figure 21. Mesh with draft classes, Navtor, German Bight and South Baltic, cargo, 128 partitions, size M

4.6 Length

Length constitutes one of the variants for mesh generation. We have divided length into 5 classes as described in section **Fehler! Verweisquelle konnte nicht gefunden werden.**. Each edge is annotated with i nformation what was the maximum length of the vessel that travelled along given edge. The coding of colors is as follows.

The figure below shows the mesh colored with the length value. The main routes for the biggest ships are visible in dark violet.

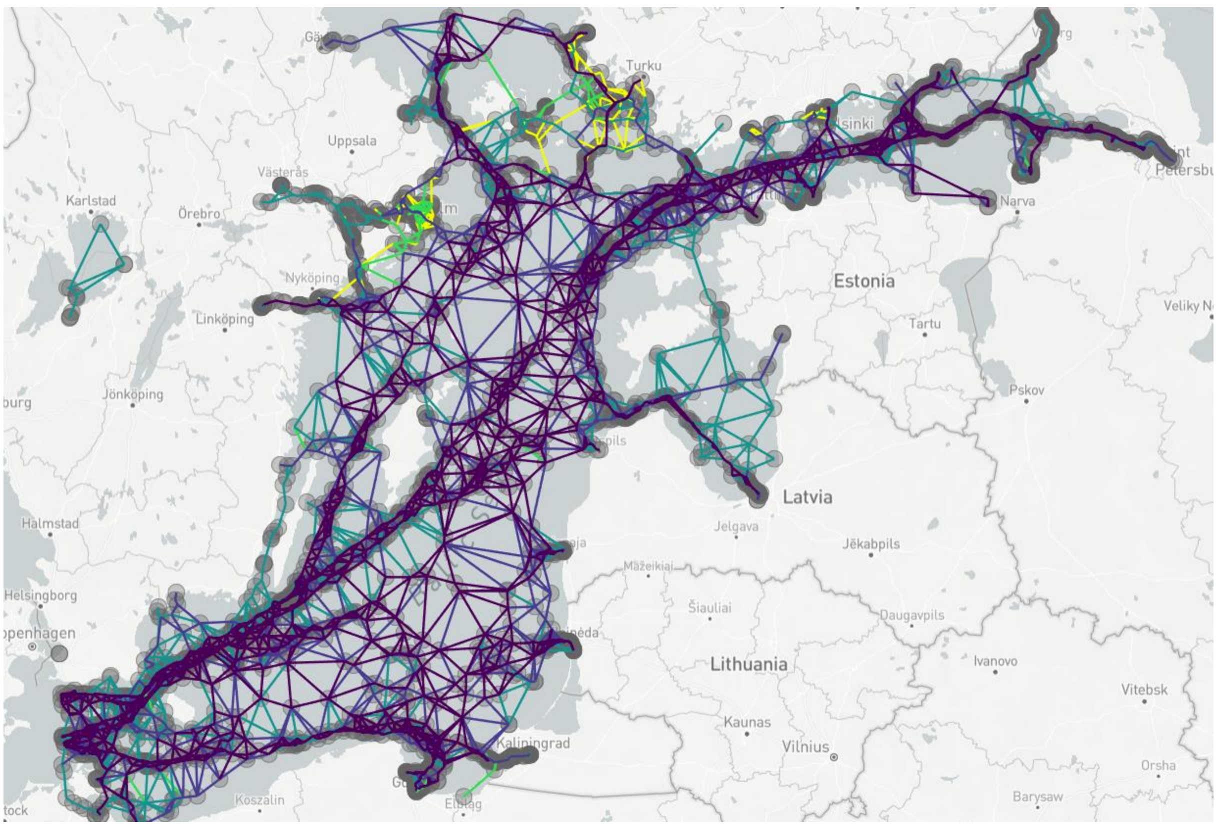


Figure 22. Mesh with length classes, Sprint, South Baltic, cargo, 128 partitions, size M

4.7 Width

Width is the last variant for mesh generation. We have divided width into 5 classes as described in section **Fehler! Verweisquelle konnte nicht gefunden werden.**. Each edge is annotated with information what w as the maximum width of the vessel that travelled along given edge. The coding of colors is as follows.

The figure below shows the mesh colored with the width value. The main routes for the biggest ships are visible in dark violet. It is noteworthy that unlike the draught and length, the width best shows the main routes followed by the biggest ships.

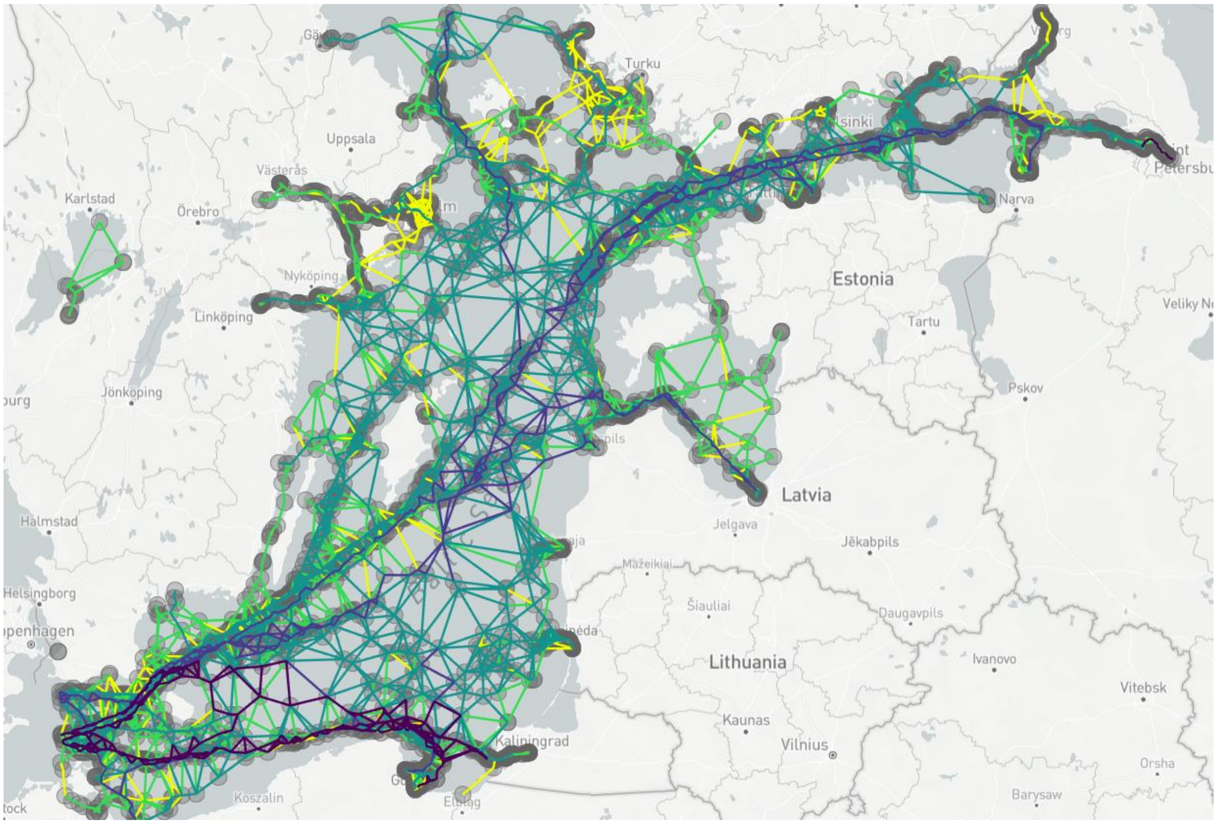


Figure 23. Mesh with length classes, Sprint, South Baltic, cargo, 128 partitions, size M

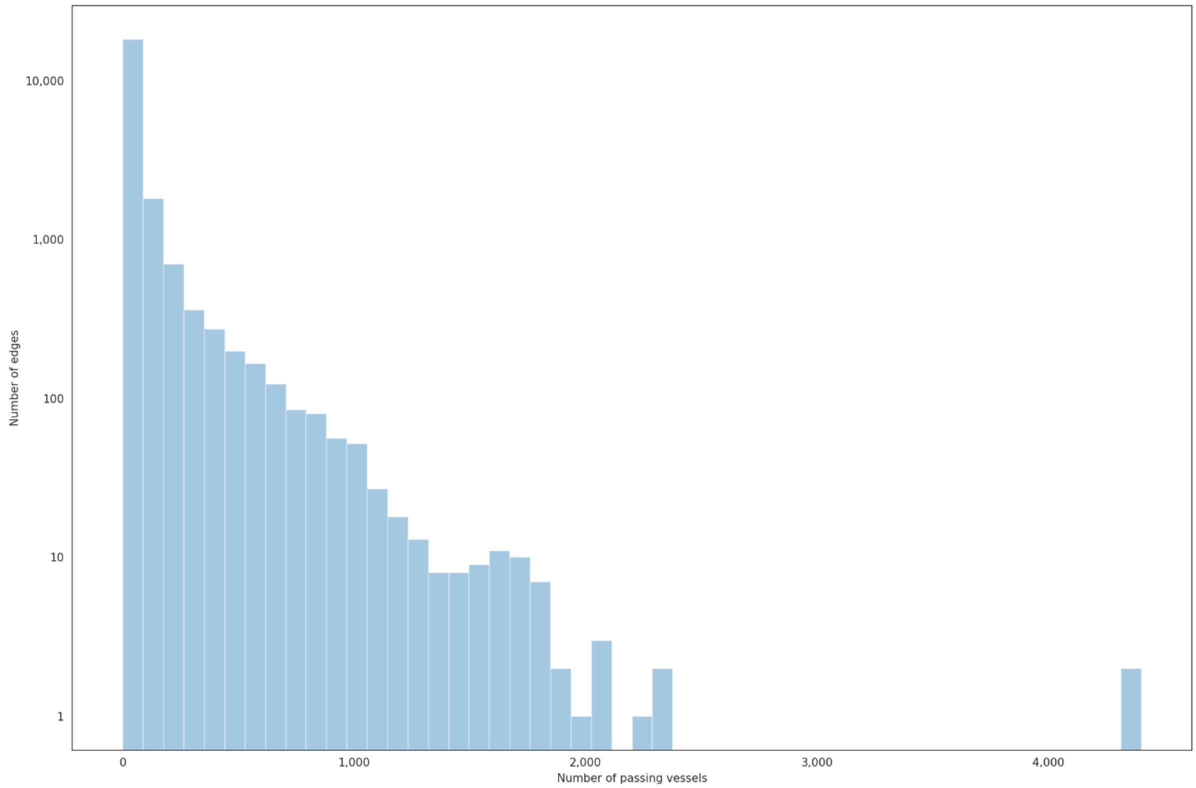
4.8 Histograms

Several histograms (distplots) were prepared to help evaluate the generated mesh. They are gathered below. The idea of histograms is as follows: the X-axis shows the aggregated attribute we would like to characterize; the Y-axis shows number of nodes or edges that have a certain value of the attribute. The Y-scale is in majority of cases logarithmic, as the phenomena we visualize are better characterized by exponential growth.

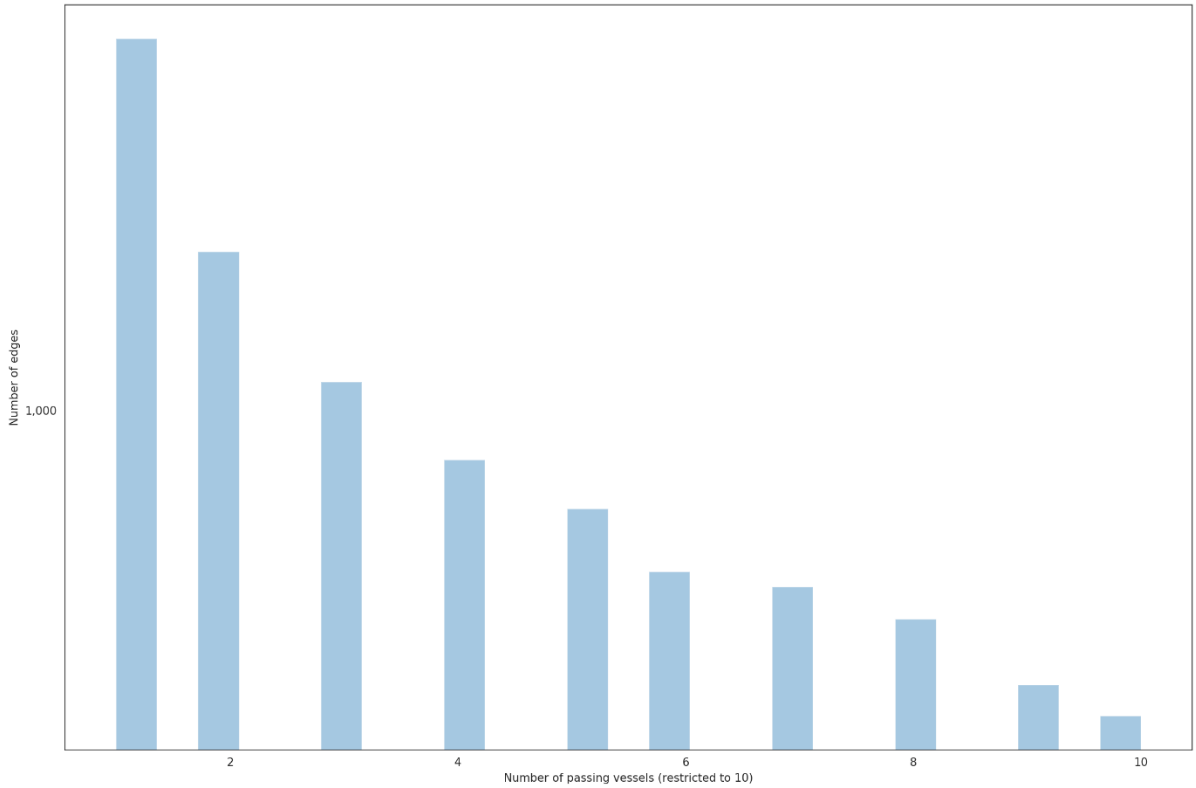
All histograms in this section are prepared for: Sprint, South Baltic, cargo, 128 partitions, size M.

4.8.1 Distribution of cnt

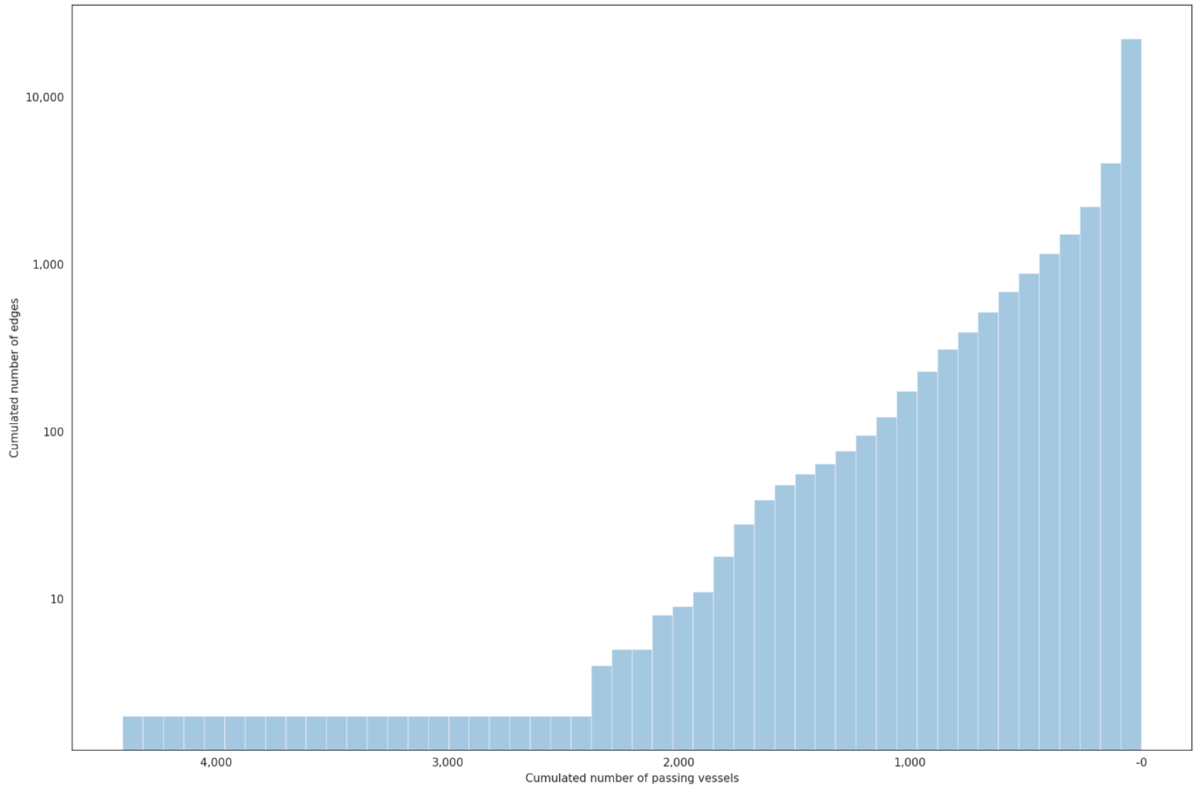
There are many edges followed just by few ships and few edges followed by many ships. This phenomena is explained by the plot below.



There are just 3 edges that have ‘cnt’ value bigger than 4000. At the same time there are over 10,000 edges that have cnt close to zero. In order to better characterize the numbers, we can visualize only bars with count<=10.



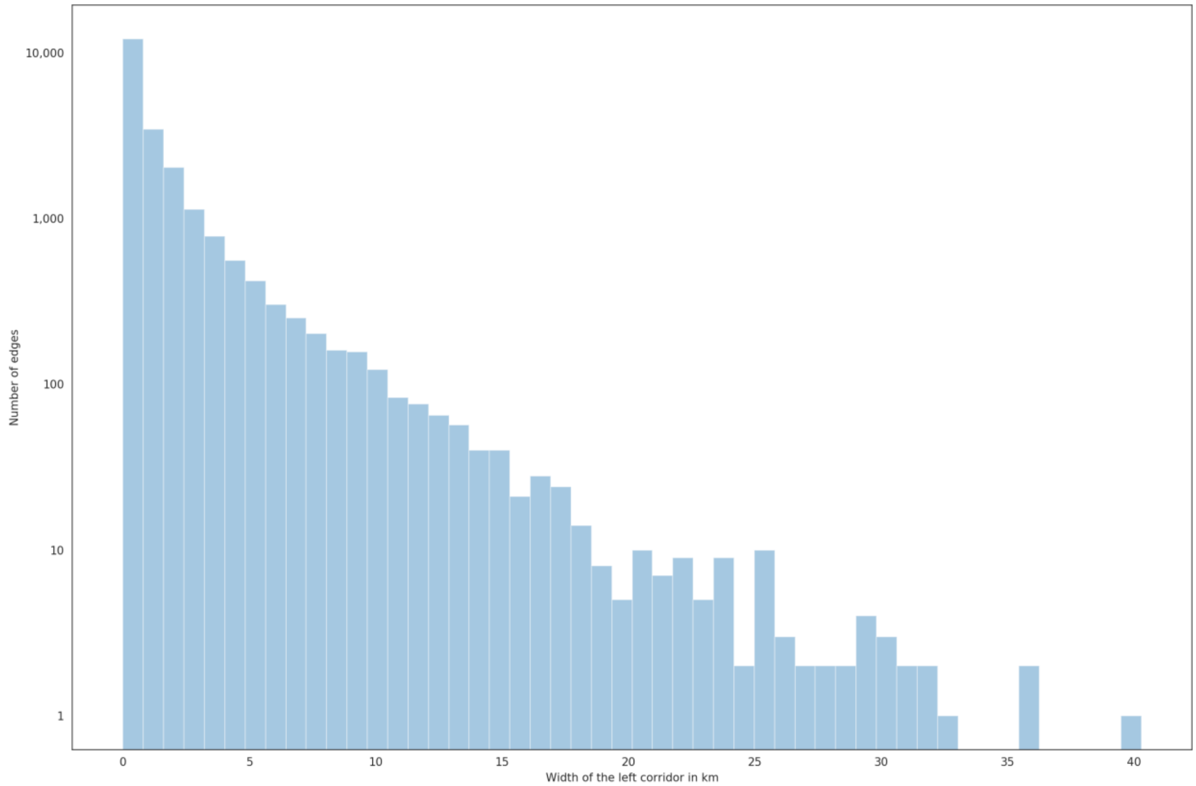
When constructing the mesh, we would like to focus on the most popular edges. The lower the cnt number, the higher the probability that the connection is accidental. By decreasing the minimum cnt value we also increase the mesh size. The growth can be better characterized in the chart below, when we show the cumulated count.

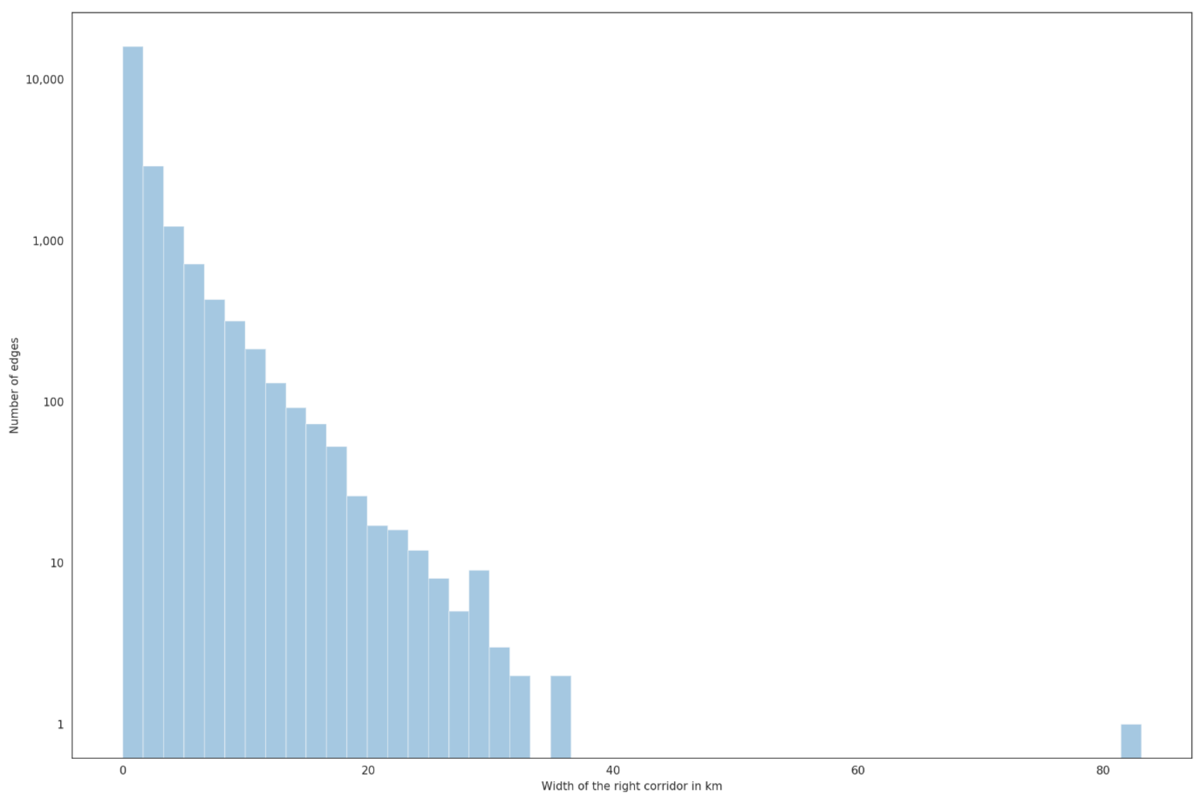


If we filtered just at cnt>500, we would get no more than 1000 edges in the mesh.

4.8.2 Corridor Width

For each edge in the mesh we provide the left and right corridor. The charts below show the distribution of the width of the corridors.

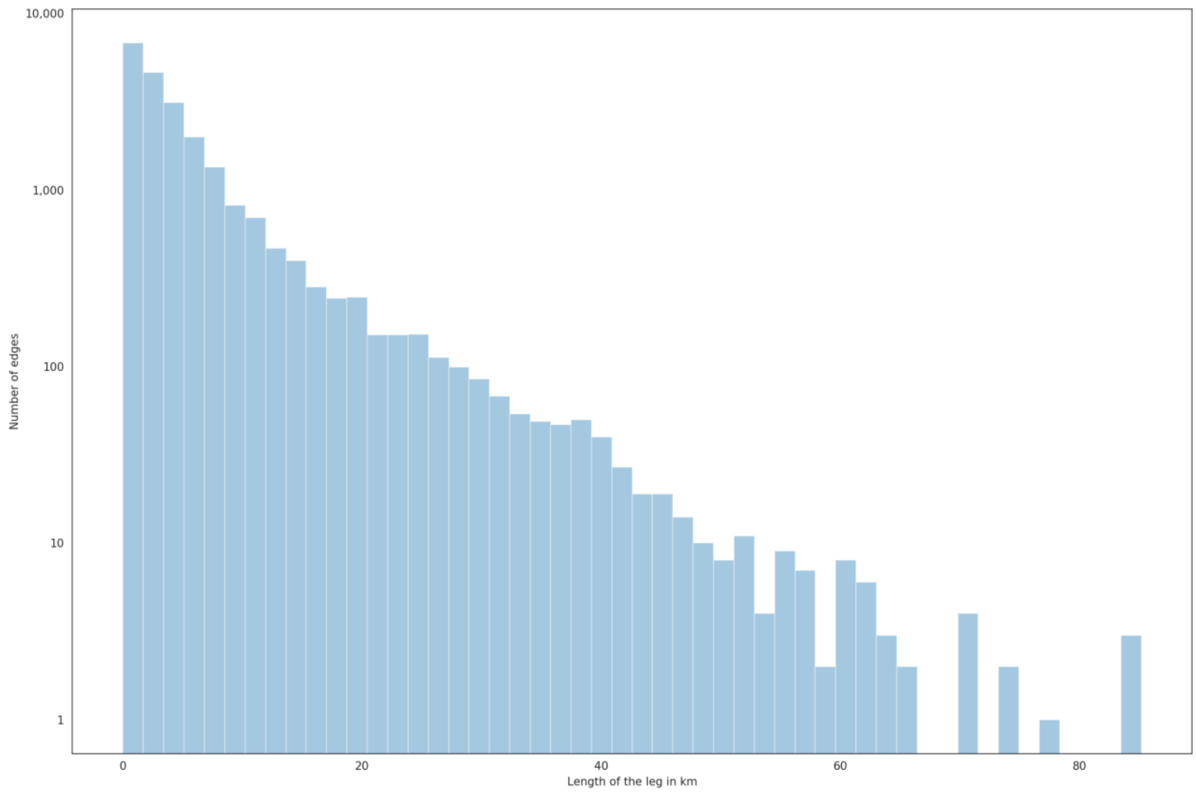

10,000
1,000
100
10
1
Number of edges
0
5
10
15
20
25
30
35
40
Width of the left corridor in km


10,000
1,000
100
10
1
Number of edges
0
20
40
60
80
Width of the right corridor in km

As we can see in the case of a right corridor, there is one outlier where the width is over 80 km. In majority of cases it is ca. 15 km. The width was set in a way to encompass 90% of the traffic.

4.8.3 Length of Legs

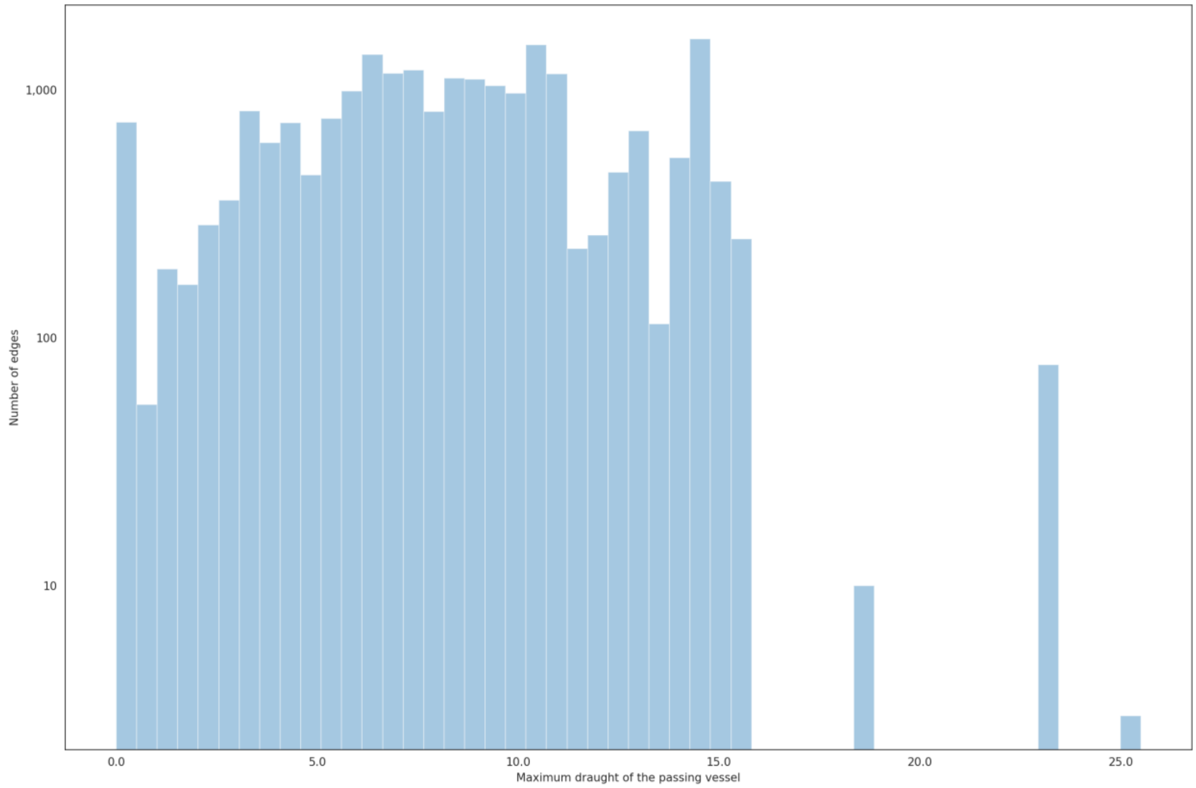
This chart shows what is the distribution of length of the legs.

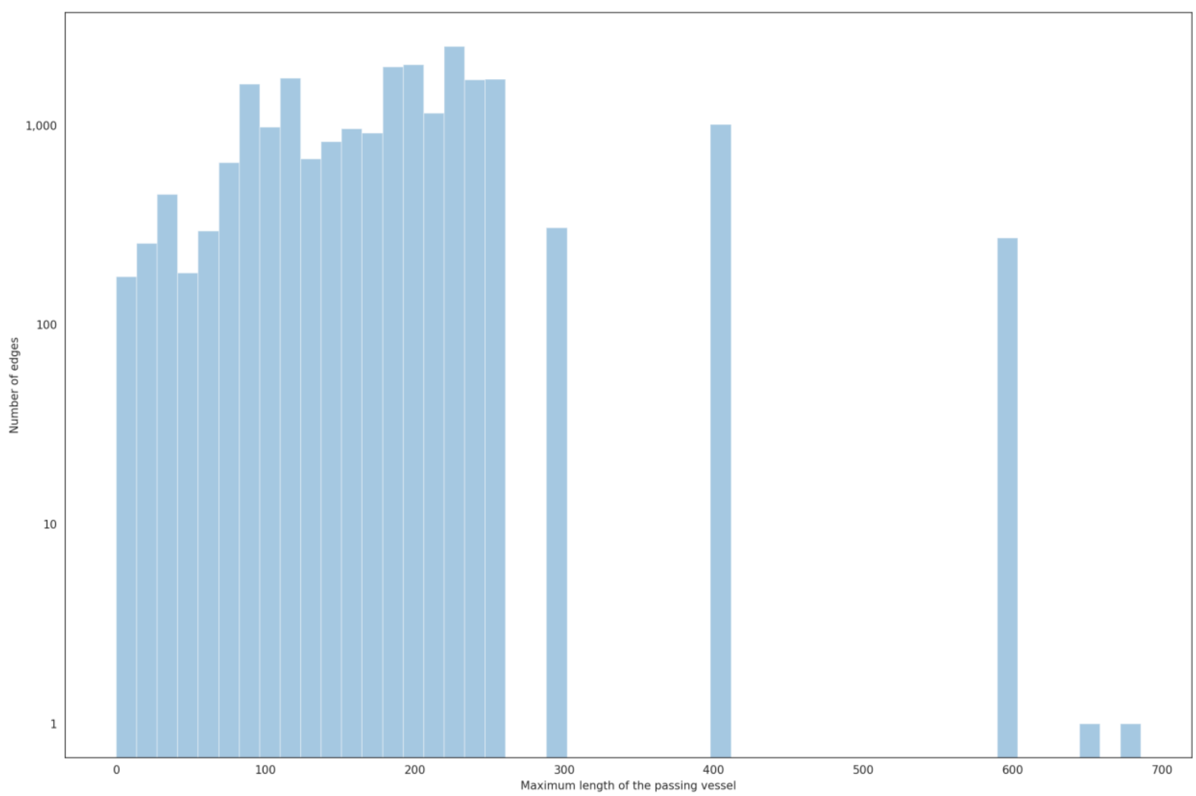


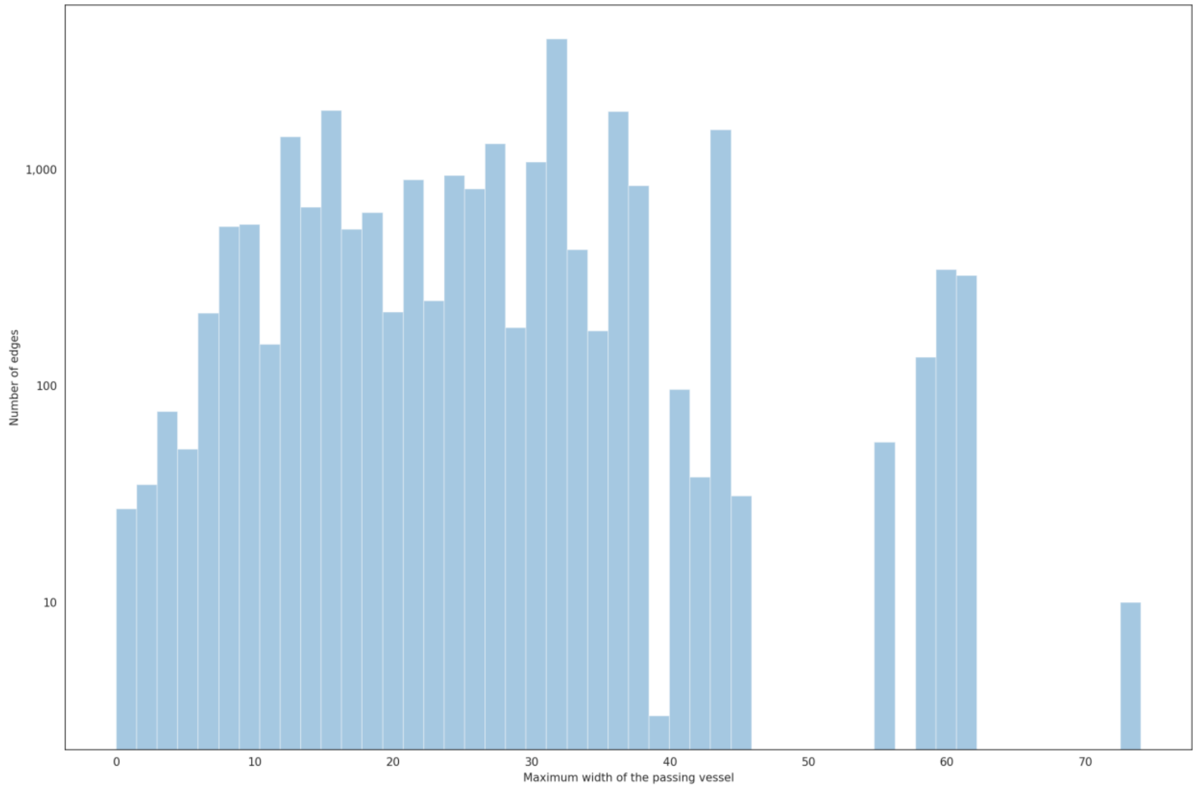
Short distances between waypoints are necessary to model complex trajectories, e.g. near a port, on the river. However, too many short legs can also mean that the mesh is too complex. However, simplification cannot be conducted just based on the distance.

4.8.4 Maximum Draught

The three charts below characterize the distribution of edges with regard to attributes characterizing the size the vessels.

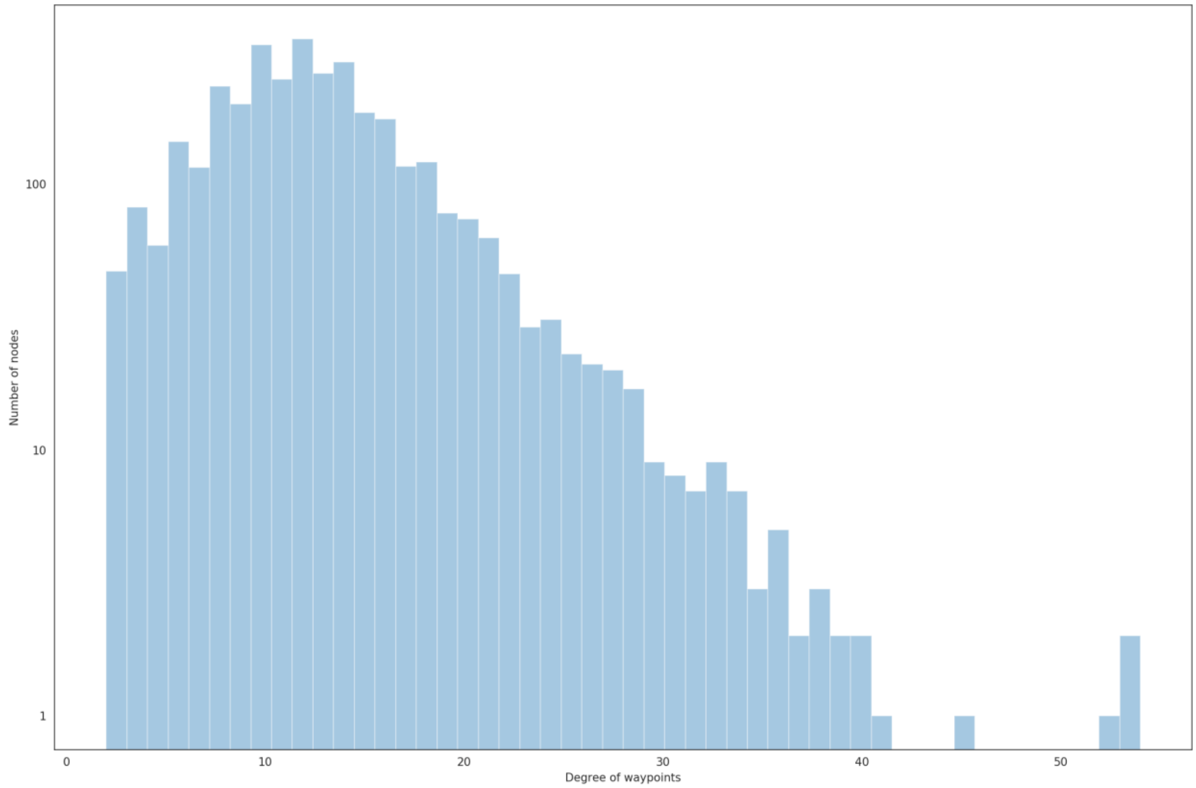

1,000
100
10
Number of edges
0.0
5.0
10.0
15.0
20.0
25.0
Maximum draught of the passing vessel


1,000
100
10
1
Number of edges
0
100
200
300
400
500
600
700
Maximum length of the passing vessel



4.8.5 Degree

Degree characterizes how many connections a given node has. In an ideal mesh, there should not be too many connections. As we can see from the figure below, there are some waypoints that are connected to over 50 other waypoints. This is a potential for simplification.

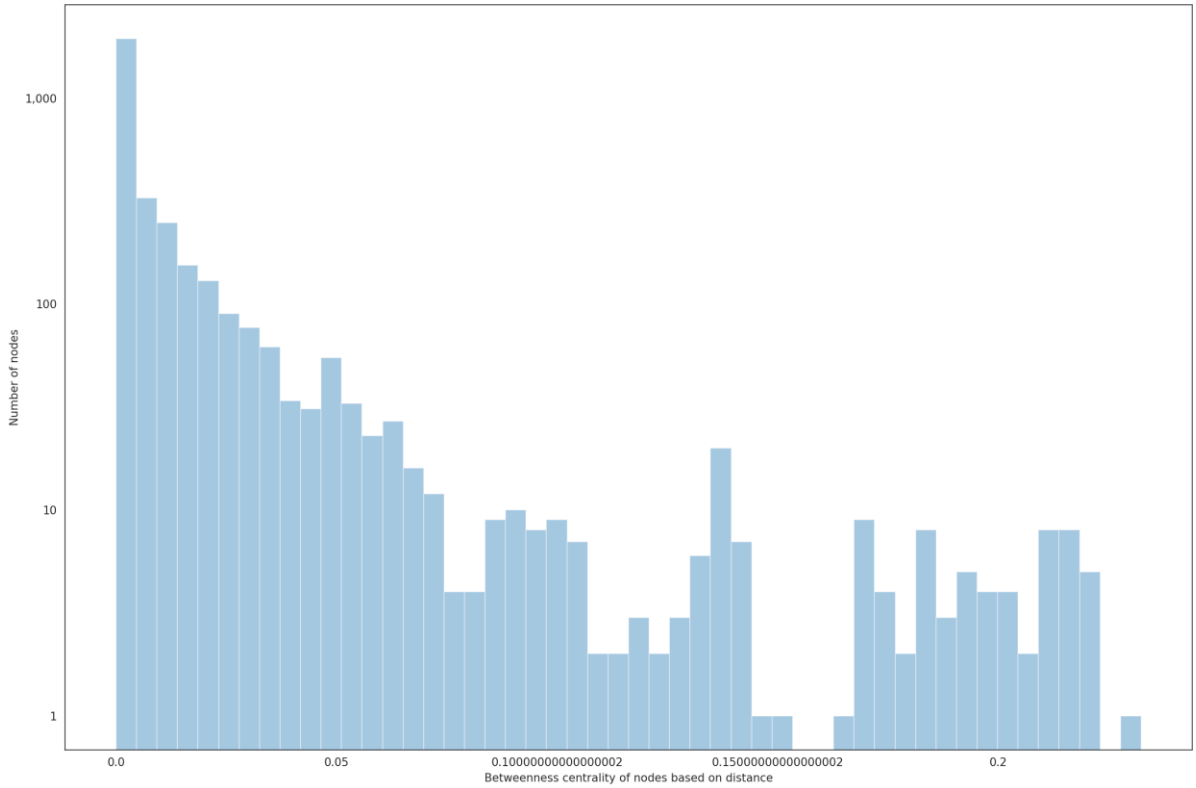


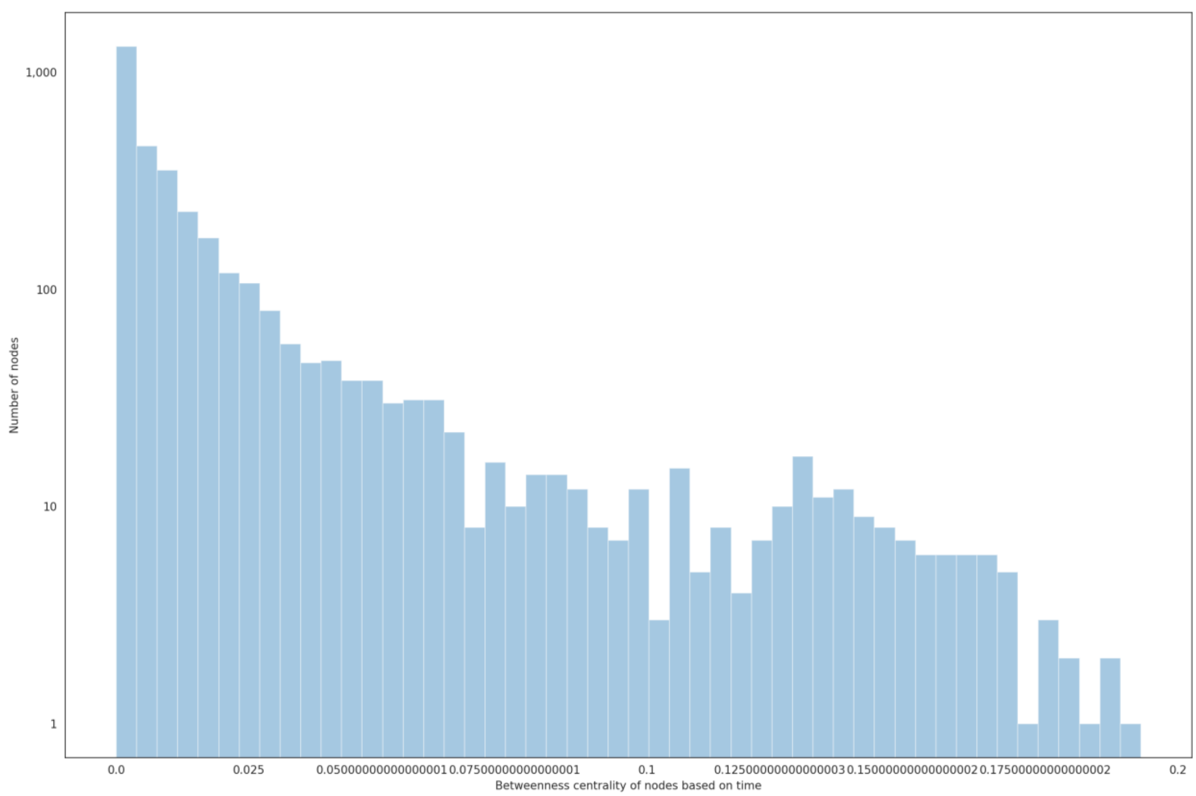
4.8.6 Centrality Measure

For simplification of the mesh it is important to know which waypoints are necessary and which are not. The main goal of the construction of the mesh is planning of a route, the shortest or the fastest, taking into account additional conditions.

Betweenness centrality of a node *v* is the sum of the fraction of all-pairs shortest paths that pass through *v*. If for a given node or edge the value of centrality equals zero, it means that such a node will never be returned as a result of a recommended corridor planning. Effectively, such a node/edge can be removed from the edge.

In order to assess the potential of the simplification of the mesh, we have prepared the distribution of betweenness centrality, showed in the following figures.


1,000
100
10
1
Number of nodes
0.0
0.05
0.10000000000000002
0.15000000000000002
0.2
Betweenness centrality of nodes based on distance


1,000
100
10
1
Number of nodes
0.0
0.025
0.050000000000000001
0.0750000000000000001
0.1
0.12500000000000003
0.15000000000000002
0.17500000000000002
0.2
Betweenness centrality of nodes based on time

4.9 Mesh Characteristics

The following characteristics are used:

- NumNodes – number of nodes/waypoints in the mesh (dataframe),
- NumNodesG – number of nodes/waypoints in the mesh (graph, for control purposes),
- NumEdges – number of edges in the mesh (dataframe),
- NumEdgesG – number of edges in the mesh (graph, for control purposes),
- NumEdgesCnt1 – number of edges with attribute cnt=1 (the potential for reduction of accidental connections),
- NumNodesBtwnDist0 – number of nodes with betweenness centrality equals 0, i.e. number of nodes not belonging to any shortest path between any other pair of nodes,
- NumNodesBtwnTime0 – number of nodes with betweenness centrality equals 0, i.e. number of nodes not belonging to any fastest path between any other pair of nodes,
- NumEdgesBtwnDist0 – number of edges with betweenness centrality equals 0, i.e. number of edges not belonging to any shortest path between any other pair of nodes,
- NumEdgesBtwnTime0 – number of edges with betweenness centrality equals 0, i.e. number of edges not belonging to any fastest path between any other pair of nodes,
- EdgesMaxCnt – maximum value of cnt attribute,
- MaxDegree – maximum number of connections of a single node,
- EdgesMaxCorridorLeftKm – maximum width of the left corridor (portboard) in kilometers,
- EdgesMaxCorridorRightKm – maximum width of the right corridor (starboard) in kilometers,
- EdgesMaxDistanceKm – maximum distance between waypoints connected with an edge,
- Density – number of edges to the total possible number of edges,
- IsStronglyConnected – if it is possible to go from any waypoint to any other waypoint, i.e. if all waypoints are connected,

- NumStronglyConnectedComponents – number of separate subgraphs that are strongly connected,
- IsWeaklyConnected – if it is possible to go from any waypoint to any other waypoint regardless of the edge direction,
- NumWeaklyConnectedComponents – number of separate subgraphs that are weakly connected.

Sample characteristics for mesh ‘Sprint, South Baltic, cargo, 128 partitions, size M, all variants’ is as follows:

NumNodes=3,435

NumNodesG=3,435

NumEdges=22,322

NumEdgesG=22,322

NumEdgesCnt1=3377

NumNodesBtwnDist0=382

NumNodesBtwnTime0=127

NumEdgesBtwnDist0=5576

NumEdgesBtwnTime0=2465

EdgesMaxCnt=4402

MaxDegree=54

EdgesMaxCorridorLeftKm=40.29352

EdgesMaxCorridorRightKm=83.08646

EdgesMaxDistanceKm=85.17483

Density=0.0018923700744079032

IsStronglyConnected=False

NumStronglyConnectedComponents=3

IsWeaklyConnected=False

NumWeaklyConnectedComponents=3

5 Algorithm for the combination of corridors

5.1 The principle of the algorithm

A basic problem when using the mesh data determined by PUEB for the purpose of traffic monitoring is that the mesh data consists of a very large number of mutually overlapping segments. However, IN's evaluation algorithms are optimized for larger, contiguous, elongated corridor areas that are shaped like a typical shipping route with a lateral catchment area.

The basic principle of traffic monitoring is to recognize when a ship leaves the area of the safe corridor valid for this ship class. It must therefore be possible to clearly evaluate for a certain class of ship and a predominant direction of travel whether the ship is inside or outside the safe corridor. Mutually overlapping mesh segments pose a problem for processing because leaving a mesh segment does not automatically mean that the ship has also left the safe corridor. Rather, it could simply have just switched to another adjacent or overlapping mesh segment.

As the results from the data processing from PUEB deliver too many and furthermore mutually overlapping corridors, IN had to implement a means of reducing the number of structures for processing and unify overlapping areas. This conversion algorithm is for demonstration purposes based on the PUEB results and is not intended for commercial use.

To simplify the results further IN had to convert the result into exclusion zones rather than corridors. The result from this algorithm are warning areas that display a warning when ships enter or are within the area. Therefore, it would not be possible to detect when a vessel is travelling in a wrong direction, too slow or too fast.

After receiving the results from PUEB they are interpreted as middle lines with parallel borders on each side. The principle is depicted in Figure 24 where the middle lines are depicted in black as solid line and the borders on each side as dotted line. The legs of these corridors are converted into rectangles as shown in Figure 25.

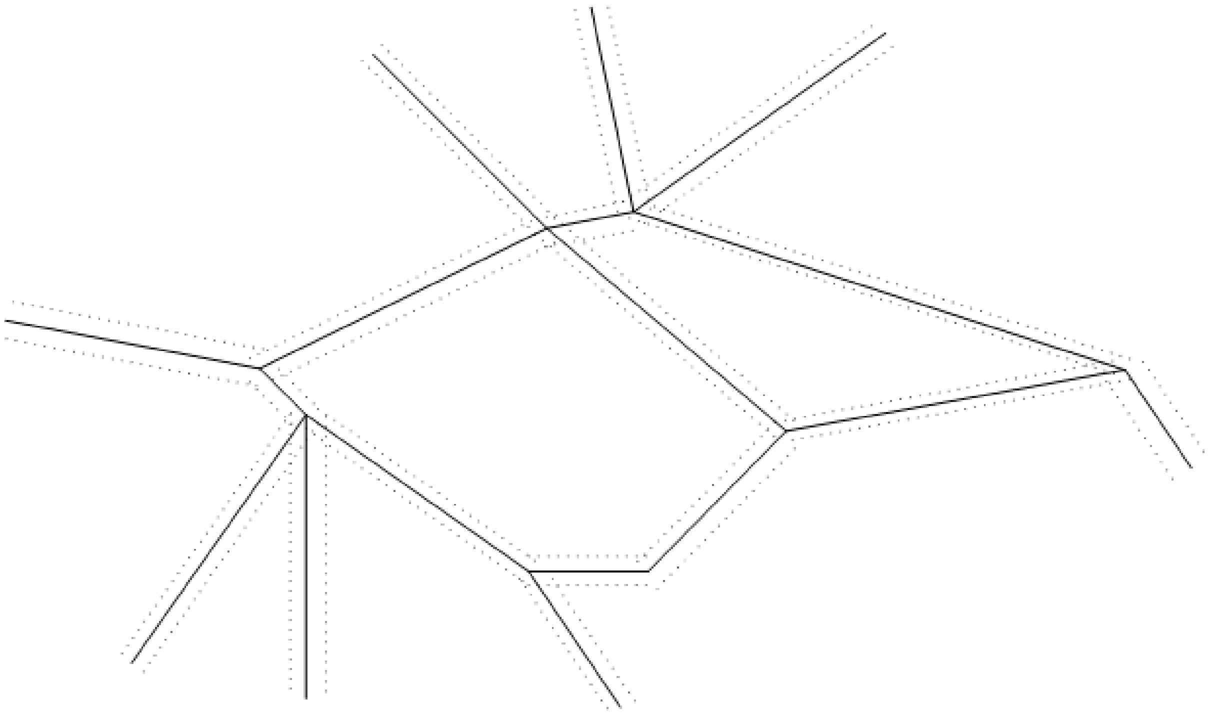


Figure 24: Mesh as received from PUEB

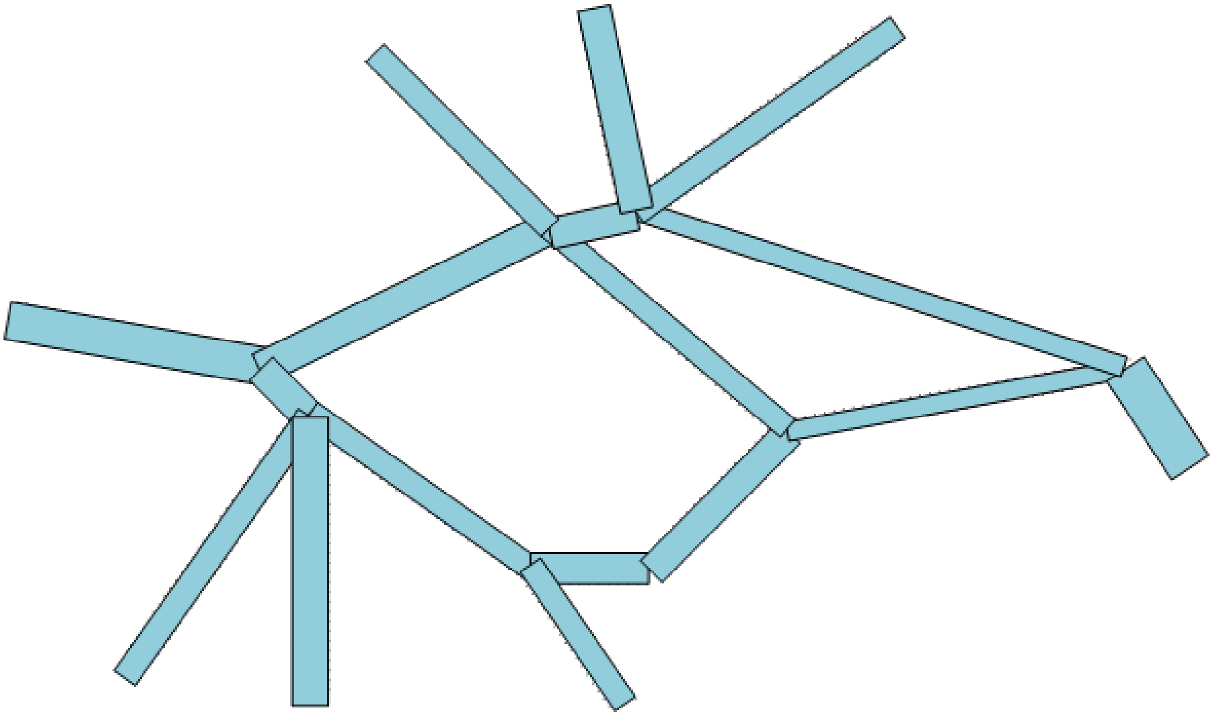


Figure 25: Mesh legs converted into rectangles

The rectangular areas are joined as one graph with “inner” and “outer” borders. For the joining operation and the follow-on operations, the GPC library was used. For the joining operation the Union function was used. Figure 26 depicts the result of this operation.

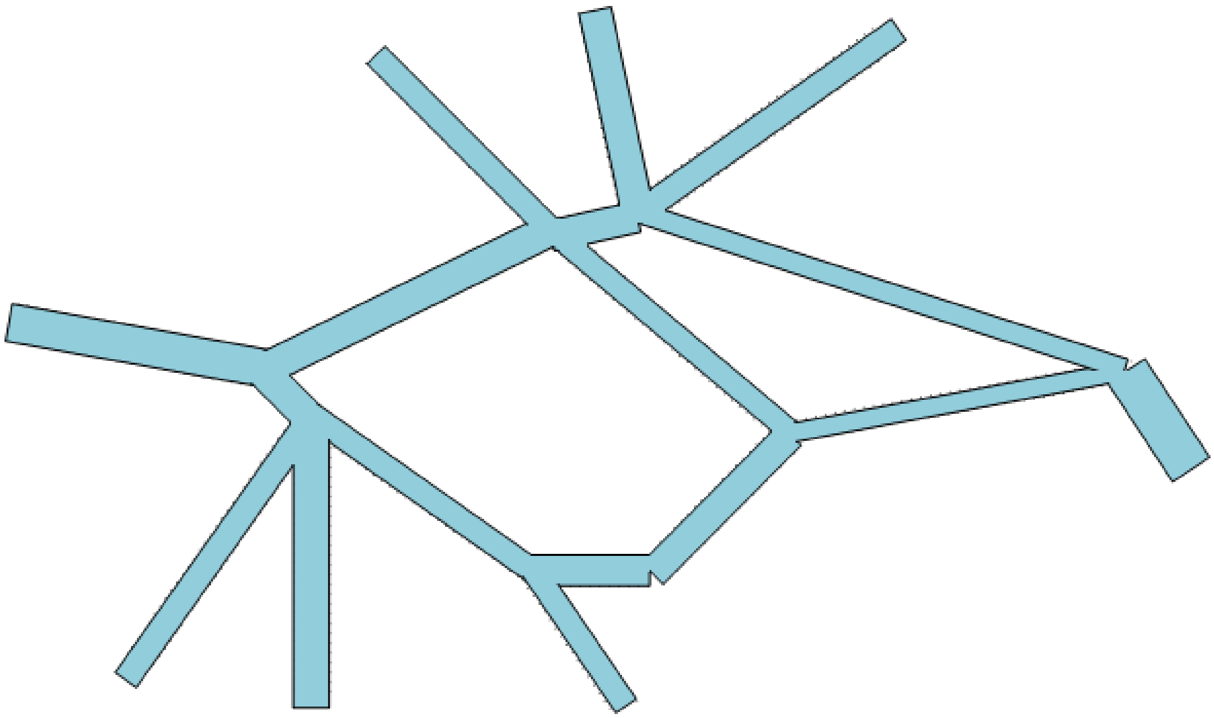


Figure 26: Mesh converted into a unified area

Once the rectangles were joined the borders of the inner exclusion zones were calculated. These are depicted in Figure 27. Together with the outer border of the graph this area defines at least 4 outer restriction areas as depicted in light red in Figure 28. To calculate the borders of the outer exclusion zones a “rectangular area” around the graph is constructed as follows:

1. Find the most western point of the graph and store the longitudinal coordinate as LON_w.The longitudinal coordinate of the most eastern point of the graph is stored equally as LON_e.
2. Find the most northern point of the graph and store the longitudinal coordinate as LAT_n.The latitudinal coordinate of the most southern point of the graph is stored equally as LAT_s.
3. The area around the graph is then defined with the following corners:
    - LAT_n , LON_e
    - LAT_s, LON_e
    - LAT_s, LON_w

- LAT_n, LON_w

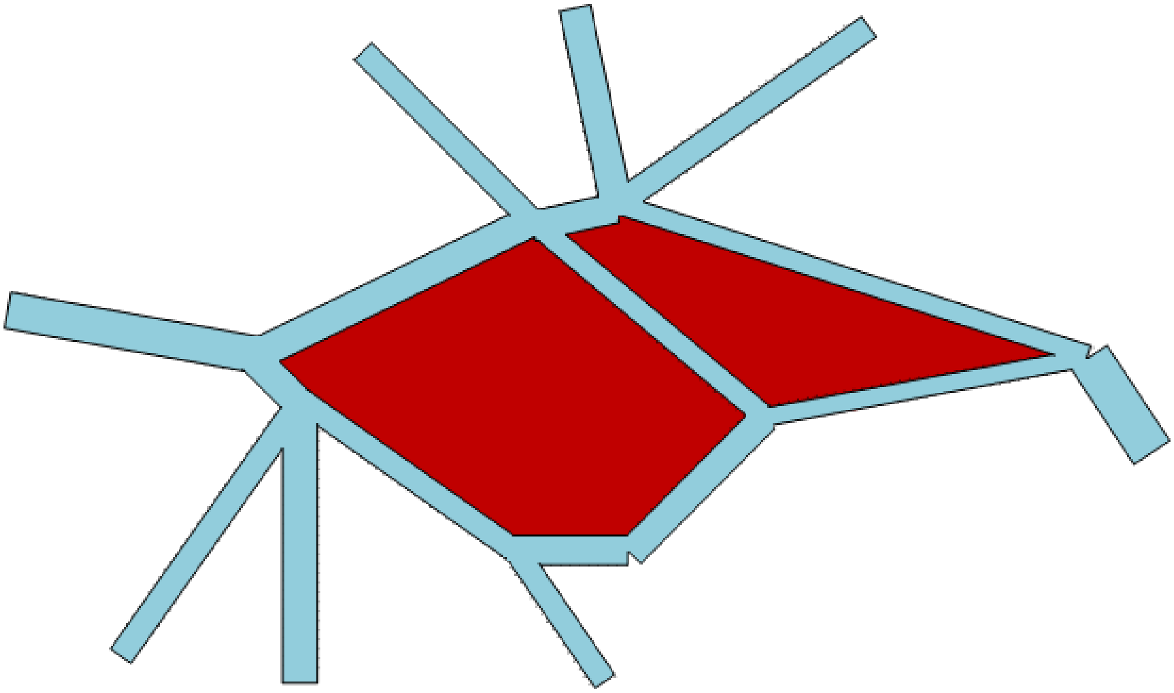


Figure 27: Inverted unified mesh area with exclusive zones (red)

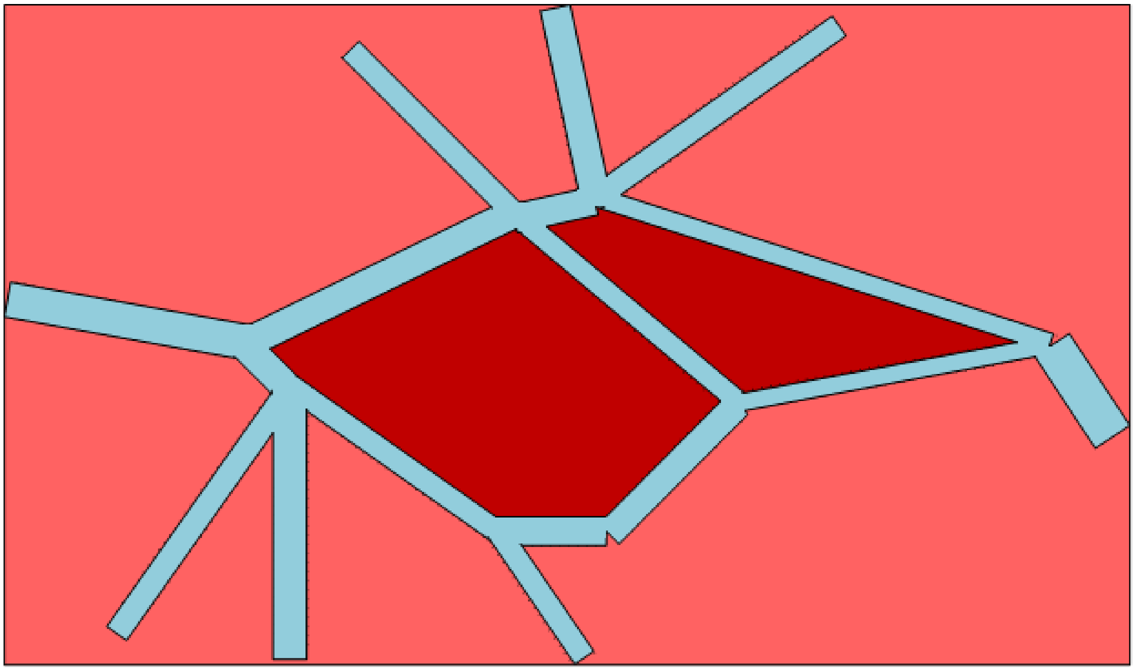


Figure 28: Inverted mesh area, including outer and inner exclusive zones (light red and dark red)

To obtain the outer and inner exclusion zones the exclusive-or function is used. The light and dark red areas together are then used as exclusion zones or warning areas.

5.2 Representation with real data

After receiving the results from PUEB the mesh looked as shown in Figure 29. It consisted of thousands of corridors, which are often overlapping strongly.

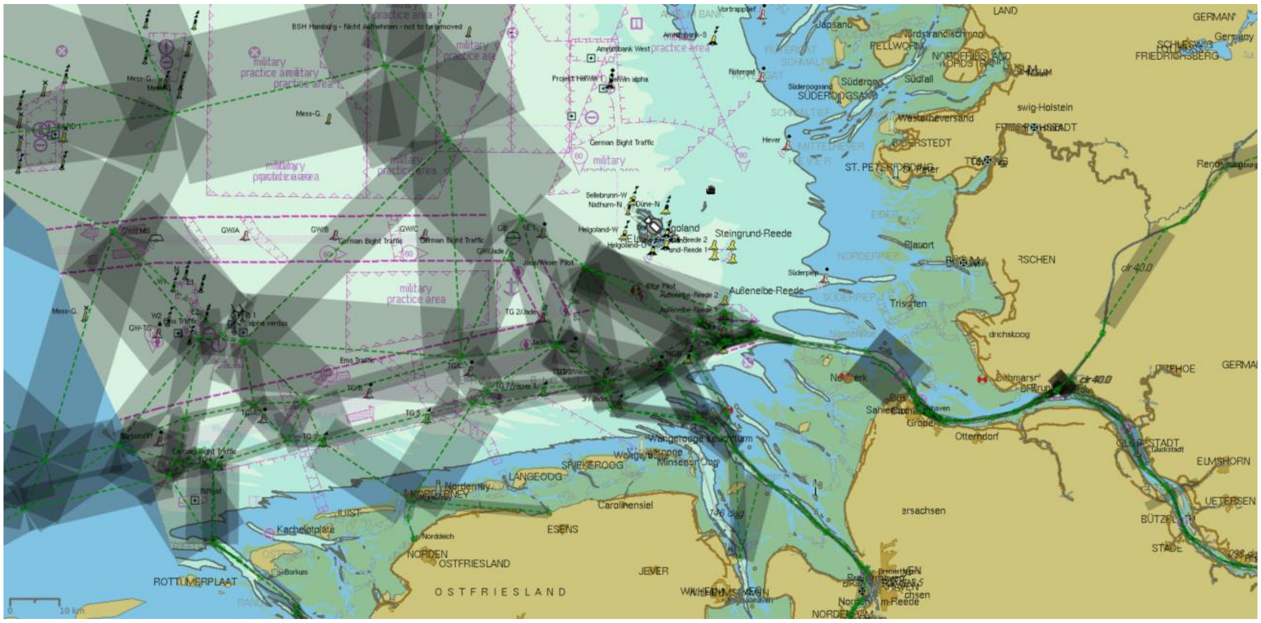


Figure 29: Mesh in the North Sea

Using the algorithm from section 5.1 the number of objects to be monitored was reduced significantly and the area looked as shown in Figure 30.

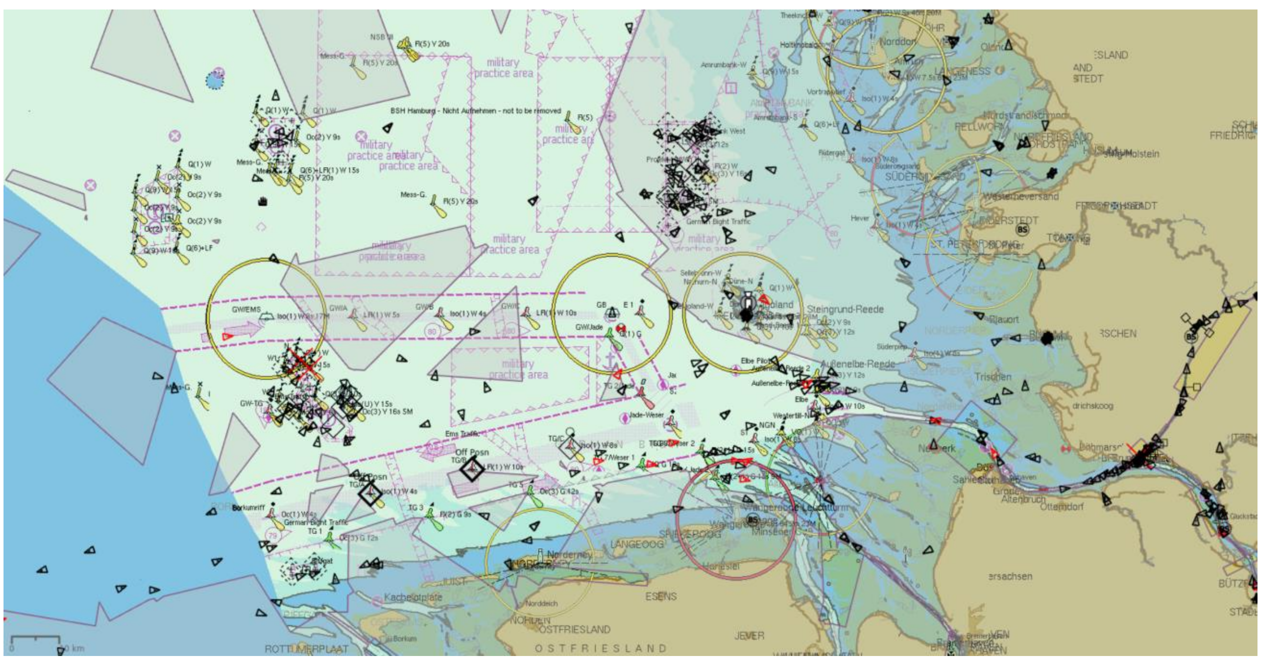


Figure 30: Warning zones after combining the mesh

6 Discussion

The method presented requires extensive preprocessing of the mesh data obtained. A typical mesh for a ship class contains thousands of segments, so that the algorithms can only determine geometrically correct exclusion zones after many hours. Basically, this relatively long processing time is not a problem since it can be assumed that the shipping corridors will not change at high frequency.

To validate the quality of the results, ship density data provided by EMSA were used and compared with the corridors. This revealed some weaknesses that must be remedied before a practical use.

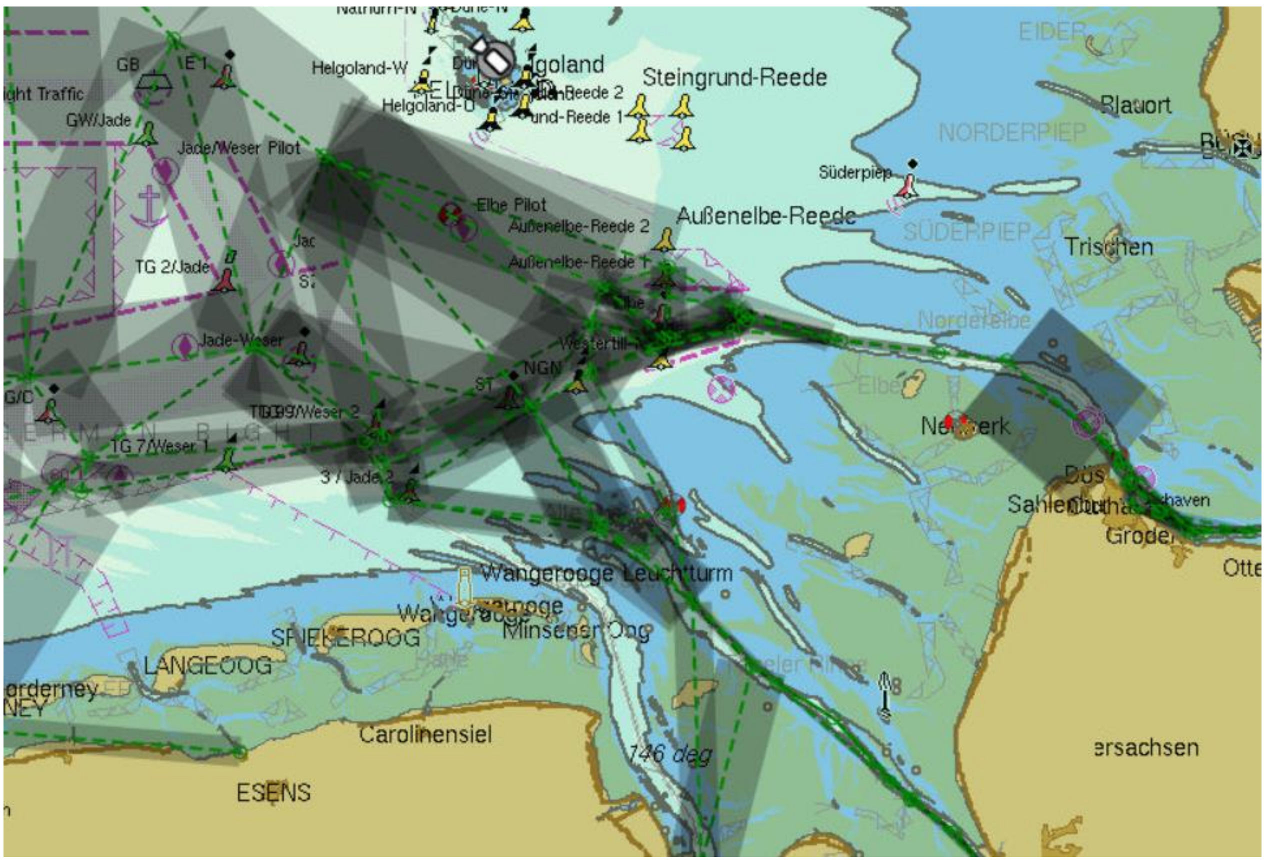


Figure 31: Mesh with multiple routes for shipping routes

1. Actual shipping routes (e.g. shipping separation areas) are sometimes not translated into a single "route", but into a very large number of parallel routes. This can be seen very well in Figure 31.

2. In curves there is the phenomenon that a multitude of apparent "abbreviations" are shown, which are wrong. This can also be seen in Figure 31.

3. There are areas where almost any direction appears to be allowed in the mesh, but the ship density data says that the ships travel along a few paths. In these areas, the network-like structure does not seem to adequately reflect the actual shipping traffic. This can be seen in Figure 32.

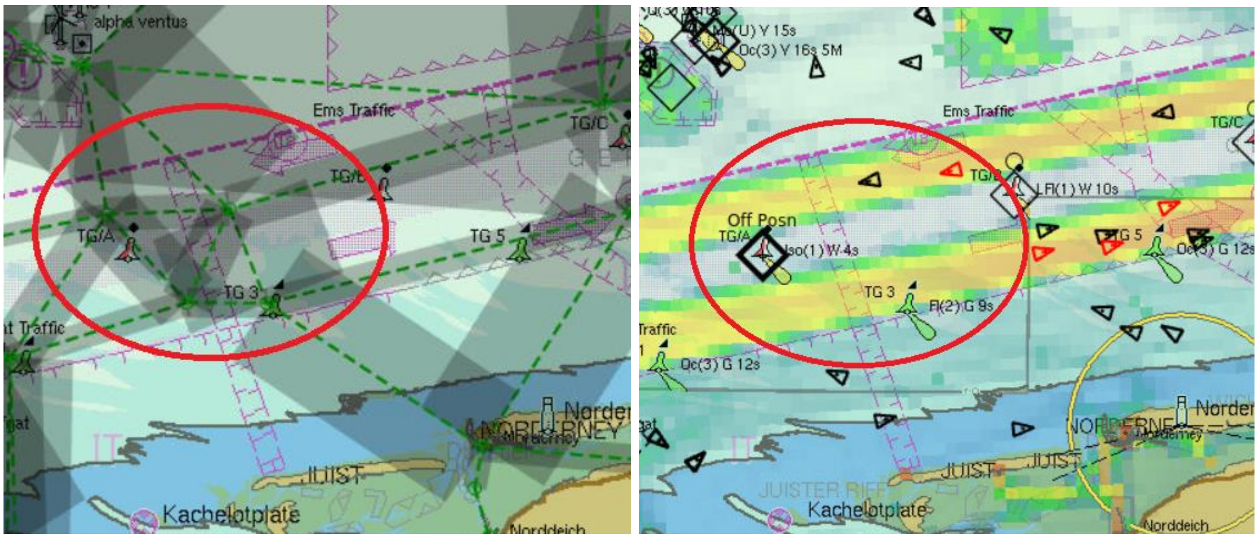


Figure 32: Comparison of Mesh representation on the left, heat map on the right

4. There are some mesh edges in areas where ships are very rarely, but certainly not at all regularly observed. You would not have expected mesh entries here, but there are some. This can also be seen in the bottom right corner in Figure 32.

5. There are edges in the mesh which cannot be explained at all, because it can be excluded that a ship could have followed this edge. This can be seen in Figure 33.

Figure 33: Corridor leading over land

6. There are some known routes that are used regularly but are not included in the mesh. For example, the deep-water area and in Figure 33 (light water color) and the approach to Wilhelmshaven.

7. There are some star-shaped structures in the mesh, for which the ship density analysis provides no explanation. It is generally not likely the vessels frequently pass “crossing points” as they try to avoid any crossings for security reasons.

In a larger number of iterations with different parameters, partial improvements could be achieved regarding the phenomena mentioned, but the weaknesses nevertheless always remained recognizable.

A comparison of the warning areas that are generated by INs algorithm from section 5.1 with an underlying heat map of vessel traffic in Figure 34 shows that false alarms would be very frequent and inevitable.

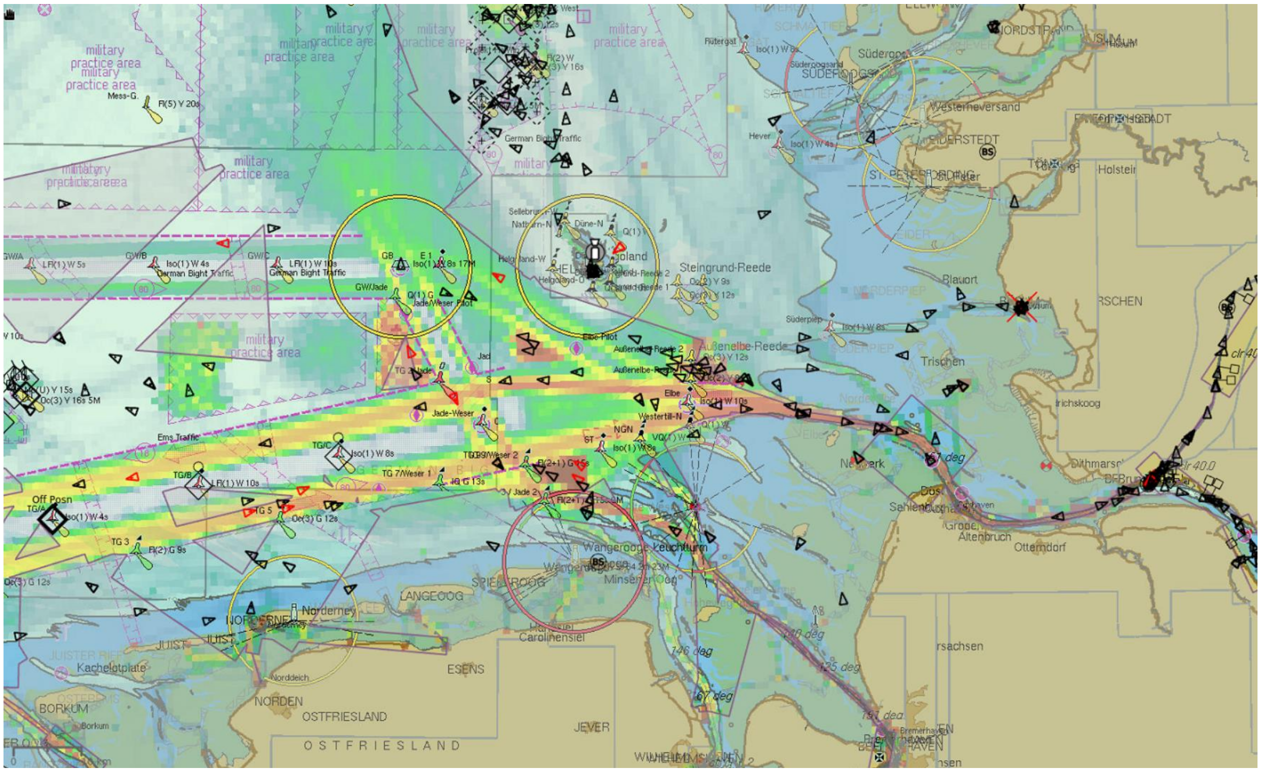


Figure 34: Warning areas compared with heat map

7 Manually created corridors

To be able to prove the correct function of traffic monitoring, corridors were manually generated for test purposes based on ship density maps. In contrast to the automatically generated corridors, these corridors are permitted areas with a predominant driving direction. For a given class of ship, the manually generated corridors only overlap at actual ship crossings. In traffic separation areas there is a corridor for each direction of travel.

A corridor has a longitudinal axis and a catchment area. It can be uni-directional (e.g. in a traffic separation area) or bi-directional (e.g. in a river without traffic separation). Corridors have a maximum permissible course deviation from the longitudinal axis. If a ship drives along a corridor and changes its course so that the maximum permissible course deviation is exceeded, the ship has left the corridor and a warning is issued.

A minimum speed can also be assigned to a corridor. If a ship falls below this speed, again an alarm is issued.

Manually created corridors look as shown in Figure 35.

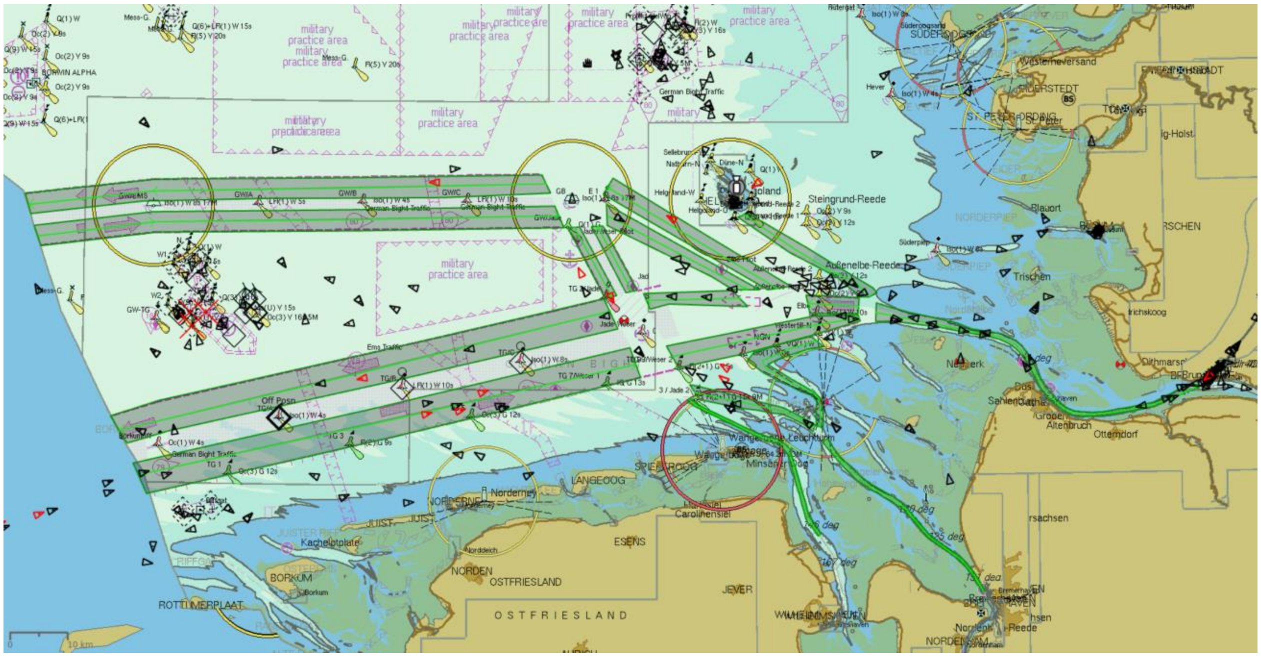


Figure 35: Manually created corridors

The comparison with an underlying heat map is shown in Figure 36.

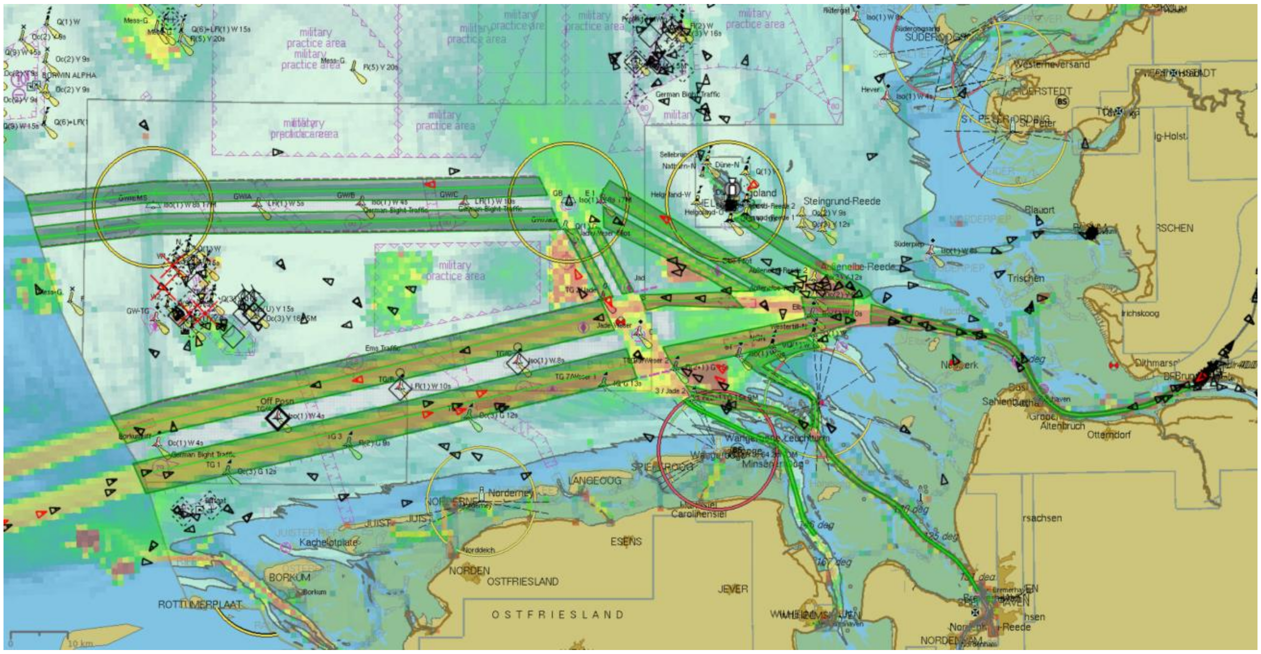
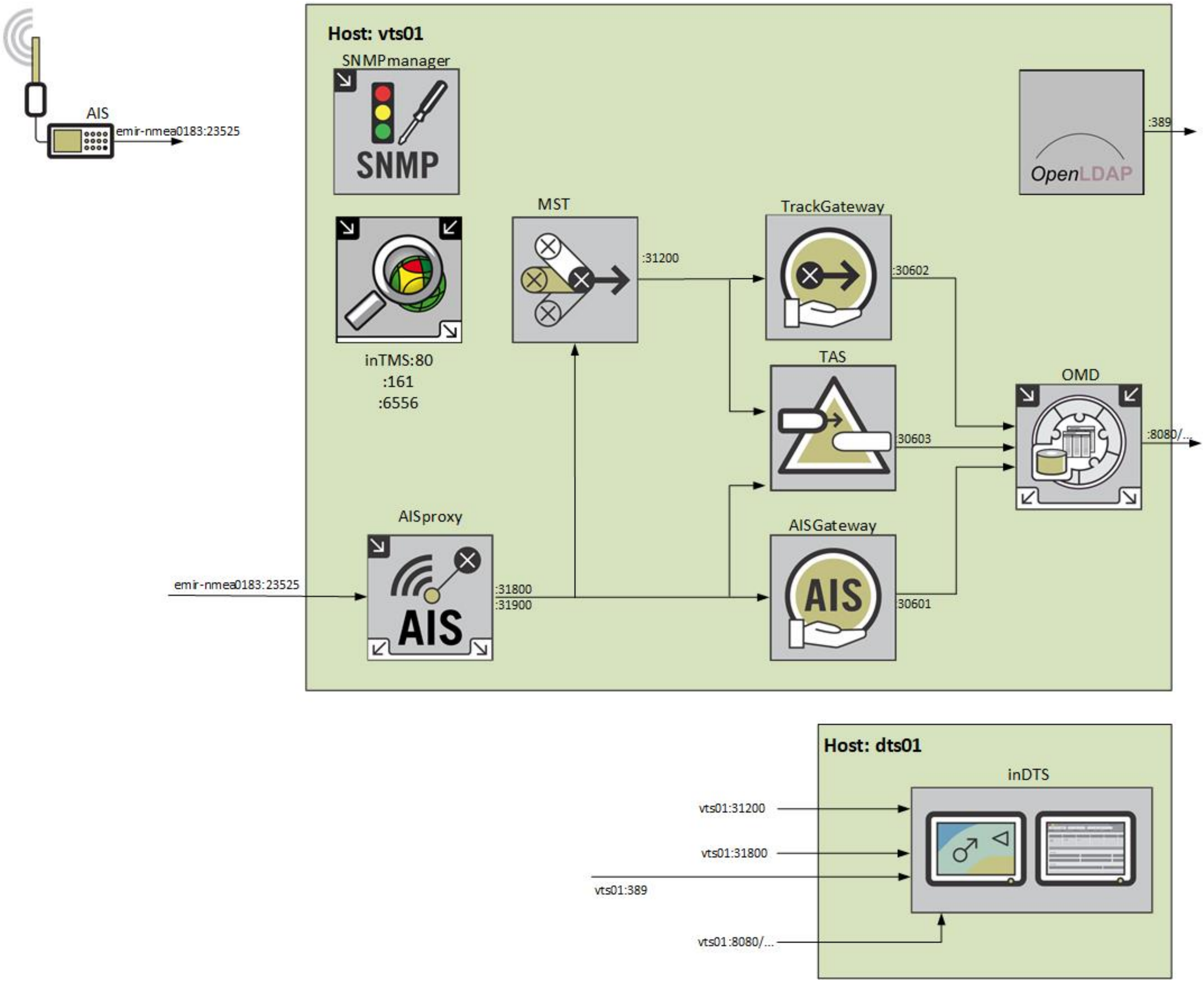


Figure 36: Manually created corridors in comparison with a heat map

8 Architecture of inVTS

IN is using a modular approach to process AIS and where applicable radar information. For the demonstration of the research results of the HANSA project IN has chosen a relatively simple setup as follows.



The module that has been adapted to implement the surveillance of recommended corridors is TAS (Traffic Analysis System). This module uses topologies (e.g. areas, lines, or similar structures) to detect unwanted behavior or vessels.

8.1 Description of Server Modules

8.1.1 AISproxy

AISproxy serves as single point for clients to access a network of AIS sources. The data of all AIS sources (in this case just one) is combined into a single data stream. Duplicate NMEA messages are filtered and reduced to one. It is possible to provide several output streams with filters based on AIS message type, geographical filters, MID, MMSI, frequency and more.

8.1.2 MST – Multi Sensor Tracker

MST uses an extended Kalman filter algorithm to correlate tracks from AIS and radar sources (when configured). MST offers an ASTERIX Cat 62 interface for further processing.

8.1.3 AISgateway

AISgateway uses AISproxy as an input and converts the NMEA Data into SOAP data for further processing.

8.1.4 TrackGateway

TrackGateway uses MST as input and converts the ASTERIX Data into SOAP data for further processing.

8.1.5 TAS – Traffic Analysis System

TAS uses AISproxy's and MST's output data as input and uses topologies to trigger traffic events. The following types are possible:

- CPA between two tracks and between tracks and topology objects.
- Warning areas (entering, leaving or inside the area) with optional filters like:
    i. Speed, heading or other dynamic attributes.

   ii. Size of the vessel or other vessel attributes

- Halos events
- Guard Line events
- Swing Circle events
- Route events based on:
   i. Direction of travel
   ii. Speed
   iii. Size/type of vessel

8.1.6 OMD

OMD is the central processing unit and has a database attached to it. It has extensions for loading the mesh of recommended routes and for requesting RTZ corridors for a single vessel. It provides track, vessel and AIS data via SOAP connections for clients.

8.1.7 Administration Modules

- SNMPmanager provides monitoring information for inTMS
- inTMS is a monitoring and administration tool (if configured) based on Check_MK
- OpenLDAP Server is used for authentication within the inVTS System

8.2 Description of Client Modules

8.2.1 inDTS

inDTS is the fully featured display application for the traffic scenarios detected via AIS – and if configured via radar. The application has been adapted to manage the display of the various routes intelligently, e.g. a route would be set visible as soon as at least one track violates the rules attached to it (wrong heading, speed, or leaving the area altogether).

Appendix 1 List of mapping between combinations and labels

The combinations are derived from the genetic algorithm's parameters and the scope of data.

The list is contained in the file: **HANSA Meshes-Labels and analysis-Appendix1-dir_to_label-mapping.txt**

A line in a file consists of two values separated with semicolon:

- first value: name of the folder on the HANSA server,
- second value: part of the label.

Sample line:

sprint_1011_2019-c-kdb-part128e300cl10p10r3.0mf0.1-graphweather;sprint-midbaltic-cargo-part128M

Appendix 2 List of available labels with variants

The variants are derived from draught, length and width of the label. When no variant is provided, the suffic is "_all".

The list is contained in the file: **HANSA Meshes-Labels and analysis-Appendix2-mesh_labels.txt**

A line in a file consists of one value – a label.

Sample line: sprint_midbaltic_cargo_part128M_draught_very_small

Appendix 3 Sample code to retrieve a mesh

The following curl command can be used. Please note that sometimes long timeouts are necessary.

curl --compressed -v -i -d @test_request_mesh.json -H "Content-Type: application/json" -H "Accept-Encoding: gzip" -X POST localhost:54321/hansa/mesh/ > result.json

The contents of the file **test_request_mesh.json** is as follows:

```
{
  "ll": {
    "lat": 0,
    "lon": -90
  },
  "ur": {
    "lat": 90,
    "lon": 90
  },
  "vtype": "navtor_all_cargo_part256L_draught_very_small"
}
```

The following code in Python can be used to retrieve all meshes. Such operation is time consuming (several hours). It is better to use cached files on the server.

```
import os
import requests
from time import time
from datetime import timedelta

URL = "http://api:8000/hansa/mesh/"

request_params = {
  "ll": {
    "lat": 0,
    "lon": -90
  },
  "ur": {
    "lat": 90,
    "lon": 90
  },
  "vtype": "baltic_cargo_navtor1011_corridors"
}


with open('mesh-labels.txt', 'r') as infile:
  for label in infile:
    label = label.strip()
    FILENAME = f'mesh_json/{label}.json'
```

```
if not os.path.exists(FILENAME):
    print(label)
    request_params['vtype'] = label
    try:
        time0 = time()
        response = requests.post(URL,json=request_params,timeout=(5, 300))
        if response.status_code==200:
            with open(FILENAME, 'w') as outfile:
                outfile.write(response.text)
            tt = str(timedelta(seconds=(time()-time0)))
            print(f"\u001b[1A{label} \u001b[32mOK {tt} sec.\u001b[0m")
        else:
            print(f"\u001b[1A{label} \u001b[31mMissing\u001b[0m")
    except Exception as ex:
        tt = str(timedelta(seconds=(time() - time0)))
        print(f"\u001b[1A{label} \u001b[31mConnection error {tt} sec.\u001b[0m")
```

The longest timeouts are mentioned below:

```
root@ba1f0559e763:/app# python download_meshes.py
[Errno 17] File exists: 'mesh_json'
navtor_all_cargo_part512M_all OK 0:01:12.052096 sec.
navtor_all_cargo_part512L_draught_small OK 0:01:39.907015 sec.
navtor_all_cargo_part512L_width_very_small OK 0:02:04.652446 sec.
navtor_all_cargo_part512L_width_small OK 0:01:19.331162 sec.
navtor_all_cargo_part512L_length_medium OK 0:01:53.529226 sec.
navtor_all_cargo_part512L_length_large OK 0:01:25.118654 sec.
navtor_all_cargo_part512L_all OK 0:02:34.186655 sec.
navtor_all_passenger_part256L_all OK 0:01:05.014408 sec.
navtor_all_passenger_part512M_all OK 0:01:05.611700 sec.
navtor_all_passenger_part512L_draught_small OK 0:01:19.303081 sec.
navtor_all_passenger_part512L_length_very_small OK 0:01:24.423567 sec.
navtor_all_passenger_part512L_width_very_small OK 0:01:49.265392 sec.
navtor_all_passenger_part512L_all OK 0:02:01.974596 sec.
navtor_all_tanker_part256L_length_small OK 0:01:08.623250 sec.
navtor_all_tanker_part256L_all OK 0:01:14.591994 sec.
navtor_all_tanker_part512M_length_small OK 0:01:05.814503 sec.
navtor_all_tanker_part512M_all OK 0:01:11.355284 sec.
navtor_all_tanker_part512L_draught_small OK 0:01:58.073362 sec.
navtor_all_tanker_part512L_width_very_small OK 0:01:49.754296 sec.
navtor_all_tanker_part512L_length_small OK 0:02:12.227224 sec.
navtor_all_tanker_part512L_width_small OK 0:01:19.432081 sec.
navtor_all_tanker_part512L_all OK 0:02:24.749675 sec.
navtor_german_cargo_part512L_all OK 0:01:34.812813 sec.
```