HÖGSKOLAN SKÖVDE
Institutionen för kommunikation och information

Pattern Parameterization with Granules in Ship Movements

Describing identifying aspects of movement patterns with varying levels of granularity

John Adolfsson

Pattern Parameterization with Granules in Ship Movements

Examensrapport inlämnad av John Adolfsson till Högskolan i Skövde, för Kandidatexamen (B.Sc.) vid Institutionen för kommunikation och information.

2010-01-01

Härmed intygas att allt material i denna rapport, vilket inte är mitt eget, har blivit tydligt identifierat och att inget material är inkluderat som tidigare använts för erhållande av annan examen.

Signerat: _____

Pattern Parameterization with Granules in Ship Movements

John Adolfsson

Student-email: a07johad@student.his.se

Summary

This report aims to explore a possible transparent alternative to the black box approach of machine learning in identifying a ship's type from simple movement data, consisting of a set of coordinates with timestamps. This is achieved by an application that converts the set of coordinates to vectors and assigns them various traits, such as turn radius, speed and distance traveled, and then identifying the correlation between collections of different values of these traits, called granules, and different ship types. The results show a definite connection between certain kinds of granules and certain ship types and lay the foundation for building a more well defined syntax for ship identification.

Keywords: Pattern detection, AIS, ship surveillance, machine learning

Contents

1 Introduction

Saab Microwave Systems is a developer of software for use with surveillance and radar technology. Their products facilitate interpretation and organization of the large amounts of data associated with multisensory input, as well as information fusion functions needed to consolidate the dissimilar input from several different kinds of sensors. One such product called the IBD is currently under development. One of its intended features is to be able to describe the type of a ship by the sensory input it receives about it. For most ships, this is no problem as their in-built AIS transmitter constantly relays data about its bearing, speed, destination and type.

But in some instances, the only available data on a vessel is where it is at what time; coordinates and timestamps. Identifying these is a non-trivial feat, as the movement of one boat could conceivably be emulated by any smaller, faster boat. This fact alone makes definite identification impossible. It is possible, however, to assign each vessel a probability distribution based on how well its attributes compare to those specified for various ship types. The problem then changes to specifying interesting attributes that different ships tend towards, and creating ways to identify these from the bare minimum data.

Problems of this nature are usually addressed through a field called machine learning, where a computer learns to associate certain patterns or parts of patterns with specific values. It can, for example, be shown a picture of a sloop and thereafter output high values when it is shown a sailboat. The downside is that it may have associated a sloop with tall structures and will also output high values when it's shown a lighthouse, or it may have seen only large white shapes and will include clouds in the identification. The inherent uncertainty in machine learning makes it tricky to control and to teach it the specific patterns that the operators want it to learn. There is a lot of data for the computer to pick up on when making comparisons, and too many levels of pattern complexity to search through them all.

This problem is not impossible to overcome. A possible solution is to define a number of small patterns beforehand that fit the sets of raw data. These patterns can be as simple as the ship going in a straight line for ten minutes, or as complex as a 500 m^2 oval-shaped spiral. They would then be fitted to the larger movement pattern and could be used to describe parts of the vessel's journey to cluster the interesting parameters to a more lucid description. It is this approach that this report will attempt to describe, from the identification of interesting behaviour to the description of the parameters.

2 Background

The identification and description of patterns interests numerous fields and areas of research. Data mining, machine learning, cluster analysis, computer vision and related fields all rely heavily on the ability to convert raw data to meaningful information (Hegland, 2003 p. 6; Jain, 1999a p. 1-2; Jain, 1999b p. 265; Campos, 1998). Its applications can be found in areas as diverse as medical diagnosis, traffic control, video games and marketing (Jain, 1999a p. 2-3; Kang, 2004; Adriaans, 1996 p. 7-8).

The IBD or Intelligent Behaviour Detector is a product developed by Saab Microwave Systems as a means for interpreting surveillance data, finding interesting patterns and behaviour, and notifying relevant parties. These three main areas are further divided into subcategories based on the type of sensory input and the objects being observed, implemented with a type of entity called modules.

The architecture of the IBD allows for modules to be added arbitrarily depending on the sought feature, such as detecting loitering in people on airport surveillance footage or an imminent collision of two approaching vehicles. These modules have access to other module output, all sensory input, and the overall database of the IBD (Saabgroup, 2009).

The goal of this report is to provide a method for describing basic aspects of movement patterns of ships for use in movement analysis modules.

2.1 Relevant Fields

This report will draw on resources from various existing fields for methodology, theory and inspiration. This section will give an overview of these fields, with an emphasis on how they relate to this report.

2.1.1 Pattern Recognition

Pattern Recognition is a broad, vaguely defined field that attempts to find interesting correlations between and within sets of data. It spans a number of subfields and is primarily considered in light of its applications in other fields, such as computer science, physics, neurobiology, psychology, engineering, statistics, mathematics or cognitive science (Pal, 2001 p. 2).

Pattern Recognition attempts to emulate in computers the human ability to make accurate distinctions between objects and concepts based on vague and unreliable data. As movement analysis deals with highly situational and flexible behaviour, it is a prime example of the kind of discrete yet relational data set that makes general solutions to the problem of Pattern Recognition so elusive (Jain, 1999a p. 2).

2.1.2 Granular Computing

Granular Computing is an emerging field attempting to consolidate and extrapolate on standards for grouping related atomic variables into lower resolution components. It builds on the view that human perception and cognition intuitively focus on that level of detail in a given system which matches potential, known patterns, and attempts to describe these levels in terms of fundamental information granules (Pedrycz, 2007).

A granule is defined as a meaningful abstraction of data, a pattern that emerges when considering a certain level of resolution, but is either lost in noise or in information entropy when viewed at a higher or lower resolution respectively. An example is the proverbial forest that can't be seen for the trees.

Common terms to describe granules are large and small, coarse and fine, low and high resolution, low and high granularity. These refer to examining data in large and small sets; building on the previous example the forest would be large, coarse or low while the trees would be small, fine or high.

Granules are not universal, but arbitrarily defined for each instance, which gives rise to two separate subdisciplines; Granule Construction and Granule Calculation (Yao, 2000; Pedrycz, 2007). These deal with the identification of atomic descriptors in given patterns; and the use of these in describing more complex variations, respectively.

2.1.3 Data Mining & Knowledge Discovery in Databases

The field of Data Mining is concerned with methods for distilling interesting, new information from large sets of data (Hegland, 2003 p. 5). The process can be broken down into three steps (Fayyad, 1997 p. 102):

- Normalize, refine or complement the data in a pre-processing stage
- Identify and extract patterns
- Analyze the relevance of the results

Data mining is sometimes considered a subset or a step of a broader subject called Knowledge Database Discovery (KDD) (Adriaans, 1996 p. 5; Fayyad, 1997 p. 102), in which case its scope is limited to just identifying and extracting patterns. In many publications the two are considered synonymous (Adriaans, 1996 p. 5; Hegland, 2003), but in this report we will use the former definition.

3 Aim

As previously written, pattern recognition is a popular field with a wide variety of applications. Substantial research has been done in these areas, especially over the past few decades when new business practices and standards, coupled with the rise in ubiquity and capability of hardware, has significantly increased the amount of data available to commercial and administrative entities, while intuitive comprehension has understandably decreased (Adriaans, 1996 p. 2; Olafsson, 2006). Areas like cluster analysis employ multiple parameters to group features into multidimensional categories, where computers excel, as arbitrary numbers of dimensions hold a similar complexity to them, but humans find it difficult, if not impossible, to visualize connections in hyperdimensional space (Höppner, 1999 p. 1; Olafsson, 2006; Fayyad, 1997 p. 101).

Focus has therefore been put on Machine Learning, to strip as much of the interpretation away from the human aspect and allow autonomous programs to interpret the data themselves and output the correlations they find. The benefit of this approach is the aforementioned comprehension and speed; computers are much better at finding connections between entities with hyperdimensional parameters, and are capable of processing far greater sets of data. The disadvantage is the uncertainty of the results. Machine learning approaches are good at finding correlations, but a similarity in one or more aspects is not inherently interesting, nor is its relevance obvious (Fayyad, 1997 p. 102-103; Kodratoff, 2001 p. 15). If the human operator cannot understand the connection, the data is little more than additional noise in the system.

There are several ways to deal with this uncertainty. Some approaches consider it a part of the recognition criteria to link the correlations to previously explored and established patterns (Jain, 1999a p. 3), whereas some limit the scope of the search by selecting a suitable subset of dimensions to use for clustering (Fayyad, 1997 p. 101).

Another approach to addressing the problem is a combination of the two, as described in works on Granular Computing, namely breaking down the description of types of patterns into so called granules, which will both limit the number of dimensions and make them more interesting to an observer.

A similar method to detect anomalies in ship movements was proposed by Ekman and Holst (Ekman, 2003) in their SICS evaluation for SaabTech. In their report, they propose that a set of basic sensory input about a ship can be used in conjunction with adjacent ships, statistics of area and changes over time to satisfactorily describe interesting, anomalous behaviour.

This report aims to explore Ekman and Holst's proposition by identifying the granules required for describing the defining aspects of movement patterns utilized by different ship types. We will concentrate on a level of granularity that minimizes the number of dimensions needed, but is fine enough to not miss interesting behaviour. We will also assume that the data we have to draw from will be limited to a set of coordinates with related timestamps.

3.1 Objectives

3.1.1 Implementation

The purpose of the implementation is to provide a way to define granules, as well as an interface for matching granules to ship movements from groups of ships of similar type. As the result of this report is meant to be incorporated into the IBD at Saab Microwave Technologies, the application will need to be built using its interface, as well as access its databases of stored ship movements. It will utilize the existing graphical interface of the IBD and will be written in Java.

The implementation will be divided into several components. The three main components will be:

- Ship type group data
- Granule Library
- Matching algorithm

Ship type group data

The ship type group data refers to the data that will be collected from the IBD AIS database, and will be sorted in several groups based on type. The groups are selected from the AIS registry based on uniqueness of movement patterns and prevalence in the Gothenburg port area. The groups selected are:

- Fishing ships
- Cargo ships
- Passenger ships
- Pilot ships
- Pleasure / private ships
- Sailing ships
- Rescue ships

Fishing ships have distinct movement patterns that are usually centered on small areas with many sharp turns and short, jerky movements. This behaviour is called trawling, where the boats attempt to cover as much of a fish-rich area as possible.

Cargo ships are characterized by large, slow movements with long acceleration stretches, few, wide turns and generally stable bearings. This behaviour is indicative of high fuel consumption in maneuver adjustments.

Passenger ships keep within a defined area, going back and forth with few deviations between two or more points. Their repetitive movement patterns are easily identified.

Pilot ships move much like fishing ships, but within a smaller area.

Pleasure / private ships are any non-commercial ships with an AIS tracker. They do not follow any standardized patterns but are useful as a comparison since they generally exhibit certain capabilities such as speed and high turn rate.

Sailing ships have some limitations that make them interesting as a comparison to the other groups.

Rescue ships move similarly to pilot ships and fishing ships, going long or small distances before occupying an area for a certain amount of time.

Ship data of the selected type will be extracted from the IBD AIS database at SAAB using SQL queries and stored in custom data types

Granule Library

The Granule Library stores the definitions of granules, defined in a way that allows them to be matched to the format used to store the movement data. The format must therefore be designed along with the Granule Library, or with the interface of it in mind.

Matching Algorithm

The matching algorithm or functions will find occurrences of granules described in the Granule Library in the ship data, and provide feedback regarding occurrence of granules in a ship's movements, as well as metadata collected during the comparison.

Initial granules

The possible movement range of any given ship is significantly limited and well described. The possible data of a ship at any time, given a set of positions with timestamps, can be described with two vectors; its position and its velocity vector given its previous position. These can be further extended over longer spans to calculate for each time interval:

- Position
- Speed
- Acceleration
- Direction
- Rate of Turn

As such, detecting certain thresholds of these variables would be prime candidates for atomicity; the highest level of granularity.

3.1.2 Iteration

Once the implementation is complete we must interpret the data and use it to find better descriptions for each group.

3.2 Method

It is difficult to prove understandability of a system, but it is possible to show how well different ship types can be described by the proposed granules. The methodology used will be Correlational as described in (Ellis, 2009 p. 327), as the focus is to find a correlation between more or less basic aspects of ship movements and their type classification. In a Correlational method it is necessary to have both a solid

framework with which to describe the data you wish to correlate, and to know what signs to look for.

In 3.2.1 Implementation we describe the mechanism we will use to both extract the data and to describe it in relation to the ship types. In 3.2.2 Iteration we use this data to create better descriptions and refine our searches.

3.2.1 Implementation

The implementation will serve to illustrate the correlation between granules and different kinds of ship. By matching granules to the movement patterns of groups of ships, we can extract data of granule usage, and compare the prevalence. The occurrence of a granule in one group compared to others would indicate the degree to which it can be used for identification, called the accuracy of the granule.

Further data that can be drawn from the matching algorithm is the span of variable variation in granule matches. If a granule finds many matches in a given ship's movement based on one criterion, several other criteria may be overlooked that, in conjunction with the first, could increase the granule's accuracy. E.g.: both fishing boats and ferries make many turns based on Rate of Turn criteria, but ferries slow to a stop beforehand based on Speed, while fishing boats keep a constant Speed through all their turns.

The important aspects of the implementation are those that can provide meaningful feedback of the granule matches, as these are key to the next step in the verification: the iteration.

3.2.2 Iteration

Each time we extract and analyze the data from the granule matches, we will use it to create new granules or modify existing ones, based on the characteristics of the ones that are most uniquely matched to the groups. That is to say, if a granule is shown to occur more often for a specific group of ships we can spawn a number of new granules based on it with slightly altered parameters and compare in the next iteration if their accuracy for a specific group has changed. This way the granule library will iteratively evolve to more uniquely describe ships of different types.

The granule library and each of the more accurate - or otherwise interesting - granules will be evaluated each iteration.

The granules will be evaluated on their accuracy, which is a measure of its highest percentile in relative distribution of group matches. A granule with 20, 200 and 200 matches for groups A, B and C respectively will show a ~4.8%, ~48% and ~48% accuracy. The advantage of this metric as opposed to measuring the difference between the highest and second highest accuracy is that it allows for analysis of granules that have a high accuracy in two groups, which could lead to the discovery of defining aspects in the group with the lowest accuracy (group A in the example given above).

The library will be evaluated on its ability to identify classes of ships, and the parameters that are unique to these.

4 Implementation

4.1 The produced system

4.1.1 Object overview

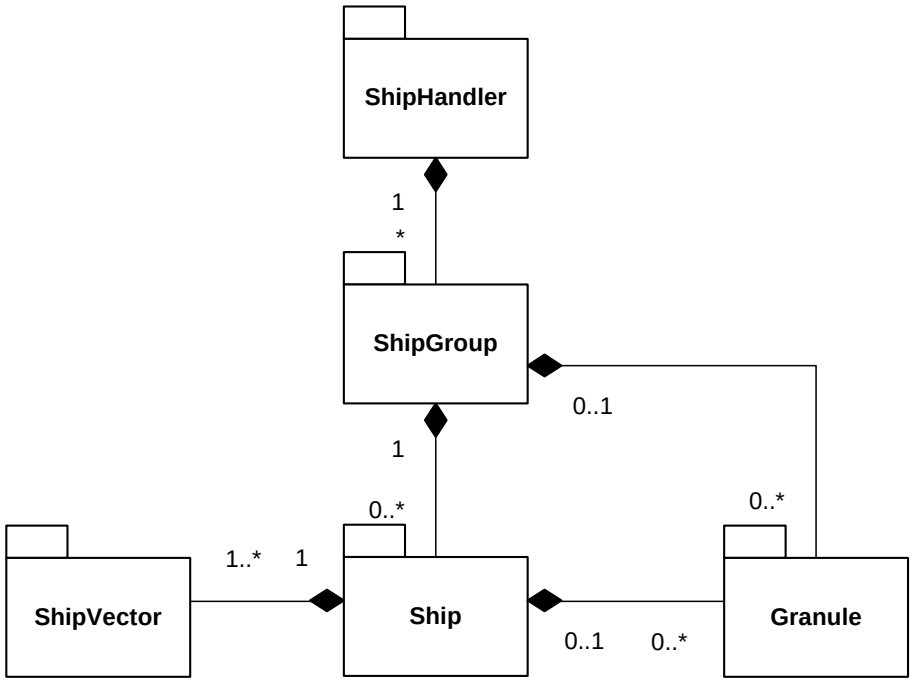
The application is comprised of three main types of objects. Below is a summary of their names and overall functionality, with more detailed descriptions in later sections.

- Management objects
- Utility objects
- Granules and granule utilities

The management objects are responsible for retrieving and collating the movement data from the database, and facilitating the usage of granules. They provide the framework for the granules to work in, and have been explicitly constructed separately to make the design of granules as modular as possible. These include Ship, ShipGroup, ShipHandler, DatabaseConnector, DatabaseRetriever and Filemanager.

The utility objects are the classes that store movement data and provide related functionality. These are more closely tied to the design of the granules, as they are used as intermediary objects between them and the management objects. The members are ShipType, DataPoint and ShipVector.

The granules and granule utilities perform searches and calculations on ShipVectors, and return data regarding occurrence and various locally defined parameters. Each granule extends the abstract base class Granule which provides the interface for accessing the granules. The classes included under this definition are Granule, DistanceGranule, MeanStatsGranule, RoTGranule, SpeedGranule, Demarcation, Statistics, Turn and TurnFinder.



4-1 Management, Utility and Granule objects

4.1.2 Data loading and interpretation

The program starts with retrieving movement data from an external source and converting it to instances of the PointData class. This holds the timestamp, the longitude and latitude, and the ID and type of the ship. The data is retrieved in one of two ways:

- From an SQL database via java-supported query functions
- Through locally defined .pdts files that store LinkedLists of PointData

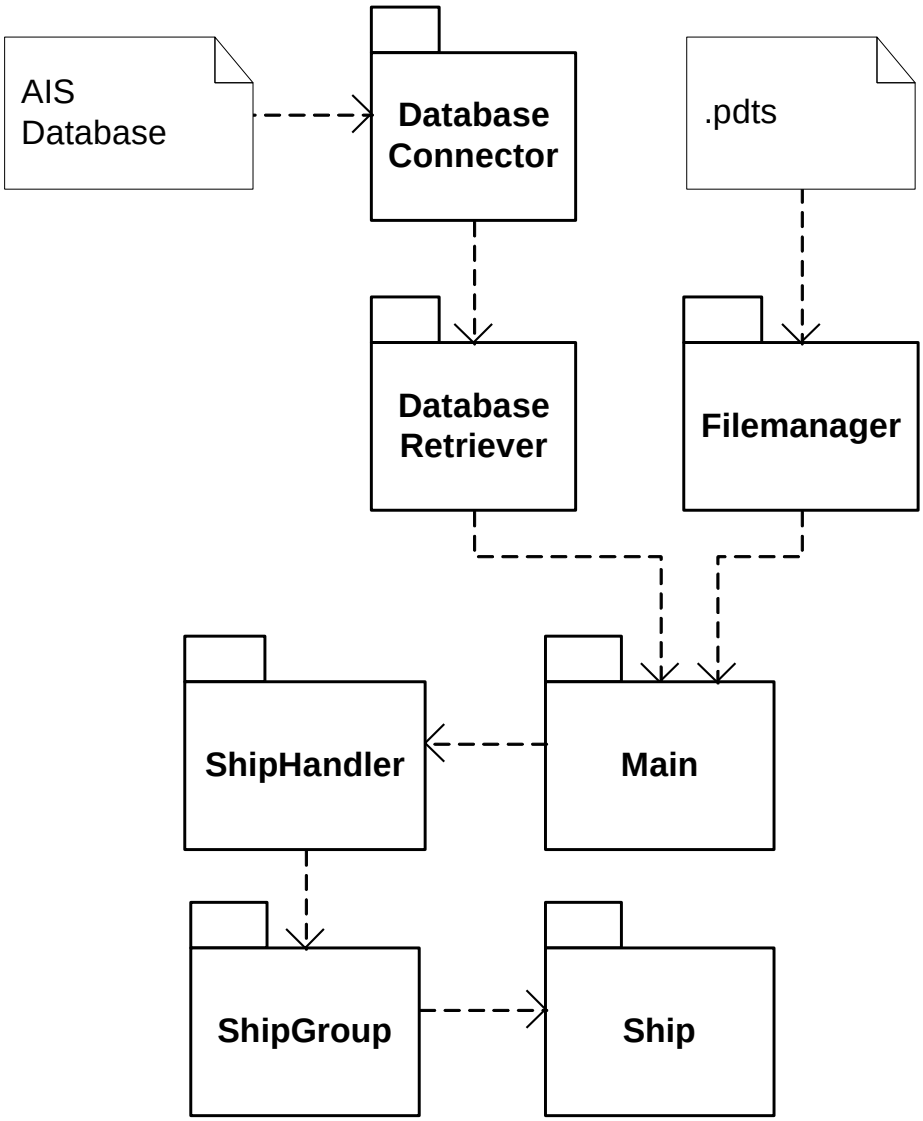
The former uses the DatabaseConnector and DatabaseRetriever classes to connect to a PostGIS database. Queries retrieve coordinates, timestamps and ID in the requested table and create a new instance of PointData for each entry. If the database is absent and no connection is made, the program attempts to access four .pdts files located in the project folder, which are loaded one by one and cast to LinkedLists with PointData that was previously retrieved from queries. This latter option is to allow the program to execute on computers without the SQL database.

These PointData are given to the singleton ShipHandler, that holds four instances of the class ShipGroups. ShipHandler is responsible for distributing PointData and Granules to each of the four ShipGroups, and to collect return data from the granules. Each ShipGroup corresponds to a specific ship type and hold collections of Ships of their respective type. When a DataPoint is given to the ShipHandler, it checks its type and sends it on to the proper ShipGroup, which either adds it to the Ship with the corresponding ID, or creates a new Ship with that ID.

As mentioned in 3.1.1 the application required that the format used to store movement data was compatible with, or usable by, the granules. The ShipVector class was created to accomplish this, which extrapolates on DataPoints to provide a common representation of basic navigational shifts, such as changes in direction and speed. When a Ship is given a PointData, it stores it in a list in wait for a call to the createShipVectors function, which converts the stored PointData to a list of ShipVectors that represent the routes of the ship.

This conversion from PointData to ShipVector is complicated by erroneous entries in the AIS database, and the realities of ship behaviour. Many ships keep transmitting position data even at rest, which results in several vectors with no variation except for time. The AIS system also has a default position outside of the geographic coordinate system, located beyond the geographic North Pole, that is used for faulty signals, and is randomly distributed among the regular entries. Additionally, some ships leave the range of the AIS scan and return later, creating large gaps in the sequence of data points in both time and distance.

To avoid these anomalies, Ship discards those PointData that are too close to the previous ones, as well as those that are out of range. It also inserts breaks in the movement stretches where too much time has passed, so that a Ship might contain a number of stretches.



4-2 Data points being transferred from external source to Ships

4.1.3 Granules

Granules are inherited from the abstract base class Granule. They are defined and added to the ShipHandler singleton in the Main class. The ShipHandler creates a copy with the same settings for each of the four ShipGroups and pass them on to them. The ShipGroups hold the given copy as a master copy and create additional copies for each Ship they contain. The Ships hold a collection of granules which, when requested, they provide with their ShipVector collection in the Granule function MatchShipRoute. The granules in turn perform their algorithms on this movement data and save the results. When prompted, the Ships return their lists of granules to their ShipGroup, where they are fed to their respective master copy using the Granule function MergeGranule. This function merges the results from all granules of their type and stores statistical and meta data.

Most Granules use the utility class Statistics to calculate one or more of three kinds of averages; the mean, median and mode. Further discussion of these follows later in the text.

MeanStatsGranule

The MeanStatsGranule collects elementary statistics from the sets of ShipVectors, regarding average length, acceleration, deceleration, rate of turn and time per vector. The results of this makes it easier to set the parameters and thresholds for further granules, as they reveal the approximate spans that the ships act in.

RoTGranule

The RoTGranule uses the utility TurnFinder to extract the turns in the movement patterns based on the given parameters RoTThreshold and RoTSensitivity. These define the minimum total size of a turn in radians and the minimum size of a turn between ShipVectors divided by their length, respectively, for the stretch to be counted as a turn. It then collects statistics regarding these turns; their length, size, duration, occurrence, area covered and how far apart each turn is from the others.

This Granule is acutely sensitive to parameter changes and requires a great deal of calibration to provide the desired results. If the RoTSensitivity is set too low it might register uninteresting standard bearing adjustments, if it is set too high it might exclude long, gradual turns. RoTThreshold is equally delicate, too high values will only register long, gradual turns and gloss over sharp, extreme ones, whereas a low value will potentially sum several turns into one, where they might have been interesting individually.

The parameters set will define what kind of turns will be found, and thus what kind of ships it will be most applicable to. It is in finding these desired values that the output of other granules like the MeanStatsGranule can be useful.

DistanceGranule

The DistanceGranule finds sets of ShipVectors that keep within a certain distance of one another, while fulfilling criteria of exceeding a distance travelled treshold, not exceeding a certain time limit, both or neither, depending on the parameters set.

This allows for finding movements that are contained in a small area but might still have travelled a long way or stayed in one place for a long time. This is behaviour that can be indicative of specialized ships that perform location-specific tasks as opposed to shipping over large areas.

Like with the RoTGranule, the parameters are sensitive and will require calibration to function properly, not least to ascertain what size areas to investigate or how long the interesting tasks might take to complete.

4.1.4 Granule utilities

The granule utilities are classes that hold data and functions specific to one ore more granules.

Turn

The Turn class represents the stretch of ShipVectors that together constitute a turn. It provides data regarding the turn's size, distance travelled, distance from the start of the turn to the end, and the turn's duration. The class can calculate this data in the Calc function after a start and end ShipVector have been set.

TurnFinder

As mentioned in RoTGranule, the TurnFinder locates and returns the Turns in a set of ShipVectors. It does this by checking for stretches of ShipVectors whose Rate of Turn exceeds the specified sensitivity and then summarizing it to see if the total exceeds the specified threshold.

Demarcation

Demarcation is a simple holding class for the data which the DistanceGranule looks for. It stores a Start and an End ShipVector describing the relevant stretch of movement, along with the radius of the demarcated area, the distance travelled within it and the time spent there.

Statistics

The Statistics class provides functionality for finding averages in collections of float values or integers. The three available averages are the arithmetic mean, the median and the mode.

Mean

The arithmetic mean is the summarized value of a collection of elements, divided by the size of the collection.

Median

The median is the middlemost element in an ordered collection.

Mode

The mode is the most represented value in a collection. This is a simple calculation with integers, but for floating point values, each value is most often unique and simply counting occurrences would be ineffectual. Instead, the Statistics class calculates the most represented value incrementally per decimal, narrowing down the collection to a defined precision and returning one of its values.

All three of these representations of the average have disadvantages as well as benefits. The arithmetic mean works best when the values do not differ much from one another, whereas extreme values on either end can skew results. The median works best when the value distribution is evenly spread out over all elements, but it can also give irrelevant data if the values are too extreme or centered to either end. The mode gives a usable metric to compare the other two against by showing the most common value, which can indicate if the collection is skewed in one direction or the other. But mode is per definition uncertain, since several different values can all be equally represented, and the result might just be one of many.

In providing all three, more meaningful analysis is possible, giving the option to discard certain data if it is evident that it falls prey to outliers or skewed distributions.

4.2 Measurements

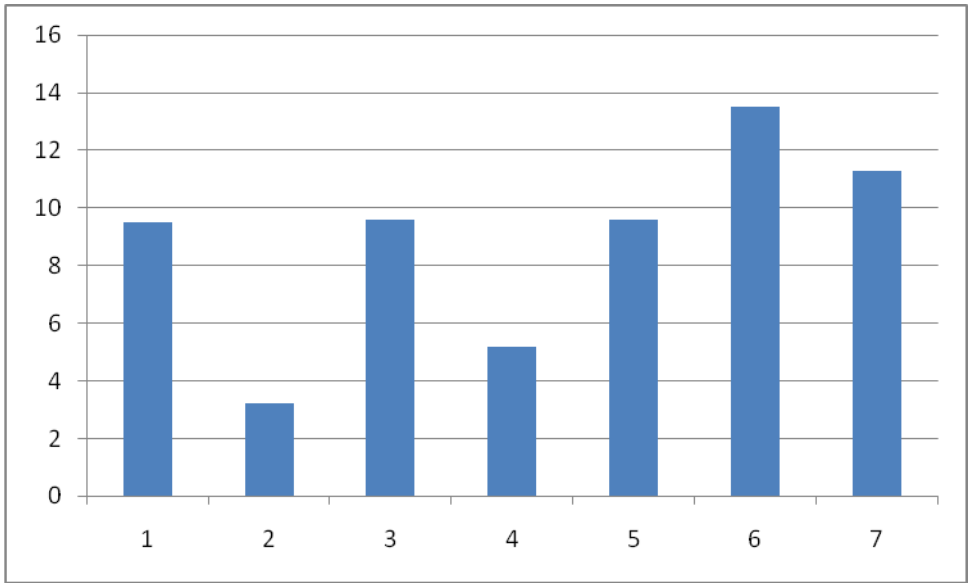
We've tested variations of the parameters for the RoTGranules' and DistanceGranules' inputs to find the combinations that yield the highest number of occurrences for each ship type. The results are presented below. All units are arbitrary but internally consistent regarding the comparison between the ships.

4.2.1 Rate of Turn Granule

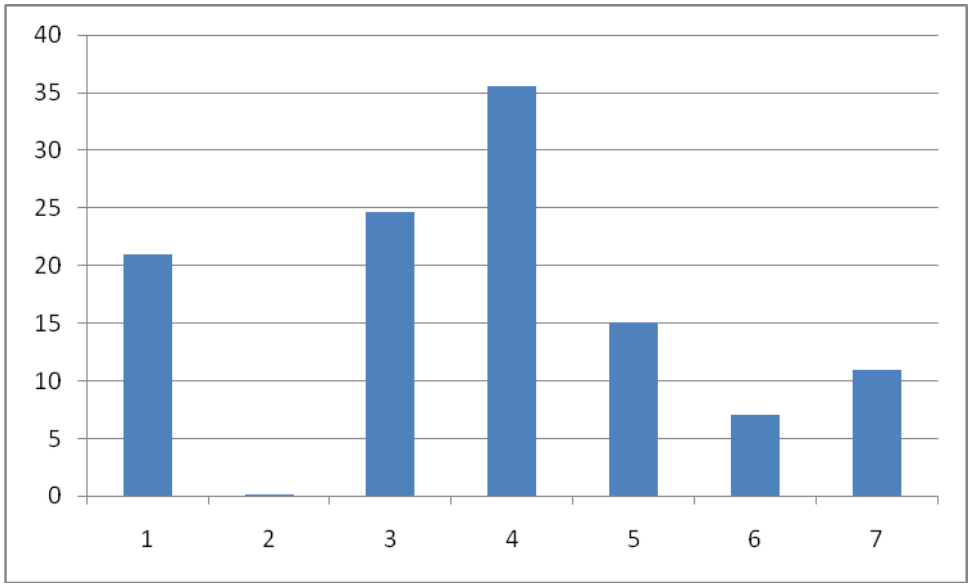
The charts represent the Rate of Turn Sensitivity values for:

1. Fishing ships
2. Cargo ships
3. Pilot ships
4. Passenger ships
5. Pleasure / private ships
6. Sailing ships
7. Rescue ships

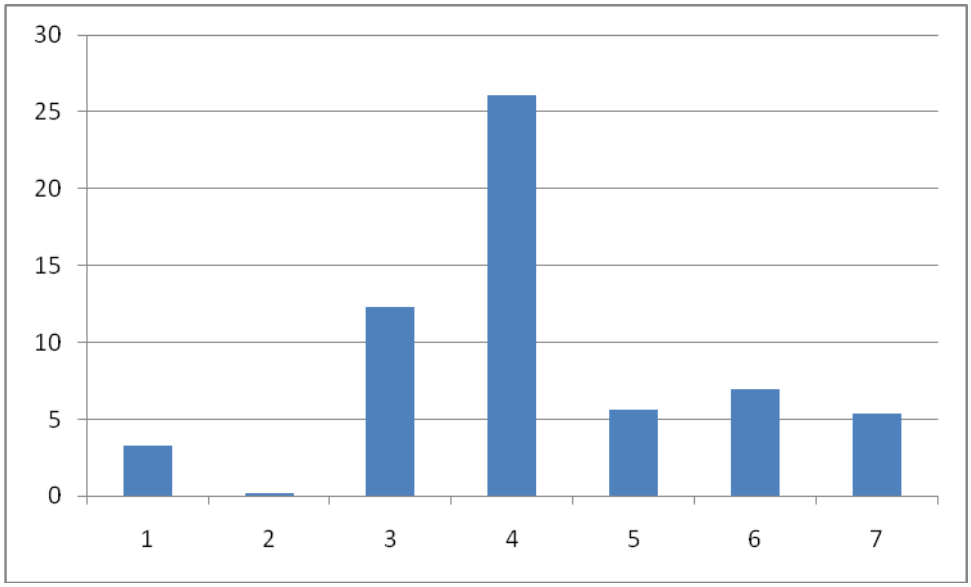
The Rate of Turn Sum Threshold value tested is displayed beneath each chart.



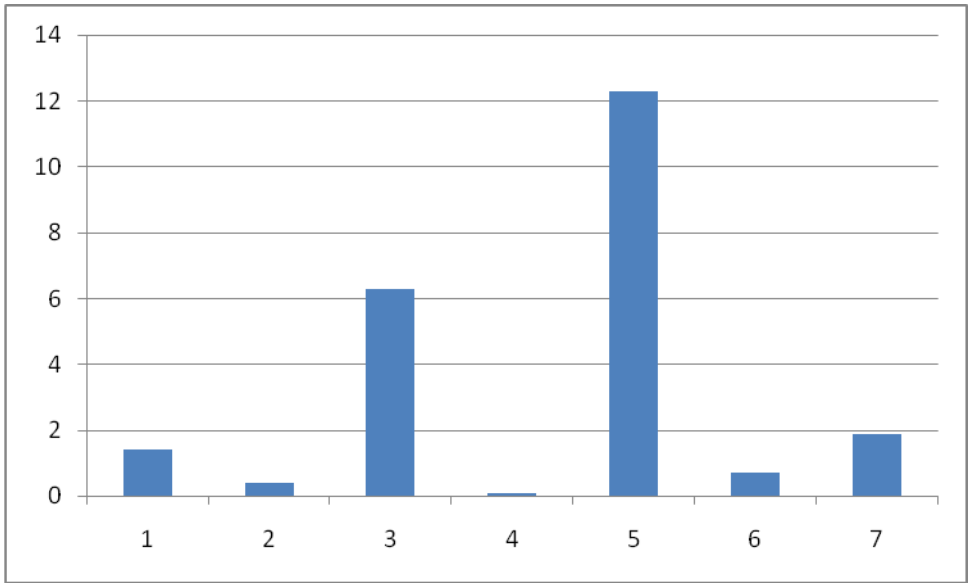
4-3 Rate of Turn Sum Threshold: 0.1



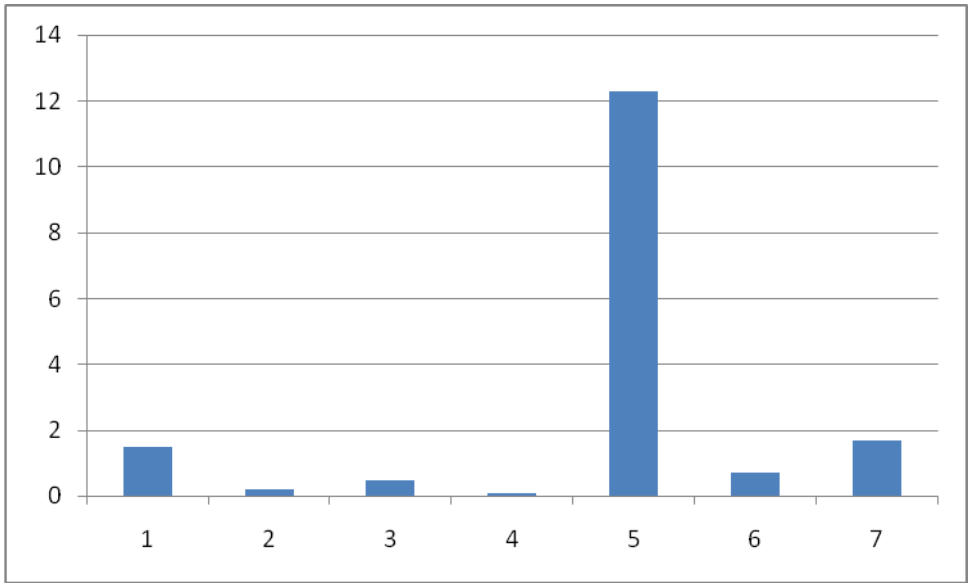
4-4 Rate of Turn Sum Threshold: 2.0



4-5 Rate of Turn Sum Threshold: 3.0



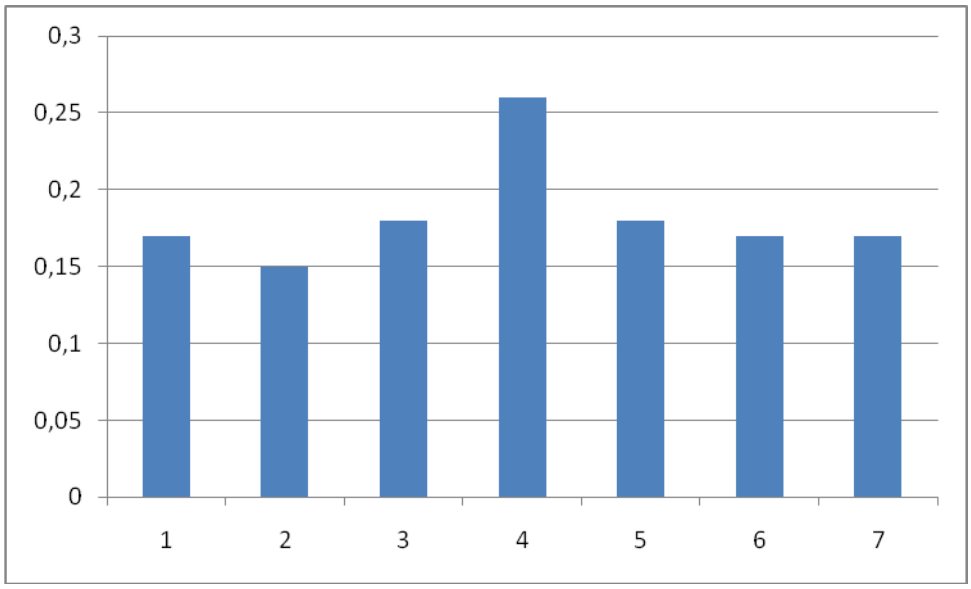
4-6 Rate of Turn Sum Threshold: 4.0



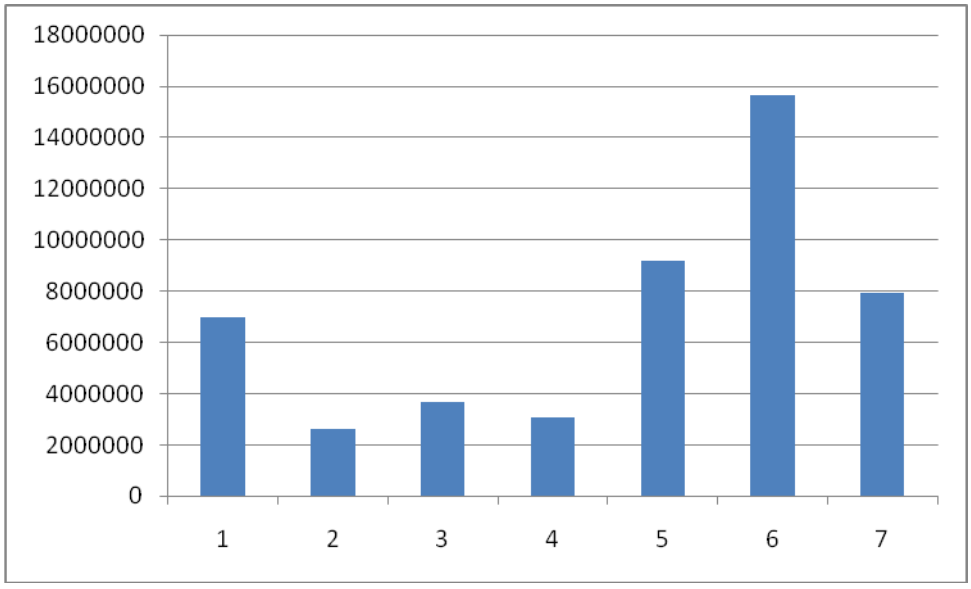
4-7 Rate of Turn Sum Threshold: 5.0

4.2.2 Distance Granule

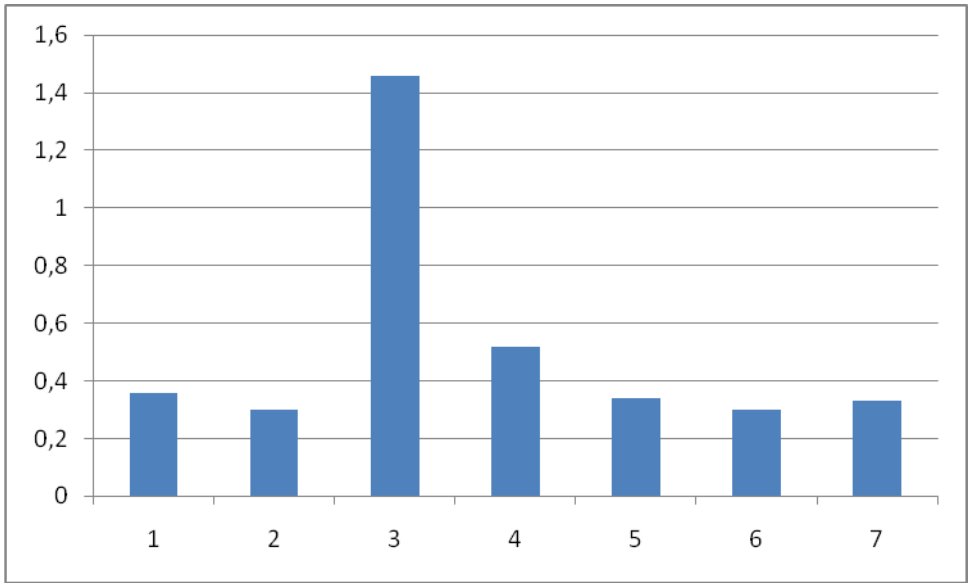
The DistanceGranule was tested on the mean distance travelled and mean time travelled per Demarcation, with the parameters set to the values shown to yield the most occurrences. The numbering of ship types is the same as above.



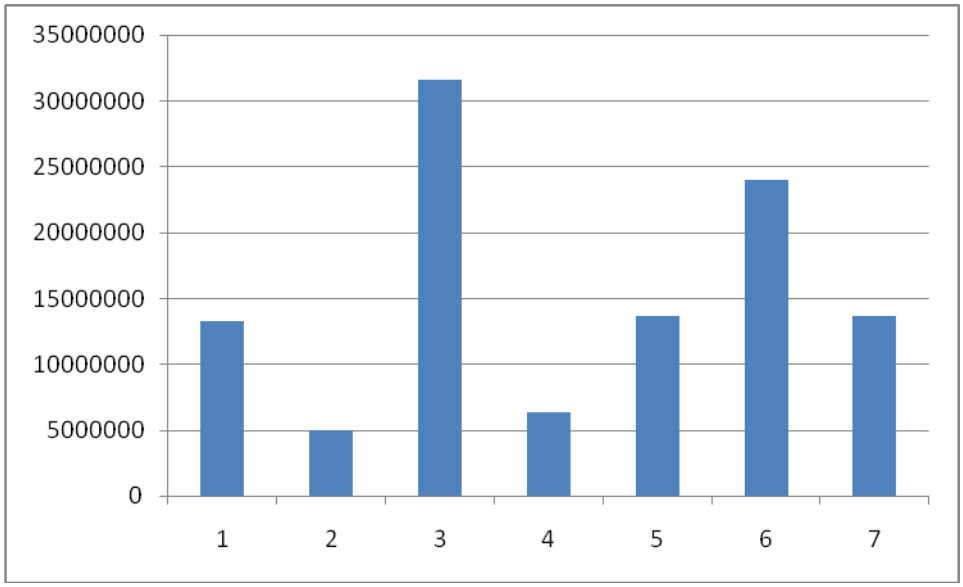
4-8 Mean distance travelled. Demarcation Radius: 0.15, Distance Threshold: 0.11



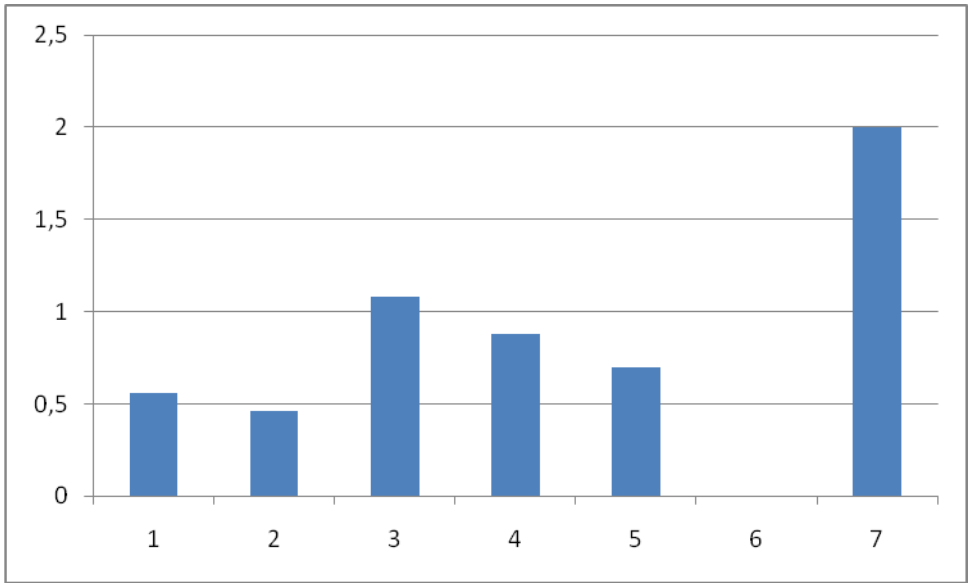
4-9 Mean time travelled. Demarcation Radius: 0.15, Distance Threshold: 0.11



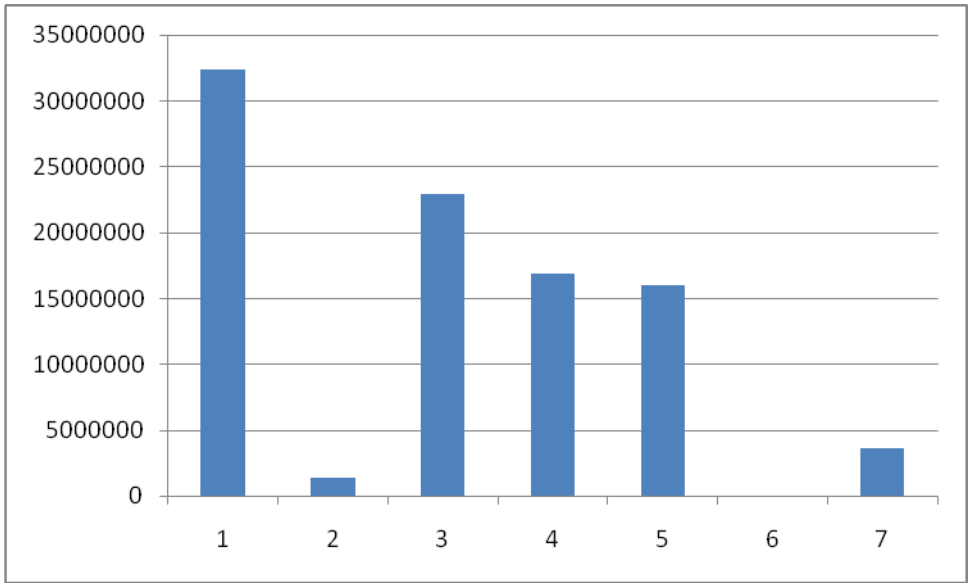
4-10 Mean distance travelled. Demarcation Radius: 0.3, Distance Threshold: 0.11



4-11 Mean time travelled. Demarcation Radius: 0.3, Distance Threshold: 0.11



4-12 Mean distance travelled. Demarcation Radius: 0.3, Distance Threshold: 0.4



4-13 Mean time travelled. Demarcation Radius: 0.3, Distance Threshold: 0.4

4.3 Measurement analysis

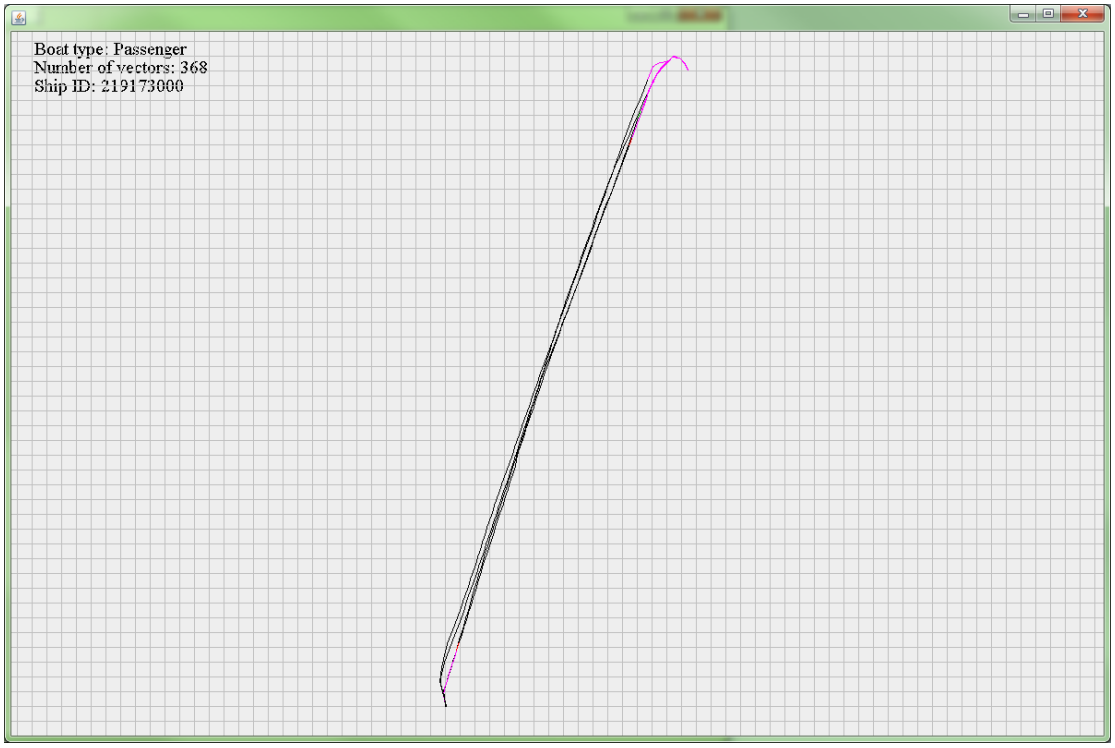
4.3.1 Rate of Turn Granule

What's most striking about the RoTGranule results is that not all optimal sensitivity values are zero. Since it represents the lowest value needed for a turn to count, the lowest possible value ought to be the most represented, but instead we find that for the smallest turns (e.g. threshold values) the sensitivity levels are quite high for certain ships. The higher the sensitivity, the more a big turn can be broken down into further,

smaller turns and thus increasing the occurrence of the granule for ship types that favor small, jerky movements over large, cumulative ones.

The cargo ship category has predictably the lowest sensitivity values on average, having the highest prevalence of long turns and not many small ones. The passenger ships are on the other hand characterised by their straight courses that terminate in an abrupt 180° turn, and therefore have the highest values in 2.0 and 3.0 RoTSum Threshold groups.

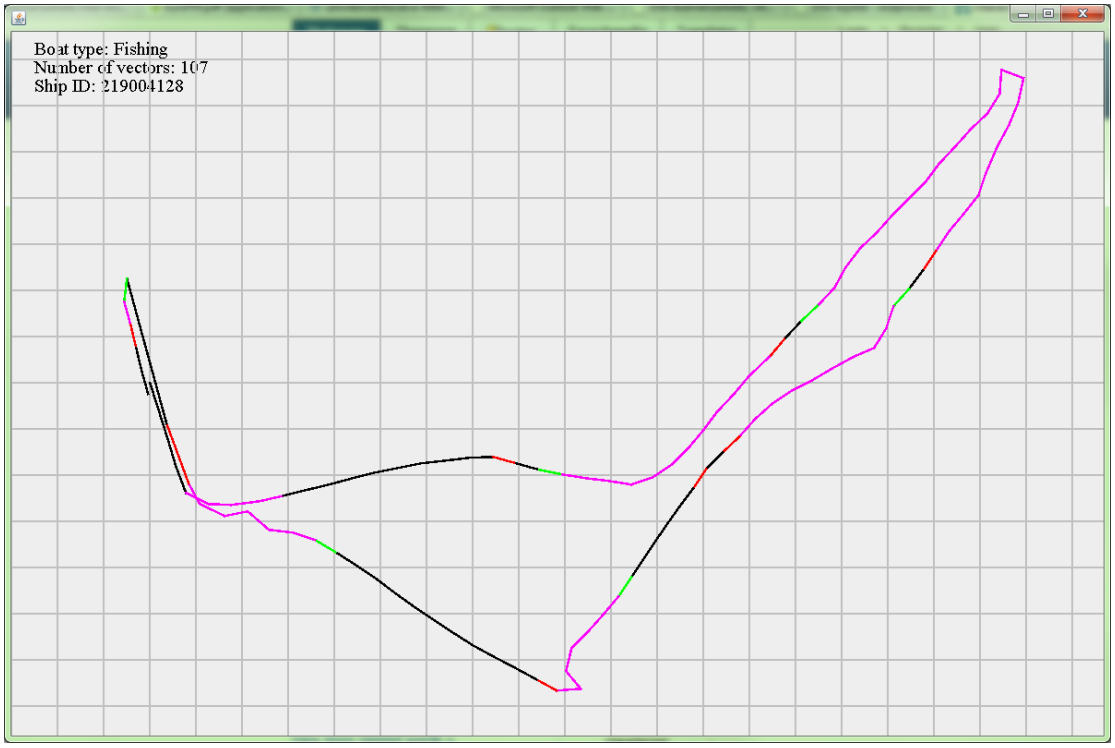
4-14 Course for a cargo ship with marked turns. The purple areas denote a turn, and the green and red denote the start and end of one respectively.



4-15 Course for a passenger ship. Note the purple end points, denoting very isolated turns

Fishing, pilot and rescue ships display varied behaviour, travelling long, straight stretches and then covering an area with smaller, sharper turns. This blend of high and low sensitivity behaviour can be seen in the initially high values that shrink steadily as the threshold is increased.

4-16 Course for a fishing ship with marked turns, sensitivity 9.5 and threshold 0.1. Stand alone red vectors indicate a very small turn



4-17 The same ship as 4-11, but with a sensitivity of 3.2. Note that the number of turns are fewer, but much longer on average.

Pleasure / private ships are by nature erratic and unpredictable. This category consists of any ship large enough to require an AIS transmitter, but without a specified commercial purpose. There is no consistent behaviour among these ships as they have no consistent agenda, but the higher speeds and greater rates of turn attainable by them in contrast to typically larger, bulkier ships like Cargo ships makes it possible to differentiate them in that respect.

RoTGranule Sensitivity 2.0 Threshold 0.2	**Pleasure Ships**	**Cargo Ships**
Mean size of turns	2.97	0.72
Mean distance travelled per turn	0.076	0.035
Mean distance travelled between turns	0.029	0.32
Mean duration of turns	3474720	903655

4-18 **Comparison of mean stats between Pleasure ships and Cargo ships. Note the wider, longer turns of the Pleasure ships, and the greater distance travelled between turns of the Cargo ships**

4.3.2 Distance Granule

As the Distance Granule finds instances where the ships have travelled a certain length in a small area it is not surprising to see that the Cargo ships once more have the lowest values. But a more interesting observation is that of the relative difference between the length and time of a Demarcation. In 4-12 and 4-13 we see a major difference in the rank of the Fishing ships' time, as opposed to its rank in distance, indicating that it spends longer time in small areas than the other ships, easily explainable by its trawling behaviour.

The Pilot ships exhibit a similar behaviour, also ranking very high on the time scales. As the pilot ships are area dependent like the Fishing ships, they linger in one area while performing their job.

5 Conclusions

5.1 Synopsis

In order to find unique characteristics of ship movements that could be described in an intuitive way, we created an application that takes point data from an AIS tracking system and converts them to less granular abstractions. The abstractions take the form of ShipVectors and Granules; algorithms that create vector data from the GPS signals and assigns each vector traits and parameters. These traits were then correlated and compared to find patterns with transparent parameters that could yield an explainable connection between the ship types and the traits. The output of the application did point towards several interesting identifiers, but it is still in the form of data that needs to be further interpreted by the user.

5.2 Discussion

The task of identifying ships based solely on limited information about their movements is problematic for a number of reasons. The most obvious one is that ships are controlled by people rather than predictable algorithms, and are subject to chaotic influences such as traffic, weather, bureaucracy, business or personal agendas.

As such, one must keep in mind that aberrations will be plentiful and unpredictable, and to consider the data in a larger perspective. Large sample groups are key in finding common trends, and the data used in this project was perhaps not sufficient for an in-depth analysis.

Abstracting the point data from intermittent reported GPS signals to a smooth, uniform description of a ship's course is itself an issue as well. One must be careful in selecting and discarding data, and choosing a suitable level of abstraction. Many entries in the initial database were worthless; some ships had kept reporting long after anchoring while others had left the covered area only to reappear days later, wreaking havoc on the time values of the vectors. Further ship entries were far too short or broken to be of any use and skewed the averages heavily. The majority of the work gone into this project was spent on gradually improving the accuracy of the abstraction by locating and analyzing these errors and incompatibilities, and rectifying them or changing the system to accomodate them.

In this process it is important to avoid, but hard to notice, confirmation bias. When correcting an error or changing one's approach to allow it in the system, it is possible to change the application into one that provides the desired output. As this project began with assumptions regarding ship movements and behaviours, there has always been the risk of designing solutions that would accomodate that kind of result, but discard others. There is no simple fix for this problem, and without the feature to automatically test the efficacy of the application on individual ships there are no quantifiable certainties. The output of the application must be considered in light of this.

Nevertheless, it has provided a general framework and certain indicators for characterising ship movements, such as the particulars for frequency, size and duration of turns, as per the Correlational model described in (Ellis, 2009 p. 327), where we attempt to find a predictive relationship between the various factors of ship movement and ship type.

5.3 Future work

Larger sample groups over longer times and larger areas could smooth out the errors in the abstraction, and provide more accurate statistics. More representation from the smaller ship groups could improve the comparisons inbetween the ships.

Several projects could be imagined with this work as their base. Further effort could be put into applying the data to individual ships and develop the granules to provide percentages and likelihood that a certain ship belongs to a certain type. This could be a stepping stone towards an application that detects these kinds of behaviours in real time for a number of ships in or around a port area or other interesting points of surveillance.

This particular brand of detection is possible to carry over to other types of vehicles, such as cars and even airplanes. Ship detection has the advantage of dealing with gradual, comparatively slow changes over time whereas cars and airplanes are capable of much more discrete manoeuvring and also contend with a third dimension, but these problems can be overcome by finer measurements and more frequent updates. In a similar vein, any 2D AI system could benefit from this approach, in games or other simulations, to find patterns. The traits that are investigated are inherent in any 2D movement and can find distinct patterns wherever a similar system is used.

The application lends itself well towards genetic algorithms, for finding optimal parameter inputs and interesting behaviour. A simple version of this was used in the final stages to test the granules, but was very time consuming and not powerful enough to find more than the most rudimentary combinations.

With more conclusive results, the data could be used to create a language for describing ship movements, using collections of several behaviours as single, syntactic units. This language could be used not only for identifying ship types in real life scenarios, but for guiding simulated ships in virtual environments. The same principles used to find notable characteristics could be used to define limitations and typical manoeuvres.

References

Adriaans P., Zantinge, D. (1996), *Data Mining*
Addison-Wesley Longman Ltd.

Campos, F.M., Elfes, A., (1998), *Introduction to the Special Issue on Robotics and Computer Vision*
Journal of the Brazilian Computer Society, vol. 4 (3)

Ekman, J., Holst, A., (2003), *Avvikelsedetektion av Fartygsrörelser*
Internal report Saab Systems, SICS by request of SaabTech

Ellis, T., Levy, Y., (2009), *Towards a Guide for Novice Researchers on Research Methodology: Review and Proposed Methods*
Issues in Informing Science and Information Technology, vol. 6, p. 323 - 337

Fayyad, U., Stolorz, P. (1997), *Data Mining and KDD: Promise and Challenges*,
Future Generation Computer Systems, vol. 13(2-3), p. 99-115

Hegland, M. (2003), *Data Mining - Challenges, Models, Methods and Algorithms*
Scientific Literature Digital Library, Accessed 08-02-2010 from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.57.9510

Höppner, F., Klawonn, F., Kruse, R. & Runkler, T. (1999), *Fuzzy Cluster Analysis*
John Wiley & Sons Ltd.

Jain, A.K., Duin, R.P.W., Mao, J., (1999a), *Statistical Pattern Recognition: A Review*
IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22 (1), p. 4-37

Jain, A.K., Murty, M.N., Flynn, P.J., (1999b), *Data Clustering: A Review*
ACM Computing Surveys (CSUR), vol. 31 (3), p. 264-323

Jesan, J.P., (2004), *The Neural Approach to Pattern Recognition*
ACM Ubiquity, vol. 5 (7)

Kang, H., Lee, C.W., Jung, K., (2004), *Recognition-based gesture spotting in video games*
Pattern Recognition Letters, vol. 25 (15), p. 1701-1714

Kodratoff, Y., (2001), *Comparing Machine Learning and Knowledge Discovery in Databases : An Application to Knowledge Discovery in Texts*
Springer-Verlag, Lecture Notes in Computer Science, vol. 2049, p. 1–21.

Olafsson, S. (2006), *Introduction to operations research and data mining*
Computers and Operations Research, vol. 33 (11), p. 3067-3069

Pal, S., Pal, A. (2001), *Pattern Recognition, From Classical to Modern Approaches*
World Scientific Publishing Co. Pte. Ltd.

Pedrycz, W. (2007), *Granular Computing - The Emerging Paradigm*
Journal of Uncertain Systems, vol. 1 (1), p.38-61

Ramakrishnan, N., Grama, A.Y. (1999), *Data Mining: From Serendipity to Science*
Computer, vol. 32 (8), p. 34-37

Saabgroup (2009), *IBD Internal Product Specification 1301*
Internal Product Specification at Saab Microwave Systems

Yao, Y. (2000), *Granular Computing: basic issues and possible solutions*
Proceedings of the 5th Joint Conference on Information Sciences, vol. 1, p. 186-189