

Criação de trajetórias de navios autônomos, em ambiente de simulação virtual, baseada no histórico de dados AIS

Antônio L. C. Doneda^a, Ricardo S. Bastos^a, Julio Cesar Duarte^b

^aCentro de Análise de Sistemas Navais

Ed. 23 do AMRJ - R. da Ponte, s/n, 20.091-000, Centro,
Rio de Janeiro - RJ - Brasil

^bInstituto Militar de Engenharia, Seção de Engenharia da Computação – SE/9

Praça General Tibúrcio, 80, 22290-270, Praia Vermelha,
Rio de Janeiro, RJ, Brasil

RESUMO: Simulações virtuais têm crescido muito em importância para treinamento de recursos humanos, mas é importante que representem a realidade com um grau satisfatório de fidelidade. Neste trabalho, aplicamos o método de clusterização DBSCAN modificado, em conjunto com pré e pós-processamentos de dados, para obter derrotas representativas do comportamento comum de navios em uma zona portuária, a partir de mensagens AIS, para utilizá-las no controle de NPC em uma simulação de treinamento de navegação da Marinha do Brasil. O algoritmo mostrou-se efetivo, gerando 19 diferentes trajetórias representativas de 5 tipos de navios selecionados, a partir de 76.179 mensagens AIS coletadas durante 48 horas.

PALAVRAS-CHAVE: Simulações Virtuais. Clusterização. DBSCAN. AIS. Trajetórias.

ABSTRACT: Virtual simulations have been growing in importance for manpower training, but it is important to portray reality with a satisfactory degree of fidelity. In this work, we applied a modified DBSCAN clusterization method, along with pre and post data processing, to obtain routes representative of ship's common behavior in a port zone from AIS messages, in order to use them to control non-playable characters in a Brazilian Navy's navigation training simulation. The algorithm was effective to generate 19 different routes depicting 5 selected types of ships, from 76.179 AIS messages collected for 48 hours.

KEYWORDS: Virtual Simulations. Clusterization. DBSCAN. AIS. Routes.

1. Introdução

O uso de simuladores para recriar experiências de ambientes reais auxilia o treinamento e reduz custos operacionais. Seu uso na Marinha do Brasil permite recriar as condições de navegação e manobra de um navio, bem como suas interações com o tráfego marítimo, promovendo uma redução de custos operacionais. Além disso, há uma preservação maior dos meios aliada a uma ampla gama de procedimentos que podem ser treinados, beneficiando o aprestamento da Força Naval, com grande redução de custos, de riscos e de desgaste dos equipamentos.

A Marinha do Brasil, no seu Centro de Análises de Sistemas Navais (CASNAV), desenvolveu um Simulador de Passadiço para atender às demandas de

treinamento de navegação do seu pessoal [1]. Neste simulador, o tráfego marítimo que interage com o usuário é controlado pelo instrutor, que deve definir trajetórias a serem seguidas pelos navios autônomos (NPC - “non-playable character” - personagem não-jogável). Cada NPC pode ser associado a uma trajetória predeterminada. Esta solução é interessante apenas para uma simulação específica. Para simular um navio entrando e saindo de um porto, o ideal é que o tráfego marítimo do ambiente simulado seja o mais parecido possível com a realidade.

Dessa forma, verifica-se certos padrões nas trajetórias de determinados navios (cargueiros, rebocadores, *offshores*, passageiros etc.). A criação manual das trajetórias para os exercícios exige do instrutor conhecimento acerca das vias navegáveis comuns de uma zona portuária (ZP) e de qual categoria

de navios as utilizam. Caso o instrutor não detenha este conhecimento, pode ser que crie exercícios que não condizem com a realidade neste aspecto.

Uma ferramenta que possa auxiliar na geração de trajetórias de determinados tipos de navios permite ao instrutor, por meio de comandos básicos, povoar o ambiente de simulação com NPCs que retratem as trajetórias encontradas no dia a dia o mais fidedignamente possível. Sendo assim, o instrutor pode focar nos procedimentos realizados pelo usuário, sem se preocupar em construir uma trajetória para cada NPC.

Este artigo tem como objetivo propor a utilização de uma técnica de Aprendizado de Máquina Não Supervisionada, o DBSCANSD (Clusterização Espacial Baseada em Densidade de Aplicações com Ruído considerando Velocidade e Direção) [2], para gerar trajetórias representativas da movimentação comum dos navios baseadas em um histórico de navegação de um determinado porto. Essas trajetórias seriam armazenadas no sistema de simulação e o instrutor apenas determinaria o tipo de navio, deixando por conta do sistema a determinação de uma trajetória.

Assim, seria possível simular o tráfego marítimo em qualquer porto do Brasil ou do exterior, bastando apenas realizar a coleta dos dados históricos de navegação, ampliando, portanto, o nível de adestramento dos usuários do simulador. Além disso, é de suma importância que as trajetórias geradas tenham as características das trajetórias observadas, a fim de gerar um maior realismo.

A seção 2 deste artigo descreve trabalhos relacionados de predição de trajetórias. Já a seção 3 apresenta como foram obtidos e tratados os dados históricos de movimentações de navios na Baía de Guanabara. Na seção 4, é descrita a utilização do método DBSCANSD para predição de trajetórias. A seção 5 fornece os resultados obtidos pela proposta apresentada, e por fim, na seção 6 expomos as considerações finais e conclusões.

2. Trabalhos relacionados

Há exemplos de trabalhos para predição de trajetórias de navios com diversas finalidades.

Todos têm como premissa a busca por prever possíveis posições futuras com base em trajetórias passadas obtidas por equipamentos AIS (*Automatic Identification System* - Sistema de Identificação Automático). Tais equipamentos fornecem, dentre outras informações, as posições em latitude e longitude de uma embarcação. Os dados históricos podem ser divididos em grupos (*clusters*), de forma a extrair e classificar as trajetórias em um modelo que possa representá-las.

Perera et al. [3] implementaram um algoritmo que utiliza Redes Neurais Artificiais (ANN) e Filtro de Kalman estendido para a predição de trajetórias, focando principalmente na detecção e acompanhamento de navios.

Vries e Someren [4] definiram um método de Aprendizado de Máquina baseado em kernel para agrupar e classificar tráfego marítimo e detectar anomalias de comportamento. Inicialmente, os dados AIS são comprimidos em trajetórias lineares utilizando operações geométricas. Em seguida, comparam-se as similaridades das trajetórias agrupando-as com *kernel k-means* e classificando-as com *Support Vector Machines* (SVMs).

Duca et al. [5] utilizaram um algoritmo baseado em um classificador *K-Nearest Neighbor* (K-NN) para prever a posição de um navio após 30, 45 e 60 minutos. Esse algoritmo recebe os dados correntes do navio como entrada e retorna uma matriz de probabilidades para uma posição futura em uma grade pré-estabelecida. Este trabalho atingiu uma precisão de 79.4%, abrangência de 78.5% e acurácia de 93.1%.

Pallotta et al. [6] utilizaram o algoritmo *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN), primeiramente proposto por Ester et al. [7], para agrupar pontos de posição baseados nos atributos dos dados AIS. Dado o estado corrente do navio, uma região circular é inserida em torno de sua posição. Todas as rotas que passam por esse círculo são consideradas compatíveis com o navio. Uma vez que as posições são agrupadas, predições de trajetórias podem ser feitas baseadas nesse histórico.

O trabalho de Liu et al. [2] é fruto de uma abordagem da técnica DBSCAN em conjunto com

a utilização dos atributos não espaciais como a Velocidade e Direção dos navios, a fim de promover uma comparação das trajetórias mais utilizadas com as previstas nas regras e regulamentos de cada porto.

Fica evidente a preocupação desses trabalhos em prever onde o navio estará com a finalidade de acompanhamento dos meios navais, de detecção de anomalias e de busca e salvamento.

Diferentemente de [2], o presente trabalho utiliza pré e pós-processamento dos dados, a partir das características da ZP e dos tipos de navios que compõem seu tráfego marítimo, com a finalidade de obter trajetórias realistas para serem inseridas em um ambiente de simulação e treinamento.

3. Base de dados utilizada

Os dados utilizados pelo algoritmo são as informações dinâmicas e estáticas emitidas pelos navios por meio do equipamento AIS. Este equipamento é um transponder que transmite, em frequência VHF, uma série de informações, tais como: latitude, longitude, velocidade, direção, registro, nome e data-hora de coleta destas informações.

O trabalho aborda a região da Baía de Guanabara, no Estado do Rio de Janeiro, que abriga o porto do Rio de Janeiro, terminais de óleo e gás, e estações de barcas de passageiros que perfazem o trajeto Rio de Janeiro x Niterói. Delimitou-se uma área de interesse para a aquisição das informações limitada pelos pontos $22^{\circ}45.0'S$ e $043^{\circ}14.0'W$, $22^{\circ}45.0'S$ e $043^{\circ}05.0'W$, $23^{\circ}0.0'S$ e $043^{\circ}14.0'W$, $23^{\circ}0.0'S$ e $043^{\circ}05.0'W$, equivalente à carta náutica 1501, produzida pelo Centro de Hidrografia da Marinha.

Em seguida, foram obtidos os dados AIS de todos os navios naquela área, referentes ao período de 14 a 16 de agosto de 2019, totalizando 111.015 mensagens AIS. Posições em que o navio estava com uma velocidade abaixo de 0,5 nós foram retiradas, a fim de se utilizar apenas os navios em real movimento, permanecendo, após a remoção, um total de 76.179 mensagens.

Esses dados foram processados a partir das mensagens originais em formato NMEA-183, extraindo-se as seguintes informações de interesse:

identificação do navio (MMSI), latitude, longitude, velocidade, direção (rumo) e data-hora.

A Figura 1 apresenta os dados obtidos na região da Baía de Guanabara, plotados na carta náutica 1501.



Fig. 1 – Posições de todos os navios.

4. Utilizando o DBSCAN para a predição de trajetórias

DBSCAN é a abreviação do termo 'Density Based Spatial Clustering of Application with Noise' (Clusterização Espacial Baseada em Densidade de Aplicações com Ruído) e consiste em um método de clusterização não paramétrico baseado em densidade [7]. Este método é efetivo para identificar *clusters* de formato arbitrário e de diferentes tamanhos, identificar e separar os ruídos dos dados e detectar *clusters* "naturais" e seus arranjos dentro do espaço de dados, sem qualquer informação preliminar sobre os grupos.

A noção de clusters e o algoritmo DBSCAN se aplicam para espaços Euclidianos de duas e três dimensões, como para qualquer espaço característico de alta dimensão [7]. A ideia do método DBSCAN é que, para cada ponto de um *cluster*, a vizinhança

para um dado raio contém, no mínimo, certo número de pontos.

4.1 Definições

ϵ -vizinhança de um ponto p ($N_\epsilon(p)$): A vizinhança de um objeto p com raio ϵ é dada por $N_\epsilon(p) = \{q \in \mathbf{D} \mid \text{dist}(p, q) < \epsilon\}$. Na **Figura 2**, os círculos representam, respectivamente, a ϵ -vizinhança do ponto q e ϵ -vizinhança do ponto p [8].

Ponto Central: Se a ϵ -vizinhança de um objeto p contém ao menos um número mínimo ($MinPts$) de objetos, então o objeto p é chamado de ponto central.

Pontos de borda: Se a ϵ -vizinhança de um objeto p contém menos que $MinPts$, mas contém algum ponto central, então o objeto p é chamado de ponto de borda.

Alcance Direto por Densidade: Um objeto p é alcançável por densidade diretamente do objeto q , com respeito aos valores ϵ e $MinPts$, se p está na ϵ -vizinhança de q e q é um ponto central.

Conexão por densidade: Um objeto p é conectado por densidade ao objeto q com respeito à ϵ e $MinPts$ em um conjunto de objetos \mathbf{D} , se existe um objeto o em \mathbf{D} , tal que ambos, p e q , são alcançáveis por densidade do objeto o com respeito à ϵ e $MinPts$.

Cluster DBSCAN: É o conjunto de pontos conectados por densidade.

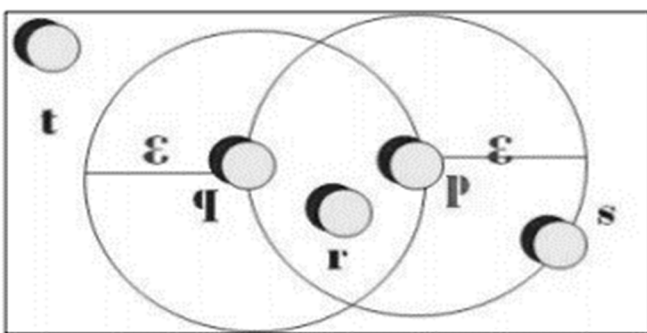


Fig. 2 – ϵ -vizinhança de um ponto p . Fonte: [8].

4.2 Método

Ainda segundo [8], o método DBSCAN encontra *clusters* verificando a vizinhança de cada ponto na base de dados, começando por um objeto arbitrário p . Se p é um ponto central, um novo *cluster* com p como

um centro é criado. Se p é um ponto de fronteira, nenhum ponto é alcançável por densidade a partir de p e o algoritmo visita o próximo ponto na base. O método DBSCAN, então, iterativamente coleta objetos alcançáveis por densidade diretamente de pontos centrais, que podem envolver a união de alguns *clusters* alcançáveis por densidade. O processo termina quando nenhum novo ponto pode ser adicionado a qualquer *cluster*.

Para o algoritmo DBSCAN, quaisquer dois pontos centrais com distância menor ou igual a ϵ são colocados no mesmo *cluster*. Qualquer ponto de borda que está perto de um ponto central é colocado no mesmo *cluster* do ponto central. Pontos que não são diretamente atingíveis por algum ponto central são classificados como ruído.

Liu et al. [2], além da ideia básica do DBSCAN, adotaram dois fatores, a variação máxima de velocidade ($MaxSpd$) e a variação máxima de direção ($MaxDir$). O intuito foi abranger não só os vizinhos por proximidade, mas também por similaridade de direção (*Course Over Ground*, COG) e de velocidade (*Speed Over Ground*, SOG), criando, portanto, o DBSCAN-D.

Com isso, modificaram a definição de ϵ -vizinhança para: A vizinhança de um objeto p com raio ϵ é dada por $N_\epsilon(p) = \{q \in \mathbf{D} \mid \text{dist}(p, q) < \epsilon, |p.SOG - q.SOG| < MaxSpd \text{ e } |p.COG - q.COG| < MaxDir\}$. Considera-se $\text{dist}(p, q)$ como a distância geográfica entre p e q , considerando estarem em um círculo máximo de raio aproximado ao do planeta Terra.

Após a aplicação do DBSCAN-D aos dados coletados, posições geográficas próximas com velocidade e rumo similares são agrupadas em um *cluster*. Então, formatos arbitrários de *clusters* podem ser formados, bem como os *clusters* podem ser subdivididos em função das velocidades e rumos.

Aplicou-se também um Vetor de Gravidade (*Gravity Vector*, GV), particionando um *cluster* em múltiplas partes. Com isso, cada *cluster* pode ter múltiplos GVs. O GV é um vetor formado por: COG médio, SOG médio, latitude média, longitude média e distância média [2].

A escolha, portanto, deste método levou em consideração sua robustez, a apresentação de melhores

resultados e processamento mais rápido que alguns outros algoritmos de clusterização, tal como o método *k-means*.

No entanto, a aplicação isolada do método não se mostrou suficiente para obter as trajetórias representativas de uma zona portuária, produzindo também trajetórias incomuns ou impraticáveis. Foi essencial o desenvolvimento e emprego de pós-processamentos, a fim de otimizar o resultado final, eliminando trajetórias ineficazes e otimizando as trajetórias obtidas.

5. Aplicação e resultados obtidos

O algoritmo requer 5 parâmetros como entrada: *DatasetM* (número de pontos de uma trajetória), *Épsilon* (ϵ -vizinhança), *MinPts*, *MaxDir* (máxima variação de direção) e *MaxSpd* (máxima variação de velocidade). A complexidade desse algoritmo é $O(n^2)$, onde n é o tamanho de *DatasetM* [2].

Os valores dos parâmetros dependerão das características das vias navegáveis encontradas em cada ZP [2]. De fato, observamos que a ZP do Rio de Janeiro tem algumas particularidades capazes de influenciar o resultado, como, por exemplo, a existência de um canal principal orientado no sentido norte – sul, em que praticamente todos os tipos de navio trafegam em algum momento de sua demanda, ao mesmo tempo em que há várias vias transversais utilizadas em maior ou menor escala por navios de diferentes tipos e finalidades.

Em virtude do baixo tempo médio de processamento do algoritmo, foi possível efetuar diversas tentativas, a fim de determinar a melhor combinação desses parâmetros. A partir da experimentação, observamos que, além das características inerentes à conformação das vias navegáveis da ZP, a melhor combinação dos valores desses parâmetros é também dependente do comportamento típico dos navios-alvo, quanto à sua variabilidade de rumos assumidos, vias mais comuns de navegação, velocidades assumidas e dispersão na área geográfica abrangida.

À vista dos resultados obtidos, atentamos para a possibilidade de se aprimorar a discriminação das

trajetórias comuns a partir da aplicação de filtro prévio nos dados de entrada quanto ao tipo de navio para aplicação isolada do algoritmo a cada um desses tipos. Esta escolha justifica-se pelo fato de que navios de uma mesma categoria tendem a ter origens, destinos e vias de circulação similares, enquadrando-se em uma faixa similar de rumos assumidos, além de geralmente observarem uma mesma faixa de velocidades.

Resolvemos, portanto, gerar modelos para cada uma das cinco categorias de navios definidas, possibilitando ao algoritmo explorá-las separadamente: Barca (passageiros), Cargueiro (contêiner e graneleiro), *Offshore* (navio de apoio a plataformas), Navio-Tanque (óleo e gás), e Rebocador.

A Figura 3 mostra a quantidade de mensagens AIS por tipo de navio. Observa-se uma grande quantidade de mensagens de barcas, que realizam o transporte de passageiros, pois as barcas possuem intervalos regulares e constantes entre as viagens. Em seguida, há um número considerável de mensagens de navios *Offshore*, por conta da proximidade do porto do Rio de Janeiro com a Bacia Petrolífera de Campos.

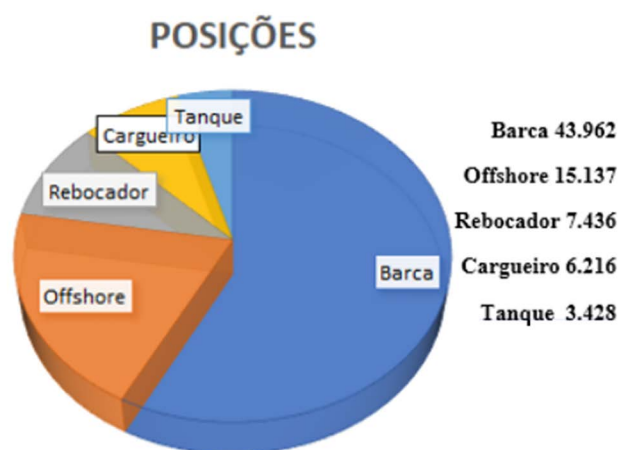


Fig. 3 – Divisão dos dados de entrada nas cinco categorias de navios.

A Tabela 1 demonstra a combinação de parâmetros adotada para cada categoria processada. A seleção dos parâmetros iniciou-se pelo arbitramento de um valor de *Épsilon*, baseado no valor aproximado da

largura das vias navegáveis consideradas. Reparámos que, no caso específico das Barcas, foi necessário reduzir este raio, posto que o cruzamento de diversas trajetórias favorecia a massificação de trajetórias diferentes em um mesmo *cluster*. Nesta Zona Portuária escolhida, há grandes variações de rumo nas trajetórias típicas dos navios que entram ou saem dos diversos terminais portuários. A escolha de valores baixos de *MaxDir* ocasionaria a partição de uma mesma trajetória em diversos *clusters*, razão pela qual optou-se por um valor alto deste parâmetro, que foi repetido em todos os tipos de navios. Os valores possíveis de *MaxSpd* foram limitados a uma faixa de valores factíveis para cada tipo de navio escolhido e definidos experimentalmente em combinação com valores de *MinPts*, chegando-se às combinações de parâmetros apresentadas na **Tabela 1**.

Tab. 1 - Seleção de Parâmetros.

Categoria	DatasetM (mensagens)	Épsilon (Milhas Náuticas)	MinPts (posições)	MaxDir (graus)	MaxSpd (nós)
Barca	43.962	0.001	2	20.0°	15.0
Cargueiro	6.216	0.003	6	20.0°	15.0
Offshore	15.137	0.003	6	20.0°	8.0
Rebocador	7.436	0.003	4	20.0°	10.0
Navio-Tanque	3.428	0.003	6	20.0°	15.0

5.1 Pós-processamento dos resultados

A abordagem de processamento por categorias melhorou a discriminação das trajetórias mais comuns, havendo, no entanto, algumas discrepâncias que poderiam ainda ser resolvidas, a fim de otimizar os resultados.

Observou-se, inicialmente, a geração de *clusters* com poucas posições. As trajetórias com apenas uma posição foram automaticamente excluídas. Naquelas compostas por duas a quatro posições, foi estabelecido o critério da distância percorrida pela trajetória, eliminando automaticamente as que percorriam menos de 100 jardas por posição. A **Tabela 2** especifica as trajetórias processadas.

Tab. 2 - Trajetórias geradas, eliminadas e mantidas.

Categoria	(A)	(B)	(C)	(D)	(E)
Barca	76	36	28	1	4
Cargueiro	23	2	4	0	3
Offshore	56	21	8	4	4
Rebocador	29	8	10	2	5
Navio-Tanque	30	3	7	0	3

Nota: (A) – Total de trajetórias geradas; (B) – trajetórias com apenas uma posição; (C) – trajetórias com menos de quatro posições; (D) – trajetórias sobre terra; (E) – Trajetórias mantidas.

Outro problema observado foi a ocorrência de trajetórias passando sobre pontos de terra ou áreas não-navegáveis. A fim de detectar automaticamente estas ocorrências, e eliminá-las previamente, foi criada uma imagem da carta náutica, conforme mostra a **Figura 4**, em que as áreas de terra e não-navegáveis foram coloridas em vermelho no padrão **RGB** = {255, 0, 0}. As latitudes e longitudes dos pontos das trajetórias foram, então, mapeados para corresponder às posições dos *pixels* da imagem gerada. Para cada ponto de uma trajetória checa-se o *pixel* equivalente e, caso o valor de **R** seja igual a 255, aquela trajetória inteira é desprezada.

Para aplicação no Simulador de Passadiço desenvolvido pelo CASNAV, não é interessante que se tenham pontos muito próximos representando uma trajetória, uma vez que os NPC são programados para sempre buscar o próximo ponto da trajetória, e ao alcançar este, corrigir o rumo e a velocidade para buscar o que se segue. Trajetórias com pontos muito próximos causariam comportamentos erráticos nos NPC, diminuindo a fidelidade do seu comportamento à realidade.

A fim de solucionar este problema, otimizando a trajetória pela diminuição de pontos colineares ou com pouco afastamento do trajeto, foi aplicado o algoritmo de Douglas-Peucker [9].

Este algoritmo percorre os pontos componentes de uma trajetória, avaliando-os três a três, e eliminando os pontos intermediários que estão dentro de um valor pré-determinado de afastamento do segmento de reta formado pelos pontos extremos.

Ao aplicar este algoritmo, pode-se criar segmentos de trajetórias que passam sobre pontos de terra ou não navegáveis. O valor máximo de afastamento para que um ponto seja eliminado deve ser ajustado para evitar que isso aconteça. Nos experimentos relatados, foram adotados valores menores que a metade da largura do canal principal da Baía de Guanabara, ou seja, aproximadamente 100 jardas.

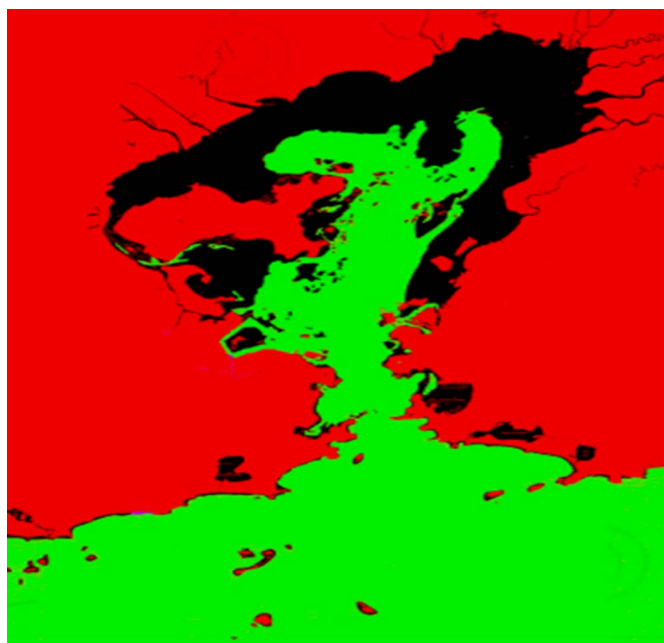


Fig. 4 - Imagem gerada para detectar pontos de trajetória sobre terra ou áreas não-navegáveis, representadas em vermelho, e áreas navegáveis, em verde.

Após todo esse pós-processamento automático, como há de se notar pela análise da **Tabela 2**, restam ainda muitas trajetórias. A determinação de critérios objetivos para avaliação dos resultados obtidos não é simples. A melhor análise parece ser a avaliação subjetiva das trajetórias obtidas por especialistas que conheçam as características do tráfego marítimo na região em pauta, que foi o último critério utilizado para se chegar à quantidade final de trajetórias constante da **Tabela 2**.

Valores mais altos de ϵ e mais baixos de $MinPts$ diminuem a discriminação entre trajetórias, causando a união de trajetórias diferentes. No entanto, ao se evitar esta junção de trajetórias

diferentes pelo ajuste dos parâmetros, surge a tendência de separar trechos de uma mesma trajetória em diferentes clusters.

Ainda com o intuito de obter critérios objetivos, foi calculado para cada trajetória o desvio padrão e o coeficiente de variação dos rumos e das velocidades, buscando obter tendências que ajudassem a escolher as trajetórias de maior qualidade. Nota-se que, para alguns casos, as melhores trajetórias eram aquelas com menores variâncias, enquanto, para outras categorias, eram aquelas com os valores mais altos, denotando não ser um bom critério de escolha.

As **Figuras 5 a 9** mostram as trajetórias obtidas antes e após o pós-processamento.

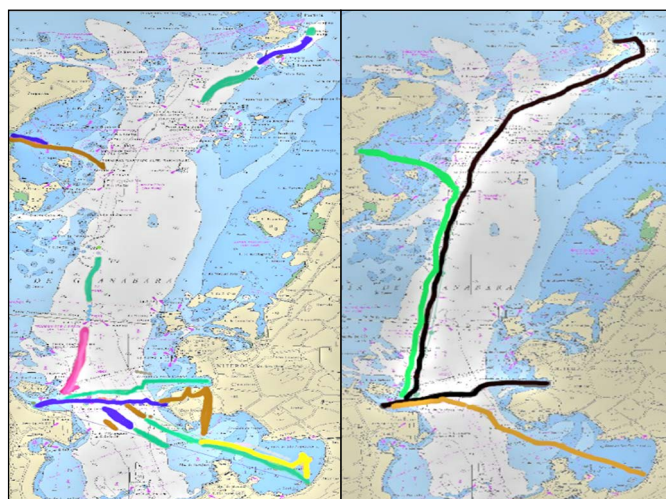


Fig. 5 - Trajetórias geradas para Barcas. À esquerda antes do pós-processamento e à direita após o pós-processamento.

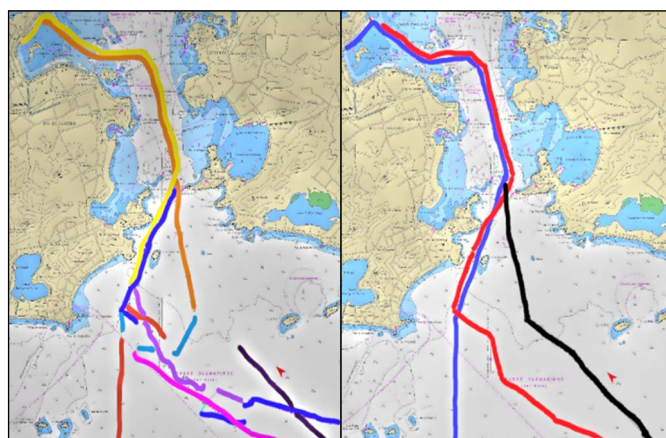


Fig. 6 - Trajetórias geradas para Cargueiros. À esquerda antes do pós-processamento e à direita após o pós-processamento.

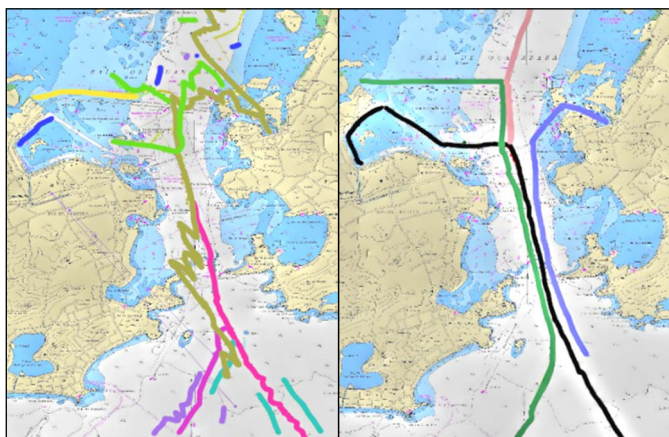


Fig. 7 - Trajetórias geradas para navios Offshore. À esquerda antes do pós processamento e à direita após o pós-processamento.

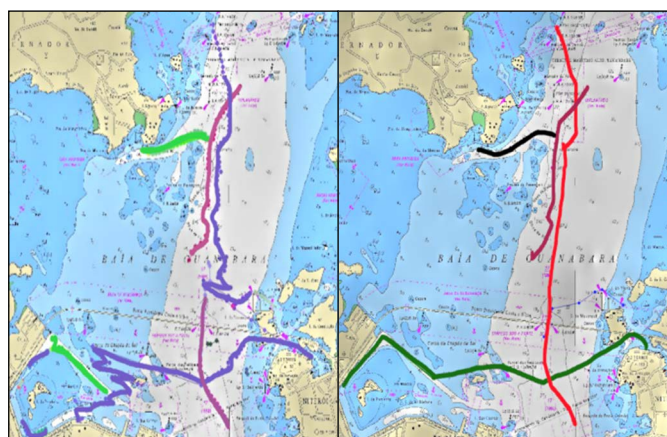


Fig. 8 - Trajetórias geradas para Rebocadores. À esquerda antes do pós processamento e à direita após o pós-processamento.

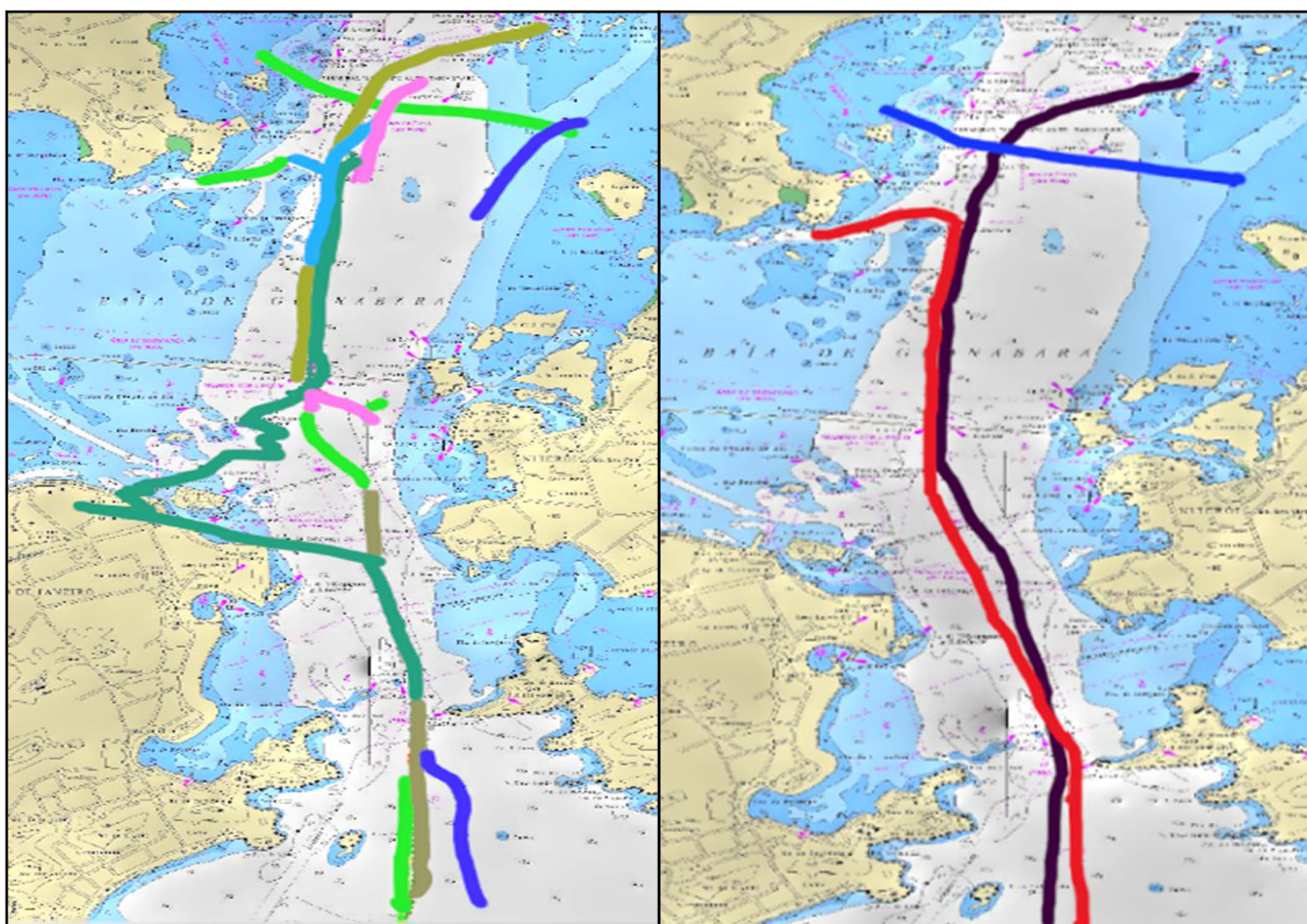


Fig. 9 - Trajetórias geradas para Navios-Tanque. À esquerda antes do pós processamento e à direita após o pós-processamento.

6. Considerações Finais

O objetivo deste trabalho foi o de obter, a partir de uma grande massa de dados de mensagens AIS, trajetórias representativas da movimentação comum dos navios que frequentam uma ZP, e utilizar as trajetórias obtidas para controlar navios NPC no Simulador de Passadiço desenvolvido pelo CASNAV.

Utilizando o algoritmo proposto no trabalho de Liu et al. [2], em conjunto com pré-processamentos e pós-processamentos dos dados, foi possível obter 19 trajetórias representativas de 5 categorias de navios

selecionados, sendo os resultados obtidos considerados muito satisfatórios e aplicáveis aos exercícios de simulação conduzidos no âmbito da Marinha do Brasil.

Em trabalhos futuros, pode-se explorar a conjugação deste método com o TraClus [10] e se avaliar os resultados. Há grande espaço para os aprimoramentos dos métodos de pré e pós-processamento dos dados, a fim de otimizar os resultados obtidos. Seria interessante estudar futuramente formas de união automática de trechos de uma mesma trajetória separados em diferentes *clusters* pela discriminação do algoritmo.

Referências

- [1] LAGE, M., CLUA, E., BARBOZA, D., et al. Simulador de Passadiço. **XI Simpósio Brasileiro de Jogos e Entretenimento Digital – SBGames**, 5, 2012.
- [2] LIU, Bo; DE SOUZA, ERICO N.; MATWIN, Stan, et al. Knowledge-based clustering of ship trajectories using density-based approach. **Proceedings - IEEE International Conference on Big Data**, 603, 2014.
- [3] PERERA, L. P., OLIVEIRA, P.; GUEDES SOARES, C. Maritime Traffic Monitoring Based on Vessel Detection, Tracking, State Estimation, and Trajectory Prediction. **IEEE Transactions on Intelligent Transportation Systems**, 13, 1188, 2012.
- [4] VRIES, G. K. D. De and Someren, M.; Machine learning for vessel trajectories using compression, alignments and domain knowledge. **Expert Systems with Applications**. 2012, 13.426, 2012.
- [5] DUCA, A. Lo, BACCIU, C. and MARCHETTI, A.; A K-nearest neighbor classifier for ship route prediction. **OCEANS – Aberdeen**, October, 1, 2017.
- [6] PALLOTTA, G., VESPE, M.; BRYAN, K.; Vessel Pattern Knowledge Discovery from AIS Data: A Framework for Anomaly Detection and Route Prediction. **Entropy**, 15, 2218, 2013.
- [7] ESTER, M., KRIEGEL, H.-P., SANDER, J.; XU, X. Density-based spatial clustering of applications with noise. **Int. Conf. Knowledge Discovery and Data Mining**, 240, 6, 1996.
- [8] MARA CASSIANO, K.; **Análise De Séries Temporais Usando Análise Espectral Singular (SSA) E Clusterização De Suas Componentes Baseada Em Densidade**, Tese (Doutorado em Engenharia Elétrica), Pontifícia Universidade Católica do Rio de Janeiro, Brasil, 2014.
- [9] PEUCKER, T.; DOUGLAS, D. H. Reflection Essay: Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature Classics in Cartography: **Reflections on Influential Articles from Cartographica**, 29, 2011.
- [10] LEE, J., HAN, J. and WHANG, K.-Y. Trajectory Clustering: A Partition-and-Group Framework. **Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data – ACM**, 593, 2007.