

# ACOMPANHAMENTO AUTOMÁTICO POR CLUSTERIZAÇÃO

FÁBIO OLIVEIRA BAPTISTA DA SILVA\*, FABRÍCIO DE ABREU BOZZI\*, FERNANDO DE SOUZA PEREIRA MONTEIRO\*

\**Rua Ipiru, n 02, Cacuia, Ilha do Governador, Rio de Janeiro*

Emails: `fabio.oliveira@marinha.mil.br`, `bozzi@ipqm.mar.mil.br`,  
`fernando.monteiro@marinha.mil.br`

**Resumo**— Este trabalho propõe um algoritmo de clusterização *online* em dois níveis para detecção e acompanhamento de contatos em um sistema sonar passivo. O algoritmo, após pré-processado o dado bruto do sonar, se divide em duas etapas de clusterização: uma no espaço das *features* e outra temporal, denominada de *tracking*. Genericamente, a primeira etapa de clusterização do algoritmo foi desenvolvida para clusterização de *features* onde a expressão de um *cluster* tenha em uma dada *feature* a característica de um máximo local que representa o *cluster*. Enquanto a segunda etapa, *tracking*, acompanha a evolução temporal dos *clusters* da primeira etapa. As características utilizadas para clusterização no algoritmo desenvolvido foram modeladas por conhecimento especialista para a aplicação neste cenário específico. Para teste do algoritmo proposto foram considerados dados simulados seguindo o modelo proposto, enquanto, para validação, foram utilizados os dados disponibilizados para realização do Encontro de Tecnologias em Acústica Submarina 2018 (XIII - ETAS).

**Palavras-chave**— sonar passivo, cluterização, cluterização online, acompanhamento, algoritmo

**Abstract**— This work proposes a two-level online clustering algorithm for detection and tracking of contacts in a passive sonar system. The algorithm, after preprocessing the raw sonar data, it divides into two stages of clustering: one in the space of features and the other in time, called tracking. Generally, the first step of clustering the algorithm was developed for feature clustering where the expression of a cluster has in a given feature the characteristic of a local maximum representing the cluster. While the second stage, tracking, accompanies the temporal evolution of the clusters of the first stage. The characteristics used for clustering in the developed algorithm were modeled by knowledge application in this specific scenario. In order to test the proposed algorithm, simulated using the proposed model, while, for validation, the data available for the Encontro de Tecnologias em Acústica Submarina 2018 (XIII - ETAS).

**Keywords**— passive sonar, clustering, online clustering, tracking, algorithm

## 1 Introdução

Sonares passivos são utilizados, a muitos anos, como sistemas robustos e confiáveis para detectar e acompanhar a presença de outros navios, principalmente por submarinos. Tipicamente a detecção e o acompanhamento de contatos é feita por operadores humanos, experientes em distinguir o ruído e o sinal do meio acústico submarino. Uma ferramenta gráfica para auxílio desta função é o gráfico de energia *waterfall*. Neste gráfico, é visualizada a intensidade de energia por marcação ao longo do tempo, sendo a energia representada em uma escala de cores. Neste cenário, este trabalho propõe o desenvolvimento de um algoritmo de detecção automática de contatos para auxílio do operador sonar.

## 2 Proposta de Algoritmo

O algoritmo proposto trata de clusterização para acompanhamento sobre o gráfico de energia *waterfall*, a extração de detecções do gráfico de energia assim como a descrição das etapas de processamento será descrito na seção 2.1.

As detecções, recebidas em pequenos conjuntos no ritmo de atualização do gráfico de *waterfall*, serão clusterizadas seguindo uma heurística de busca de picos de energia dispersos em marcação, conforme abordará a seção 2.2.

Os *clusters*, que serão discutidos na seção 2.3, terão sua evolução acompanhada temporalmente por outra etapa de clusterização online, que para facilitar a nomenclatura, será chamada, a partir de agora, de acompanhamento, ou *tracking*.

### 2.1 Condicionamento

Os dados reais utilizados neste trabalho foram adquiridos por um arranjo, aproximadamente, linear de 32 *staves* e em blocos de 4096 amostras para cada canal.

O primeiro passo para cálculo da energia, exibida em um gráfico *waterfall*, consiste na conformação de feixe (*beamforming*). O método utilizado neste estudo é o *delay and sum* no domínio do tempo, que

basicamente atrasam sinais e somam dado uma direção de interesse. Desta forma, sinais correlacionados tendem a ser amplificados e sinais descorrelacionados tendem a ser atenuados.

Em seguida, o sinal é filtrado em uma banda de interesse, para este trabalho foi utilizada uma banda de 800 Hz a 8.5 kHz. A energia de um bloco é calculada como o valor absoluto deste sinal filtrado. Para estimar o nível de ruído de fundo em cada direção é utilizado um TPSW (*Two-Pass Split Window*) e o sinal é recalculado como uma razão deste ruído estimado, passando a representar quantas vezes o sinal está acima do nível de ruído.

Para reduzir o encobrimento de sinais mais fracos é aplicado um logaritmo e posteriormente, cada bloco é normalizado entre 0 e 1, representando a menor e a maior energia naquele bloco.

Por fim, toda direção que a energia calculada estiver acima de um limiar (*threshold*) pré-estabelecido, gera uma detecção, ou seja, uma coordenada marcação, nível de sinal.

## 2.2 Clusterização

Esta seção aborda a eurística proposta para clusterização explorando a característica dos dados de serem dispersos ao longo da marcação. Portanto, considera-se um modelo onde cada contato (*cluster*) é composto de um pico e seja monotonicamente decrescente a partir dele. Os níveis de clusterização são ilustrados pela Figura 1.

A modelagem típica de detecções para um problema de clusterização, envolveria uma dissimilaridade em função de marcação e nível de sinal, o que poderia gerar *clusters* com elementos não contíguos. Nesta proposta, a dissimilaridade será calculada apenas em termos de marcação e o nível de sinal será considerado a parte, tentando formar um *cluster* a partir dos picos contíguos espacialmente.

A heurística proposta consiste em percorrer a lista de detecções ordenadas por nível, analisando portanto, primeiro as maiores energias e progressivamente descendo a análise no gráfico.

Para um dado elemento em análise tem-se duas opções, ele já fazer parte de um *cluster* ou de ainda estar isolado. Se o elemento em análise já pertence a um *cluster* ele deve adicionar ao *cluster* todos os vizinhos menos dissimilares que um hiperparâmetro **máxima dissimilaridade**, desde que estes vizinhos já não pertençam a nenhum *cluster*.

Por outro lado caso o elemento não faça parte de um *cluster*, ele deve criar um *cluster* e ser o representante deste, além de incorporar os vizinhos menos dissimilares que a **máxima dissimilaridade** indica que não pertençam a outro *cluster*.

Ao final deste processo ter-se-á um conjunto de *clusters* monotonicamente decrescentes, na prática, o sinal não será tão bem comportado dado o ruído de toda a cadeia, portanto, propõe-se um passo de consolidação dos *clusters* obtidos anteriormente, conforme ilustrado na Figura 1.

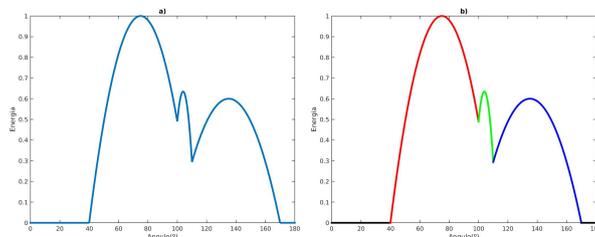


Figure 1: Primeiro nível de clusterização

Percorrendo os *clusters* na ordem de criação, ou seja, primeiro os *clusters* com elementos representantes com maior nível, tentar incorporar todos os *clusters* ainda não analisados. O *cluster* menor deve ser incorporado ao maior se o elemento do cluster maior menos dissimilar ao representante do *cluster* menor for menos dissimilar que a **máxima dissimilaridade** e o nível do representante do *cluster* menor não for maior que este elemento mais o hiperparâmetro **margem**. Voltando ao exemplo da Figura 1, se a margem for de 0.2 (considerando a energia normalizada) o *cluster* verde seria incorporado ao *cluster* vermelho.

Além disso, para redução de falso alarme é estabelecido que um *cluster* deve possuir um número de detecções mínima e portanto outro hiperparâmetro **tamanho mínimo**

## 2.3 Clusterização online - Acompanhamento

Um bloco de detecções geradas pelo processamento dos dados gera um bloco de *clusters*, nesta seção será explicado o algoritmo de clusterização online destes *clusters*, chamado de acompanhamento ou *tracking*, caracterizado como:

Nesta seção, foi utilizado uma variação dos algoritmos básicos de clusterização online. Similar ao ART, cada *cluster* é inserido sequencialmente. Cada inserção atualiza a posição de um *track* ou incorpora um *cluster* a um *track*, de acordo com o hiperparâmetro **máxima dissimilaridade *cluster track***, equivalente ao radio de vigilância do ART.

A atualização do *cluster* é feita pela interpolação dos parâmetros: *level*, *bearing* e *size* entre o *cluster* inserido e o *track* existente. O *sequency\_id* é substituído. A *direction* é estimada. A atualização da posição segue a métrica da equação 3, baseado no hiperparâmetro **memória**, definido como complementar ao **fator de aprendizado**.

$$\mu_k = \mu_{k-1} + \eta(x - \mu_{k-1}) \quad (1)$$

$$\mu_k = (1 - \eta)\mu_{k-1} + \eta x \quad (2)$$

$$\mu_k = (m_o)\mu_{k-1} + (1 - m_o)x \quad (3)$$

A direção é estimada pela interpolação da variação de marcação entre *track* e *cluster* e da própria direção do *track*, com a mesma mecânica de memória dos demais parâmetros. A direção age como estimador simplificado do movimento do alvo.

Para cálculo da dissimilaridade entre *cluster* e *track* foi ponderada a dissimilaridade das detecções dos representantes, dos níveis, dos tamanho e das direções.

### 3 Avaliação em cenários simulados

Para validação, refinamento e análise dos hiperparâmetros do algoritmo foram construídos dois cenários básicos de teste: um contato em movimento, um cruzamento de dois contatos.

Todas as análises serão feitas sobre o gráfico de *waterfall* para facilitar a compreensão gráfica das etapas do algoritmo, variando o cenário de detecções normalizadas com adição de ruído branco, verificando o efeito da variação da intensidade do ruído.

#### 3.1 Um contato

Neste cenário um contato faz um movimento senoidal ao longa da marcação. Na Figura 2 podemos ver o *waterfall* para cenário sem ruído *a* e com ruído de desvio padrão 0.4 *b*. Analogamente, na Figura 3 as respectivas clusterizações e na Figura 4 os *tracks*.

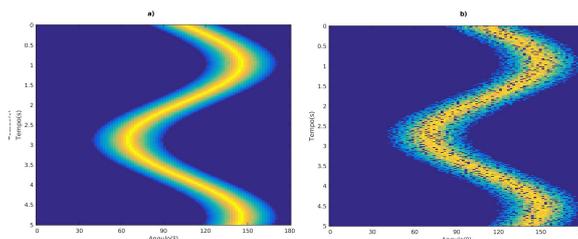


Figure 2: Primeiro cenário - Detecção

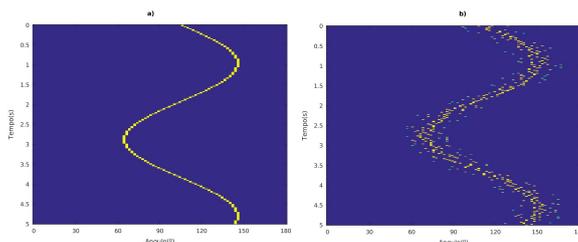


Figure 3: Primeiro cenário - Clusterização

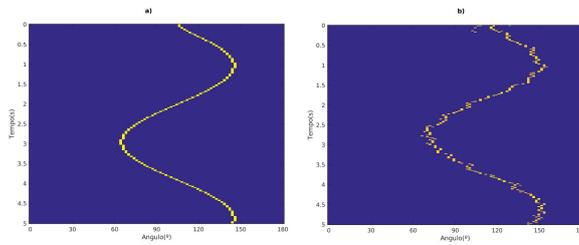


Figure 4: Primeiro cenário - *Tracking*

Neste cenário fica favorável a visualização do efeito do hiperparâmetro **memória** e o dilema plasticidade vs estabilidade. Na qual, o aumento da memória, ou a redução do **fator de aprendizado**, levam a um refinamento da estimativa de movimento do alvo, tornando o conjunto mais resistente ao ruído. Entretanto, tornando cada vez menos plástico o sistema, até o extremo onde o *track* é tão estável que não consegue acompanhar o movimento do alvo e acabam sendo criados vários *tracks* ao longo do movimento.

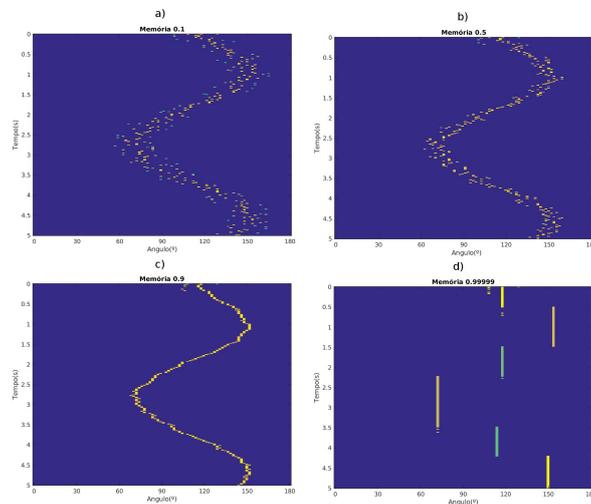


Figure 5: Efeito do parâmetro memória

### 3.2 Cruzamento

Esta seção, analogamente a seção 3.1 analisa o resultado do algoritmo com diferentes níveis de ruído. Neste cenário dois contatos com intensidades diferentes, dispersões diferentes e movimentos opostos se cruzam em movimentos senoidais, conforme ilustra a Figura 6. Analogamente, na Figura 7 as respectivas clusterizações e na Figura 8 os *trackings*.

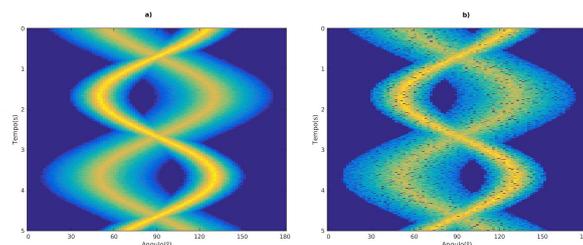


Figure 6: Segundo cenário - Detecção

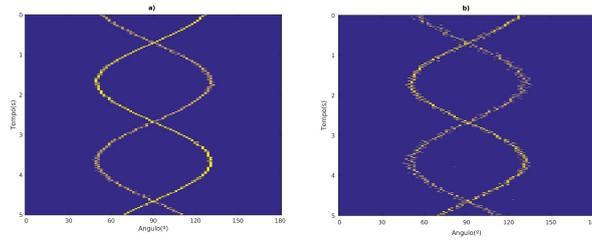


Figure 7: Segundo cenário - Clusterização

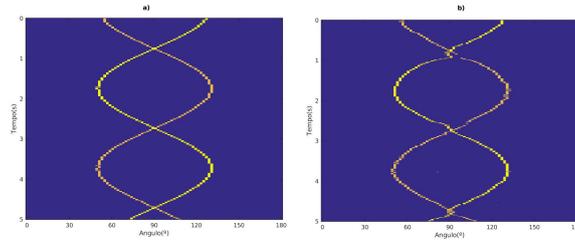


Figure 8: Segundo cenário - *Tracking*

Os tracks referentes a Figura 8a encontram-se separados na Figura 9, enquanto os *tracks* referentes a Figura 8b na 10. Podemos ver que o algoritmo foi capaz de resolver o cruzamento neste cenário.

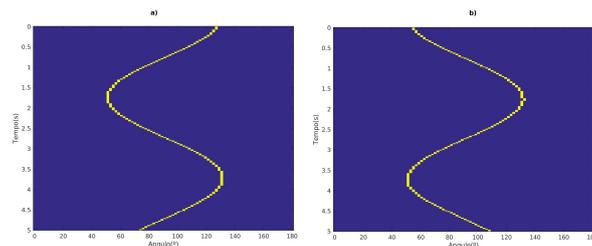


Figure 9: Segundo cenário - *Tracking* sem ruído separado

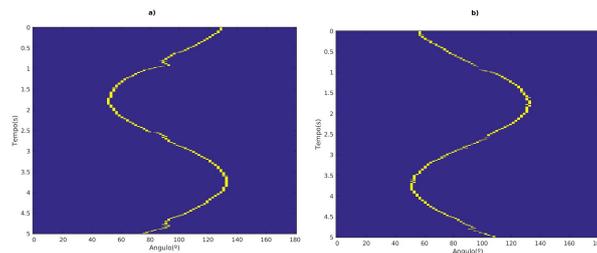


Figure 10: Segundo cenário - *Tracking* com ruído separado

#### 4 Avaliação em cenários real

Nesta seção, o algoritmo será testado com dados reais, disponibilizados para o XIII Encontro Tecnológico em Acústica Submarina (ETAS 2018). Os dados disponibilizados são duas gravações feitas com um arranjo de 32 canais.

O primeiro cenário é um *offshore* cruzando o canal, com cerca de seis minutos. O segundo é um cruzamento de lancha de trajetórias opostas de cerca de um minuto. Para obtenção da energia e da detecção dos blocos de dados gravados foi feito o condicionamento descrito na seção 2.1.

#### 4.1 Offshore

Na Figura 11a pode-se ver o gráfico *waterfall* da detecção, onde se identifica a trajetória do navio e a presença de duas regiões de ruído uma no início e outro no final da gravação. Na clusterização, Figura 11b, esse efeito já é reduzido e consegue-se ver a trajetória claramente. Pode-se verificar também que em ambos os limites de marcação o sinal é mais ruído e disperso, isso ocorre devido a linearidade do arranjo e características da formação de feixe, sendo um resultado prático esperado.

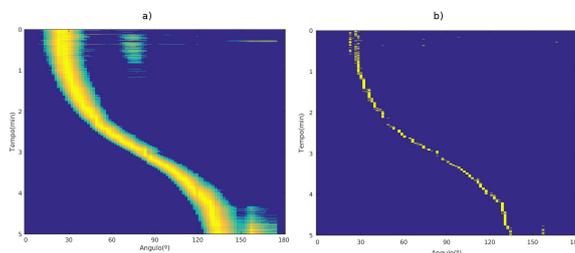


Figure 11: Terceiro cenário - Detecção e Clusterização

Diferente do sinal simulado, não é possível afirmar o erro da estimação de marcação por não se ter a marcação real, restando uma verificação visual do resultado do acompanhamento. Na Figura 12 pode-se ver o resultado do *tracking* e a sobreposição do *tracking* ao gráfico de detecção para avaliação visual do resultado. Observa-se que o algoritmo foi capaz de seguir satisfatoriamente o navio.

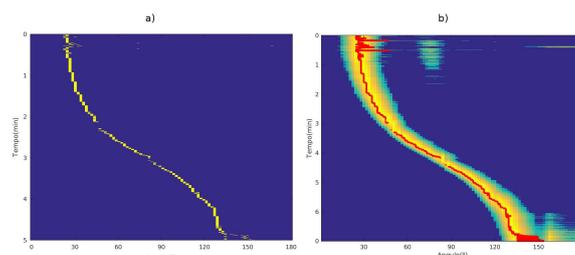


Figure 12: Terceiro cenário - Acompanhamento e Avaliação do resultado

#### 4.2 Cruzamento de Lanchas

Na Figura 13 pode-se ver o gráfico de detecções, onde se verifica um cenário bem mais desafiador, muito mais ruidoso, com cruzamento e com redução de ruído logo após o cruzamento por parte de uma das lanchas gerando quase um y.

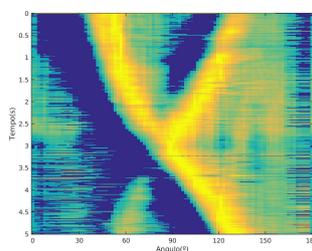


Figure 13: Quarto cenário - Detecção

Na Figura 14a podemos ver a clusterização do cenário. Na Figura 14b foi modificado o hiperparâmetro **tamanho mínimo** do *cluster* para dez detecções, reduzindo a quantidade de falsas detecções, principalmente na parte direita do gráfico, em compensação, diminuindo duas partes do sinal, a recuperação da lancha é encoberta com o início do caminho da outra lancha.

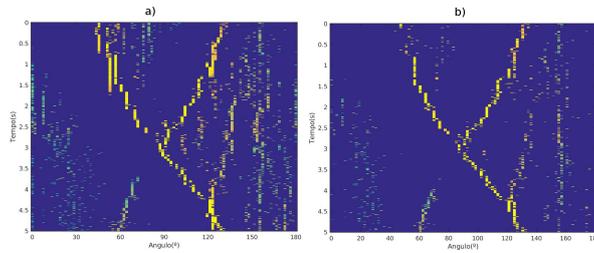


Figure 14: Quarto cenário - Cluster

O resultado do *tracking* pode ser visto na Figura 15, e a análise do resultado final do algoritmo pode ser avaliado pela Figura 16. Pode-se ver que o algoritmo conseguiu acompanhar ambos os alvos, mas a estimação da marcação ficou prejudicada pela cenário tão ruidoso.

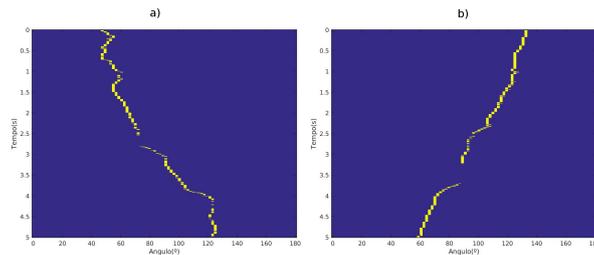


Figure 15: Quarto cenário - *Tracking*

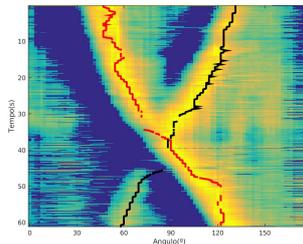


Figure 16: Quarto cenário - Avaliação do resultado

## 5 Conclusões

Neste trabalho foi proposto, explicado e analisado um algoritmo de clusterização que se mostrou funcional e útil tanto para o cenário simulado quanto para o dado real. O algoritmo proposto, busca *clusters* contíguos baseados em picos de uma dado dimensão, pode ser aplicado em outra aplicações inclusive na área de sonar passivo, como para extração automática de parâmetros de lofar e demon, ou uma extensão deste algoritmo para três dimensões para sonares ativos, na detecção automática de ecos.

Se faz necessário uma melhor análise e definições de métricas para aprimoramento da técnica proposta.

## 6 Trabalhos Futuros

Como melhoria para este algoritmo em termos de clusterização, poderia-se utilizar uma técnica de interpolação entre as detecções presentes em um *cluster* para refinamento da estimativa de marcação, ao invés de usar a marcação de maior energia. Isso provavelmente tornará a geração de detecções mais precisa e menos oscilante, melhorando a estimativa de marcação do *tracking*.

Além disso algumas outras heurísticas podem ser analisadas para melhora do algoritmo de *tracking*, como, por exemplo, hiperparâmetros transientes e adaptativos, assim como, distinguir diferentemente momentos onde o haja conflito entre *tracks*.

Pode-se ainda modificar a forma de calcular o efeito do movimento utilizado no cálculo de dissimilaridade entre *track* e *cluster* para considerar uma técnica mais robusta que envolve estimativas de posição futura do alvo.

### References

- [1] C. S. Jensen, D. Lin e B. C. Ooi, “Continuous Clustering of Moving Objects,” IEEE Transactions on Knowledge and Data Engineering, vol. 19, n<sup>o</sup> 9, pp. 1161-1174, set 2007.
- [2] O. Ossama, H. M. Mokhtar e M. E. El-Sharkawi, “An extended k-means technique for clustering,” Egyptian Informatics Journal, vol. 12, n<sup>o</sup> 1, pp. 45-51, pct 2010.
- [3] H. J. e. Y. J. Yifan Li, “Clustering moving objects,” Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 617-622, 2004.
- [4] E. Alpaydin, Introduction to Machine Learning, 2nd ed. London: The MIT Press, 2014, pp. 280–285.
- [5] S. Theodoridis, and K. Koutroubas, “Pattern Recognition”, 4th ed. California: Academic Press, pp. 739-740.