

SLAM mais Preciso utilizando Filtro Controlado Aprimorado por Alinhamento Ativo

Luigi Maciel Ribeiro* Nadia Nedjah**
Paulo Victor R. de Carvalho***

* *Instituto Computação, Universidade Federal do Rio de Janeiro, RJ,
(e-mail: luigimaciel@dcc.ufrj.br).*

** *Faculdade de Engenharia, Universidade do Estado do Rio de
Janeiro, RJ, (e-mail: nadia@eng.uerj.br)*

*** *Instituto Computação, Universidade Federal do Rio de Janeiro, RJ,
(e-mail: paulov@ien.gov.br)*

Abstract: This work presents a novel method, called Controlled Filter with Active Alignment (CFAA), to address the Simultaneous Localization and Mapping (SLAM) problem. SLAM aims to map an unknown environment while estimating the trajectory of a mobile agent moving within that environment. CFAA integrates the two fundamental pillars of SLAM, scan alignment and loop closure, into a unified process. The method draws inspiration from human perception of self-localization and employs a mental map for guidance. CFAA employs a Gaussian distribution to estimate potential poses and carries out the alignment process in cycles, where each cycle is influenced by the outcomes of the preceding one. The Active Alignment mechanism is employed in each cycle to determine the quality of each potential pose, enabling more accurate simultaneous localization and mapping. Benchmark tests were conducted on five public datasets, verifying the effectiveness and efficiency of the proposed method.

Resumo: Este trabalho apresenta um novo método, chamado Filtro Controlado com Alinhamento Ativo (FCAA), para abordar o problema de Localização e Mapeamento Simultâneos (SLAM). O SLAM tem como objetivo mapear um ambiente desconhecido enquanto estima a trajetória de um agente móvel que se move dentro desse ambiente. O FCAA combina os dois pilares fundamentais do SLAM, alinhamento de varredura e fechamento de laço, em um único processo. O método é inspirado na percepção humana de se localizar e para tal utiliza um mapa mental para se guiar. O FCAA utiliza uma distribuição gaussiana para estimar possíveis poses e realiza o processo de alinhamento em ciclos, onde cada ciclo é influenciado pelos resultados do ciclo anterior. O mecanismo de Alinhamento Ativo é utilizado em cada ciclo para determinar a qualidade de cada uma das possíveis poses, permitindo uma localização e mapeamento simultâneos mais precisos. Foram realizados testes de *benchmarking* em 5 conjuntos de dados públicos que comprovam a eficácia e eficiência do método proposto.

Keywords: Localization; mapping; perception; robots; alignment;

Palavras-chaves: Localização; mapeamento; percepção; robôs; alinhamento;

1. INTRODUÇÃO

Localização e mapeamento simultâneos (*Simultaneous Localization and Mapping* - SLAM) é um problema computacional de alta complexidade, que tem como objetivo mapear um ambiente desconhecido e estimar simultaneamente a trajetória de um agente móvel que está se locomovendo nesse ambiente. Os pilares fundamentais do SLAM são o alinhamento de varreduras e o fechamento de laços, que são amplamente abordados por inúmeros trabalhos Durrant-Whyte and Bailey (2006); Davison and Murray (2002).

O processo de alinhamento é comum no nosso dia a dia. Desde cedo somos treinados e estimulados a realizar este tipo de processo, tal como colocar a chave na fechadura, conectar o carregador do celular ou rosquear a tampa de uma garrafa, entre outros. Os processos de alinhamento

realizados pelos seres humanos mostram uma metodologia sistemática, onde o processo é realizado em ciclos, com etapas de ação, observação e ajuste. Esse conceito inspirou a criação de um novo método para alinhamento de varreduras, chamado Filtro Controlado (FC), que utiliza uma distribuição Gaussiana para estimar possíveis poses e realiza o processo de alinhamento de varreduras em etapas, com a etapa atual sendo realimentada com os resultados da etapa imediatamente anterior. As principais inspirações do FC são o Filtro de Partículas Ristic et al. (2003) e os sistemas de controle realimentados Van de Vegte (1994).

O outro pilar do SLAM é o fechamento de laços, que consiste no processo de reconhecimento de locais já visitados e mapeados pelo agente móvel. Com o objetivo de otimizar todo o processo do SLAM, desenvolveu-se um alinhamento de varredura inspirado na percepção humana, que une os dois pilares do SLAM em um único processo. Para

comprovar a eficácia do método, foram realizados testes de *benchmarking* com cinco conjuntos de dados públicos diferentes.

Este trabalho está dividido em cinco seções. Na Seção 2 são apresentados trabalhos que apresentam abordagens para otimizar o processo do SLAM. Na Seção 3 são demonstradas todas as etapas do alinhamento de varreduras utilizando o FC. Em seguida, na Seção 4 é explicado como a percepção é implementada para melhorar o processo do SLAM. Na Seção 5 é destacada toda a metodologia do teste de *benchmarking* e concluída com os resultados do teste. O trabalho é finalizado com a Seção 6, onde são analisados os resultados obtidos e propostos trabalhos futuros.

2. TRABALHOS RELACIONADOS

Em Matsuki et al. (2023), os autores apresentam uma abordagem recentemente introduzida na área de mapeamento e localização simultâneos (SLAM). Esse estudo inovador aborda a limitação dos sistemas SLAM neurais atuais ao oferecer uma solução dinâmica que se distancia das representações de mapas centrados no mundo, ao mesmo tempo que enfatiza a construção de campos neurais com base em observações em tempo de execução. Em um contexto semelhante de busca por inovações no SLAM, o presente trabalho propõe o Filtro Controlado com Alinhamento Ativo (FCAA), uma abordagem que se diferencia ao combinar o alinhamento de varredura e fechamento de laço em um único processo. Enquanto a abordagem mencionada destaca-se por sua perspectiva de construção de campos neurais dinâmicos, o FCAA se destaca ao adotar o conceito de percepção ativa e uso de mapas mentais para guiar o processo. Essa quebra de paradigma na forma como o SLAM é abordado demonstra uma tendência de busca por soluções mais eficientes e precisas, resultando em abordagens mais adaptáveis às características dinâmicas dos ambientes e aos desafios enfrentados na estimação da trajetória e mapeamento.

Em Zhao et al. (2022), é apresentada uma abordagem para o SLAM ativo em ambientes dinâmicos usando veículos micro-aéreos (MAVs). Neste trabalho, a percepção é utilizada por um planejador de caminhos sensíveis, chamado de planejador "Next-Best-View" (NBVP), para evitar obstáculos estáticos e dinâmicos em ambientes desconhecidos. Em conjunto com um fechamento de laço ativo, chamado de planejador de fechamento de laço ativo (ALCP), é possível reduzir a incerteza do SLAM e melhorar a precisão na localização. O método é demonstrado em ambientes simulados e apresenta resultados que comprovam a eficácia desta metodologia.

Em Deng et al. (2018) os autores apresentam uma abordagem de SLAM visual (VSLAM) com o objetivo principal de rastrear possíveis falhas e preveni-las durante o processo de navegação e mapeamento. Por meio da identificação e associação contínua de pontos do mapa durante o movimento, esta proposta consegue estruturar um planejamento que leva em consideração as restrições sensoriais, resultando em uma navegação confiável. A percepção é utilizada neste método para prever o número de pontos do mapa associados a uma determinada pose, permitindo planejar um caminho de distância ideal de modo que o número de pontos associados esteja sempre acima do limite. Os

resultados dos experimentos realizados em dois ambientes demonstram o desempenho desta abordagem.

Em Nedjah et al. (2020) é apresentado uma abordagem inovadora para solucionar o problema de localização e mapeamento simultâneo (SLAM) em robótica autônoma. A pesquisa tem como foco a correspondência de varreduras, etapa crucial para o sucesso do SLAM. Três métodos de otimização de inteligência de enxame foram investigados: otimização de enxame de partículas, colônia artificial de abelhas e algoritmo de vaga-lume. A solução proposta foi testada em vários cenários e os resultados mostraram que a colônia artificial de abelhas foi mais precisa em um amplo espectro de circunstâncias. Este trabalho foi uma das principais fontes de inspiração para o desenvolvimento do método Filtro Controlado com Alinhamento Ativo, que será apresentado neste artigo. A principal diferença entre os trabalhos é o uso da percepção no novo método, o que pode levar a melhorias significativas na precisão e eficiência do SLAM em robótica autônoma.

3. FILTRO CONTROLADO

O processo de alinhamento compara duas varreduras, uma que será alinhada e outra que serve de referência. O objetivo é encontrar uma pose (x, y, ϕ) que minimize o erro médio quadrático entre as duas varreduras. O Filtro Controlado (FC) atualiza um parâmetro por vez, o que torna o processo mais eficiente. Por exemplo, se o algoritmo tiver que analisar 50 partículas para encontrar a melhor pose, seriam necessárias cerca de 125 mil execuções da função objetivo. No entanto, se a análise for feita separadamente para cada grau de liberdade (x, y, ϕ) , serão necessárias apenas 150 execuções da função objetivo por ciclo. Os resultados apresentados mais adiante mostram que o FC converge em poucos ciclos e que, usando a estratégia adotada, seria necessário mais de 800 ciclos para executar a função objetivo 125 mil vezes. Isto ilustra a maior eficiência do FC em relação a outras abordagens.

O FC é composto por cinco etapas: inicialização, ajuste de ϕ , ajuste de x , ajuste de y e critério de parada. O Algoritmo 1 apresenta o funcionamento completo do filtro. O algoritmo do FC possui alguns parâmetros de entrada, tais como: o mPose (melhor pose), que é a pose que se acredita possuir um bom alinhamento; a vAtual, que são as medições captadas pelo robô na pose atual; e a vReferência, que são as medições captadas pelo robô na pose anterior.

3.1 Inicialização

Nesta etapa, é realizada a definição da pose que se acredita possuir um bom alinhamento (melhor pose), a qual pode ser a pose de referência ou uma pose calculada a partir da odometria no primeiro ciclo de alinhamento. Após esta definição, é realizada a distribuição gaussiana em torno da melhor pose, considerando os três graus de liberdade da pose (x, y, ϕ) , em que σ e μ são valores reais maiores que zero, sendo σ a média da distribuição (melhor pose) e μ , o desvio padrão, conforme a equação:

$$\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (1)$$

Algorithm 1 Filtro controlado

Require: mPose, vAtual, vReferência
pControle \leftarrow mPose
dP \leftarrow 0.1 ▷ Desvio padrão
nAmostras \leftarrow 100
nMaxCiclos \leftarrow 100
nMaxEstagnação \leftarrow 3
estagna \leftarrow 0
for $k \leftarrow 1$ **to** $nMaxCiclos$ **do**
 a Φ , aX, aY \leftarrow **Inicialização**(mPose, dP, nAmostras)
 mPose \leftarrow **Ajuste** Φ (a Φ , mPose, vAtual, vReferência)
 mPose \leftarrow **Ajuste**X(aX, mPose, vAtual, vReferência)
 mPose \leftarrow **Ajuste**Y(aY, mPose, vAtual, vReferência)
 finalizar, estagna, pControle \leftarrow **critérioPa-**
rada(pControle, mPose, estagna, nMaxEstagnação)
 if finalizar = *Sim* **then**
 k \leftarrow nMaxCiclos ▷ Finaliza
 end if
end for
return mPose

3.2 Ajuste de coordenadas

Para encontrar a melhor ϕ , é preciso manter fixos os outros parâmetros (x e y) da melhor pose. Em seguida, para cada ϕ , é realizada uma nova varredura e uma avaliação com a função objetivo. A função objetivo utilizada é o erro médio quadrático entre a varredura de referência e a nova varredura:

$$\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2, \quad (2)$$

após obter o melhor resultado, atualiza-se o parâmetro ϕ da melhor pose e avança-se para a próxima etapa.

Para atualizar o valor de X, é necessário manter os outros parâmetros fixos (y e ϕ) da melhor pose. Em seguida, para cada valor de X, é recalculada a varredura e realizada uma avaliação com a função objetivo. Após obter o melhor resultado, atualiza-se o parâmetro x da melhor pose e avança-se para a próxima etapa.

Para atualizar o valor de Y, é necessário manter os outros parâmetros fixos (x e ϕ) da melhor pose. Em seguida, para cada valor de Y, é recalculada a varredura e realizada uma avaliação com a função objetivo. Após obter o melhor resultado, atualiza-se o parâmetro y da melhor pose e avança-se para a próxima etapa.

3.3 Critério de parada

Esta etapa controla o momento de encerramento do processo de alinhamento do FC, verificando se houve alguma atualização na melhor pose. Em caso afirmativo, o processo retorna para a primeira etapa de inicialização. Se não houve atualização, o contador de estagnação é incrementado. Em seguida, é verificado se o número máximo de estagnações foi atingido. Se ainda não foi atingido, o processo retorna para a etapa de inicialização, caso contrário, o processo é finalizado.

4. ALINHAMENTO ATIVO

A localização, assim como os diferentes processos de alinhamento, são processos comuns em nosso cotidiano, como mencionado na introdução deste trabalho. Nós nos localizamos o tempo todo, seja indo para o trabalho, indo do quarto para o banheiro ou passeando em um shopping. Para nos localizarmos, utilizamos duas coisas: a nossa percepção em relação ao espaço que nos cerca - normalmente proveniente da nossa visão, mas também pode ser através de outros sentidos, como tato, audição ou até o olfato - e um mapa mental que construímos, caso seja um local novo, ou um mapa que já está em nossa memória (Thornton et al., 1998).

O Alinhamento Ativo desenvolvida neste trabalho é inspirada na nossa percepção ao nos localizarmos. O robô, ao navegar em um ambiente desconhecido, utiliza o mapa que já construiu para se localizar. Com isto, ocorre a junção dos processos de alinhamento de varredura e fechamento de laço. Ao alinhar a varredura da pose atual ao mapa existente, o processo resolve o problema de acúmulo de erros comum em processos de alinhamento tradicionais (Mo et al., 2021) e identifica simultaneamente locais já visitados. O SLAM pode adotar diferentes estratégias, como não adicionar novas leituras em áreas já mapeadas, evitando a inclusão de ruídos no mapa.

A função *AlinhamentoAtivo* é apresentada no Algoritmo 2. Esta função será utilizada como função objetivo dos processos de alinhamento de varredura realizada pelo Filtro Controlado. Como parâmetros de entrada, a função recebe a *pAjustada*, que representa a pose a ser avaliada, a vAtual e o mapa. Vale ressaltar que no primeiro ciclo de SLAM não há um mapa prévio, portanto, não é necessário realizar o alinhamento. Nesse cenário, o mapa é construído com base na primeira varredura realizada pelo robô. A partir desse ponto, o alinhamento é realizado normalmente, utilizando o mapa construído durante as varreduras anteriores.

O primeiro processo realizado pela função é a conversão da varredura em coordenadas. A varredura é uma representação das distâncias dos obstáculos em relação ao robô e esta representação precisa ser convertida para uma representação em coordenadas, compatível com o mapa, para que seja possível realizar uma análise. A *pAjustada* é definida como $P = (x_p, y_p, \phi_p)$, a coordenada como $c_i = [x_i, y_i]$, onde x e y representam uma posição no mapa, e a varredura como $V = [v_1, v_2 \dots v_i]$, onde v_i é definido como $v_i = (d_i, \theta_i)$, sendo d a distância e θ o ângulo de medida. A função *vParaCoordenadas* é definida pela seguinte equação:

$$c_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} x_p + d_i \cos(\phi_p + \theta_i) \\ y_p + d_i \sin(\phi_p + \theta_i) \end{bmatrix}, \quad (3)$$

após a conversão da varredura em coordenadas, é realizada uma verificação ponto a ponto. Para isto, a função *verificarP* é utilizada, analisando um ponto específico do mapa e retornando *Sim* caso o ponto esteja ocupado ou *Não* caso o ponto esteja vazio. A primeira verificação ocorre no próprio ponto que está sendo analisado. Se a coordenada do ponto estiver ocupada, o processo passa para o próximo ponto sem incrementar nenhuma punição. Caso contrário, uma punição de 100% é incrementada, e a análise de todos

Algorithm 2 *AlinhamentoAtivo*

Require: pAjustada, vAtual, mapa
coordenadas \leftarrow **vParaCoordenadas**(pAjustada, vAtual)
erro, nPunidos \leftarrow 0
punicao \leftarrow 4
for ponto **in** coordenadas **do**
 pOcupado \leftarrow **verificarP**(ponto, mapa)
 if pOcupado = Sim **then**
 continue \triangleright Passa para o próximo ponto
 else
 erro \leftarrow erro + punicao
 nPunidos \leftarrow nPunidos + 1
 for vX \leftarrow -2 to 2 **do**
 for vY \leftarrow -2 to 2 **do**
 vizinho \leftarrow (ponto[x] + vX, ponto[y] + vY)
 pOcupado \leftarrow **verificarP**(vizinho, mapa)
 if pOcupado = Sim **then**
 distanciaVizinho \leftarrow |vX| + |vY|
 erro \leftarrow erro + $\frac{\text{punicao}}{4} \frac{1}{\text{distanciaVizinho}}$
 end if
 end for
 end for
 end if
end for
return erro / nPunidos

os vizinhos é iniciada. Para cada coordenada dos vizinhos desocupada, é incrementada uma punição referente em função da distância.

Um exemplo do funcionamento da função *AlinhamentoAtivo* no processo de alinhamento de varredura é apresentado na Figura 1. Inicialmente, é apresentado um alinhamento ruim, como mostrado na primeira figura. Esse alinhamento possui vários pontos de punição, como pode ser observado na segunda figura. Um alinhamento melhor é apresentado na terceira figura, com conseqüentemente um número menor de pontos com punição, como mostrado na quarta figura. O objetivo do processo de alinhamento é minimizar os pontos de punição, o que só ocorre com um alinhamento eficiente da varredura atual ao mapa.

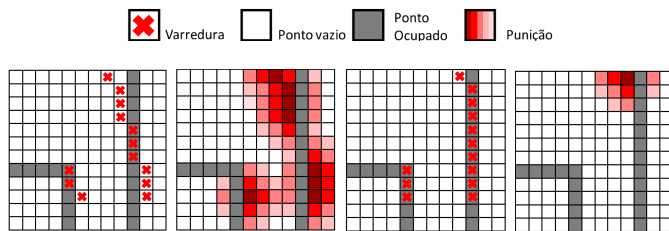


Figura 1. Alinhamento de varredura utilizando *AlinhamentoAtivo*

5. RESULTADOS DE DESEMPENHO

Nesta seção, apresentaremos os detalhes de implementação do método proposto neste trabalho, Filtro de Controle com Alinhamento Ativo (FCAA). Avaliamos o desempenho do FCAA, com o objetivo de apontar os resultados obtidos nos testes de *benchmarking* e comprovar sua eficácia. Para isso, comparamos os resultados com outros trabalhos que utilizaram os mesmos testes de *benchmarking*. A implementação foi realizada na linguagem Python, utilizando a

biblioteca Pandas para análise e conversão de dados, e a biblioteca PyGame para criar os ambientes de simulação. As simulações foram executadas em um computador equipado com um processador Intel Core i7-10750H e 8GB de memória RAM.

5.1 Avaliação de desempenho do FCAA

Foi definida como metodologia de avaliação o método proposto por Kümerle em (Kümerle et al., 2009). Esta metodologia é amplamente aplicada em outros trabalhos que propõem soluções para SLAM, e, portanto, ao utilizá-la, torna-se possível comparar o desempenho do método proposto com outros trabalhos de forma padronizada, facilitando sua avaliação. É importante ressaltar que a metodologia de Kümerle não avalia apenas o resultado final dos mapas, mas também o processo de criação deles, definindo um modelo para comparar a variação entre poses consecutivas. Para avaliar a acurácia, dois parâmetros são definidos: o erro translacional absoluto e o erro rotacional absoluto. O erro translacional absoluto é definido por:

$$\lambda_{\text{translacional}} = \frac{1}{N} \sum_{i,j} |T(\delta_{i,j} - \delta_{i,i}^*)|, \quad (4)$$

onde N é o número total de poses consideradas; $\delta_{i,j}$ é a variação entre as poses, definida por:

$$\delta_{i,j} = \begin{bmatrix} \Delta_x \\ \Delta_y \\ \Delta_\phi \end{bmatrix} = \begin{bmatrix} (x_i - x_j) \cos \phi_j + (y_i - y_j) \sin \phi_j \\ -(x_i - x_j) \sin \phi_j + (y_i - y_j) \cos \phi_j \\ \phi_i - \phi_j \end{bmatrix}, \quad (5)$$

$\delta_{i,j}^*$ é a variação entre poses do conjunto de referência; e $T(\cdot)$ separa o componente translacional (x, y) da pose relativa e calcula a resultante utilizando o Teorema de Pitágoras. O erro rotacional absoluto, onde $R(\cdot)$ separa o componente rotacional (ϕ) da pose relativa, é definido por:

$$\lambda_{\text{rotacional}} = \frac{1}{N} \sum_{i,j} |R(\delta_{i,j} - \delta_{i,i}^*)|. \quad (6)$$

Com o objetivo de possibilitar a comparação dos resultados obtidos pelo FCAA com outros trabalhos e de estabelecer um método de avaliação padronizado, optamos por utilizar conjuntos de dados públicos disponíveis em <http://www.ipb.uni-bonn.de/datasets/>. Na Figura 2, é possível visualizar os mapas de grade e as trajetórias dos robôs correspondentes aos conjuntos de dados INTEL, MIT CSAIL, ACES3, FREIBURG BLDG 79 e MIT KILLIAN COURT, dispostos da primeira à última linha, respectivamente.

O FCAA demonstrou sucesso em todos os testes de *benchmarking* realizados, comprovando sua capacidade de lidar com diversas estruturas de dados, como diferentes números de leituras do Lidar, variações na precisão das medições e topologias distintas. Essa diversidade é apresentada nos conjuntos de dados utilizados nos testes, fornecendo notoriedade ao teste. O método apresentou uma melhora significativa no mapa gerado quando comparado com os

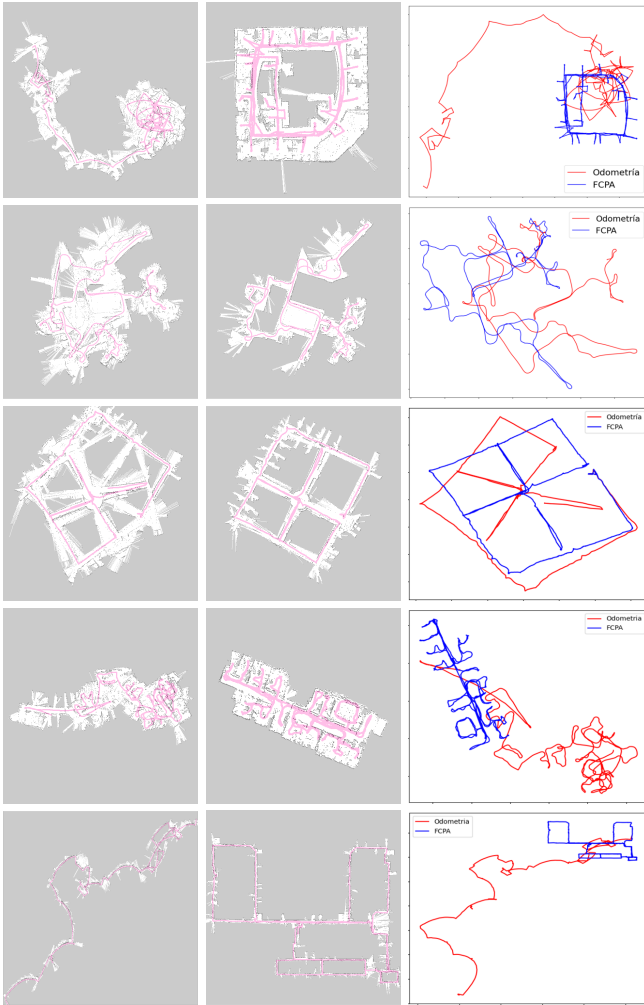


Figura 2. Mapa de grade e trajetória do robô (linha rosa) obtidos usando somente a odometria (à esquerda) e usando o FCAA (ao centro) e a comparação das trajetórias (à direita), simulações referentes ao conjunto de dados: INTEL, MIT CSAIL, ACES3, FREIBURG BLDG 79 e MIT KILLIAN COURT, da primeira linha à última respectivamente

mapas gerados apenas com a odometria. No entanto, foram identificadas dificuldades em conjuntos de dados com corredores muito longos, devido à própria metodologia do Alinhamento Ativo. Em corredores longos, o alinhamento sofre grande influência da posição atual do robô, o que pode levar a uma estagnação, causando o problema de encurtamento de corredores. Outro desafio com corredores longos é o fechamento de laços distantes, o Alinhamento Ativo possui um alcance limitado, o que significa que, na ocorrência de pequenos erros de alinhamento no início do laço, este erro poderá causar um distúrbio significativamente grande, pelo fato da proposta não possuir uma estratégia específica para otimizar os fechamentos de laço. Nesse cenário, o Alinhamento Ativo não terá alcance para corrigir o erro.

5.2 Análise comparativa

A análise comparativa realiza uma avaliação do FCAA em função de outros dois métodos para a resolução do SLAM:

o *Cartographer* (Hess et al., 2016) e o *Graph Mapping* (GM) (Kümmerle et al., 2009). A seleção dos conjuntos de dados foi baseada em sugestões apresentadas nesses trabalhos. O *Cartographer* propõe uma metodologia para fechamento de loops em tempo real com alta precisão, utilizando estruturas ramificadas e submapas para limitar as correspondências de varredura. Já o GM apresenta uma abordagem para SLAM baseada em otimização de grafo, construindo o grafo a partir de medidas sequenciais e otimizando a configuração do grafo utilizando uma abordagem de minimização do erro médio quadrático.

De acordo com os resultados apresentados na Tabela 1 e ilustrado como gráfico de barras na Figura 3, o método FCAA apresenta resultados promissores quando comparado aos métodos *Cartographer* e GM. No conjunto de dados INTEL, o FCAA superou ambos os métodos em termos de erros rotacionais e translacionais. Especificamente, o FCAA apresentou uma melhora de 5,67% e 58,27% nos erros translacionais e rotacionais, respectivamente, em comparação com o *Cartographer*. Em comparação com o GM, o FCAA apresentou uma melhora de 30,32% e 85,46% nos erros translacionais e rotacionais, respectivamente.

No conjunto de dados MIT CSAIL, o FCAA ainda superou o *Cartographer* em termos de erros translacionais e rotacionais, com melhorias de 67,08% e 65,58%, respectivamente. No entanto, o FCAA apresentou uma diminuição significativa no desempenho quando comparado ao GM, com um aumento de 262% e 254% nos erros translacionais e rotacionais, respectivamente. Isso destaca a importância da avaliação específica do conjunto de dados ao considerar o desempenho dos métodos SLAM.

No conjunto de dados ACES3, o FCAA apresentou uma diminuição no desempenho em comparação com o *Cartographer*, com aumento nos erros translacionais e rotacionais de 43,0% e 33,13%, respectivamente. Em comparação com o GM, o FCAA também apresentou uma diminuição no desempenho, com um aumento de 30,66% e 25,65% nos erros translacionais e rotacionais, respectivamente.

No conjunto de dados FREIBURG BLDG 79, o FCAA apresentou melhorias em relação ao *Cartographer* e GM em termos de erros rotacionais e translacionais. Especificamente, o FCAA apresentou uma melhora de 10,61% e 13,01% nos erros translacionais e rotacionais, respectivamente, em comparação com o *Cartographer*. Em comparação com o GM, o FCAA apresentou uma melhora de 27,85% e 22,0% nos erros translacionais e rotacionais, respectivamente.

Por fim, no conjunto de dados MIT KILLIAN COURT, o FCAA apresentou piora de 49,93% no erro translacional absoluto em comparação com o *Cartographer*. No entanto, houve uma melhora de 8,8% no erro rotacional absoluto em comparação com o *Cartographer*. Além disso, o FCAA apresentou uma piora de 36,62% no erro translacional absoluto e uma melhora de 35,8% no erro rotacional absoluto em comparação com o GM. Esses resultados sugerem que o FCAA é um método promissor para SLAM e pode fornecer maior precisão em certos conjuntos de dados quando comparado a outros métodos comumente utilizados.

Tabela 1. Comparação quantitativa do erro com (Hess et al., 2016; Kümmerle et al., 2009).

	FCAA	Cartographer	GM
INTEL			
λ_{tra}	0.0216 ± 0.0302	0.0229 ± 0.0239	0.031 ± 0.026
λ_{rot}	0.189 ± 1.854	0.453 ± 1.335	1.3 ± 4.7
MIT CSAIL			
λ_{tra}	0.0105 ± 0.0114	0.0319 ± 0.0363	0.004 ± 0.009
λ_{rot}	0.127 ± 0.255	0.369 ± 0.365	0.05 ± 0.08
ACES3			
λ_{tra}	0.0658 ± 0.0708	0.0375 ± 0.0426	0.044 ± 0.044
λ_{rot}	0.538 ± 0.365	0.373 ± 0.469	0.4 ± 0.4
F. BLDG 79			
λ_{tra}	0.0404 ± 0.0258	0.0452 ± 0.0354	0.056 ± 0.042
λ_{rot}	0.468 ± 0.571	0.538 ± 0.718	0.6 ± 0.6
MIT K.C.			
λ_{tra}	0.0789 ± 0.0651	0.0395 ± 0.0488	0.050 ± 0.056
λ_{rot}	0.321 ± 0.278	0.352 ± 0.353	0.5 ± 0.5

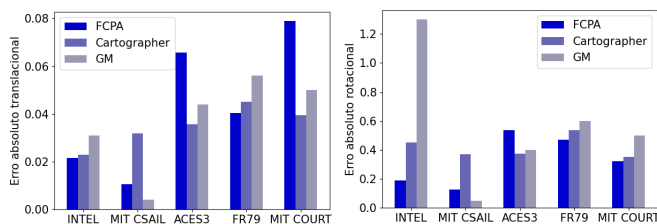


Figura 3. Gráfico comparativo do erro translacional absoluto e rotacional absoluto, respectivamente.

6. CONCLUSÃO

Este trabalho apresentou uma nova abordagem para o problema de localização e mapeamento simultâneos (SLAM). A metodologia sistemática de alinhamento realizada pelos seres humanos inspirou a criação de um novo método para o alinhamento de varreduras, denominado Filtro Controlado (FC). Esse método utiliza uma distribuição Gaussiana para estimar possíveis poses e realiza o processo de alinhamento de varreduras em etapas. Além disso, foi desenvolvido um alinhamento de varredura inspirado na percepção humana que une os dois pilares do SLAM em um único processo.

Para avaliar o método proposto, utilizou-se a metodologia proposta por Kümmerle, que avalia tanto o resultado final dos mapas quanto o processo de criação deles. Foram utilizados conjuntos de dados públicos para permitir a comparação dos resultados do FCAA com outros trabalhos e um método de avaliação padronizado. O FCAA demonstrou sucesso em todos os testes de *benchmarking* realizados, mas apresentou dificuldades em conjuntos de dados com corredores muito longos, devido à própria metodologia do Alinhamento Ativo.

Realizou-se uma análise comparativa com outros métodos de SLAM, como o Cartographer e o Graph Mapping, e o FCAA obteve em média um resultado melhor que os demais. Contudo, reconhecemos que há espaço para melhorias no FCAA, especialmente em relação às dificuldades enfrentadas em corredores longos. Nesse sentido, futuros trabalhos poderiam explorar o desenvolvimento de um sistema auxiliar ao FCAA que detecte esse tipo de topologia utilizando a própria percepção, permitindo a adoção de

abordagens para aprimorar o desempenho e a precisão do método.

Em resumo, este trabalho apresentou resultados promissores e abriu caminho para futuros estudos na área de SLAM, contribuindo para o desenvolvimento de novos métodos e técnicas que possam ser aplicados em diversas áreas, como robótica, mapeamento e navegação autônoma.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Faperj e da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

REFERÊNCIAS

- Davison, A.J. and Murray, D.W. (2002). Simultaneous localization and map-building using active vision. *IEEE transactions on pattern analysis and machine intelligence*, 24(7), 865–880.
- Deng, X., Zhang, Z., Sintov, A., Huang, J., and Bretl, T. (2018). Feature-constrained active visual slam for mobile robot navigation. In *2018 IEEE international conference on robotics and automation (ICRA)*, 7233–7238. IEEE.
- Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2), 99–110.
- Hess, W., Kohler, D., Rapp, H., and Andor, D. (2016). Real-time loop closure in 2d lidar slam. In *2016 IEEE international conference on robotics and automation (ICRA)*, 1271–1278. IEEE.
- Kim, A. and Eustice, R.M. (2013). Perception-driven navigation: Active visual slam for robotic area coverage. In *2013 IEEE International Conference on Robotics and Automation*, 3196–3203. IEEE.
- Kümmerle, R., Steder, B., Dornhege, C., Ruhnke, M., Grisetti, G., Stachniss, C., and Kleiner, A. (2009). On measuring the accuracy of slam algorithms. *Autonomous Robots*, 27, 387–407.
- Matsuki, H., Tateno, K., Niemeyer, M., and Tombari, F. (2023). Newton: Neural view-centric mapping for on-the-fly large-scale slam. *arXiv preprint arXiv:2303.13654*.
- Mo, J., Islam, M.J., and Sattar, J. (2021). Fast direct stereo visual slam. *IEEE Robotics and Automation Letters*, 7(2), 778–785.
- Nedjah, N., de Oliveira, P.J.A., et al. (2020). Simultaneous localization and mapping using swarm intelligence based methods. *Expert Systems with Applications*, 159, 113547.
- Ristic, B., Arulampalam, S., and Gordon, N. (2003). *Beyond the Kalman filter: Particle filters for tracking applications*. Artech house.
- Thornton, I.M., Pinto, J., and Shiffrar, M. (1998). The visual perception of human locomotion. *Cognitive Neuropsychology*, 15(6-8), 535–552.
- Van de Vegte, J. (1994). *Feedback control systems*. Prentice-Hall, Inc.
- Zhao, Y., Xiong, Z., Zhou, S., Wang, J., Zhang, L., and Campoy, P. (2022). Perception-aware planning for active slam in dynamic environments. *Remote Sensing*, 14(11), 2584.