

Detection and localization of mobile robots in intelligent spaces using artificial neural networks

Mateus Sobrinho Menines¹, Antonio Marcos Mutz Entringer², Leornador de Assis Silva³,
Fabricio Bertolini de Sá⁴, Vinícios Hernech Izabel⁵, and Raquel Frizera Vassallo⁶

^{1, 5, 6}Universidade Federal do Espírito Santo, Vitória, Brasil

^{2, 3, 4}Instituto Federal do Espírito Santo, Guarapari, Brasil

mateus.menines09@gmail.com¹

Resumo—The increasing impact of technology on daily and industrial activities, with a special focus on mobile robotics, has become increasingly evident in scenarios where collaboration between humans and machines is essential for the well-being of society. This work proposes the development of a detection and position reconstruction method for mobile robots in intelligent spaces with multiple cameras, using convolutional neural networks instead of physical markers. This aims to provide an alternative means of robot localization and monitoring, with the goal of generating precise position and orientation data, eliminating the need to rely on markers for this purpose. The article describes the localization method using convolutional neural networks and includes location error experiments to validate the proposed approach.

Index Terms—mobile robotics, intelligent space, detection, convolutional neural networks

I. INTRODUÇÃO

Com o avanço da tecnologia, a capacidade de replicar as características humanas se tornou fundamental em sistemas robóticos. Por conta disso, atividades por muitas vezes realizadas por seres humanos podem ser efetuadas com maior rapidez e eficiência por robôs em diversos ambientes [1].

Neste cenário, a colaboração entre o ser humano e o avanço tecnológico na área da robótica se torna fundamental para a harmonia nos procedimentos de fabricação na era da Indústria 4.0. Isso evoluiu para o que é conhecido como Indústria 5.0, um ambiente no qual a colaboração entre a máquina e a humanidade se torna o epicentro das operações [2].

Os Espaços Inteligentes (IS - *Intelligent Spaces*) têm se tornado um conceito com característica harmônicas entre homem e a máquina para as Indústrias 4.0 e 5.0. Esses locais possuem sensores e atuadores e uma infraestrutura de hardware e software para a obtenção de informações do ambiente, permitindo a interação humana e a execução de possíveis tarefas no local [3]. Alguns dos sensores utilizados são as câmeras, que capturam imagens do espaço para a realização de técnicas envolvendo visão computacional e processamento de imagens [4].

A utilização destas câmeras oferece a possibilidade de evitar sensores adicionais no ambiente. Quando devidamente calibradas, as câmeras podem fornecer a localização de objetos com precisão. No contexto da robótica, por exemplo, a utilização de câmeras é eficaz para a localização de robôs móveis, devido a não acúmulo de erros no processo de localização, o que

é comum em métodos baseados em sensores proprioceptivos (odometria) [5].

Deste modo, o trabalho de Carmo *et al.* [6] realiza um sistema para controlar a formação de dois robôs móveis em espaços inteligentes. Sua postura (posição e orientação no mundo) são obtidas por meio do reconhecimento de marcadores formados por centróides coloridos. Essas posições são convertidas em pontos tridimensionais usando visão computacional, com base no modelo de câmera *pinhole*. No entanto, o controle de robôs móveis requer o acompanhamento constante e ágil de sua postura no espaço. Normalmente, são utilizados marcadores e técnicas como ArUco para a detecção e localização desses padrões específicos nas imagens.

Um exemplo é o trabalho de Izabel [7], que apresenta um sistema de localização de robôs móveis em espaços inteligentes. Esta aplicação é constituída por serviços baseados em visão computacional que permitem a detecção e reconstrução da postura do robô móvel utilizando o marcador do tipo ArUco para localizá-lo em um ambiente multi-câmeras. A sua funcionalidade é de guiar pessoas com deficiência visual em um espaço com a utilização do robô como guia.

Outra abordagem possível para a detecção de objetos em imagens é o uso de redes neurais convolucionais (CNN - *Convolutional Neural Network*), treinadas para identificar objetos e classificá-los [8]. Sua vantagem na detecção de robôs é a independência em relação a marcadores fixos, podendo descartar a necessidade de acoplar marcadores diretamente ao robô. Essa independência é um ponto positivo visto que fixar um marcador no robô pode acabar sendo complicado dependendo do formato dele. Além disso, a detecção de marcadores é muito sensível a obstruções em relação às câmeras, algo que pode acontecer corriqueiramente em aplicações com obstáculos dinâmicos.

Diante desses desafios, neste artigo é proposto um sistema de localização capaz de reconstruir a postura de robôs móveis sem a necessidade de marcadores de reconhecimento no Espaço Inteligente do Instituto Federal do Espírito Santo, Campus Guarapari. Essa abordagem se baseia em técnicas de visão computacional, processamento de imagem e redes neurais artificiais para detectar, estimar e reconstruir a postura de robôs móveis em espaços inteligentes.

II. TRABALHOS RELACIONADOS

Uma aplicação de odometria visual em um sistema de localização pode ser vista em Cotta [5], em que a startup capixaba Vixsystem desenvolveu o robô Lysa, um robô móvel que desempenha funções semelhantes às de um cão-guia. Para implementar essas funcionalidades, foi desenvolvido um sistema denominado MobiLysa, que oferece serviços de localização em espaços inteligentes com múltiplas câmeras, utilizando marcadores do tipo ArUco acoplados ao robô e câmeras do ambiente para realizar a detecção desses marcadores.

Outra abordagem viável envolve a utilização de câmeras no robô em conjunto com marcadores no ambiente. No trabalho de Santos [9], a localização de robôs em ambientes internos é possível ao equipar o robô com uma câmera que detecta os marcadores, permitindo uma comparação com as leituras dos sensores de odometria.

Além da odometria visual para a identificação de marcadores, o trabalho de Foroughi [10], utiliza a CNN para localizar robôs móveis em ambientes internos. Neste trabalho, o robô possui uma câmera integrada que permite identificar objetos no ambiente utilizando a detecção e classificação providas pela CNN. Esses dados são então integrados a um mapa topológico do ambiente, que determina a posição do robô. Essa abordagem oferece uma alternativa para a localização por odometria visual baseada em marcadores, eliminando a sua necessidade e eliminando a necessidade de fixá-lo ao robô.

III. PROPOSTA

Nesta seção, será abordada a metodologia empregada no desenvolvimento deste trabalho, assim como os procedimentos adotados para validar a eficácia da detecção e localização de robôs móveis por meio de uma CNN. A principal intenção é realizar uma análise comparativa, especialmente em relação ao uso de marcadores visuais do tipo ArUco, com foco na localização em ambientes que possuam múltiplas câmeras.

A. Espaços Inteligentes

Os Espaços Inteligentes podem ser caracterizados pela integração de sensores e atuadores em um ambiente, sendo gerenciados por uma infraestrutura composta por hardware e software. Esses espaços tem como principal objetivo a coleta dados, processo de informações, tomada de decisões e a realização de tarefas de maneira eficiente e automatizada. Carmo [11] propõe a idealização de um Espaço Inteligente que combina a estrutura de ambientes programáveis com funcionalidades de orquestração multinível entre os serviços existentes. A sua arquitetura é baseada em microsserviços (MSA - *microservices architecture*), que é caracterizada pela divisão em estruturas pequenas e pela independência entre os microsserviços existentes [12].

Esses ambientes são definidos como Espaços Inteligentes programáveis (PIS - *Programmable Intelligent Space*), sendo adotados para a realização deste trabalho com o uso de visão computacional aplicada nas imagens providas pelo ambiente multicâmara. A Figura 1 exibe a estrutura física e virtual do PIS utilizado neste trabalho.

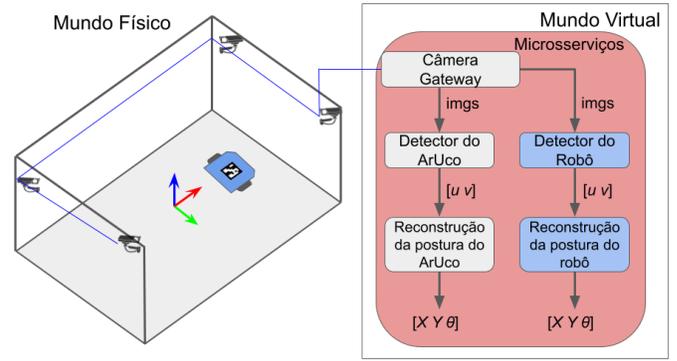


Figura 1. Estrutura física e virtual do PIS utilizado neste trabalho.

A Figura 1 exibe os microsserviços existentes na cor cinza que representa o método de localização do robô móvel utilizada atualmente no espaço. Já em azul, os microsserviços desenvolvidos para este trabalho utilizando CNN para a detecção do robô, sendo eles:

- **Câmera Gateway:** Responsável pela publicação de imagens das câmeras para os demais microsserviços, sendo um *Gateway* para cada câmera;
- **Detector do ArUco:** Responsável pela detecção do ArUco nas imagens, disponibilizando a localização dos seus vértices em u e v (coordenadas da imagem);
- **Reconstrução do ArUco:** Responsável pela reconstrução da postura do ArUco no referencial do mundo em X , Y e θ (coordenadas do mundo em metros e radianos) a partir da posição dos vértices do ArUco;
- **Detector do robô:** Responsável pela detecção do robô utilizando uma CNN, disponibilizando vértices de uma caixa delimitadora u e v ;
- **Reconstrução do robô:** Responsável pela reconstrução da postura do robô no referencial do mundo e X , Y e θ (coordenadas do mundo em metros e radianos) a partir da posição dos vértices da caixa delimitadora.

B. Estimativa de localização utilizando visão computacional

A localização por meio de visão computacional é realizada utilizando as câmeras do PIS, com o auxílio do processamento digital de imagens. Essa técnica de odometria visual é adotada devido à sua precisão na localização, ao mesmo tempo em que evita o acúmulo de erros, que é comum na odometria por *encoders* realizada pelo próprio robô [5].

Para garantir a precisão na estimativa da postura, é esperado que o sistema de múltiplas câmeras esteja devidamente calibrado. Durante o processo de calibração, um conjunto de parâmetros são gerados, os quais podem ser categorizados em dois grupos: parâmetros intrínsecos e extrínsecos. Os parâmetros intrínsecos englobam a distância focal f , os fatores de escala s_x , s_y , a posição do centro óptico da câmera no plano de projeção u_o , v_o e o cisalhamento entre os eixos da câmera τ . Já os parâmetros extrínsecos são determinados pela matriz de rotação $R_{3 \times 3}$ e um vetor de translação $T_{3 \times 1}$ em relação a um referencial global [13].

A eq. (1) utiliza esses parâmetros para transformar as coordenadas homogêneas de um ponto do referencial do mundo $\tilde{M} = [X, Y, Z, 1]^T$ em coordenadas homogêneas da imagem em pixels $\tilde{m} = [u, v, 1]^T$.

$$\lambda \cdot \tilde{m} = \begin{bmatrix} fs_x & f\tau & u_0 \\ 0 & fs_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot [R \mid T] \cdot \tilde{M} \quad (1)$$

A partir das coordenadas de um ponto na imagem, é possível recuperar suas coordenadas tridimensionais no referencial do mundo, com a exceção de um fator de escala desconhecido, denotado como λ pela eq. (1). Portanto, é necessário calcular esse fator por meio de um método adequado. Para abordar esse problema, o PIS emprega duas estratégias de reconstrução que envolvem o uso de câmeras [14]:

- Uma câmera: É possível reconstruir o ponto tridimensional quando se conhece o plano onde o ponto está localizado, o que permite determinar o valor de Z . Caso o plano do chão seja considerado igual ao plano $Z=0$ e a altura do ponto constante, a partir da equação da eq. (1), obtém-se o sistema de equações da eq. (2). Esse sistema possui solução única para as incógnitas X , Y e λ , por Z ser constante, $\tilde{m} = [u, v, 1]^T$ é obtido pela imagem e as colunas a_n do resultado da multiplicação das matrizes dos parâmetros intrínsecos com os extrínsecos são conhecidas pelo fato do sistema estar calibrado.

$$\begin{bmatrix} a_1 & a_2 & \tilde{m} \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ \lambda \end{bmatrix} = \begin{bmatrix} -z \cdot a_3 & a_4 \end{bmatrix} \quad (2)$$

- Duas ou mais câmeras: Utilizando a eq. (3), que envolve o fator de escala λ e o ponto \tilde{m} da câmera, é possível empilhar as equações de cada câmera, formando um sistema de equações. A relação R e T das câmeras com o referencial do mundo permitem reconstruir o ponto tridimensional em coordenadas métricas sem ser necessário saber em qual o plano Z o ponto está.

$$\begin{bmatrix} -I & (K \cdot R)^{-1} \tilde{m} \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ \lambda \end{bmatrix} = R^{-1} \cdot T \quad (3)$$

O termo I e K da eq. (3) representam respectivamente a matriz identidade e a matriz dos parâmetros intrínsecos.

As duas estratégias são necessárias pelo fato de nem sempre ser possível detectar o objeto por mais de uma câmera.

C. Método de localização via marcador ArUco

Até o momento, a abordagem utilizada para determinar a localização de um robô no Espaço Inteligente envolve a identificação de um marcador ArUco que está acoplado ao robô, como ilustrado na Figura 2. Essa estratégia é apropriada para fornecer dados ao microsserviço de reconstrução devido à sua detecção eficiente pelo microsserviço encarregado de identificar o ArUco.

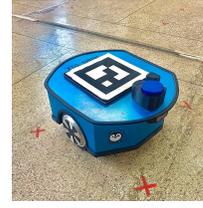


Figura 2. Robô móvel e o ArUco utilizado neste trabalho

O algoritmo de detecção do ArUco é capaz de fornecer informações de postura do objeto ao qual está vinculado em relação à câmera. O algoritmo identifica os vértices do marcador para determinar o ponto central e segmenta as bordas para obter sua orientação [15]. Por meio das técnicas de reconstrução descritas na Seção III-B, é possível estimar a postura deste ponto central em relação ao referencial do espaço. No entanto, é fundamental enfatizar que a precisão dessas informações só pode ser garantida quando o sistema está devidamente calibrado.

D. Método de localização via Redes Neurais Convolucionais

A detecção do robô da Figura 2 utilizando CNN, ao contrário do algoritmo de detecção do ArUco, necessita que seja treinada uma rede neural com um conjunto de dados do espaço que incluam imagens do robô e suas respectivas localizações nas imagens. Para isso, utilizou-se o YOLOv5, um detector de objetos em imagens conhecido por sua eficácia em comparação às versões mais antigas. A escolha se baseou em sua arquitetura aprimorada e melhor desempenho em comparação a outros detectores [16].

Em virtude disso, a rede neural foi treinada utilizando 3000 imagens, 1600 com a presença do robô e 1400 sem sua presença. Também foram usadas 500 imagens diferentes do treinamento para validar o modelo. Essa separação é fundamental para realizar a seleção do melhor modelo do treinamento e dos melhores ajustes dos hiper-parâmetros. Durante o processo, o melhor modelo é selecionado como modelo final [17]. Exemplos de imagens utilizadas no processo de treinamento estão apresentados na Figura 3.



Figura 3. Exemplo de imagens usadas no treinamento

Conforme Arjovsky *et al.* [18], para melhorar a capacidade de generalização do modelo, foram utilizadas imagens com variação em relação à iluminação e objetos presentes. A Figura 4 exibe uma imagem das câmeras com a detecção do robô no Espaço Inteligente utilizando a caixa delimitadora.



Figura 4. Imagem da detecção do robô no espaço

Após a detecção, o microserviço encarregado de reconstruir a postura do robô recebe os vértices da caixa delimitadora gerada pela rede neural que identifica o robô nas imagens. Em seguida, utiliza-se a mesma abordagem de reconstrução utilizada pelo algoritmo do ArUco, baseando-se nas técnicas de visão computacional. Dessa forma, é possível estimar a posição tridimensional do mundo do ponto central dessa caixa delimitadora.

No entanto, ao utilizar os vértices da caixa delimitadora, não é possível reconstruir a orientação do robô, uma vez que a detecção pela CNN treinada não é sensível às mudanças de orientação que o robô exerce durante seu deslocamento. Isso se difere com o ArUco, que permite a recuperação dessa orientação em relação a um referencial específico. Esse aspecto pode ser observado na Figura 5.

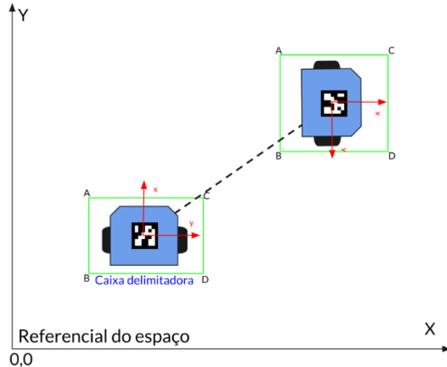


Figura 5. Representação da orientação do robô no espaço

Para resolver esse problema, são utilizadas posições passadas e atuais do robô para calcular um vetor de deslocamento e estimar o ângulo em relação a referência do espaço. No entanto, isso requer que o robô se desloque linearmente com velocidade positiva para obter estimativas precisas. Velocidades negativas resultam em orientações incorretas que se difere com a frente real do robô.

A eq. (4) é utilizada para realizar o cálculo dos ângulos usando as posições atuais e passadas estimadas pelo serviço de

reconstrução. É realizado um processo que utiliza a distância entre a posição atual $P(x, y)_t$ e a posição em instantes passados $P(x, y)_{t-n}$, em que o subíndice $t - n$ representa valores anteriores ao tempo t . Se o valor da norma deste vetor calculado entre os pontos for maior que NT , o ângulo θ_t é calculado. Caso contrário, o ângulo θ_t mantém seu valor anterior, sendo esse ângulo representado por θ_{t-n} .

$$\theta_t = \begin{cases} \arctan\left(\frac{\Delta P_y}{\Delta P_x}\right), & \sqrt{\Delta P_y^2 + \Delta P_x^2} \geq NT \\ \theta_{t-n}, & \sqrt{\Delta P_y^2 + \Delta P_x^2} < NT \end{cases} \quad (4)$$

O parâmetro NT é um valor arbitrário que evita o cálculo de ângulos quando o robô percorre distâncias curtas em relação à taxa de amostragem da detecção, que é cerca de 25 Hz. Isso ajuda a reduzir o impacto de ruídos, uma vez que a função \arctan é sensível a pequenas variações. Porém, aumentar o NT torna a reconstrução atrasada em relação ao ângulo real que o robô está direcionado.

IV. EXPERIMENTOS E RESULTADOS

Nesta seção, são apresentados os experimentos realizados para comparar o desempenho de localização proposto por este trabalho em relação ao método já existente de localização pelo ArUco no PIS do Campus Guarapari. Os dados da estrutura física utilizada para a realização deste experimento podem ser vistos na Tabela I.

Tabela I
TABELA DA ESTRUTURA FÍSICA UTILIZADA

| Quantidade | Equipamento | Descrição |
|------------|-------------|------------------------------------------------|
| 4 | Câmeras | Intelbras VIP 1230 D G2 f |
| 1 | Servidor | CPU: Intel i5-10400 GPU: 1x NVIDIA RTX 3050 |
| 1 | Robô | CPU: NVIDIA Jetson Nano |

A. Detecção do Robô móvel

Para a realização deste experimento, o robô foi programado para executar uma trajetória circular com um raio de 2 metros no espaço, sendo este percurso considerado como *ground truth*. Durante essa trajetória, foram registrados os pontos nos quais tanto o ArUco quanto o robô foram detectados, com o objetivo de obter métricas comparativas de quantidades de detecção, verdadeiros positivos e falsos negativos.

As câmeras do espaço registraram toda a trajetória ao longo de 397 quadros, e as comparações entre a detecção do ArUco e a detecção via CNN podem ser visualizadas na Tabela II. Nessa tabela, são apresentados os seguintes parâmetros para cada método de detecção: verdadeiros positivos (VP), que indicam detecções corretas do robô durante a trajetória; falsos negativos (FN), que correspondem às detecções não realizadas do robô na trajetória; e *Recall*, um parâmetro que determina a relação entre VP e a soma de VP + FN. Quanto maior o valor do *recall*, maior é a taxa de detecção.

Tabela II
PARAMETROS DE DETECÇÃO DO ARUCO E DO ROBÔ VIA CNN

| Câmeras | Deteção Robô via CNN | | | Deteção ArUco | | |
|----------|----------------------|----|--------|---------------|-----|--------|
| | VP | FN | Recall | VP | FN | Recall |
| Câmera 1 | 390 | 7 | 0.982 | 327 | 70 | 0.823 |
| Câmera 2 | 389 | 8 | 0.979 | 320 | 77 | 0.806 |
| Câmera 3 | 397 | 0 | 1 | 239 | 158 | 0.602 |
| Câmera 4 | 394 | 3 | 0.992 | 245 | 152 | 0.617 |

Conforme evidenciado na Tabela II, em todas as câmeras, a deteção por meio da CNN apresentou um desempenho superior em comparação com o algoritmo de deteção ArUco. Esse desempenho pode ser atribuído ao fato de que a deteção do ArUco pode falhar em casos nos quais uma parte do marcador esteja obstruída na imagem. Além disso, as condições de iluminação no ambiente podem influenciar as deteções. Já a CNN mostrou-se mais robusta nesse aspecto, uma vez que foi treinada com imagens com variações na iluminação.

A Figura 6 (a) exibe a trajetória realizada pelo robô com a deteção via CNN e a Figura 6 (b) a deteção do ArUco.

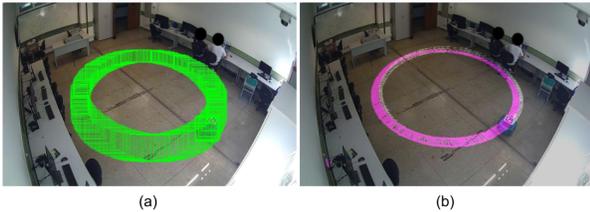


Figura 6. Deteção (a) do robô e (b) do ArUco vista pela câmera 1 do espaço

B. Reconstrução da postura

O propósito deste experimento é avaliar a precisão da reconstrução da postura do robô, comparando o desempenho do algoritmo do ArUco com a deteção via CNN. A partir dos pontos identificados pelo experimento da Seção IV-A, foi possível estimar a posição tanto do ArUco quanto do robô em relação ao referencial do espaço. Ambas as trajetórias foram representadas graficamente junto ao *ground truth* na Figura 7.

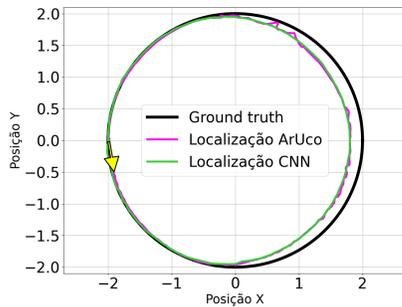


Figura 7. Representação da posição estimada pelos diferentes métodos

Em relação à trajetória de referência (*ground truth*), a Figura 8 exibe o erro euclidiano de ambas trajetórias. Neste caso, é necessário assumir que o robô percorreu de fato o que é representado pelo *ground truth*, isso porque, para o

experimento, o robô percorreu apenas uma volta da trajetória circular, assim, não acumulando um erro que o tirasse do percurso previsto.

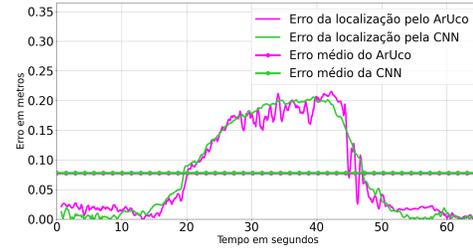


Figura 8. Representação do erro euclidiano pelos diferentes métodos de estimação da posição.

Conforme visto na Figura 8, o erro máximo da posição foi aproximadamente de 22 cm para o ArUco e 20,5 cm para o detector via CNN. O erro médio ao longo do percurso foi cerca de 8 cm para ambas as reconstruções. Esses resultados indicam que a reconstrução da deteção do robô via CNN fornece estimativas de posição aceitáveis, comparáveis ao microserviço de reconstrução do ArUco já utilizado no PIS.

Entretanto, na Figura 7, pode-se notar uma diferença na estimativa de posição em relação ao *ground truth* em uma parte do trajeto. Essa diferença pode ser atribuída à falta de precisão na calibração das câmeras nesta região, considerando também outros fatores, como a qualidade da imagem das câmeras e sua distorção radial. Esses elementos têm um impacto direto na reconstrução da posição desses pontos.

Por meio dessas posições, a segunda parte deste experimento teve como objetivo, reconstruir a orientação com base nos dados de posição estimados pelo microserviço de reconstrução via CNN. Isso envolve o uso dos pontos passados e atuais para calcular o ângulo do vetor de deslocamento entre esses dois pontos. A Figura 9 exibe as curvas cossenoidais dos ângulos estimados do *ground truth*, do ArUco, com *NT* igual a 1 e *NT* igual a 3.

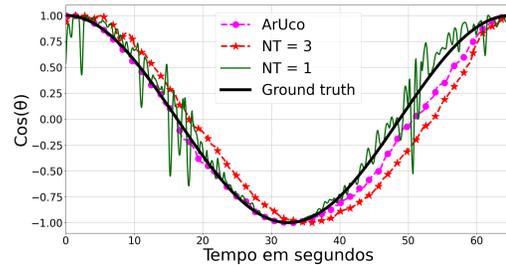


Figura 9. Curvas cossenoidais dos ângulos estimados

Na Figura 9, é notável que utilizando *NT* igual a 1, a orientação calculada apresenta muita variação em comparação com o ArUco e com a utilização do *NT* igual a 3. Isso se deve à sensibilidade da função *arctan* na eq. (4). Por meio das curvas cossenoidais exibidas na Figura 9, a Figura 10 apresenta os erros euclidianos do ArUco e do *NT* igual a 3 por alcançarem um desempenho aceitável em comparação com o *ground truth*.

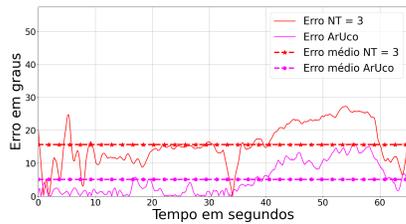


Figura 10. Erro euclidiano das estimativas dos ângulos

De acordo com a Figura 10, o erro médio do ArUco foi de aproximadamente 6 graus, enquanto o do NT igual a 3, chegou a cerca de 18 graus. Isso ocorre porque o ArUco calcula o ângulo instantaneamente após a reconstrução da sua postura, enquanto o microserviço de reconstrução angular com NT igual a 3 apresenta um atraso devido à condição imposta pela eq. (4). Portanto, é válido ressaltar que quanto maior o valor de NT , isto é, quanto maior for a norma do vetor de deslocamento, maior será o atraso do ângulo em relação à orientação real do robô, porém, melhor será a filtragem de ruídos. Quanto menor o valor de NT , maior o ruído, no entanto, menor será o atraso. Neste caso, o ArUco tem um desempenho melhor para estimar a orientação do robô.

V. CONCLUSÃO

Neste artigo, foi apresentada uma abordagem de localização para estimar a postura de robôs móveis em ambientes com múltiplas câmeras, como parte de um projeto de pesquisa no PIS do Campus Guarapari.

O principal objetivo deste trabalho foi desenvolver um serviço que empregasse um método de odometria visual que não dependesse da utilização de marcadores ArUco. O método proposto neste artigo utilizou uma CNN treinada para detectar o robô em imagens capturadas pelas câmeras do espaço e, posteriormente, estimar suas coordenadas métricas. Para a estimativa da orientação do robô, foi necessário criar um microserviço adicional.

O método proposto demonstrou ser promissor, apresentando uma melhor taxa de detecção em relação ao método utilizando marcadores ArUco, além de obter uma estimativa de postura similar. Como já comentado, o método utilizando CNN retorna ao espaço uma nova estimativa a uma taxa de 25 Hz, ou seja, num tempo adequado para as aplicações de controle de robôs, permitindo a substituição do método atual, não necessitando mais fixar um marcador visual à estrutura do robô.

Uma limitação deste trabalho é a estimativa de orientação, que não permite atualmente o cálculo do ângulo do robô durante movimentos em ré ou quando está girando em seu próprio eixo. Como desenvolvimento futuro, pode-se explorar a detecção simultânea de mais partes do robô pela rede neural. Isso acabaria com a limitação da estimativa de orientação do robô, não sendo mais necessário aguardar um deslocamento mínimo dele para a atualização do ângulo.

Por fim, em aplicações como a de cães-guia para deficientes visuais, o sistema poderia utilizar o detector atual do robô juntamente com um detector de pessoas, no caso para a

detecção do usuário humano. Isto, inclusive, pode ser usado para a estimativa conjunta da orientação do sistema (robô e humano).

VI. AGRADECIMENTOS

Este trabalho é parte do Projeto “Cooperação entre VANT e robô terrestre para SLAM em espaços inteligentes”, financiado pela FAPES - Fundação de Amparo à Pesquisa e Inovação do Espírito Santo, por meio do Edital N° 03/2021 - Universal.

REFERÊNCIAS

- [1] Almeida, C. M. d. S., “A importância da aprendizagem da robótica no desenvolvimento do pensamento computacional: um estudo com alunos do 4º ano.”, Tese (Doutorado) — UNIVERSIDADE DE LISBOA, 2015.
- [2] Cotta, W.A.A.; Lopes, S.I.; Vassallo, R.F. Towards the Cognitive Factory in Industry 5.0: From Concept to Implementation. *Smart Cities* 2023, 6, 1901-1921. <https://doi.org/10.3390/smartcities6040088>
- [3] Rampinelli, M. et al. An intelligent space for mobile robot localization using a multi-camera system. *Sensors*, v. 14, n. 8, p. 15039–15064, 2014. ISSN 1424-8220.
- [4] MILANO, D. de; HONORATO, L. B. Visão computacional. Faculdade de Tecnologia, Universidade Estadual de Campinas, 2010
- [5] Cotta, W. A. A. et al. Mobilysa-sistema de localização e controle do cão-guia robô lysa para ambientes internos baseado em visão computacional. In: SBC. Anais Estendidos do XXV Simpósio Brasileiro de Sistemas Multimídia e Web. [S.l.], 2019. p. 159–162
- [6] Carmo, A. do et al. Uso de um Espaço Inteligente baseado em visão computacional para o controle de formação de robôs móveis. In: Anais do XII Simpósio Brasileiro de Computação Ubíqua e Pervasiva. Porto Alegre, RS, Brasil: SBC, 2020. p. 171–180. ISSN 2595-6183. Disponível em: <https://sol.sbc.org.br/index.php/sbcup/article/view/11223>.
- [7] Izabel, V. et al. Implantação do sistema mobilysa em espaços inteligentes programáveis. In: Anais do XIV Simpósio Brasileiro de Computação Ubíqua e Pervasiva. Porto Alegre, RS, Brasil: SBC, 2022. p. 61–70. ISSN 2595-6183. Disponível em: <https://sol.sbc.org.br/index.php/sbcup/article/view/20611>.
- [8] Lecun, Y.; Kavukcuoglu, K.; Farabet, C. Convolutional networks and applications in vision. In: Proceedings of 2010 IEEE International Symposium on Circuits and Systems. [S.l.: s.n.], 2010. p. 253–256.
- [9] G. d. S. Santos, E. Cardoso, and M. A. d. Reis, “Localização de robôs móveis em ambiente internos usando marcadores fiduciais,” in Anais do V Simpósio Internacional de Inovação e Tecnologia., 2019, pp. 226–233.
- [10] Foroughi, F.; Chen, Z.; Wang, J. A cnn-based system for mobile robot navigation in indoor environments via visual localization with a small dataset. *World Electric Vehicle Journal*, v. 12, n. 3, 2021. ISSN 2032-6653. Disponível em: <https://www.mdpi.com/2032-6653/12/3/134>.
- [11] Carmo, A. P. do. Uma Arquitetura de Microserviços centrada na Observabilidade Multinível para Espaços Inteligentes baseados em Visão Computacional. Tese (Doutorado) — Universidade Federal do Espírito Santo Centro Tecnológico - Departamento de Engenharia Elétrica Programa de Pós-Graduação em Engenharia Elétrica, Janeiro 2021.
- [12] Nadareishvili, I., Mitra, R., McLarty, M., and Amundsen, M. (2016). Microservice architecture: aligning principles, practices, and culture. “O’Reilly Media, Inc.”.
- [13] Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334.
- [14] Queiroz, F. *et al* Estimating Tridimensional Coordinates of Skeleton Joints in a Multicamera System. In: Anais do XIV Workshop de Visão Computacional [S.l.:s.n.], 2018
- [15] Sampathkrishna, A. (2022). ArUco Maker based localization and Node graph approach to mapping. arXiv preprint arXiv:2208.09355.
- [16] Redmon, J. et al. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], 2016.
- [17] Géron, A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O’Reilly Media, Inc., 2019. Disponível em: <https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/>.
- [18] Arjovsky, M., Bottou, L., Gulrajani, I., Lopez-Paz, D. (2019). Invariant risk minimization. arXiv preprint arXiv:1907.02893.