SLAM Visual Em Ambientes Dinâmicos Usando Segmentação Panóptica

Gabriel F. Abati[1], **João Carlos V. Soares**[2], **Marco Antonio Meggiolaro** [1]

[1]Departamento de Mecanica – Pontificia Universidade Catolica do Rio de Janeiro (PUC-Rio)
Rio de Janeiro – RJ – Brazil

[2]Departamento de ciencias mecanicas e engenharia – Universidade de Illinois
Illinois, U.S.A.

fischerabati@gmail.com, virgolinosoares@gmail.com, meggi@puc-rio.br

Abstract. *The majority of visual SLAM systems are not robust in dynamic scenarios. The ones that deal with dynamic content in the scenes usually rely on deep learning-based methods to detect and filter dynamic objects. However, these methods cannot deal with unknown objects. This work presents Panoptic-SLAM, a visual SLAM system robust to dynamic environments, even in the presence of unknown objects. It uses Panoptic Segmentation to filter dynamic objects from the scene during the state estimation process. The proposed methodology is based on ORB-SLAM3 [Campos et al. 2021], a state-of-the-art SLAM system for static environments. The implementation was tested using real-world datasets and compared with several systems from the literature, including DynaSLAM, DS-SLAM and SaD-SLAM and PVO.*

Resumo. *A maioria dos sistemas de SLAM visual não é robusta em cenários dinâmicos. Aqueles que lidam com conteúdo dinâmico nas cenas geralmente dependem de métodos baseados em aprendizado profundo para detectar e filtrar objetos dinâmicos. No entanto, esses métodos não conseguem lidar com objetos desconhecidos. Este trabalho apresenta o Panoptic-SLAM, um sistema de SLAM visual robusto para ambientes dinâmicos, mesmo na presença de objetos desconhecidos. Ele utiliza a Segmentação Panóptica para filtrar objetos dinâmicos da cena durante o processo de estimativa de estado. A metodologia proposta é baseada no ORB-SLAM3, um sistema SLAM estado-da-arte para ambientes estáticos. A implementação foi testada usando conjuntos de dados do mundo real e comparada com vários sistemas da literatura, incluindo DynaSLAM, DS-SLAM e SaD-SLAM e PVO.*

Submissão ao CTDR (mestrado)
Data de conclusão do mestrado: 17 de Maio de 2023

1. Introdução

Mapeamento e localização simultâneos (SLAM) é um problema crucial para diversas aplicações, como veículos autônomos, drones e robôs móveis, pois permite que os robôs naveguem e operem em ambientes previamente desconhecidos. SLAM consiste na construção de um mapa de um ambiente desconhecido e na determinação simultânea da posição do robô dentro desse ambiente.

Os sistemas de SLAM visual estão recebendo cada vez mais atenção na comunidade de robótica devido ao baixo custo e à riqueza de informações fornecidas pelas câmeras. Existem vários sistemas de SLAM visual na literatura com alta precisão usando câmeras monoculares [Mur-Artal et al. 2015][Engel et al. 2014] estéreo [Mur-Artal and Tardós 2017] e câmeras RGB-D [Engel et al. 2018].

No entanto, esses sistemas de SLAM não estão preparados para atuar em cenários com objetos em movimento, o que resulta em localização imprecisa e mapeamento inconsistente. Para lidar com isso, existem várias abordagens para incorporar elementos dinâmicos a um sistema de SLAM visual. Recentemente, métodos baseados em aprendizado profundo tem sido explorados para sistemas de SLAM em ambientes dinâmicos, fornecendo informações de alto nível sobre as cenas.

Alguns métodos[Soares et al. 2021][Yang et al. 2020] utilizam detecção de objetos, como o YOLO [Redmon and Farhadi 2018], para localizar e rastrear objetos rotulados, filtrando os *keypoints* dos objetos dinâmicos. Outros métodos [Yu et al. 2018] [Liu and Miura 2021] [Zhang et al. 2022] utilizam segmentação semântica combinada com geometria epipolar para filtrar os *keypoints* dinâmicos. Os métodos abordados por [Bescos et al. 2018] [Yuan and Chen 2020] [Vincent et al. 2020] dependem de técnicas de segmentação de instâncias, como Mask R-CNN [He et al. 2017] ou YOLOACT [Bolya et al. 2019]. O principal problema com todas as abordagens anteriores é a necessidade de ter um número pre-determinado de classes que podem ser detectadas e, consequentemente, filtradas. Em outras palavras, se um objeto não rotulado estiver se movendo na cena, ele não seria filtrado e suas características causariam um desvio na localização e valores discrepantes no mapa.

A Segmentação panóptica [Kirillov et al. 2019b] é uma tarefa de visão computacional que combina tanto a segmentação de instâncias quanto a segmentação semântica. Na segmentação de instâncias, os objetos na imagem são identificados e segmentados ao nível de *pixel*, enquanto na segmentação semântica, cada *pixel* na imagem é rotulado com uma categoria semântica. O objetivo da segmentação panóptica é unificar essa duas tarefas em uma única, onde todos os *pixels* na imagem são rotulados com um rótulo de instâncias, indicando a qual objeto ele pertence, ou um rótulo semântico, indicando a qual categoria ele pertence.

Recentemente, alguns métodos de SLAM visual têm sido propostos usando segmentação panóptica para lidar com cenários dinâmicos, como o PVO [Ye et al. 2022], que utiliza o sistema de odometria visual do DROID-SLAM [Teed and Deng 2021] e combina a segmentação panóptica com o fluxo óptico para criar uma representação de fluxo óptico panóptico do ambiente. Embora seu módulo de odometria visual possa obter poses precisas da câmera, os autores não especificaram como as informações desconhecidas do modelo de segmentação panóptica afetam seu sistema. Além disso, o PVO pode funcionar de forma robusta em cenas dinâmicas, mas ignora a detecção de fechamento de loop quando a câmera retorna a uma localização previamente visitada

Este trabalho propõe Panoptic-SLAM, um sistema de SLAM visual para ambientes dinâmicos que utiliza a segmentação panóptica para detectar e filtrar objetos em movimento, mesmo na presença de objetos desconhecidos em movimento. As contribuições deste artigo podem ser resumidas da seguinte forma.

- Um sistema de SLAM visual open-source que utiliza a segmentação panóptica.
- Um método robusto para lidar com objetos desconhecidos em movimento

- A partir dos testes utilizando conjuntos de dados que explicitamente possuem objetos dinâmicos desconhecidos na cena, são obtidos os melhores resultados em comparação com outros sistemas em diversas sequências.

O artigo está organizado na seguinte forma: a seção 2 detalha trabalhos anteriores de SLAM visual em ambientes dinâmicos utilizando abordagens baseadas em aprendizado profundo. A seção 3 descreve a metodologia proposta. Na seção 4, o Panoptic-SLAM é avaliado utilizando os conjuntos de dados dinâmicos TUM e BONN, comparativamente com sistemas da literatura. Por fim, as conclusões e trabalhos futuros são apresentados na seção 5.

2. Revisão Bibliográfica

Trabalhar com objetos dinâmicos é um desafio na pesquisa de SLAM visual. A maioria dos pesquisadores trata os objetos dinâmicos como valores discrepantes (*outliers*), e diversos sistemas de SLAM visual foram propostos para lidar com esses valores discrepantes. Essas soluções podem ser amplamente categorizadas em dois grupos principais: métodos baseados em geometria e métodos baseados em aprendizado. Os métodos baseados em geometria dependem de técnicas clássicas de visão computacional para detectar e filtrar conteúdo dinâmico. Em geral, eles possuem acurácia inferior em comparação com os métodos baseados em aprendizado. Sua principal vantagem é não exigir conhecimento prévio sobre os objetos na cena.

Por outro lado, os métodos baseados em aprendizado requerem um modelo pre-treinado. Geralmente, eles utilizam detecção de objetos, segmentação semântica, segmentação de instâncias ou, mais recentemente, segmentação panóptica. O trabalho entitulado Crowd-SLAM [Soares et al. 2021], baseado no ORB-SLAM2, utiliza um modelo de detecção de objetos personalizado com arquitetura YOLO-Tiny [Redmon and Farhadi 2018] para localizar pessoas na imagem e remover características dinâmicas dentro das caixas delimitadoras previstas. [Li and Chen 2022] propuseram um sistema que combina detecção de objetos e uma abordagem baseada em geometria. O sistema também é baseado em ORB-SLAM2 com rastreamento DeepSort e geometria epipolar para detectar pontos estáticos de cada objeto na cena.

O DS-SLAM [Yu et al. 2018] combina a rede de segmentação semântica SegNet com um algoritmo de verificação de consistência de movimento baseado em fluxo óptico para reduzir o impacto de objetos dinâmicos e gera um mapa semântico denso em formato de *octree* do ambiente. O RDS-SLAM [Liu and Miura 2021] é um método baseado em ORB-SLAM3 que inclui um *thread* paralela de segmentação semântica com modelo SegNet. Isso permite que o processo de rastreamento opere continuamente sem esperar por novas informações semânticas. Além disso, os autores introduziram uma estratégia de seleção de quadros-chave para segmentação semântica cujo objetivo é obter as informações semânticas mais recentes possíveis, independentemente da velocidade do método de segmentação. Para atualizar e propagar informações semânticas, os autores utilizaram a probabilidade de movimento para detectar e eliminar discrepâncias de rastreamento por meio de um algoritmo de associação de dados.

Nenhum dos métodos citados anteriormente consegue lidar com objetos moveis desconhecidos. [Ji et al. 2021] apresentaram uma abordagem de SLAM semântico com câmera RGB-D que opera em ambientes dinâmicos, extraindo informações semânticas apenas dos quadros-chave. Apesar de ser capaz de lidar com objetos desconhecidos usando *k*-

means e reprojeção de profundidade, sua precisão é inferior a outros métodos, como o DynaSLAM, em ambientes com pessoas. O DynaSLAM [Bescos et al. 2018] é um dos primeiros trabalhos a utilizar segmentação de instâncias para detectar e identificar pessoas na cena ao nível de *pixel*, o que é usado para filtrar características dinâmicas. Este trabalho é um dos sistemas de SLAM visual com melhor precisão no conjunto de dados de referência da TUM [Sturm et al. 2012].

SaD-SLAM proposto por [Yuan and Chen 2020] integra informações de profundidade e segmentação de instâncias do Mask R-CNN para identificar características dinâmicas em imagens. O algoritmo classifica cada ponto de características como estático, dinâmico ou estático e móvel. O SaD-SLAM apresenta alta precisão, superando o DynaSLAM em certos cenários. No entanto, sua principal limitação é que a segmentação semântica é realizada de forma *offline*. O Dot-Mask [Vincent et al. 2020], utiliza a segmentação de instâncias para obter informações de *pixel* sobre objetos em uma imagem e um filtro de Kalman Estendido para rastreá-los. Os autores buscaram desenvolver um sistema de SLAM mais rápido em detrimento de uma precisão menor em comparação com outras abordagens. O principal problema das abordagens baseadas em segmentação de instâncias para estimativa de pose visual é a falta de informações de cenário de fundo (*background*), o que diminui a possibilidade de inferir sobre a existência de objetos moveis desconhecidos.

Recentemente, tem havido um aumento no uso da segmentação panóptica em sistemas de estimação de estado visual, como o mencionado PVO. O SVG-LOOP [Yuan et al. 2021] apresenta um algoritmo de detecção de *loop closure* que utiliza uma combinação de um modelo semântico de *bag-of-words* processado com segmentação panóptica para minimizar o impacto das características dinâmicas, e um modelo de vetor de *landmark* semânticos para codificar as conexões geométricas dentro do grafo semântico.

O autores de [Zhu et al. 2022] propuseram um método baseado em ORB-SLAM2 que incorpora segmentação panóptica e informações de geometria. Para minimizar o impacto de objetos desconhecidos em movimento, os autores propõem uma abordagem de classificação de objetos dinâmicos baseados em geometria epipolar. Apesar de afirmarem robustez contra objetos em movimento desconhecidos, isso não foi avaliado numericamente em seu artigo. Além disso, existe a eliminação *a priori* de todos os *keypoints* pertencentes ao que consideram altamente dinâmico, sem considerar objetos moveis conhecidos que podem ser estáticos (como veículos, por exemplo).

3. Metodologia

A Figura 1 mostra um diagrama da abordagem proposta. O sistema SLAM é baseado no ORB-SLAM3 e é composto por quatro *threads* executadas em paralelo: segmentação panóptica, rastreamento (*tracking*), mapeamento local (*local mapping*) e *loop closure*. Primeiramente, os *frames* são processados tanto na *thread* de rastreamento quanto na *thread* de segmentação panóptica. Os recursos ORB são extraídos na *thread* de rastreamento, e a imagem é segmentada em objetos, plano de fundo e informações desconhecidas na *thread* de segmentação panóptica. Os objetos conhecidos são enviados para um algoritmo de *Short-Term data association* para determinar se são novos ou estavam presentes no último *frame*. Os recursos associados ao plano de fundo são correspondidos com os do último *frame* e usados para calcular uma matriz fundamental. Usando essa matriz fundamental, os *keypoints* associados a objetos conhecidos e desconhecidos são classificados

como dinâmicos ou estaticos. Apenas recursos estáticos são usados para o mapeamento e *loop closure*.

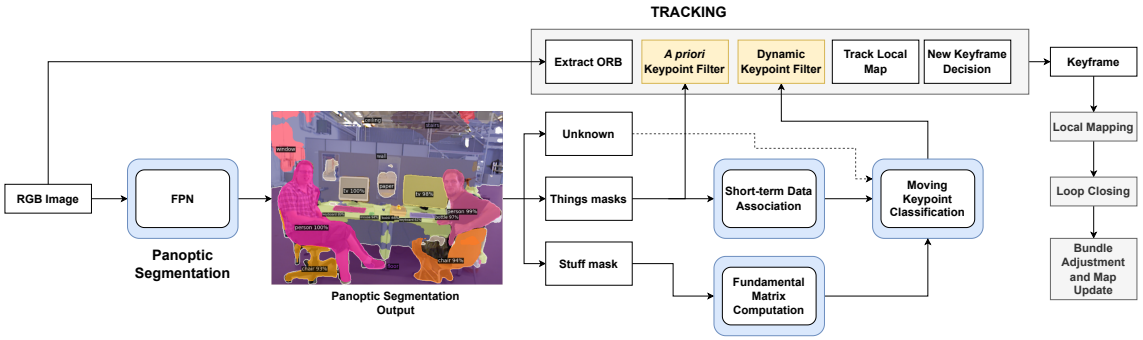

Figura 1. Diagrama de blocos do sistema proposto

3.1. Segmentação Panóptica

Na segmentação panóptica, os *pixels* são classificados como *Thing* ou *Stuff*. Os objetos *Thing* são objetos contáveis com fronteiras bem definidas e potencialmente móveis, como pessoas, animais ou veículos. Os objetos *Stuff*, por outro lado, referem-se a regiões amorfas, não contáveis da imagem, principalmente "imóveis", como céu, chão ou paredes. A segmentação por instâncias identifica objetos individuais na categoria *Stuff*, enquanto a segmentação semântica rotula todos os *pixels* da imagem com sua categoria correspondente. Com alta precisão e eficiência, esse método de segmentação supera modelos anteriores baseados em caixas *box-based* ou sem caixas *box-free*.
Além disso, algumas partes da imagem são desconhecidas, ou seja, regiões em que modelo de segmentação panóptica não conseguiu prever nenhum rótulo. As máscaras desconhecidas podem ocorrer devido a desfoque de movimento ou objetos não rotulados na cena. Também pode haver detecções incorretas, ou seja, objetos com classificação errada. Isso pode ocorrer se um objeto na cena for rotulado, mas houver outra classe com características semelhantes (por exemplo, TV e monitor), ou se um objeto não rotulado for semelhante a uma classe rotulada. Este trabalho utiliza o PanopticFPN[Kirillov et al. 2019a] para inferência, treinado com o conjunto de dados COCO [Lin et al. 2014], que pode segmentar até 80 rótulos diferentes de objetos *Thing* e 91 rótulos de objetos *Stuff*.

3.2. Filtro de *Keypoint* Dinâmicos

O método proposto para detecção e filtragem de *keypoints* dinâmicos é dividido em quatro processos: filtragem de *keypoints a priori*, cálculo da matriz fundamental, *short-Term data association* e classificação de *keypoints* móveis de *Things* e objetos desconhecidos. Primeiramente, o modelo panóptica gera todas as previsões de máscaras. Os *keypoints* das pessoas são filtrados a priori, pois os seres humanos podem ser considerados altamente dinâmicos e apenas em situações raras eles permanecem completamente imóveis por um longo período. Os *keypoints* pertencentes a máscara *Stuff* são usadas para calcular a matriz fundamental com o RANSAC. Para calcular a matriz fundamental, é necessário identificar características correspondentes em ambas as vistas. O algoritmo de *Feature Matching* envolve encontrar pares de pontos nas duas vistas que correspondem ao mesmo ponto 3D na cena. Sem uma correspondência de características precisa, a matriz fundamental não

pode ser calculada com precisão, o que pode resultar em erros em tarefas subsequentes. O algoritmo de *Feature Matching* combina os descritores ORB dos *keypoints* nas imagens de referência e de consulta usando uma abordagem de vizinho mais próximo *Nearest Neighbor*. Especificamente, para cada *keypoint* na imagem de referência, o algoritmo procura pelo *keypoint* mais próximo na imagem de consulta com base na similaridade de seus descritores ORB.
A matriz fundamental é usada para mapear os pontos de características do *frame* anterior para o seu domínio de pesquisa correspondente no *frame* atual, ou seja, a linha epipolar. Supondo que os pontos correspondentes nos *frames* atual e anterior sejam p1 e p2, respectivamente, sua forma de coordenada homogênea pode ser representada como P1 e P2.

$$P1_j = [u_1, v_1, 1], P2_j = [u_2, v_2, 1] \tag{1}$$

Onde "u"e "v"representam as coordenadas de *pixel* no *frame* da imagem, e "j"simboliza a categoria do ponto de características (*Thing* ou desconhecido). Usando esses valores, pode-se calcular a linha epipolar, denotada por L:

$$L \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = FP1_j \tag{2}$$

Dado que X, Y, Z representam vetores de linha e F representa a matriz fundamental, a distância entre um ponto correspondente e sua linha epipolar correspondente pode ser determinada como:

$$D_j = \frac{P2_j^T F P1_j}{\sqrt{||X||^2 + ||Y||^2}} \tag{3}$$

A ideia é ter a matriz fundamental calculada com *keypoints* estáticos e usá-la com as correspondências de características de *Thing* e objetos desconhecidos para determinar se a distância de um ponto correspondente a sua linha epipolar correspondente é menor que um certo limite. Se a distância for menor que o limite, o *keypoint* correspondente é considerado estático.

3.3. *Short-Term Data Association*

A classificação de *keypoints* em movimento de objetos *Thing* usa um algoritmo de *Short-Term data association* para lidar com o problema de múltiplos objetos com o mesmo rótulo em um *frame*. Este algoritmo avalia, usando a métrica de interseção sobre a união (IoU), mostrada na Eq. 4, se uma nova máscara de *Thing* detectada no *frame* atual corresponde a uma máscara *Thing* no *frame* anterior. Cada máscara de instância prevista com sua respectiva *bounding box*. Para cada par de *frames* novos no tempo de k e k-1, o algoritmo verifica se há sobreposição entre duas *bounding boxes* do mesmo rótulo. Uma vez detectada uma sobreposição, o algoritmo extrai os contornos do objeto C^k e C^{k-1} para calcular a IoU e determinar se ambos os contornos possuem uma associação.

$$IoU = \frac{|C^k \cap C^{k-1}|}{|C^k \cup C^{k-1}|} \tag{4}$$

Após a associação de dados, os *keypoints* do *frame* atual e do *frame* anterior pertencente a objetos *Thing* com o mesmo rótulo e mesmo ID de rastreamento são pareados. D_{Thing} é calculado usando características pareadas e a matriz fundamental para determinar quais pontos são dinâmicos e, consequentemente, filtrados.
As características pertencentes aos *pixels* desconhecidos são localizadas adicionando todas as máscaras conhecidas em uma unica imagem e analisando as áreas pretas nela, conforme mostrado na Figura 2. O processo é semelhante à classificação de *keypoints* moveis de *Thing*. Após os *keypoints* correspondentes do *frame* atual e do *frame* anterior serem encontrados na máscara desconhecida, esses pontos são verificados usando a matriz fundamental para calcular a distância epipolar D_{unk} e determinar se eles são estáticos ou dinâmicos. Os pontos não correspondidos são filtrados.

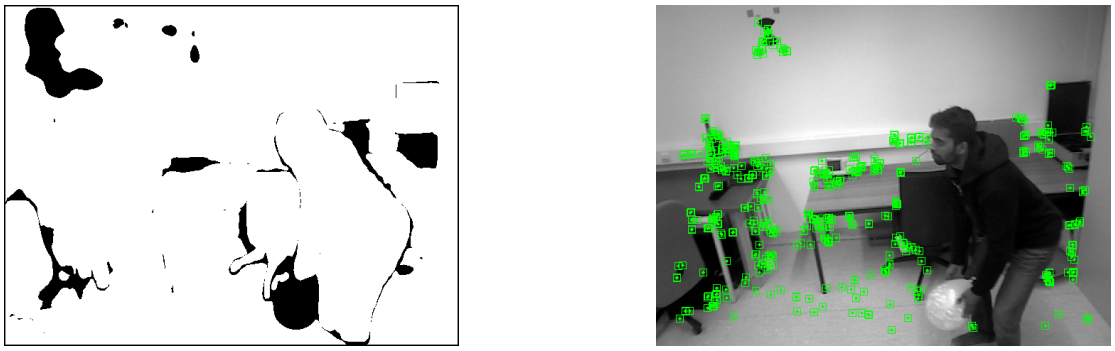

Figura 2. Exemplo de um objeto móvel desconhecido sendo filtrado em uma sequência do *BONN Dataset*

4. Resultados

Foram utilizadas sequência de dois conjuntos de dados de referência diferentes para avaliar o sistema e compará-lo com métodos estado-da-arte. Todos os testes foram realizados cinco vezes e os valores medianos foram selecionados.

4.1. TUM Dataset

O conjunto de dados TUM [Sturm et al. 2012] é usado para avaliar a robustez do sistema em ambientes dinâmicos. Estão inclusas sequências de imagens RGB e de profundidade capturadas por uma câmera *Microsoft Kinect*, juntamente com trajetórias de referência correspondentes. Os dados foram gravados com uma resolução de 640×480 e frequência 30 Hz. Quatro sequências foram escolhidas para a avaliação: *fr3 w static*, *fr3 w xyz*, *fr3 w rpy*, *fr3 w halfsphere*. Essas sequências retratam duas pessoas caminhando por uma sala, passando atrás de uma mesa, passando na frente da câmera e sentando em cadeiras. A principal diferença entre cada sequência é o movimento da câmera. Na sequência xyz, a câmera é movida ao longo dos tres eixos mantendo uma orientação fixa. Na sequência rpy, a câmera é girada em torno dos eixos de *roll*, *pitch* e *yaw*, mantendo uma posição fixa. Na sequência *halfsphere*, a câmera segue uma trajetória ao longo de uma meia esfera. Na sequência *static*, a câmera permanece parada. A consistência global da trajetória

estimada é analisada usando o Erro Absoluto de Trajetória (ATE). Essa métrica compara as distâncias absolutas entre os componentes de translação das trajetórias de referência e estimadas.

A Tabela 1 mostra a comparação do ATE entre o Panoptic-SLAM e diversos métodos da literatura. Os melhores resultados são destacados em negrito. Com base nos resultados da comparação, o Panoptic-SLAM alcançou uma precisão similar ao DynaSLAM. Isso provavelmente se deve ao fato de que o DynaSLAM filtra as pessoas antecipadamente, enquanto o conjunto de dados *fr3 walking* da TUM tem principalmente pessoas em movimento. O método de [Ji et al. 2021] possui uma precisão inferior ao sistema proposto, apesar de também ser robusto em relação a rótulos desconhecidos nos ambientes. O método proposto supera [Zhu et al. 2022], que também utiliza segmentação panóptica.

Tabela 1. Comparação dos valores RMSE da métrica ATE [m] do sistema proposto contra resultados de ORB-SLAM3, PVO, ReFusion, DynaSLAM, DS-SLAM, SaD-SLAM, DOT-Mask, Ji *et al.* [Ji et al. 2021], e Zhu *et al.* [Zhu et al. 2022] utilizando o *TUM Dataset*

Sequência	*fr3_w_static*	*fr3_w_xyz*	*fr3_w_rpy*	*fr3_w_half*
Panoptic-SLAM	0.009	**0.014**	**0.032**	**0.025**
ORB-SLAM3	0.038	0.819	0.957	0.315
PVO	0.007	0.018	0.056	0.221
ReFusion	0.017	0.099	—	0.104
DynaSLAM	**0.006**	0.015	0.035	**0.025**
DS-SLAM	0.008	0.024	0.444	0.030
SaD-SLAM	0.017	0.017	**0.032**	0.026
DOTMask	0.008	0.021	0.053	0.040
Ji *et al.* [Ji et al. 2021]	0.011	0.020	0.037	0.029
Zhu *et al.* [Zhu et al. 2022]	0.013	0.018	0.039	0.030

4.2. BONN Dataset

O conjunto de dados BONN [Palazzolo et al. 2019] é usado para avaliar a robustez em relação a objetos em movimento, pessoas e objetos em movimento não rotulados. Também utiliza as mesmas métricas de avaliação do conjunto de dados TUM. Foram escolhidas seis sequências, filmadas em um ambiente interno, para avaliação: *Balloon*, *Balloon2*, *non-obstructing box* (*non-obst box*) 1 e 2, *placing non-obstructing box* (*placing no box*) 1 e 2. As sequências *Ballon* mostram uma pessoa caminhando e interagindo com um balão flutuante. Na sequência *Non-obst box*, uma pessoa move uma caixa de papelão de um lugar para outro. Nas sequências de *placing no box*, uma pessoa aparece e coloca uma caixa de papelão no chão. É importante ressaltar que as classes de balão não estão presentes

no conjunto de dados COCO e, consequentemente, não podem ser detectadas pelo nosso modelo de segmentação.
A Tabela 2 mostra o RMSE da comparação de ATE entre o Panoptic-SLAM e o ORB-SLAM3, DynaSLAM e ReFusion. Os resultados do ReFusion e do DynaSLAM foram obtidos em [Palazzolo et al. 2019]. Nosso sistema superou o ORB-SLAM3 em todas as sequências. Nosso sistema também superou os outros sistemas em todas as sequências, exceto na última, onde alcançou resultados semelhantes ao DynaSLAM, com uma diferença de 1 mm. Apesar do fato de o DynaSLAM ter alcançado resultados semelhantes aos do Panoptic-SLAM no conjunto de dados TUM, o mesmo não ocorre no conjunto de dados BONN devido à presença de objetos em movimento desconhecidos. Isso é evidente nos resultados de *non-obst box* e *placing-no-box*, onde o DynaSLAM é dez vezes maior que os resultados do Panoptic-SLAM em ordem de magnitude.

Tabela 2. Comparação dos valores RMSE da métrica ATE [m] do sistema proposto contra resultados ORB-SLAM3, DynaSLAM, e ReFusion utilizando o *BONN Dataset*

Sequência	Panoptic-SLAM	ORB-SLAM3	DynaSLAM	ReFusion
Non-obst box	**0.027**	0.347	0.232	0.071
Non-obst box2	**0.033**	0.043	0.039	0.179
Balloon	**0.029**	0.092	0.030	0.175
Balloon2	**0.027**	0.215	0.029	0.254
placing_no_box	**0.044**	0.842	0.575	0.106
placing_no_box2	0.022	0.023	**0.021**	0.141

Para mostrar a importância de cada etapa de nossa metodologia, foram realizados experimentos utilizando três configurações diferentes do sistema: com apenas o filtro de pessoas, o filtro de pessoas juntamente com o filtro de objetos em movimento conhecidos e o filtro de pessoas juntamente com o filtro de objetos em movimento desconhecidos. Foram escolhidas duas sequências desafiadoras do conjunto de dados BONN que contêm objetos em movimento desconhecidos: *non-obst box* e *placing-non-box*. A Tabela 3 mostra o RMSE do ATE das diferentes configurações, comparado ao sistema combinado. Devido à presença de objetos em movimento conhecidos e desconhecidos nessas cenas, todas as configurações falham, exceto o sistema combinado.

4.3. Análise de Desempenho

Todos os testes foram realizados em um notebook com processador Intel i7, 16 GB de RAM e GPU NVIDIA RTX3060 com 8 GB de VRAM, executando o sistema operacional Ubuntu 20.04 Linux. O sistema SLAM é implementado em C++, e a inferência de

Tabela 3. Avaliação do ATE no *BONN Dataset* utilizando Panoptic-SLAM com diferentes configurações [m]

Filtro	non-obst box	placing_no_box
Filtro de pessoas	0.481	0.707
Filtro de objetos moveis	0.029	0.765
Filtro de objetos desconhecidos	0.302	0.721
Panoptic-SLAM	**0.027**	**0.044**

segmentação panóptica é implementada em Python, utilizando o framework Detectron2 [Wu et al. 2019]. A Tabela 4 mostra o tempo médio de rastreamento do sistema em quatro sequências do conjunto de dados TUM. A média do tempo de inferência da segmentação panótica foi de 0,2 segundos por *frame* para todas as sequências.

Tabela 4. Tempo de *tracking* médio [s]

Sequência	Tempo de *tracking* médio [s]
fr3_w_static	0.344
fr3_w_xyz	0.344
fr3_w_rpy	0.319
fr3_w_half	0.323

5. Conclusões

Este artigo apresentou o Panoptic-SLAM, um sistema SLAM visual de *open-source* construído sobre o ORB-SLAM3, que opera em tempo real e é robusto em ambientes dinâmicos, mesmo na presença de objetos em movimento desconhecidos e não rotulados. Uma segmentação de cena panóptica foi proposta para classificar *keypoints* estáticos e dinâmicos. Demonstramos a eficácia do método proposto comparando-o com vários sistemas da literatura considerados tendo a maior precisão em ambientes dinâmicos, incluindo DynaSLAM, DS-SLAM, SaD-SLAM e PVO em sequências dinâmicas desafiadoras. Os resultados indicam que o Panoptic-SLAM não apenas alcança os mesmos níveis de precisão do DynaSLAM em cenas altamente dinâmicas, mas também supera em uma grande margem em cenários com objetos em movimento desconhecidos. Para trabalhos futuros, planejamos lidar com objetos em movimento que ocupam uma grande parte da imagem para evitar a perda de rastreamento. Além disso, temos como objetivo incluir no método a construção de um mapa semântico que possa se adaptar ao longo do tempo, explorando ainda mais as capacidades da segmentação panóptica.

Referências

Bescos, B., Fácil, J. M., Civera, J., and Neira, J. (2018). Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4):4076–4083.

Bolya, D., Zhou, C., Xiao, F., and Lee, Y. J. (2019). Yolact: Real-time instance segmentation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9156–9165.

Campos, C., Elvira, R., Rodríguez, J. J. G., Montiel, J. M. M., and Tardós, J. D. (2021). Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890.

Engel, J., Koltun, V., and Cremers, D. (2018). Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):611–625.

Engel, J., Schöps, T., and Cremers, D. (2014). Lsd-slam: Large-scale direct monocular slam. In *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II*, pages 834–849.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

Ji, T., Wang, C., and Xie, L. (2021). Towards real-time semantic rgb-d slam in dynamic environments. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11175–11181.

Kirillov, A., Girshick, R., He, K., and Dollar, P. (2019a). Panoptic feature pyramid networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6392–6401.

Kirillov, A., He, K., Girshick, R., Rother, C., and Dollár, P. (2019b). Panoptic segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9396–9405.

Li, G. and Chen, S. (2022). Visual slam in dynamic scenes based on object tracking and static points detection. *Journal of Intelligent & Robotic Systems*, 104(33).

Lin, T., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Liu, Y. and Miura, J. (2021). Rds-slam: Real-time dynamic slam using semantic segmentation methods. *IEEE Access*, 9:23772–23785.

Mur-Artal, R., Montiel, J. M. M., and Tardós, J. D. (2015). Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163.

Mur-Artal, R. and Tardós, J. D. (2017). Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262.

Palazzolo, E., Behley, J., Lottes, P., Giguère, P., and Stachniss, C. (2019). Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals. In

2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 7855–7862.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Soares, J. C. V., Gattass, M., and Meggiolaro, M. A. (2021). Crowd-slam: Visual slam towards crowded environments using object detection. *Journal of Intelligent & Robotic Systems*, 102(2).

Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. (2012). A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580.

Teed, Z. and Deng, J. (2021). Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In *Neural Information Processing Systems*.

Vincent, J., Labb'e, M., Lauzon, J., Grondin, F., Comtois-Rivet, P., and Michaud, F. (2020). Dynamic object tracking and masking for visual slam. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4974–4979.

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., and Girshick, R. (2019). Detectron2. `https://github.com/facebookresearch/detectron2`.

Yang, S., Fan, G., Bai, L., Zhao, C., and Li, D. (2020). Sgc-vslam: A semantic and geometric constraints vslam for dynamic indoor environments. *Sensors*, 20(8).

Ye, W., Lan, X., Chen, S., Ming, Y., rong Yu, X., Bao, H., Cui, Z., and Zhang, G. (2022). Pvo: Panoptic visual odometry. *ArXiv*, abs/2207.01610.

Yu, C., Liu, Z., Liu, X.-J., Xie, F., Yang, Y., Wei, Q., and Fei, Q. (2018). Ds-slam: A semantic visual slam towards dynamic environments. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1168–1174.

Yuan, X. and Chen, S. (2020). Sad-slam: A visual slam based on semantic and depth information. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4930–4935.

Yuan, Z., Xu, K., Zhou, X., Deng, B., and Ma, Y. (2021). SVG-Loop : Semantic – Visual – Geometric Information-Based Loop Closure Detection.

Zhang, J., Gao, M., He, Z., and Yang, Y. (2022). Dcs-slam: A semantic slam with moving cluster towards dynamic environments. In *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1923–1928.

Zhu, H., Yao, C., Zhu, Z., Liu, Z., and Jia, Z. (2022). Fusing panoptic segmentation and geometry information for robust visual slam in dynamic environments. In *IEEE 18th International Conference on Automation Science and Engineering*.