

ANÁLISE DA FERRAMENTA DEEP-IMAGE-MATCHING PARA LOCALIZAÇÃO TOPOLÓGICA

Carlos Eduardo A. da Silva¹ (IC), Giovani Bernardes Vitor (PQ)²

¹ Universidade Federal de Itajubá, ² Universidade Federal de Itajubá

Palavras-chave: Comparação de imagens. Extração de características. Localização topológica. Veículos inteligentes.

Introdução

No contexto dos veículos inteligentes a localização topológica, e o mapeamento e identificação de imagens e obstáculos são fatores essenciais para garantir o funcionamento adequado desses sistemas (Garcia-Fidalgo e Ortiz, 2015), a fim de garantir a sua completa autonomia.

A capacidade de extrair características de imagens e compará-las com outras, possui grande aplicação na área da localização topológica. A ferramenta Deep-Image-Matching (DIM) é capaz de realizar a extração de características e fazer a comparação (*matching*) de um conjunto de imagens qualquer (Morelli et al, 2024), além disso o software possui suporte para diferentes métodos de extração de características e de comparação (*matchers*), neste trabalho foi testada a aplicação dessa ferramenta no mapeamento e localização topológica.

Na literatura existente, são encontrados diversos trabalhos, com diferentes abordagens e aplicações relacionadas ao mapeamento e à localização topológica, cada uma explorando formas variadas de implementação. Em R. Neto et al (2024) um ambiente *outdoor*, semelhante ao analisado neste estudo, é explorado a partir de algoritmos de visão computacional clássica, como SURF, KNN e RANSAC, integrados ao framework ROS2.

Sánchez (2023) desenvolveu um algoritmo para localização topológica em ambientes externos utilizando imagens aquisitadas em um veículo. Durante o processo de localização, as imagens são analisadas pelo descritor SURF, que extrai os *keypoints*. Em seguida, o algoritmo KNN Match classifica os dados com base em clusters, e a regressão linear (RANSAC) é utilizada para determinar a correspondência entre os pontos extraídos.

Nessa perspectiva, a abordagem baseada em *Deep learning* em aplicações de visão computacional,

possui grande adaptabilidade para uma gama de diferentes de cenários de iluminação, diferenças temporais e de ângulos de vista entre imagens (Morelli et al, 2024).

As imagens foram colhidas de uma gravação de um percurso durante o dia, em um veículo convencional, a partir do vídeo foram extraídas 374 imagens presentes no *dataset* utilizado para compor o caminho. A fim de validar o sistema foi retirado de outras gravações em momentos variados um conjunto de 3 imagens para representar cada um dos nós, que corresponde a um determinado local. No total foram definidos 8 nós, entretanto, o caminho que forma o *dataset*, só possui imagens comuns com os nós 1, 2, 6, 7 e 8 respectivamente, o caminho percorrido e a disposição dos nós pode ser observado na Figura 1.

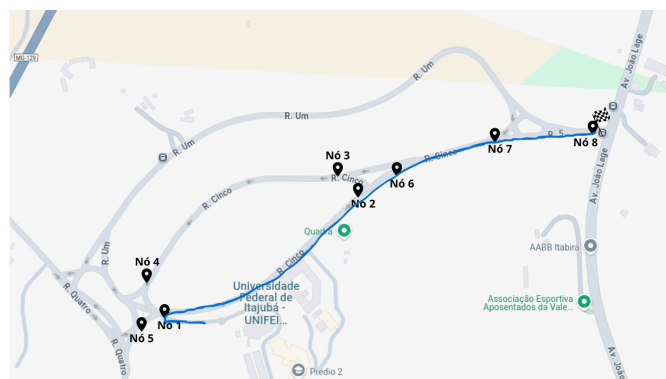


Figura 1 - Mapa do caminho, representado pela linha em azul.

Metodologia

O fluxograma mostrado na Figura 2, descreve os passos realizados para realização da proposta.

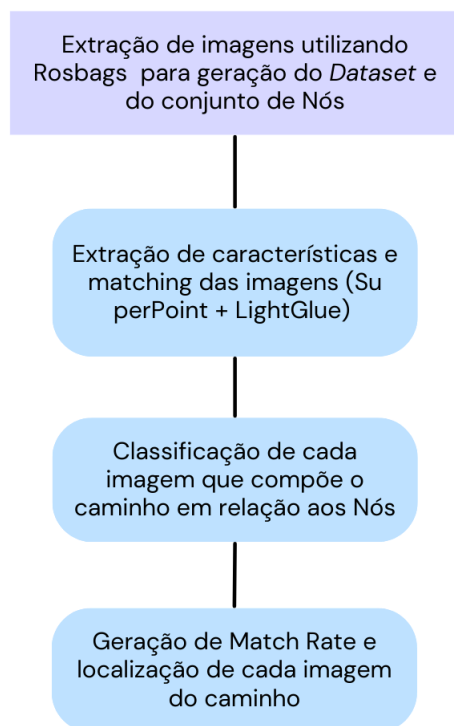


Figura 2 - Fluxograma para realização da proposta

A máquina utilizado para execução do DIM foi o sistema operacional Ubuntu 22.04 LTS com uma placa de vídeo RTX 3060, que possibilitou executar a ferramenta em CUDA, plataforma de computação paralela e modelo de programação criado pela NVIDIA, que aumenta significativamente o desempenho do algoritmo.

A extração de imagens foi realizada utilizando vídeos gravados a partir de um veículo com auxílio da câmera estéreo ZED, os frames foram retirados dos vídeos com uma frequência de 10Hz, utilizando o pacote Rosbags do framework ROS2.

A ferramenta DIM possibilita a utilização de diversos extratores e comparadores, no presente estudo foi escolhido utilizar algoritmos baseados em *Deep learning*. Como extrator foi utilizado o SuperPoint (DeTone et al., 2018) e como comparador (*matcher*) utilizado foi o LightGlue (Lindenberg et al., 2023).

Foi desenvolvido também, um algoritmo em Python, utilizando a biblioteca h5py, a fim de ler os resultados gerados a cada *matching* das imagens do caminho com cada um dos Nós e calcular seus respectivos Match Rates (MR), possibilitando analisar similaridade entre as imagens segundo o DIM. O cálculo

de MR é demonstrado pela Equação 1.

$$MR = \frac{matches}{(Kp_{nó} + Kp_{img})/2} \quad (1)$$

Onde, *matches* é a quantidade de keypoints (elementos extraídos das imagens) equivalentes entre a imagem analisada do *dataset* e a imagem comparada do conjunto de Nós. O $Kp_{nó}$ a quantidade de keypoints totais da imagem do Nó atualmente comparado e Kp_{img} a quantidade de keypoints totais da imagem do *dataset* que forma o caminho. Todos esses dados são gerados pela ferramenta e lidos pelo algoritmo Python utilizando a biblioteca citada.

Para que uma imagem pertença a um nó é necessário que o seu MR ultrapasse o *threshold* estabelecido de 15%. Nesse caso, o algoritmo classifica a imagem como pertencente ao Nó em questão.

O resultado do *match* entre duas imagens utilizando a configuração (SuperPoint + Lightglue), como extrato e comparador respectivamente, pode ser observado na Figura 3, gerada pela ferramenta.

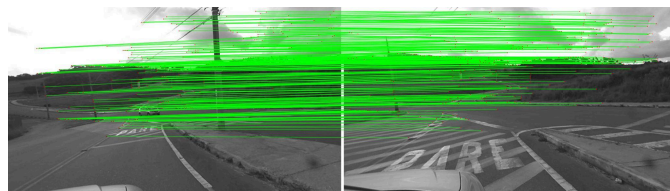


Figura 3 - Matches entre duas imagens representando pelas linhas em verde.

Resultados e discussão

A partir dos dados obtidos com o algoritmo, foi possível elaborar algumas medidas a fim de comprovar a eficácia das classificações de imagens semelhantes. O gráfico apresentado na Figura 4, apresenta o MR de todas as imagens do caminho.

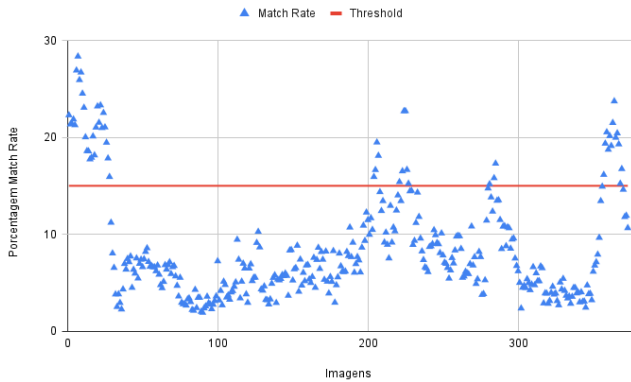
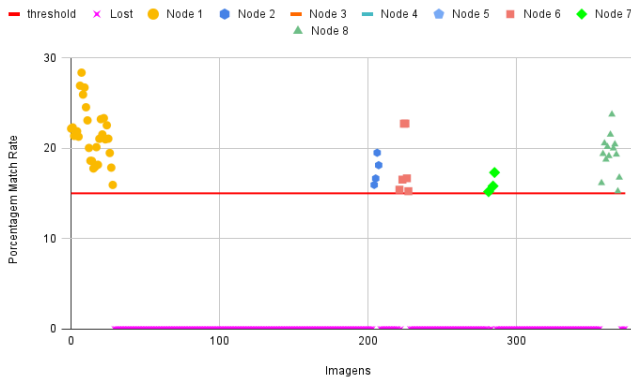


Figura 4 - Match Rate de todas as imagens dataset

É possível observar os momentos em que a porcentagem de MR ultrapassa o *threshold* estabelecido, simbolizando a correspondência entre as imagens e um determinado Nó. Na Figura 5 é mostrado apenas os Match Rates acima do *threshold*, juntamente com as identificações dos nós correspondentes.

Figura 5 - Match Rates das imagens acima do *threshold*

Como previsto, para o caminho percorrido, os Nós encontrados foram apenas os números 1, 2, 6, 7 e 8. Nos casos em que o limiar de 15% não era alcançado o Nó retornado era -1 que remete a um estado perdido, simbolizado pela linha em rosa na Figura 5.

Além disso, foi possível verificar a precisão dos acertos e falhas dos resultados gerados pelo sistema, utilizando uma matriz de confusão, exposta na Figura 6. Foi possível determinar uma acurácia de 95,72% calculada conforme a Equação 2

$$\text{Acurácia} = \frac{\text{Nº de predições corretas}}{\text{Nº total de predições}} \quad (2)$$

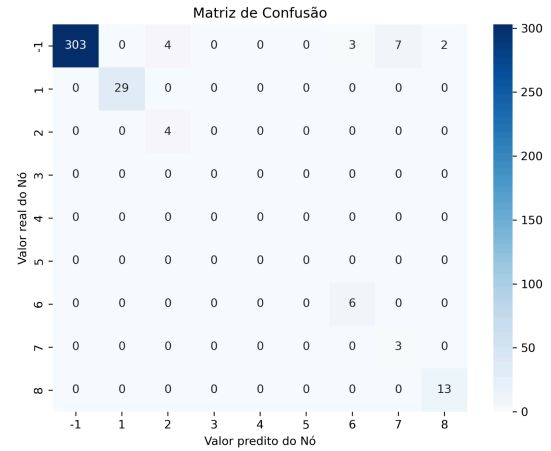


Figura 6 - Matriz de confusão

A diagonal principal da matriz corresponde a quantidade de predições corretas do DIM.

Esses resultados mostram a eficiência da ferramenta na comparação de imagens, considerando o momento do dia e as condições climáticas testadas — durante o dia e com clima limpo. Nesse contexto, a ferramenta apresentou uma boa acurácia, mesmo com apenas 3 imagens na composição de cada Nó.

Comparado aos algoritmos de visão computacional que utilizam outros métodos, como o apresentado no trabalho de R. Neto et al. (2024), que possui uma acurácia de 83,79% utilizando 20 imagens em cada nó, o DIM apresenta uma precisão de 95,72%. É importante destacar, no entanto, que técnicas baseadas em algoritmos são particularmente valiosas em cenários onde o uso de *deep learning* pode ser excessivo ou inviável, especialmente quando há escassez de dados rotulados.

Conclusões

Por meio das validações propostas, foi possível concluir que a ferramenta Deep-Image-Matching possui grande precisão no reconhecimento de similaridade entre imagens, sendo possível considerar seu uso em aplicações na navegação em veículos autônomos e localização topológica.

Contudo, é importante ressaltar a necessidade de adaptações no uso da ferramenta para a função proposta. Neste estudo, foi necessário desenvolver um script em Python para mover as imagens da pasta do *dataset* que compõe o caminho, uma de cada vez, para a pasta onde

estavam os Nós. Além disso, o script gerava um arquivo de texto, que posteriormente seria utilizado pelo DIM, contendo os pares de imagens que deveriam ser comparados pela ferramenta — nesse caso, a imagem da pasta do caminho e todas as imagens que compunham cada um dos Nós. Em seguida, o script aciona a ferramenta, lê os resultados gerados por ela, calcula e registra o valor de MR. E, por fim, apaga a imagem inicialmente copiada e repete o processo para todas as outras imagens do *dataset*.

Entretanto, a ferramenta DIM não está completamente finalizada (Morelli et al, 2024), e portanto futuras funcionalidades que podem vir a ser implementadas para facilitar seu uso na temática proposta.

Outro fator a ser considerado é o tempo de execução, que é um fator importante para a finalidade almejada. As comparações entre as imagens não ocorrem em tempo real. Para as 24 imagens que compõem o conjunto dos Nós, são necessários cerca de 3 segundos* de processamento utilizando o DIM para comparar cada uma das imagens do *dataset*. Além disso, há o tempo necessário para que o script desenvolvido realize a inclusão e exclusão das imagens a serem comparadas e acione a ferramenta via interface de linha de comando.

* *Tempo computado de execução com máquina utilizada.*

Agradecimentos

Gostaria de expressar meus mais sinceros agradecimentos a Deus pela minha vida, e por me ajudar a superar todos os obstáculos. Aos meus pais e irmãos que me apoiam e incentivam amavelmente. Um agradecimento especial ao meu orientador, Giovani Bernardes Vitor, pela paciência, oportunidade e ensinamentos. Assim como, agradeço a Universidade Federal de Itajubá (Unifei), a qual faço parte, e que tornou esse projeto possível. Também gostaria de agradecer grandemente a PIBIC Unifei, que foi o órgão financiador da minha bolsa de pesquisas, pelo incentivo e oportunidade.

Referências

Morelli, L. et al. **DEEP-IMAGE-MATCHING: A toolbox for multiview image matching of complex scenarios**. The International Archives of the

Photogrammetry, Remote Sensing and Spatial Information Sciences, v. XLVIII-2/W4, p. 309–316, 2024. DOI: 10.5194/isprs-archives-XLVIII-2-W4-2024-309-2024.

Garcia-Fidalgo, E. e Ortiz, A. (2015). **Vision-based topological mapping and localization methods: A survey**, Robotics and Autonomous Systems 64: 1–20.
URL: <https://www.sciencedirect.com/science/article/pii/S0921889014002619>

Neto, Antonio Rola; VITOR, Giovani Bernardes; OLIVEIRA, Tiago Gaiba de. (2024) **Localização topológica baseada em visão computacional para aplicação em veículos inteligentes**. CONGRESSO BRASILEIRO DE AUTOMAÇÃO (CBA), 2024.

Sánchez, N.S. (2023). **Sistema de localização topológica utilizando visão computacional**. Master's thesis, UNIVERSIDADE FEDERAL DE ITAJUBÁ

DeTone, D., Malisiewicz, T. and Rabinovich, A., (2018). **Superpoint: Self-supervised interest point detection and description**. Proc. CVPR workshops, pp. 224-236.

Lindenberger, P., Sarlin, P.E. and Pollefeys, M., (2023). **LightGlue: Local Feature Matching at Light Speed**. arXiv preprint arXiv:2306.13643.