

SISTEMAS PARA PERCEPÇÃO DO ESPAÇO LIVRE À FRENTE DE UM VEÍCULO E CÁLCULO DA DISTÂNCIA DE SEUS LIMITES

Diogo Ebert Gatti¹; Eduardo Lobo Lustosa Cabral²

¹Aluno de Iniciação Científica do Instituto Mauá de Tecnologia (IMT);

²Professor do Instituto Mauá de Tecnologia (IMT).

Resumo. *Este trabalho apresenta o desenvolvimento de um sistema de percepção para veículos autônomos focado na detecção do espaço livre e no cálculo das distâncias de seus limites a partir de imagens. Utilizando o veículo de pesquisa ORBI, foram capturadas e analisadas mais de 12.000 imagens com suporte de modelos de IA para segmentação de imagens (SAM, FastSAM e SAM2) e de cálculo de distâncias (MiDas, ZoeDepth e Depth-pro). Além disso, um sensor LiDAR foi integrado no sistema para estudo e análise. Os resultados indicam que o modelo SAM2 apresenta maior precisão e velocidade de segmentação, enquanto o Depth-pro é o modelo mais eficiente para o cálculo de distâncias métricas. A integração das imagens de câmeras com os resultados de distâncias obtidas pelo LiDAR demonstrou potencial para se ter redundância das medidas e uma maior precisão do sistema, confirmando a viabilidade da sua aplicação em veículos autônomos.*

Introdução

A pesquisa em veículos autônomos tem avançado significativamente nas últimas décadas, impulsionada pela necessidade de sistemas de transporte mais seguros e eficientes. Um dos desafios cruciais nessa área é a determinação do espaço livre à frente do veículo, o que é fundamental para a navegação e a tomada de decisões em tempo real. Este trabalho tem como objetivo investigar métodos para calcular e monitorar o espaço livre à frente de um veículo autônomo, utilizando técnicas de sensoriamento, tais como, LiDAR, câmeras e inteligência artificial, com alta precisão para garantir a confiabilidade do sistema. Ao desenvolver algoritmos que integram dados de sensores para identificar obstáculos e espaço livre, espera-se contribuir para a melhoria da segurança e eficiência na operação de veículos autônomos em ambientes dinâmicos.

A capacidade de compreender a dimensão do espaço livre à frente, ou seja, a distância possível de ser percorrida é importante para que o sistema de navegação do veículo seja viável. Existem diversas abordagens para resolver esse problema, como visto nos artigos BRAGA, José Renato et.al (2016), GONÇALVES, Luis Felipe, et.al (2011) e KLASER, Rafael Luiz, et.al (2014). Modelos de segmentação e cálculo de distâncias têm evoluído rapidamente, mas a escolha dos métodos mais adequados e sua integração em um sistema funcional continuam sendo questões em aberto, passíveis de aprimoramento.

Nesse aspecto, o desenvolvimento de um sistema para a percepção do entorno de veículos autônomos, focado na detecção e segmentação do espaço livre e na estimativa das distâncias dos limites do percurso é o desafio que se busca enfrentar. Para isso, foram avaliadas diferentes abordagens baseadas em redes neurais e na fusão com dados de um LiDAR, com o intuito de identificar os métodos mais adequados para viabilizar a navegação autônoma em tempo real.

A realização desta pesquisa justifica-se pela relevância do tema para o avanço da tecnologia aplicável aos veículos autônomos, os quais prometem benefícios amplos à sociedade, como a redução de acidentes de trânsito, assim como maior eficiência nos sistemas de transporte.

Aparato experimental

O sistema desenvolvido neste trabalho tem por objetivo, com a utilização de apenas imagens de uma única câmera, obter a percepção do entorno de um veículo autônomo, com foco na detecção

e na segmentação do espaço livre, além de estimar as distâncias dos limites que integram o espaço livre que se pode percorrer.

São utilizadas neste projeto cerca de 12.000 imagens coletadas no campus do Instituto Mauá de Tecnologia com a utilização do veículo de pesquisa ORBI (Ônibus Robótico Inteligente), conforme ilustrado na Figura 1. Esse veículo é equipado com câmeras acopladas na parte superior do para-brisa, como se observa na Figura 2, apesar de haver duas câmeras instaladas no veículo, somente utilizou-se a imagem de 1 câmera para os métodos. Além disso, na parte frontal do ônibus está instalado um sensor LiDAR, que utiliza feixes de laser para medir distâncias, conforme mostrado na Figura 3. O LIDAR utilizado é um RS-Helios-16P, que possui 16 linhas verticais de medição, operado em uma faixa de frequência entre 10-20Hz. O LIDAR tem um campo de visão de 360°, porém no ORBI foi limitado a 180°, visto que o objeto de estudo é o espaço à frente do veículo. Esta limitação foi realizada nas configurações do sensor. Todos os dados obtidos foram utilizados no estudo por meio de scripts python e softwares como MATLAB, em computadores pessoais e esporadicamente no Super computador do IMT



Figura 1 - Veículo de pesquisa ORBI



Figura 2 - Câmeras Instalados no ORBI



Figura 3 - LiDAR instalado na frente do ORBI

Modelos de IA para segmentação

Identificar e quantificar o espaço por onde o veículo pode se locomover é requisito essencial para um veículo autônomo. Inicialmente é importante entender a diferença de detecção e segmentação. Na qual a detecção identifica a presença de objetos em uma imagem e delimita de forma geral a sua posição, enquanto a segmentação vai além da detecção e identifica quais pixels pertencem a um objeto ou região específica na imagem. Na detecção do espaço livre deve-se considerar a presença de obstáculos, tais como, pessoas e outros veículos. A abordagem adotada nesse trabalho é a identificação, única e exclusivamente, da rua onde o veículo se locomove com a premissa de que o espaço detectado é o espaço livre. Nesse trabalho não é realizada a identificação do tipo de obstáculo presente no entorno do veículo. Porém, ressalta-se que isso poderia ser realizado facilmente usando as mesmas imagens usadas para determinação do espaço livre.

Os seguintes modelos de IA pré-treinados foram utilizados para a etapa de segmentação da rua:

- SAM (Segment Anything), desenvolvido por Kirillov, Mintun, et.al (2023);
- FastSAM, desenvolvida por Zhao, Ding, et.al (2023); e
- SAM2, uma segunda versão do SAM desenvolvida por desenvolvido por Ravi, Gabeur, et.al (2024).

Todos os modelos utilizados são open-source, ou seja, são abertas para o uso.

O modelo SAM foi criado para realizar a segmentação de objetos a partir de comandos do usuário, que podem ser: (i) um ponto, ou seja, a posição de um dos pixels que compõem o objeto desejado; (ii) caixas delimitadoras, que representam a região que se deseja segmentar; (iii) inserção de máscaras como comando, nas quais o SAM busca os objetos na imagem que se assemelham à máscara; e (iv) o comando por texto, que é a opção escolhida para ser utilizada nesse trabalho. No comando por texto, insere-se o nome do objeto que se deseja segmentar que, neste projeto, foi descrito como “street”. O SAM utiliza um codificador de imagem baseado no modelo ViT (Visual Transformers), que é um modelo para reconhecer na imagem tanto os padrões locais quanto os contextos globais. Após realizar o processamento da imagem, o SAM segmenta todos os objetos presentes e em seguida utiliza o comando do usuário para retornar somente a região desejada pelo usuário. O grande diferencial desse modelo foi o vasto banco de dados de treinamento, contendo um bilhão de máscaras e onze milhões de imagens.

O modelo FastSAM tomou como base o modelo SAM, porém apresenta uma velocidade de processamento mais rápida mantendo a alta qualidade. Este modelo opera com os mesmos tipos de comandos de usuários. O FastSAM utiliza uma rede CNN (Rede neural convolucional) treinada com apenas 2% do banco de dados da rede SAM.

O modelo SAM2 é uma evolução do SAM, que propõe realizar a segmentação em velocidades altas suficientes para a utilização em aplicações em tempo real. O SAM2 tem o mesmo tipo de arquitetura que o SAM, ou seja, é um modelo ViT. O SAM2 se diferencia do SAM em alguns aspectos e um deles é não possuir o comando por texto. Desta forma, nesse trabalho é utilizado o comando por pontos. Nesse caso, a utilização de poucos pixels para indicar a rua, faz com que, esporadicamente, algumas manchas presentes no asfalto sejam consideradas como objeto em face da alta qualidade da segmentação. Tal problema é resolvido com a indicação de 2 pontos em posições diferentes na parte inferior da imagem (local onde está a rua). Uma diferença entre o SAM e o SAM2 é que o SAM só funciona em imagens estáticas enquanto o SAM2 funciona também para vídeos. O SAM2 foi treinado com um banco de dados de mais de 35 milhões de máscaras e mais de 50 mil vídeos.

Independentemente do modelo utilizado, nesse trabalho foi necessário realizar a redução no tamanho das imagens com a intenção de ganhar velocidade de processamento. Após isso, a partir da máscara que o modelo retorna, são definidos os pixels que compõem o limite da rua. Estes limites são obtidos na forma de um vetor onde cada valor representa a linha do último pixel que compõe a rua para cada coluna da imagem, ou seja, em uma imagem 1024x512 (coluna x linha) é obtido um vetor de uma dimensão com 1024 valores. Como se percebe na figura 4 e 5 onde, a primeira é o gráfico obtido ao plotar o vetor, e a segunda figura mostra a rua, a qual os seus limites seriam a representação dos pontos plotados

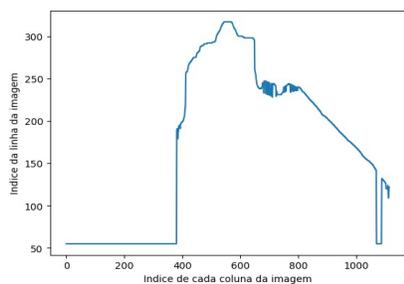


Figura 4 – gráfico obtido ao plotar os vetores obtidos das IAs



Figura 5 - Imagem utilizada para obter o vetor

Cálculo de distâncias

A próxima etapa, após a obtenção do limite da rua, é o cálculo das distâncias desses limites ao veículo. Para se realizar essa tarefa, dois métodos distintos são utilizados. O primeiro método é o uso de modelos de IA pré-treinados. São testados três modelos: 1) o MiDas, desenvolvido por Ranftl, Lasinger, et.al (2023); o ZoeDepth, desenvolvido por Bhat, Birkl, et.al (2023); e o Depth-pro, desenvolvido por Bochkovskii, Delaunoy, et.al (2024). Todos os modelos utilizados são open-source, ou seja, são abertas para o uso.

O segundo método é a fusão da imagem, com a rua segmentada, com os dados do sensor LiDAR. Esse método é baseado no trabalho de An, Ma, et al (2020), sendo que exige a calibração conjunta da câmera e do LIDAR. Essa calibração é realizada com um toolbox fornecido pelo software MATLAB.

A utilização de cada um desses métodos apresenta sequências diferentes de coleta e tratamento de dados durante a sua aplicação em um contexto real, o que afeta a velocidade do funcionamento de cada sistema. A sequência de etapas que cada método de cálculo de distância proporciona é melhor explicado posteriormente.

Modelos de IA para determinação de distâncias

Existem dois tipos de modelos pré-treinados que calculam distâncias a partir de uma única imagem. Eles podem retornar a distância relativa ou a distância métrica. Na relativa, a distância de cada pixel em relação a outro é retornada, sendo a distância real obtida pelo conhecimento da posição de dois pontos, o que é inviável neste projeto, já que o entorno de um veículo está em constante mudança. Por essa razão, os modelos utilizados retornam distâncias métricas, assim, os resultados desses modelos é uma matriz de mesma dimensão da imagem, onde cada valor da matriz corresponde à distância que cada pixel representa na cena real em relação à câmera.

Para testar a qualidade de cada modelo são utilizadas imagens com pontos com distâncias conhecidas. Foram obtidas 14 imagens, sendo que em cada uma tem 4 a 5 pontos conhecidos.

O MiDas é um modelo capaz determinar tanto a distância métrica quanto a distância relativa. Esse modelo foi originalmente treinado com imagens em ambientes internos e externos. O uso do MiDas nesse trabalho resultou em valores de distâncias absolutas com grandes erros, apesar dos erros nas distâncias relativas entre pixels não ser muito grande. Notou-se que quanto maior a distância maior o erro, estabelecendo um padrão. Assim, para tentar reduzir os erros, é aplicada uma função onde se multiplica o valor obtido por uma constante vinculada à distância, ou seja, quanto mais distante o ponto, maior a constante. Essa função é obtida usando os resultados das imagens de teste com pontos em posições conhecidas.

O modelo ZoeDepth também é capaz de realizar o cálculo de distância métrica ou de distância relativa e foi treinado originalmente com imagens de ambientes internos e externos. Esse modelo apresentou o mesmo problema que o MiDas. Assim, para tentar reduzir o erro, da mesma forma que utilizado no MiDas, é aplicado um fator de correção, baseado em uma função estabelecida a partir do padrão percebido nos resultados das imagens de teste.

O modelo Depth-pro é um modelo desenvolvido para obter distâncias absolutas com alta qualidade e velocidade. Ele cria bordas nítidas e preserva os detalhes, eliminando ruídos que não correspondem com a geometria do objeto na cena real. Estes ruídos são um problema comum em outros sistemas que calculam distâncias. Os ruídos costumam aparecer nas bordas de objetos nas imagens. O modelo utiliza a arquitetura ViT (Visual Transformers), apresentando uma alta velocidade de processamento. O banco de dados utilizado no seu treinamento original foi composto por dados reais e sintéticos, possibilitando extensa quantidade de dados de treinamento.

Ao utilizar as redes neurais para o cálculo de distâncias, é necessário reduzir o tamanho das imagens a fim de possibilitar o processamento em tempo real. O resultado das distâncias obtido pelo

modelo (imagem onde cada pixel é a distância do mesmo à câmera) é integrado com os resultados de segmentação das imagens. O processo de cálculo é mostrado na Figura 4, que apresenta a sequência de etapas, garantindo que a determinação das distâncias e a segmentação da rua sejam realizadas ao mesmo tempo, de forma que a velocidade do processo é limitada pelo processo mais lento.



Figura 4 - Fluxograma das etapas de cálculo do método que utiliza somente imagens e modelos de IA.

LiDAR

No processo de determinação das distâncias usando o LiDAR é realizada a projeção dos pontos obtidos pelo LiDAR na imagem. Para isto realizar essa projeção são necessárias algumas etapas. A primeira etapa é realizar a calibração do sistema câmera-LiDAR, visto que não só a câmera apresenta distorções inatas de sua fabricação, mas também pelo fato de o LiDAR e a câmera apresentarem campos de visão distintos e também por estarem em posições diferentes. Além disso, deve-se garantir que a obtenção da imagem e a nuvem de pontos gerada pelo LiDAR são realizadas simultaneamente.

A calibração do sistema câmera-LiDAR é feita com o software MATLAB, sendo que dois aplicativos foram testados: CAMERA-CALIBRATOR e LIDAR-CAMERA-CALIBRATOR. O aplicativo CAMERA-CALIBRATOR permite ao usuário a escolha das imagens com os melhores resultados para obter os parâmetros intrínsecos da câmera. O aplicativo LIDAR-CAMERA-CALIBRATOR obtém automaticamente tais parâmetros, contudo, não permite controle sobre o processo, o que pode acarretar em erros. Após testes dos dois métodos, conclui-se que o melhor resultado da calibração da câmera foi alcançado pelo aplicativo CAMERA-CALIBRATOR.

Na calibração do sistema, o aplicativo LIDAR-CAMERA-CALIBRATOR retorna a matrizes necessárias para realizar a projeção dos pontos do LiDAR na câmara. Para realizar essa calibração é necessário fabricar um padrão em forma de xadrez. Esse padrão foi fabricado com uma chapa de mdf cortada a laser no Laboratório de fabricação do Instituto Mauá de Tecnologia (FABlab). Nesta chapa, é colado o tabuleiro xadrez com as dimensões dos quadrados conhecida, como demonstrado na figura 5.



Figura 5 – Tabuleiro utilizado para a calibração do sistema câmera-LiDAR.

Na calibração do sistema câmera-LiDAR, o aplicativo procura as intersecções dos quadrados para determinar os parâmetros intrínsecos da câmera e a posição e rotação relativas entre a câmera e o LiDAR. Para a calibração do sistema, o aplicativo procura o retângulo onde se encontra o padrão

de calibração e combina esses pontos com o xadrez observado na imagem, como se observa na Figura 6.

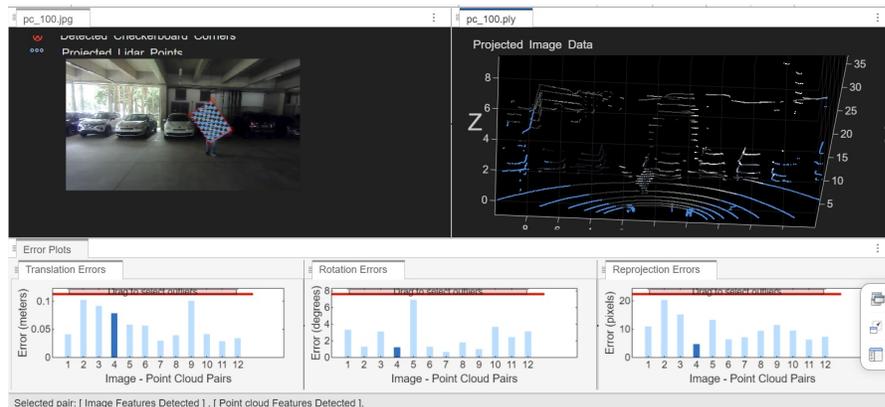


Figura 6 - Calibração do sistema câmera-LiDAR obtida pelo MATLAB

Ressalta-se que é importante realizar o tratamento dos dados do LiDAR para que possam ser usados de forma eficiente. Para isso, é necessário eliminar as distâncias maiores de 50 metros e menores de 20 centímetros por estarem fora da capacidade de detecção do sensor. Após esse processamento inicial, as matrizes de calibração são utilizadas para transformar os pontos do LiDAR de forma que estes sejam projetados no plano da câmera. Por fim, as informações da segmentação do limite da rua são utilizadas para obter as distâncias do limite do espaço livre (espaço trafegável). Essa sequência de etapas está exemplificada na Figura 7.



Figura 7 - Fluxograma das etapas de cálculo do método que usa imagens e LiDAR.

Resultados e Discussão

Após a implementação de todos os métodos, é possível analisar os resultados para selecionar os mais adequados para aplicação proposta nesse trabalho. Na etapa de segmentação, diferentes desempenhos de qualidade e velocidade de processamento foram percebidos. No aspecto da qualidade, os modelos SAM e FastSAM apresentam resultados semelhantes, apesar do modelo SAM apresentar uma qualidade ligeiramente superior, como se observa nas Figuras 8 e 9. O SAM2, por sua vez, se destaca por apresentar bordas mais definidas e com menos ruídos, como se observa na Figura 10. Outro fator que desfavorece o modelo FatsSAM é a segmentação realizada com erros, sendo que em cerca de 5% das imagens utilizadas houve a segmentação de um objeto diferente da rua, como mostrado na Figura 11, onde foi segmentado o fundo da imagem.



Figura 8 - resultado da segmentação do modelo SAM



Figura 9 - resultado da segmentação do modelo FastSAM



Figura 10 - resultado da segmentação do modelo SAM2



Figura 11 - Erro de segmentação gerado pela FastSAM

Quanto ao tempo de processamento, o modelo SAM apresenta um tempo maior, demorando cerca de 100 segundos para processar cada imagem, o que torna inviável a sua aplicação em tempo real. O modelo FastSam, por sua vez, apresenta um tempo de processamento de 2 segundos por imagem, enquanto que o SAM2 apresenta o melhor desempenho possibilitando alcançar uma frequência de aproximadamente 8 quadros por segundo, ou seja um tempo de processamento de 0,125 segundos por imagem. Observa-se que esses resultados foram obtidos em um computador comum sem uso de GPU.

Na etapa do cálculo de distâncias pelos modelos de IA, são observadas diferenças entre a qualidade dos resultados obtidos. Os resultados obtidos pelo MiDas e pelo ZoeDepth são similares, enquanto que os resultados obtidos pelo Depth-pro são superiores, com menor incidência de erro e desvio padrão inferior, como se percebe nas Tabelas 1 e 2.

Tabela 1 – Erros médios obtidos por cada rede Neural

Midas	ZoeDepth	Depth-pro
7,78%	8,00%	4,70%

Tabela 2 – Desvio médio padrão obtidos por cada rede Neural

Midas	Zoedepth	Depth-pro
1,5	0,97	0,54

Outro fator que destaca o Depth-pro é o tempo de processamento. O ZoeDepth apresenta tempo média de 4 segundos para processar cada imagem, o MiDas apresenta uma média de 2,5 segundos por imagem e o Depth-pro apresenta um tempo menor do que 1 segundo por imagem, sendo, portanto, mais adequado à aplicação em tempo real. Observa-se que esses resultados foram obtidos em um computador comum sem uso de GPU.

Ao optar pelo Depth-pro, devido ao seu desempenho superior, é realizada a fusão dos resultados de distâncias desse modelo com os limites do espaço da rua, determinado pelo SAM2. O resultado obtido para a imagem mostrada nas Figuras 8 a 10 é apresentado na Figura 12.

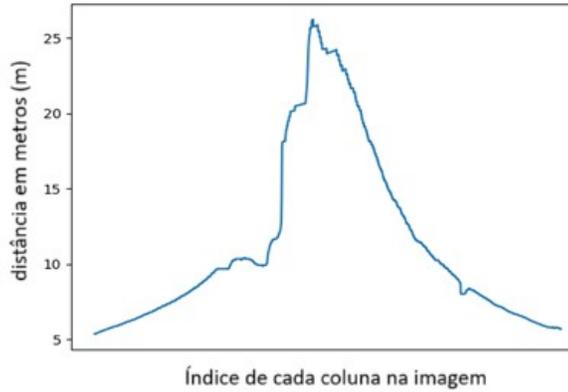


Figura 12 – distância em metros obtida dos limites da rua

A calibração do sistema câmera-LIDAR realizado com o MATLAB, determina os parâmetros intrínsecos da câmera, além da matriz de rotação e o vetor de translação que definem a posição e orientação relativa entre câmera e LIDAR. Desta forma, com a utilização das equações (1), (2) e (3) é possível realizar a projeção dos pontos do LiDAR para o plano 2D da imagem. A equação (1) realiza a transformação de coordenadas dos pontos no sistema de coordenadas do LIDAR para o sistema de coordenadas 3D da câmera.

$$M_R L + T = C \quad (1)$$

onde, M_R é a matriz de rotação entre os sistemas de coordenadas da câmera e do LIDAR, L é o vetor de posição dos pontos na cena obtidos pelo LiDAR, T é o vetor de translação entre a câmera e o LIDAR e C é o vetor de posição dos pontos da cena, obtidos pelo LIDAR, descritos no sistema de coordenadas 3D da câmera.

As equações (2) e (3) realizam a projeção perspectiva dos pontos da cena, descritos no sistema de coordenadas 3D da câmera, no plano da imagem, obtendo as posições em pixels na imagem dos pontos da cena.

$$X_p = f_x \frac{C_x}{C_z} + O_x \quad (2)$$

$$Y_p = f_y \frac{C_y}{C_z} + O_y \quad (3)$$

onde X_p , é a posição horizontal do ponto na imagem em pixel, Y_p é a posição vertical do ponto na imagem em pixel, f_x e f_y representam as distâncias focais da câmera em pixels, O_x e O_y são, respectivamente as posições horizontal e vertical do centro da imagem em pixels, C_x , C_y , e C_z são as coordenadas do ponto da cena descritas no sistema de coordenadas 3D da câmera.

Um exemplo da projeção dos pontos obtidos pelo LIDAR em uma imagem é mostrado na Figura 13. Percebe-se que os pontos determinados pelo LIDAR não preenchem toda a imagem pelo fato do LiDAR possuir apenas 16 linhas e ter um ângulo de abertura vertical menor do que a câmera. Para determinar a posição dos limites do espaço livre na frente do veículo, é necessário, fazer uma interpolação usando os pontos determinados pelo LiDAR, para aproximar essas medidas de distâncias aos limites da rua, como mostrado na Figura 14. É realizada uma interpolação linear das medidas do LIDAR aos limites da rua, obtendo um vetor de pontos que representam os limites do espaço livre para cada posição horizontal da imagem, conforme apresentado na Figura 15.

Observam-se alguns erros nesse resultado e um estudo mais detalhado é necessário para determinar a origem desses erros.

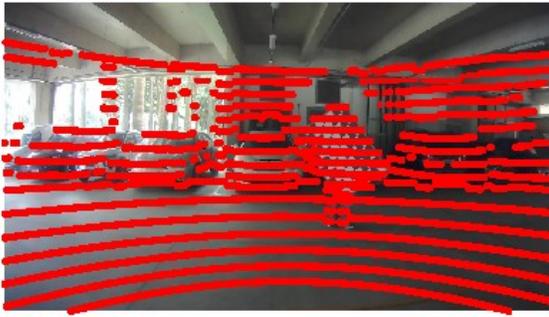


Figura 13 – interpolação dos pontos de LiDAR



Figura 14 – segmentação dos limites da rua

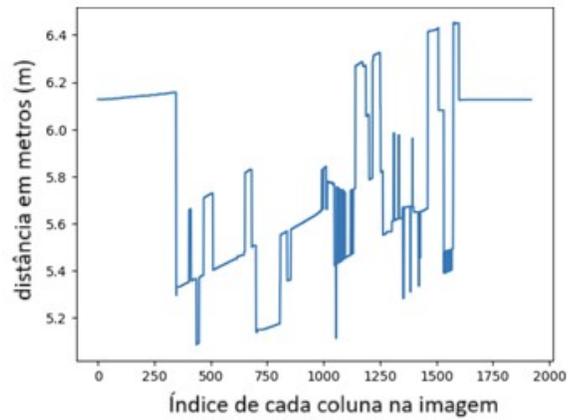


Figura 15 – distâncias em metros obtida dos limites da rua pelo sistema câmera-LiDAR.

Conclusões

Com base nos resultados obtidos ao longo desse trabalho, pode-se concluir que existem diferentes abordagens que são eficazes para determinar o espaço do entorno de um veículo autônomo, especialmente na etapa de cálculo de distâncias.

Na etapa de segmentação de imagens, conclui-se que o modelo de IA SAM2 é o mais eficiente, como era esperado, visto que é o modelo mais recente. A IA apresenta uma maior precisão nos detalhes, reduzindo a ocorrência de ruídos, além de apresentar o menor tempo de processamento, atendendo, assim, aos requisitos de aplicações em tempo real.

Na etapa de cálculo de distâncias, observa-se que tanto os métodos baseados em modelos de IA quanto o que utiliza o LiDAR são viáveis, cada qual com seus erros e limitações inerentes. Porém, observa-se que o método usando o LIDAR apresenta resultados com erros e, portanto, precisa ser melhor analisado. Contudo, pode-se concluir que ambos os métodos fornecem resultados satisfatórios, sendo a escolha dependente das necessidades específicas de cada aplicação e da disponibilidade de um LIDAR. Por exemplo projetos que necessitam altas velocidades de processamento se beneficiaram do método com o LiDAR, enquanto projetos que visem focar menos na parte do hardware se beneficiariam do método com IA sendo que não haveria a necessidade de se preocupar com calibrações e sincronizações. A integração de ambos os métodos (câmera-LIDAR e somente câmera com IA) mostra-se uma alternativa desejável para se ter redundância, confiabilidade e melhoria dos resultados, que certamente contribuiria para o aumento da robustez do sistema.

Este trabalho reforça a importância de sistemas integrados que combinam algoritmos avançados de visão computacional e sensores de alta precisão para aplicações em veículos autônomos. Além disso, demonstra que a escolha de ferramentas e modelos precisa ser cuidadosamente alinhada aos requisitos de desempenho e segurança de cada cenário.

Como próximos passos para continuação desse trabalho, sugere-se: 1) explorar essas técnicas em um veículo em condições de uso efetivo; 2) analisar melhor o método que usa câmera-LIDAR para melhorar o seu desempenho e determinar a origem dos erros detectados; e 3) realizar a fusão de dados dos resultados dos dois métodos. A instalação no veículo de pesquisa ORBI seria uma iniciativa capaz de testar o desempenho das funcionalidades dos dois métodos analisados em uma condição real. Para tanto, sugere-se a utilização do computador integrado Jetson ORIN que tem a capacidade computacional para realizar os cálculos necessários em tempo real.

Referências Bibliográficas

- An, P., Ma, T., Yu, K., Fang, B., Zhang, J., Fu, W., & Ma, J. (2020). Geometric calibration for LiDAR-camera system fusing 3D-2D and 3D-3D point correspondences. *Optics Express*, 28(2), 2122–2141. <https://doi.org/10.1364/OE.38117>
- Bhat, S. F., Birkel, R., Wofk, D., Wonka, P., & Müller, M. (2023). ZoeDepth: Zero-shot transfer by combining relative and metric depth. In arXiv [cs.CV]. <http://arxiv.org/abs/2302.12288>
- Bochkovskii, A., Delaunoy, A., Germain, H., Santos, M., Zhou, Y., Richter, S. R., & Koltun, V. (2024). Depth Pro: Sharp monocular metric depth in less than a second. Em arXiv [cs.CV]. <http://arxiv.org/abs/2410.02073>
- BRAGA, José Renato Garcia. **Navegação autônoma de VANT por imagens LiDAR**. 2016. Instituto Nacional de Pesquisas Espaciais (INPE), [s. l.], 2016. Disponível em: <http://urlib.net/sid.inpe.br/mtc-m21c/2018/05.18.16.04>. Acesso em: 8 dez. 2024.
- GONÇALVES, Luiz Felipe Sartori. **Desenvolvimento de sistema de navegação autônoma por GNSS**. 2011. Universidade de São Paulo, [s. l.], 2011. Disponível em: <http://www.teses.usp.br/teses/disponiveis/3/3138/tde-19072011-162537/>. Acesso em: 5 dez. 2024.
- Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023). Segment Anything. In arXiv [cs.CV]. <http://arxiv.org/abs/2304.02643>
- KLASER, Rafael Luiz. **Navegação de veículos autônomos em ambientes externos não estruturados baseada em visão computacional**. 2014. Universidade de São Paulo, [s. l.], 2014. Disponível em:
- Lidar and camera calibration**. Disponível em: <<https://www.mathworks.com/help/lidar/ug/lidar-and-camera-calibration.html>>. Acesso em: 24 sept. 2024
- Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., & Koltun, V. (2022). Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 1623–1637. <https://doi.org/10.1109/TPAMI.2020.30199>
- Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K. V., Carion, N., Wu, C.-Y., Girshick, R., Dollár, P., & Feichtenhofer, C. (2024). SAM 2: Segment Anything in Images and Videos. Em arXiv [cs.CV]. <http://arxiv.org/abs/2408.00714>
- Zhao, X., Ding, W., An, Y., Du, Y., Yu, T., Li, M., Tang, M., & Wang, J. (2023). Fast Segment Anything. In arXiv [cs.CV]. <http://arxiv.org/abs/2306.1215>