SISTEMA DE INTELIGÊNCIA ARTIFICIAL PARA CÁLCULO DE DISTÂNCIAS DE OBJETOS A PARTIR DE UM SISTEMA DE VISÃO ESTÉREO

Tiago Hiray Hisatugo [1]; Eduardo Lobo Lustosa Cabral [2]

[1] Aluno de Iniciação Científica do Instituto Mauá de Tecnologia (IMT);
[2] Professor do Instituto Mauá de Tecnologia (IMT).

Resumo. *Este trabalho de Iniciação Científica concentrou-se na pesquisa e desenvolvimento de um sistema de inteligência artificial voltado para o cálculo de distâncias de objetos utilizando um sistema de visão estéreo. A pesquisa foi realizada em colaboração com o grupo SMIR (Sistemas Mecatrônicos Inteligentes e Robótica), tendo o propósito de integrar o sistema de um veículo elétrico autônomo. O projeto empregou técnicas de processamento de imagens e aprendizado de máquina para analisar as informações provenientes das câmeras estéreo montadas no veículo. A partir das imagens capturadas, o sistema de IA foi treinado para identificar e medir com precisão a distância entre o veículo e objetos circundantes, possibilitando uma navegação segura e eficiente.*

Introdução

Nos últimos anos, o campo da inteligência artificial (IA) tem testemunhado avanços significativos que têm impactado positivamente uma variedade de indústrias e aplicações. Um domínio em particular que vem tendo grande notoriedade é a visão computacional para resolver problemas complexos de percepção, análise e tomada de decisões.

A medição precisa das distâncias entre objetos no espaço tridimensional é uma tarefa fundamental em diversas aplicações, incluindo automação de veículos, sistemas de navegação, robótica autônoma e segurança. Tradicionalmente, essa tarefa foi abordada com tecnologias como sensores lidar e câmeras 3D especializadas. No entanto, avanços recentes nas áreas de visão computacional e aprendizado de máquina permitiram o desenvolvimento de sistemas mais acessíveis e eficazes, que exploram as informações contidas em imagens estéreo.

Este projeto faz uso de uma abordagem multifacetada, combinando técnicas de segmentação de objetos, cálculo de limites da pista e estimativa de distâncias, com foco na eficiência e precisão. A segmentação de objetos é realizada com o auxílio de modelos de redes neurais pré-treinadas, notavelmente a FastSAM, que permite a identificação confiável de objetos em ambientes complexos. Em seguida, a biblioteca NumPy e a linguagem de programação Python foram utilizadas para calcular os limites da pista, fornecendo informações espaciais fundamentais para a localização e avaliação dos objetos.

O cerne deste projeto é a aplicação do modelo DINOv2 para calcular distâncias entre o sistema de visão estéreo e os objetos identificados. O DINOv2, com suas capacidades avançadas de auto-supervisão, oferece uma estimativa precisa das distâncias com base nas informações capturadas. Essa abordagem proporciona uma solução economicamente viável, flexível e escalável, com potencial para uma ampla gama de aplicações práticas.

Neste relatório, é detalhado o processo de desenvolvimento, as técnicas utilizadas, e os resultados obtidos. Além disso, são exploradas as implicações e aplicações potenciais desse sistema em setores como veículos autônomos, monitoramento de segurança e robótica. Ao final, é evidente como a intersecção entre IA, visão computacional e técnicas de aprendizado de máquina está moldando um futuro em que a percepção de profundidade é acessível, precisa e revolucionária.

Material e Métodos

Para a realização dos objetivos deste trabalho, é necessária a utilização de um sistema de câmeras estéreo equipado em um veículo elétrico autônomo, que pode ser observado na Figura 1.

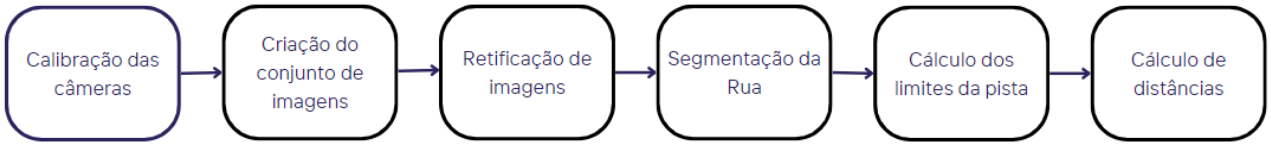
O veículo em questão, é um carrinho de golfe adaptado para o projeto. Esse carrinho suporta até 6 pessoas e foi adaptado com diversos equipamentos, como interface digital, direção elétrica para a automação, um sistema de RTK Survey para localização, além de uma câmera omnidirecional e o sistema de visão estéreo.

Figura 1 – Veículo elétrico autônomo utilizado para o projeto

O sistema é composto por um par de câmeras de 2 megapixels com resolução 1080p e entrada USB, acopladas a uma estrutura metálica centralizada na parte superior do veículo, como pode ser observado na Figura 2.

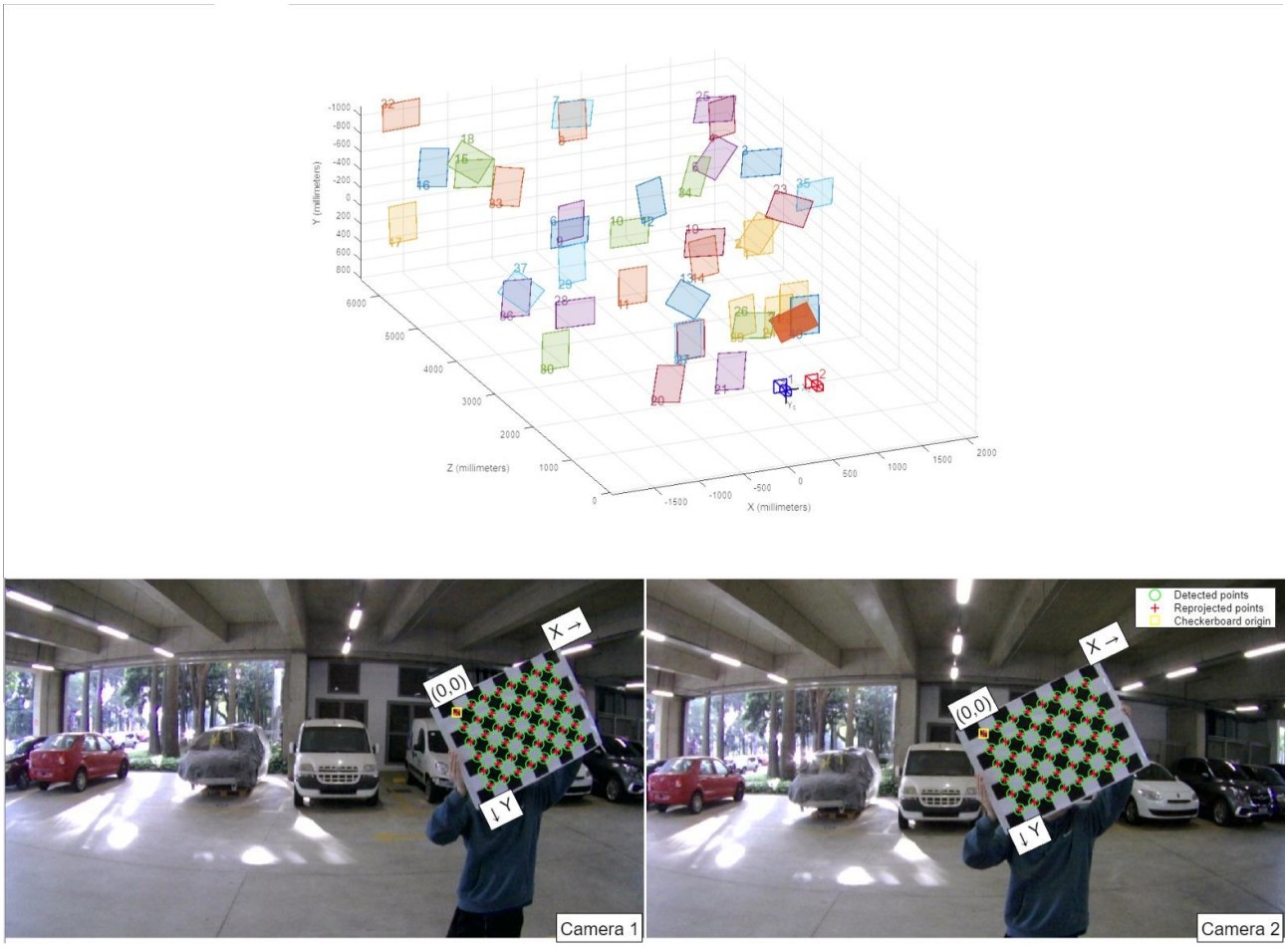
Figura 2 – Sistema de câmeras estéreo acopladas ao veículo

Figura 3 – Diagrama de blocos sobre a metodologia



Diante da utilização de um sistema estéreo, é fundamental realizar a calibração das câmeras. Esta calibração foi executada com o auxílio do software MATLAB e um tabuleiro de xadrez. O tabuleiro, que com suas dimensões ocupava entre 10% e 15% dos frames, foi posicionado em diversas posições no campo de visão das câmeras, e aproximadamente 30 imagens foram capturadas a partir de ambas. Na figura 4 é mostrada a utilização do MATLAB, que foi utilizado para a estimativa dos parâmetros intrínsecos e extrínsecos das câmeras, tais como distorções, distância focal e matriz de rotação. Isso é essencial para garantir que as informações capturadas pelas câmeras estejam devidamente alinhadas e que as distorções sejam corrigidas para uma maior acurácia nos resultados da pesquisa.

Figura 4 – Processo de calibração das câmeras estéreo no software MATLAB



Realizada a calibração das câmeras estéreo, iniciou-se o processo de geração do conjunto de imagens utilizado no projeto. Aproximadamente, 12000 pares de imagens foram capturadas no trajeto do campus do IMT a partir de um simples algoritmo utilizando Python e OpenCV, o qual integrava as câmeras estéreo a um computador e capturava aproximadamente 24 imagens por segundo. Após essa etapa, é realizada uma limpeza de dados dentro do conjunto de imagens, com o objetivo de eliminar imagens que apresentassem algum tipo de problema como baixa resolução ou iluminação ruim.

Tendo tanto a calibração das câmeras quanto o conjunto de dados prontos, é necessário realizar a retificação das imagens. Utilizou-se novamente a linguagem de programação Python e a biblioteca OpenCV para a realização desse processo. A retificação tem o propósito de ajustar as imagens de forma que as linhas horizontais em uma imagem correspondam às linhas horizontais na outra imagem, possibilitando a correspondência precisa entre os pontos nas imagens estéreo, como pode ser

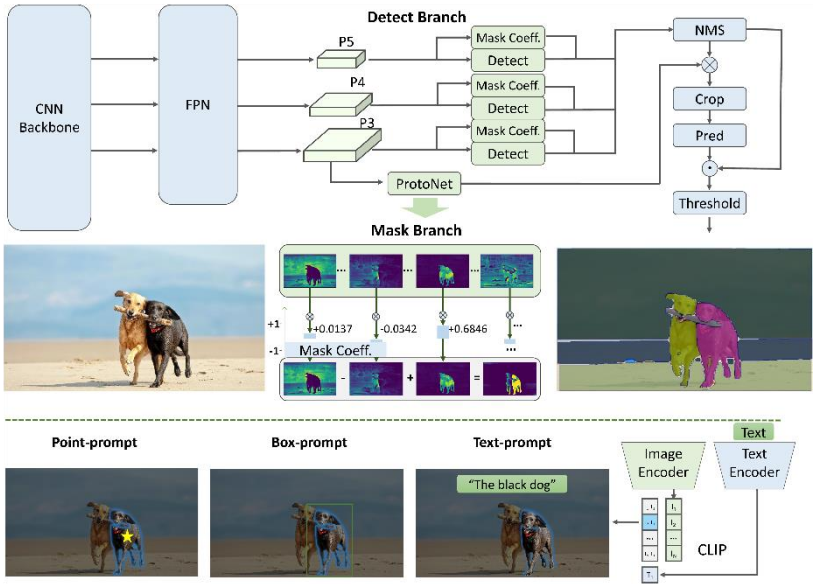
observado na Figura 5. Essas etapas metodológicas são fundamentais para garantir a eficiência do sistema de visão estéreo, proporcionando dados precisos sobre a geometria tridimensional dos objetos no ambiente e, consequentemente, contribuindo para o sucesso do projeto.

Figura 5 – Imagens estéreo retificadas

Após os processos de calibração das câmeras, criação do conjunto de dados e retificação de imagens, iniciou-se o desenvolvimento do algoritmo principal para o funcionamento do sistema. Este projeto aborda o cálculo de distâncias de objetos de uma maneira não convencional, em contraste com métodos mais populares que se concentram na detecção individual de objetos para posteriormente medir suas distâncias, essa pesquisa é direcionada para a detecção da geometria do trajeto pelo qual o veículo trafega, e seus limites. Esta abordagem permite uma maior eficiência e otimização de processamento.

Primeiramente, é realizada a segmentação das ruas com o uso da rede pré-treinada FastSAM, cuja arquitetura pode ser observada na Figura 6. A segmentação é o processo de subdivisão de imagens em partes ou regiões menores com algum propósito. Dentre esse processo, existem diversos tipos de segmentação, no caso, a FastSAM realiza a segmentação de instâncias, que identifica objetos únicos e atribui etiquetas aos mesmos.

Figura 6 – Arquitetura da rede pré-treinada FastSAM



Fonte: CASIA-IVA-Lab, 2023

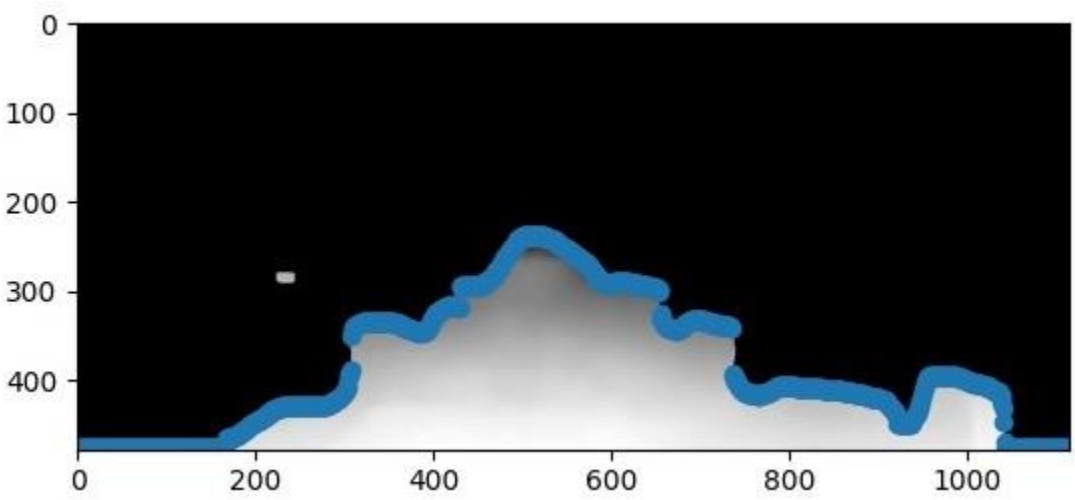
Além disso, a FastSAM possui uma ferramenta inovadora na visão computacional, o text-prompt. Esta ferramenta é um grande facilitador na segmentação precisa das ruas, visto que a

especificação do alvo para a segmentação é feita por meio de palavras, muito semelhante com os prompts das IAs Generativas. O modelo também permite o ajuste de dois parâmetros, o nível de confiança na detecção de objetos e o IOU, que se refere ao conceito de intersecção sobre união. No caso desta pesquisa, os parâmetros utilizados foram nível de confiança de 0.35 e IOU de 0.48. Tais medidas resultaram em excelentes resultados como pode ser visto na Figura 7.

Figura 7 – Imagem da rua do campus do IMT com o trajeto segmentado

Dada a segmentação da rua, é realizada a detecção dos limites da pista. Esta etapa é feita por meio de cálculos matriciais com o auxílio da linguagem Python e da biblioteca Numpy. Primeiramente é realizada a subtração da imagem segmentada com a imagem original e posteriormente, o resultado desta operação é transformado para tons de cinza. Após isso, são realizadas operações como inversão e normalização dos dados e por fim, os limites são encontrados e traçados em formato de gráfico de dispersão na imagem resultante, como observado na Figura 8.

Figura 8 – Detecção dos limites da pista

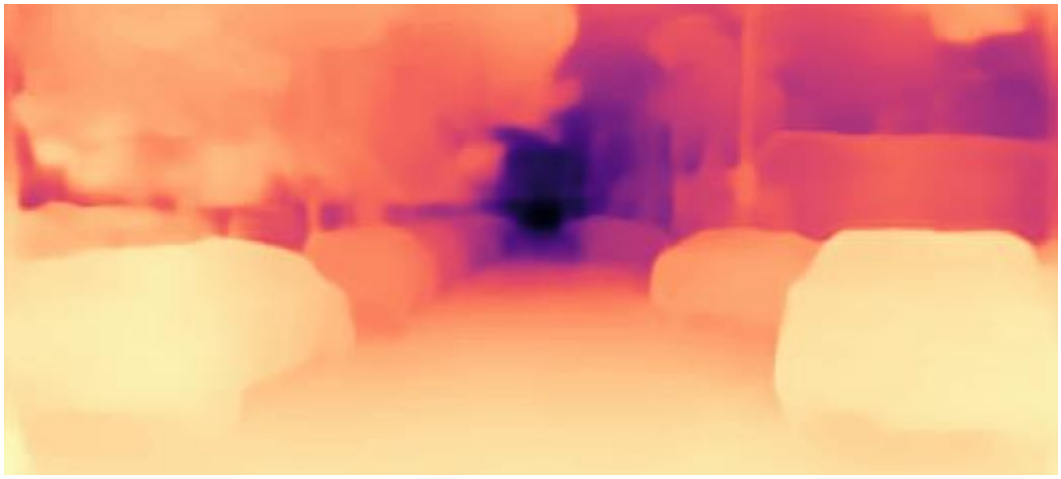


Tendo os processos de segmentação e detecção dos limites de pista, é realizada a etapa de cálculo de distâncias. Nesta etapa, é utilizado o modelo de rede pré-treinada, DINOv2. Esse modelo foi desenvolvido pela MetaAI e é uma rede neural feita com a técnica de destilação de conhecimento, em que um modelo profundo de alta complexidade e capacidade é treinado, e após isso, os resultados

são transferidos para um modelo menor, com o objetivo de melhorar a eficiência computacional para os usuários.

No cenário do projeto, a DINOv2 foi utilizada para a geração de mapas de profundidade e consequentemente o cálculo de distâncias, como pode ser observado na Figura 9. Tais mapas gerados pela rede, estimam a distância do ponto focal das câmeras até todos os objetos do cenário, destacando de forma mais clara os objetos mais próximos e de forma mais escura os objetos mais distantes. Entretanto, esse modelo estima o que é chamado de distância relativa, ou seja, em duas diferentes imagens, a mesma distância pode ser interpretada de maneiras distintas dependendo de como o cenário se comporta.

Figura 9 – Mapa de profundidade gerado pela rede DINOv2



A DINOv2 não fornece o valor da distância absoluta de cada objeto. Entretanto, com valores de cada pixel, encontrados a partir da matriz resultante do mapa de profundidade em tons de cinza, e com a utilização de uma equação linear, observada na equação (1), é possível converter tais valores para a distância real em metros.

$$P = A \times I + B \quad (1)$$

Na equação (1), a variável *P* refere-se à distância inversa, ou seja 1 / (distância real em metros). A variável *I* por sua vez, representa o valor do pixel do objeto escolhido no cenário. Por fim, as variáveis *A* e *B*, são dois escalares que representam a distância real de dois objetos utilizados como referenciais para a câmera. Dado isso, é importante notar que *A* e *B* devem ser recalculados a cada quadro, visto que as inferências são relativas aos mesmos.

Resultados e Discussão

Para uma melhor avaliação do sistema é possível observar primeiramente, os resultados da etapa de segmentação das ruas. Dentre os 12000 pares de imagens do conjunto de dados, apenas 600 pares foram segmentados de forma incorreta pelo modelo, ou seja, 95% dos pares de imagens obtiveram sucesso na fase de segmentação. Esses erros, ocorreram em sua maioria pela confusão do modelo entre o trajeto e outras partes do cenário, como o céu e árvores, que apesar de serem muito distintos possuem semelhanças, como extensão ou até mesmo cor. Na figura 10, é possível observar um desses erros.

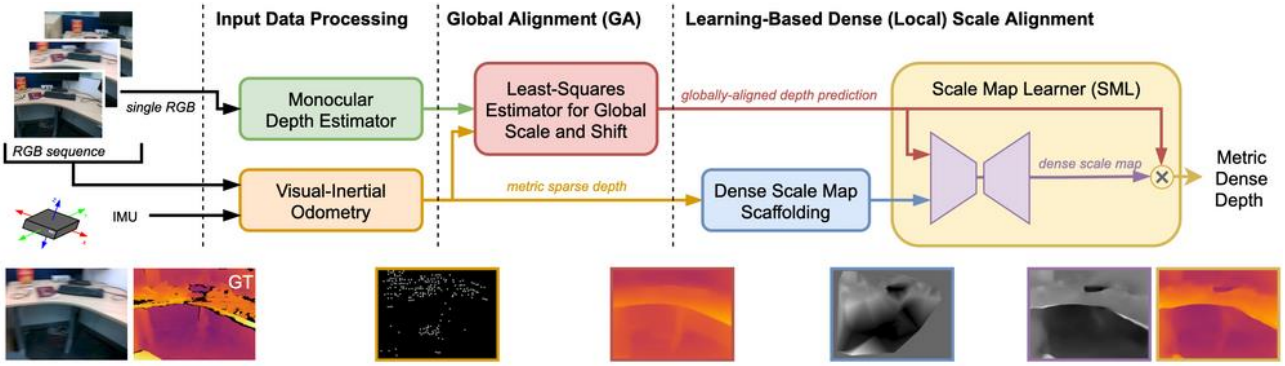
Figura 10 – Imagem com a metade superior segmentada

Apesar desse resultado apresentar um erro de 5% no que se refere a uma tarefa de extrema importância, dado que identifica o trajeto pelo qual o veículo trafegará, é possível minimizar tal erro com uma alta taxa de aquisição por segundo, conseguindo sobrepor possíveis segmentações falhas. Ademais, pode-se observar que a FastSAM possui um desempenho computacional excelente, sendo 50 vezes mais rápida e eficiente do que sua versão anterior, a SAM e permitindo seu uso em tempo real para aplicações como esse projeto.

A etapa de detecção dos limites da pista gerou resultados excelentes, tendo 100% de acerto na identificação dos limites da pista das imagens segmentadas. Grande parte desse resultado, deve-se ao fato de ele ter sido realizado estritamente com cálculos matemáticos e nenhum tipo de modelo de IA, logo, de fato são esperados resultados mais precisos.

O processo de cálculo de distâncias no projeto foi testado de duas maneiras. A primeira, foi realizada utilizando modelo pré-treinado MiDaS 3.1, da Intel Labs, cuja arquitetura pode ser observada na Figura 11. Entretanto, após alguns testes, o modelo foi descartado devido a sua alta exigência de processamento, que o torna inviável para uma aplicação em tempo real.

Figura 11 – Arquitetura da rede MiDaS 3.1



Fonte: Intel Labs, 2023

No cálculo de distâncias, a DINOv2 apresentou desempenho consideravelmente maior que a MiDaS 3.1 e resultados precisos. Na Figura 13, é mostrado o mapa profundidade utilizado para testes com a DINOv2, nele estão presentes 5 objetos referenciados nas Tabelas 1 e 2, uma lousa (L), uma

mala próxima a lousa (M1), outra mala pendurada em um pedestal (M2), uma pessoa (P) e a cadeira mais próxima na imagem (C). Esses 5 objetos podem ser observados na Figura 12, que é a imagem original utilizada para gerar o mapa de profundidade dos testes feitos com a DINOv2

Figura 12 – Imagem original para teste na DINOv2

Figura 13 – Mapa de profundidade teste gerado pela DINOv2

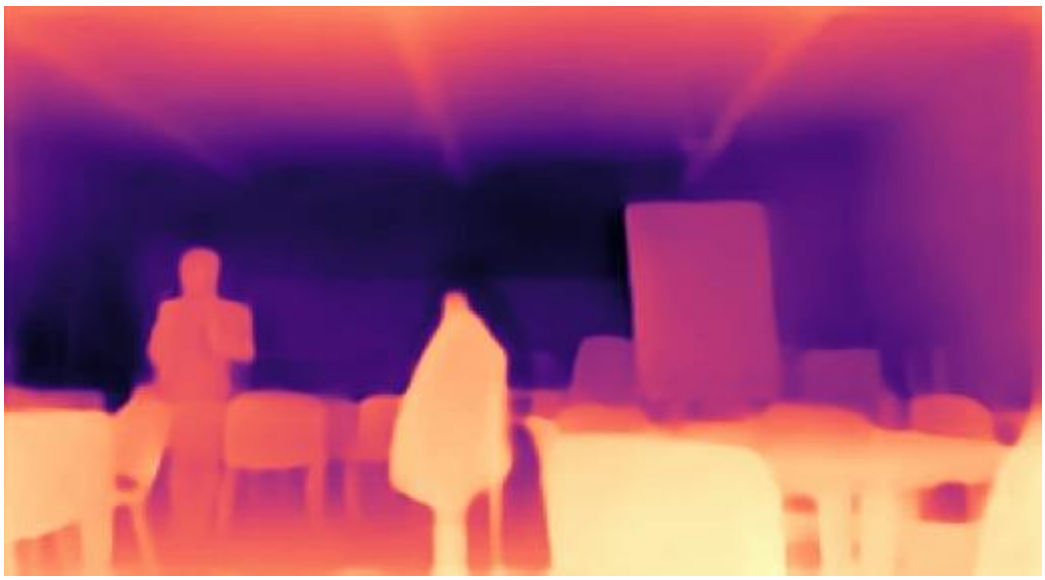


Tabela 1 – Cálculo de distâncias a partir dos referenciais P e C

Objetos	Distância real (m)	Distância calculada (m)	Erro (%)
L	2,55	2,74	7,39
M1	4,50	4,40	2,22
M2	4,20	3,89	7,33

P	3,50	3,50	0
C	1,40	1,40	0

Tabela 2 – Cálculo de distâncias a partir dos referenciais M1 e C

Objetos	Distância real (m)	Distância calculada (m)	Erro (%)
L	2,55	2,78	9,14
M1	4,50	4,50	0
M2	4,20	3,98	5,35
P	3,50	3,57	2
C	1,40	1,40	0

Utilizando esse mapa de profundidade, são utilizados como referenciais para o cálculo de distâncias, a pessoa e a cadeira, com resultados que podem ser observados na Tabela 1. Usando a mesa 1 e cadeira como referenciais, os resultados podem ser observados na Tabela 2. As distâncias reais desses referenciais são usadas para calcular as constantes *A* e *B* da equação (1) permitindo, assim, calcular as distâncias de todos os pontos da imagem. É possível observar um erro médio de 3,39% nos cálculos com os referenciais P e C, e 3,30% com os referenciais M1 e C, o que demonstra certa consistência nos cálculos, além de boa assertividade do modelo. Além disso, quatro cálculos dos dez apresentados não tiveram nenhum erro, o que é um indicador ainda melhor da precisão.

Entretanto, dois cálculos do mesmo objeto, que no caso foi a lousa, demonstraram um erro maior, com uma média de 8,265%, o que gera certa atenção, mas que no sistema pode ser evitado com limites maiores de segurança para o veículo trafegar.

Conclusões

Em suma, o projeto demonstrou boa assertividade e desempenho em seus resultados. Em uma visão geral, pode-se dizer que o sistema é confiável e funcional. Contudo, por considerações de otimização e segurança, a implementação direta no veículo autônomo ainda está pendente e o sistema permanece em fase de desenvolvimento para melhorias.

A decisão de priorizar a otimização do desempenho computacional e a garantia de níveis elevados de segurança é essencial. Essa decisão busca assegurar que o sistema funcione de forma correta em todas as situações e garanta segurança aos passageiros.

Espera-se dar continuidade ao projeto, para buscar o melhor desempenho e segurança possíveis, permitindo assim uma navegação precisa e segura a todos que utilizarem o veículo.

Referências Bibliográficas

GODARD, AODHA e BROSTOW, 2017. C. Godard, O. M. Aodha, and G. J. Brostow, Unsupervised Monocular Depth Estimation with Left-Right Consistency, arXiv:1609.03677 v3 [cs.CV], April 2017.

GODARD et al. 2019. C. Godard, O. M. Aodha, M. Firman, and G. Brostow, Digging Into Self-Supervised Monocular Depth Estimation, arXiv:1806.01260v4 [cs.CV], August 2019.

LIANG et al. 2021. Z. Liang , Y. Guo , Y. Feng, W. Chen, L. Qiao , L. Zhou, J. Zhang, and H. Liu Stereo Matching Using Multi-Level Cost Volume and Multi-Scale Feature Constancy, IEEE Transactions On Pattern Analysis And Machine Intelligence, Volv. 43,no. 1, January 2021.

VARMA et al., 2017. A. Varma, H. Chawla, B. Zonooz, and E. Arani, Transformers in Self-Supervised Monocular Depth Estimation with Unknown Camera Intrinsics, 17th International Conference on Computer Vision Theory and Applications (VISAP, 2022), arXiv:2202.03131v1 [cs.CV].

WANG et al. 2017. C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey, Learning Depth from Monocular Videos using Direct Methods, arXiv:1712.00175v1 [cs.CV], December 2017.