Detecção de Obstáculos com Câmeras Monoculares por meio de Mapas de Disparidades U/V e Redes Neurais Artificiais ⋆

Samuel Henrique Guimarães Braga * **Danilo Alves de Lima** **
Felipe Oliveira e Silva ***

Escola de Engenharia (EENG), Departamento de Automática (DAT), Universidade Federal de Lavras (UFLA), MG.

* *samuel.braga@estudante.ufla.br*
** *danilo.delima@ufla.br*
*** *felipe.oliveira@ufla.br*

Abstract: Environment perception is one of the most complex tasks to be performed autonomously. Besides depending on expensive sensors, many applications require high computational power, limiting the applicability of such solutions. This paper presents a solution for environment perception with monocular cameras, focusing on low processing costs for embedded systems or modern smartphone applications. The solution uses classical image analysis techniques (such as disparity maps and their U/V variants) with modern techniques (deep artificial neural networks) to detect and estimate the distance of objects in space. Experimental results show good accuracy in distance estimation, significant improvements in object detection, and a runtime close to that of algorithms using only classical methods.

Resumo: A percepção do ambiente é uma das tarefas mais complexas a serem realizadas de forma autônoma. Além de dependerem de sensores de custo geralmente elevado, muitas aplicações precisam de grande poder computacional, o que limita a aplicabilidade de tais soluções. Este artigo apresenta um método de percepção do ambiente com câmeras monoculares, focando no baixo custo de processamento para sistemas embarcados ou aplicações em *smartphone*s modernos. O método utiliza técnicas clássicas de análise de imagens (como mapas de disparidade e suas variantes U/V) com técnicas modernas (redes neurais artificiais profundas) para detectar e estimar a distância de objetos no espaço. Resultados experimentais demonstram boa precisão na estimativa de distâncias e melhoras significativas na detecção de objetos, além de um tempo de execução próximo ao dos algoritmos que utilizam apenas os métodos clássicos.

Keywords: obstacle detection; monocular vision; intelligent vehicles; embedded systems; *smartphone*s.

Palavras-chaves: detecção de obstáculos; visão monocular; veículos inteligentes; sistemas embarcados; *smartphone*s.

1. INTRODUÇÃO

As pesquisas no âmbito de percepção de ambiente têm obtido grandes avanços nos últimos anos, a exemplo dos veículos autônomos da Waymo, os quais já trafegam em vias públicas, e diversas aplicações em robótica móvel. Os benefícios dessa tecnologia são incontestáveis, podendo reduzir em até 90% os acidentes de trânsito (Gao et al., 2014). A percepção do ambiente também pode trazer grandes benefícios ao ser implementada no auxílio de pessoas com certo grau deficiência visual. Alguns dispositivos no mercado já realizam tarefas importantes com o foco na melhoria da qualidade de vida, como, por exemplo, o Orcam Myeye 2, um dispositivo acoplado aos óculos do usuário que oferece recurso de leitura, detecção de cores, reconhecimento de faces e papel-moeda (Feitosa Jr., 2019).Tal dispositivo, porém, ainda não oferece recurso para auxiliar no deslocamento do usuário, detectando ,por exemplo, obstáculos no caminho e sua posição relativa ao mesmo.

Os veículos autônomos, por sua vez, utilizam sensores de alto custo (Lima, 2015) e necessitam de um alto poder de processamento, com baixo consumo de energia para realizar tal tarefa (Liu et al., 2017). Devido a isso, esse tipo de implementação é muitas vezes inviável em sistemas embarcados, pelo baixo poder de processamento e a necessidade de uma estrutura portátil. Nesse quesito, os *smartphone*s modernos se destacam, fornecendo um poder de processamento robusto, um design portátil, além de sensores integrados que também podem ser utilizados para auxiliar o processo de percepção do ambiente. Outra van-

⋆ Os autores agradecem à FAPEMIG (proc. no. APQ-00202-21), ao programa PIBIC/UFLA pelo suporte financeiro e ao Programa de Pós-Graduação em Engenharia de Sistemas e Automação (PPGESISA) da Universidade Federal de Lavras (UFLA) pelo apoio fornecido.

tagem de se utilizar *smartphone*s nesse tipo de aplicação é a unidades de processamento gráfico (GPU) integrada aos dispositivos mais modernos, aumentando o desempenho de aplicações baseadas em redes neurais artificiais, por exemplo (Oh and Jung, 2004).

No Laboratório de Mobilidade Terrestre (LMT) da Universidade Federal de Lavras (UFLA), alguns estudos estão em andamento, visando aproveitar os recursos disponíveis em *smartphone*s como ferramentas de auxílio em geral. Além disso, está sendo desenvolvida a plataforma de testes em escala VIDA (Veículo Inteligente de Desenvolvimento Aplicado), na qual pretende-se incorporar sensores de baixo custo, incluindo *smartphone*s, para aplicações de navegação autônoma e de assistência.

Este trabalho apresenta uma proposta de aplicação para a percepção de ambiente a partir dos recursos de processamento e execução em tempo real em *smartphones*. Além de servir à plataforma VIDA, essa solução pretende também ser aplicável a problemas de detecção de obstáculos em geral, auxiliando pessoas que necessitem de uma assistência visual para melhor perceber o ambiente e se locomover em segurança por ele.

O restante deste trabalho está organizado da seguinte forma: Seção 2 introduz uma revisão literária dos métodos abordados neste artigo, a Seção 3 apresenta a metodologia do método proposto, a Seção 4 discute alguns resultados experimentais e a Seção 5 fornece conclusões e perspectivas para trabalhos futuros.

2. REVISÃO DA LITERATURA

Sistemas de percepção para veículos autônomos combinam informações de uma grande variedade de sensores, podendo ser divididos em duas categorias. A primeira utiliza sensores ativos e passivos do próprio veículo, i.e. sonar, radar, LiDAR (*LIght Detecting And Ranging*) e câmeras. Já a segunda utiliza dados compartilhados com outros veículos e a infraestrutura para receber informações do ambiente em uma percepção colaborativa (Rosique et al., 2019; Zhu et al., 2017). Esse artigo aborda apenas a primeira categoria, focando especificamente, na percepção de ambiente por meio de câmeras.

2.1 Visão Estéreo

Uma câmera estéreo utiliza duas ou mais câmeras espaçadas por uma curta distância, permitindo a imitação da visão binocular humana. A técnica que utiliza esse tipo de câmera na percepção foi originalmente descrita por Faugeras (1993). Ela consiste na retificação de duas imagens síncronas para encontrar correspondências entre pixels e estimar uma relação tridimensional com os pontos no mundo, formando um mapa de disparidade. No entanto, além de ser necessário conhecer com exatidão a *pose* relativa entre as câmeras, o processo como um todo é repleto de fontes de ruídos e erros de estimação. Também se mostra um problema a quantidade de pontos utilizados durante o processo, o qual pode ser um agravante na execução de todas as etapas em tempo real.

Como alternativa, existem metodologias para se trabalhar com menos pontos, combinando informações bidimensionais para se realizar melhores aproximações de regiões semelhantes em super-pixels (Lima et al., 2017). Esses super-pixels agrupam pixels semelhantes em uma mesma região, as quais normalmente correspondem a partes de objetos reais no mundo. Considerando essa propriedade, apenas alguns pixels por região (ou super-pixel) são necessários para descrever a informação tridimensional do mesmo.

2.2 Mapas de disparidades U/V [1]

O mapa de disparidade é formado pela diferença no valor de cada par de pixel das imagens, sendo geralmente representado em uma imagem com apenas 1 canal (em tons de cinza), onde os pixels mais próximos da câmera são de intensidade mais elevada e, consequentemente, aparecem em tons mais claros (Lima and Pereira, 2010). A relação de distância pode ser calculada como:

$$Z = \frac{fB}{d}, \tag{1}$$

onde Z é a distância do objeto até a câmera, B o *baseline* (distância entre as duas câmeras), d a disparidade em pixels e f a distância focal em pixels. Ou seja, quanto maior for o valor da disparidade, menor será a distância até a câmera.

Alguns trabalhos demonstraram que é possível detectar obstáculos utilizando a visão estéreo (Dornaika and Sappa, 2008; Soga et al., 2005), porém um alto poder computacional é necessário. Para contornar esse empecilho podem ser utilizados mapas de disparidade U/V(Gao et al. (2011); Lima and Pereira (2010)), que consistem em usar as informações do mapa de disparidade original como histogramas de colunas e linhas, respectivamente. No mapa V os obstáculos são projetados na imagem como linhas verticais (próximas a 90 graus), enquanto a estrada e demais planos navegáveis são projetados como linhas com inclinação superior a 90 graus. Usando o mesmo princípio, o mapa U representa em linhas horizontais a largura dos obstáculos e estruturas. Na Figura 1, fornecido um exemplo de representação de um mapa de disparidade U e V (Lima et al., 2017).

[1] Tradicionalmente as linhas na imagem são referenciadas por "v"e as colunas por "u", por este motivo o nome "Mapa de Disparidade U/V".

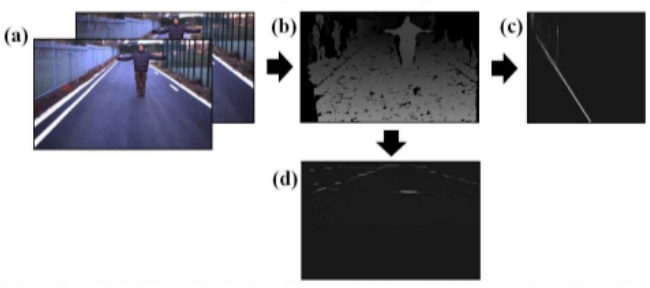

Figura 1. Exemplo de geração de mapas de disparidade U (d) e V (c) a partir de um par estéreo (a) e seu respectivo mapa de disparidade (b) (Lima et al., 2017).

2.3 Visão Monocular

A visão monocular é feita pela captura de imagens de apenas uma câmera, geralmente utilizadas em torno do veículo para monitoramento de pontos cegos, registro de acidentes, identificação de objetos, entre outras aplicações (Rosique et al., 2019). O grande problema desse método é a perda de informações de profundidade, geralmente recuperadas por meio da combinação de informações com outros sensores do veículo. No entanto, outras abordagens têm surgido com o passar dos anos, baseadas, principalmente, em redes neurais profundas, fazendo com que a detecção de obstáculos e a estimativa de distâncias com visão monocular apresentem resultados interessantes (Wu et al., 2020).

O grande problema desses métodos é o custo de processamento em função da precisão. Quanto maior a precisão, mais robusto será o modelo de aprendizado profundo e consequentemente mais poder de processamento será necessário para executá-lo em tempo real. Também é importante ressaltar que o modelo apenas detecta os objetos que consegue classificar.

3. METODOLOGIA

Neste trabalho é proposto combinar as técnicas aplicadas em visão estéreo na visão monocular, visando ao baixo custo de processamento e estimativa de distância por mapas de disparidade U/V. A solução pode ser dividida segundo o diagrama da Figura 2. Nesta Seção estão descritos os blocos desse diagrama, representando os passos necessários para a aplicação proposta.

3.1 Geração do Mapa de Disparidade

O primeiro passo para a solução proposta é a geração do mapa de disparidade apenas com a imagem da câmera monocular. Para tanto, foi utilizada uma rede neural profunda MiDaS (Ranftl et al., 2022). Essa rede foi treinada com 10 conjuntos de dados diferentes (ReDWeb, DIML, Filmes, MegaDepth, WSVD, TartanAir, HRWSI, ApolloScape, BlendedMVS, IRS) e posteriormente testada com uma alta gama de filmes 3D. Utilizando a biblioteca de aprendizado de máquina Tensorflow, o modelo foi convertido para uma versão "*lite*" reduzindo o tamanho do mesmo para ser executado em dispositivos móveis, com vista ao menor consumo de processamento possível. Na Figura 3, fornecido um exemplo da estimativa de disparidade utilizando o modelo MiDaS (Ranftl et al., 2022).

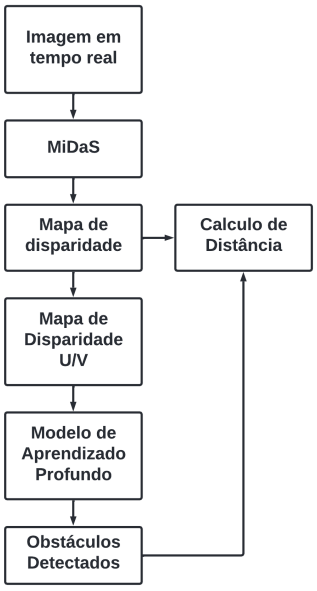

Figura 2. Fluxograma que descreve as etapas da solução proposta.

3.2 Cálculo da Distância e Posição de Obstáculos

Com a imagem da disparidade estimada, o próximo passo consistiu na formulação uma equação responsável por informar a distância dos objetos com base nos valores de disparidade fornecidos pela rede neural. Para tanto, foi necessário realizar alguns testes relacionados à consistência dos dados em termos da informação retornada. Conhecendo a posição exata de um objeto na cena, foram captadas dez imagens com o objeto sendo afastado aproximadamente cinquenta centímetros por foto. Com isso, por meio de uma regressão exponencial, foi possível encontrar a nova função geral que descreve a distância em função da disparidade, de acordo com a Figura 4. Como pode ser notado, o comportamento dessa curva está em concordância com a relação (1), onde a distância varia com o inverso da disparidade em uma curva exponencial.

A posição do objeto na cena foi, então, aproximada utilizando a igualdade de triângulos, conhecendo-se o centro óptico da imagem, o campo de visão da câmera (do inglês *Field of View* – FOV), o número de colunas em pixels e a distância do objeto até a câmera, como demonstrado na Figura 5.

Figura 3. Estimativa de disparidade pelo modelo de aprendizado profundo MiDaS (Ranftl et al., 2022).

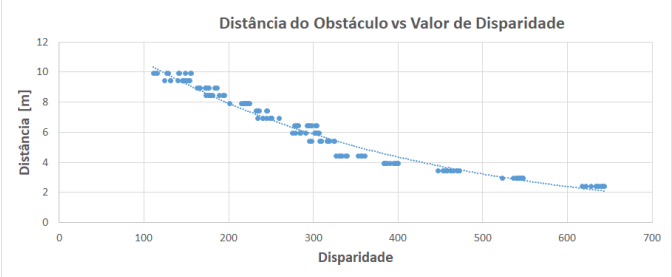

Figura 4. Gráfico da relação entre distância e disparidade.

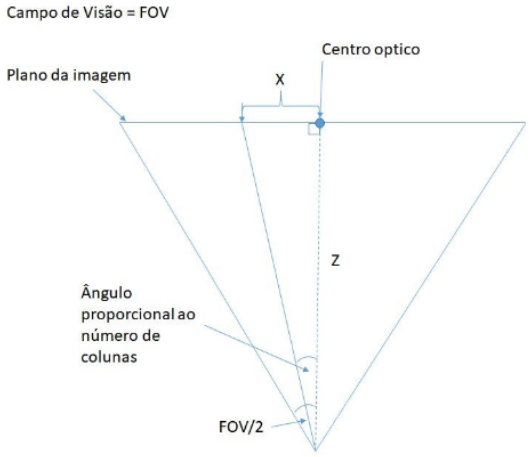

Figura 5. Relação do ângulo de visão da câmera com a posição do objeto.

3.3 Detecção de Obstáculos

Na detecção de obstáculos, foi utilizada a combinação dos mapas de disparidade U/V, como apresentado em (Lima et al., 2017), porém em conjunto com uma rede neural profunda, treinada para avaliar em conjunto os dois mapas e detectar os obstáculos. Para treinar o modelo foi utilizado uma base de dados com 5380 imagens geradas por jogos de computadores modernos (Richter et al., 2016), contendo imagens da cena e os respectivos mapas de rótulos semânticos. As imagens da cena foram passadas pelo modelo de aprendizagem profundo MiDaS, e as estimativas de disparidades fornecidas pelo mesmo, foram então convertidas em mapas de disparidades U/V. Com os mapas de rótulos semânticos, foram criados, também, mapas de disparidades U/V que não apresentam informações referentes a vias navegáveis (superfícies planas ou pouco inclinadas), ou seja, apenas com as linhas que representam os obstáculos da cena. A Figura 6 ilustra os dois processos descritos.

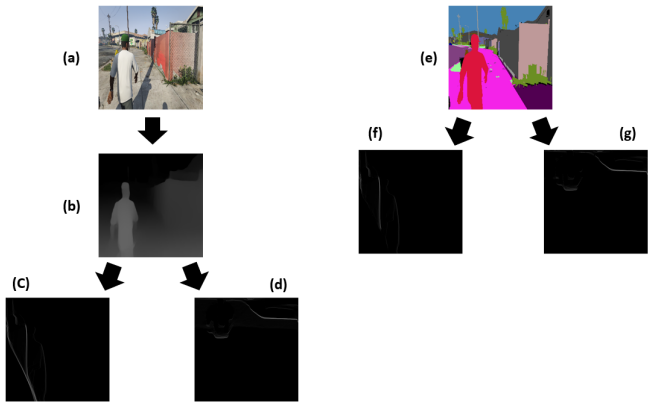

Figura 6. Processamento de imagem realizado para gerar os dados de treinamento para a rede neural, onde em (a) é a imagem original, (b) o mapa de disparidade gerado pela rede neural, (c) e (d) são os seus mapas de disparidade V e U, respectivamente, (e) é a imagem com os rótulos semânticos e (f) e (g) os respectivos mapas de disparidade V e U rotulados em área navegável e obstáculos.

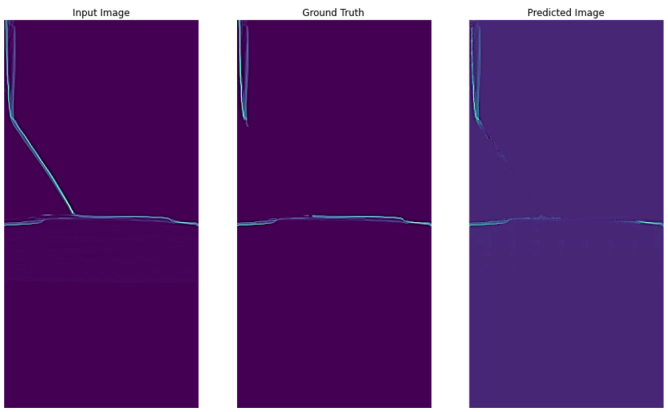

Figura 7. Entrada, imagem de referência e predição da rede neural citada anteriormente.

A arquitetura da rede convolucional escolhida foi inspirada na topologia de rede adversarial generativa condicional (cGAN), segundo o qual o modelo aprende um mapeamento de pixels das imagens de entrada e saída, chamado de peixel2peixel (Isola et al., 2017). O modelo recebeu como entrada os mapas de disparidade U/V concatenados verticalmente e foi treinado por quarenta mil épocas. A nova rede possui os seguintes parâmetros: Imagem de entrada *float* 32 com tamanho de 512x256 (altura por largura), normalizada de -1 a 1, 24 camadas e um pouco mais de 54 mil neurônios. Um exemplo da concatenação dos mapas de disparidade V/U e a predição da rede pode ser visto na Figura 7.

4. RESULTADOS EXPERIMENTAIS

4.1 Estimativa de Distância

Para se verificar a qualidade da estimativa de distância, foram realizadas algumas coletas de imagens com objetos em distâncias específicas, com pelo menos cinco medições para cada posição. Apesar de não ter sido feito um sistema rígido para garantir a repetitividade com erro mínimo, o cálculo da distância dos objetos em uma cena obteve resultados interessantes. Em um dos experimentos, dados foram coletados em uma das ruas da UFLA, com uma moto como obstáculo (Figura 8). Nesse caso, foi possível determinar a distância da moto com erro médio de 31 cm (importante ressaltar que várias fontes de erros estavam presentes no teste, como a posição da câmera, sua inclinação em relação ao solo, o ponto real de medição no obstáculo, etc.). A Figura 8 ilustra o experimento citado acima.

Figura 8. Teste realizado na avenida central da Universidade Federal de Lavras e seu respectivo mapa de disparidade.

4.2 Detecção de Obstáculos

Para a detecção de objetos, foi criado um algoritmo que faz a integração da rede neural MiDAS com o novo modelo treinado neste trabalho, visando criar máscaras com os pixels classificados, seguindo o fluxograma apresentado na Figura 2. Também foram implementados os outros dois métodos de detecção proposto neste trabalho para servir de comparação: o primeiro foi o método clássico que utiliza mapas de disparidade U/V (Lima et al., 2017), e o segundo foi a implementação de um modelo de aprendizado profundo que faz a segmentação semântica da cena. Esse último é uma variação do DeepLabV3+ (Chen et al., 2018) otimizado para *smartphone*s e que pode ser encontrado na plataforma Tensorflow Hub (Tensorflow, 2018).

Os testes foram realizados em um mesmo *notebook*, utilizando um processador Intel(R) Core(TM) i5-4200U CPU com 2,30 GHz sem o auxílio de placas de vídeo integradas. Foram realizadas as análises em 300 imagens com um algoritmo não otimizado que desconsidera os pixels que classificam o céu. Os resultados médios desses testes são apresentados na Figura 9.

Como uma segunda forma de avaliar os resultados, foi utilizado o método descrito em (Fritsch et al., 2013), que utiliza quatro métricas baseadas em pixels:

$$Precision = \frac{TP}{TP + FP}, \tag{2}$$

$$Recall = \frac{TP}{TP + FN}, \tag{3}$$

$$F_{measure} = (1 + \beta^2) * \frac{Precision * Recall}{\beta^2 * Precision + Recall}, \tag{4}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \tag{5}$$

onde TP, TN, FP e FN são os pixels com verdadeiro positivo, verdadeiro negativo, falso positivo e falso negativo, respectivamente. A *precision* representa o quanto o modelo está acertando, sendo inversamente proporcional a quantidade de falsos positivos. O *recall* funciona de fórmula análoga, sendo inversamente proporcional quantidade de falso negativo. Já *Accuracy* representa a média e o F-*measure*, a média harmônica entre a precisão e o *recall*. Os resultados dessa avaliação estão listados na Tabela 1.

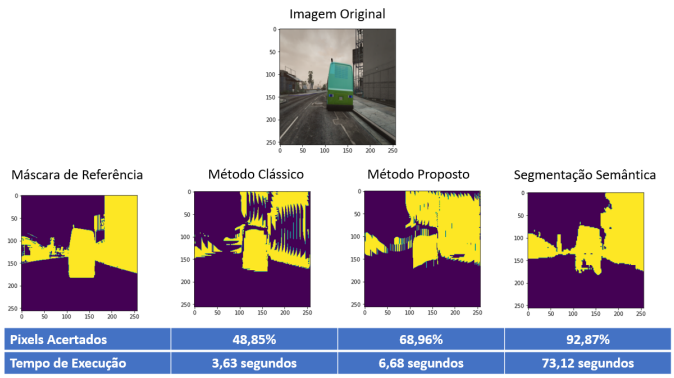

Figura 9. Resultado dos testes preliminares para os três métodos de detecção de obstáculos.

Tabela 1. Avaliação dos métodos de detecção de objetos.

Método	*Precision*	*Recall*	$F_{measure}$	*Accuracy*
Clássico	0,697	0,807	0,748	0,759
Proposto	0,745	**0,954**	0,837	**0,868**
Segmentação	**0,791**	0,930	**0,855**	0,855

Na Tabela 1, possível observar que o método proposto teve uma melhoria considerável no *Recall* e na *Accuracy* e uma alta *precision*, diminuindo significativamente os falsos positivos e falsos negativo. Também na Figura 9, possível ver que apesar do método que utiliza segmentação semântica ser superior aos outros dois métodos testados, ele tem um tempo de execução elevado, dificultando a implementação em tempo real para dispositivos com baixo poder de processamento. Já o modelo proposto, apresenta uma alta taxa de acerto e com apenas 3 segundo a mais no tempo de execução que o método clássico, apresentando um ótimo resultado, dado que foi adicionado mais uma rede neural profunda ao algoritmo.

5. CONSIDERAÇÕES FINAIS

Este artigo apresentou um método de detecção de objetos e estimativa de distância com visão monocular baseado nos métodos de visão estéreo e disparidade U/V. Além do método proposto demonstrar uma melhoria significativa na precisão da detecção de obstáculos, também foi possível estimar a distância dos objetos na cena sem a necessidade de uma segunda câmera. O tempo de execução do algoritmo em comparação com métodos convencionalmente utilizados em veículos autônomos, mostrou-se uma alternativa viável para implementações em sistemas embarcados e *smartphone*s modernos. Como trabalhos futuros, pretende-se implementar o método proposto em um *smartphone* moderno para aproveitar o processamento robusto e câmera de alta definição, com foco no desenvolvimento de aplicações que sirvam de auxílio na mobilidade de pessoas com deficiência visual e como um sensor inteligente para a plataforma VIDA.

REFERÊNCIAS

Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss (eds.), *Computer Vision – ECCV 2018*, 833–851. Springer International Publishing, Cham.

Dornaika, F. and Sappa, A.D. (2008). Real time stereo image registration for planar structure and 3d sensor pose estimation. In A. Bhatti (ed.), *Stereo Vision*, chapter 18. IntechOpen, Rijeka.

Faugeras, O. (1993). Three-dimensional computer vision: a geometric viewpoint. *MIT press*.

Feitosa Jr., A. (2019). Uma câmera que pode ser colocada em qualquer óculos É uma baita solução para deficientes visuais. URL `https://gizmodo.uol.com.br/orcam-myeye-2-ces-2019/`.

Fritsch, J., Kühnl, T., and Geiger, A. (2013). A new performance measure and evaluation benchmark for road detection algorithms. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 1693–1700. doi:10.1109/ITSC.2013.6728473.

Gao, P., Hensley, R., and Zielke, A. (2014). A road map to the future for the auto industry. *McKinsey Quarterly*, 1–11.

Gao, Y., Ai, X., Rarity, J., and Dahnoun, N. (2011). Obstacle detection with 3d camera using u-v-disparity. In *International Workshop on Systems, Signal Processing and their Applications, WOSSPA*, 239–242.

Isola, P., Zhu, J.Y., Zhou, T., and Efros, A.A. (2017). Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5967–5976.

Lima, D.A. (2015). *Sensor-based navigation applied to intelligent electric vehicles*. Ph.D. thesis, University of Technology of Compiègne.

Lima, D.A. and Pereira, G.A.S. (2010). Um sistema de visão estéreo para navegação de um carro autônomo em ambientes com obstáculos. In *Anais do XVIII Congresso Brasileiro de Automática*, 224–231.

Lima, D.A., Victorino, A.C., and Neto, A.M. (2017). A 2d/3d environment perception approach applied to sensor-based navigation of automated driving systems. In *2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)*, 1.

Liu, S., Tang, J., Zhang, Z., and Gaudiot, J.L. (2017). Computer architectures for autonomous driving. *Computer*, 50(8), 18–25.

Oh, K.S. and Jung, K. (2004). Gpu implementation of neural networks. *Pattern Recognition*, 37(6), 1311–1314.

Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., and Koltun, V. (2022). Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 44(03), 1623–1637.

Richter, S.R., Vineet, V., Roth, S., and Koltun, V. (2016). Playing for data: Ground truth from computer games. In B. Leibe, J. Matas, N. Sebe, and M. Welling (eds.), *Computer Vision – ECCV 2016*, 102–118. Springer International Publishing, Cham.

Rosique, F., Navarro, P.J., Fernández, C., and Padilla, A. (2019). A systematic review of perception system and simulators for autonomous vehicles research. *Sensors*, 19(3).

Soga, M., Kato, T., Ohta, M., and Ninomiya, Y. (2005). Pedestrian detection with stereo vision. In *21st International Conference on Data Engineering Workshops (ICDEW'05)*, 1200–1200.

Tensorflow (2018). deeplabv3-mobilenetv2-ade20k. URL `https://tfhub.dev/sayakpaul/lite-model/deeplabv3-mobilenetv2-ade20k/1/default/2`.

Wu, X., Sahoo, D., and Hoi, S.C. (2020). Recent advances in deep learning for object detection. *Neurocomputing*, 396, 39–64.

Zhu, H., Yuen, K.V., Mihaylova, L., and Leung, H. (2017). Overview of environment perception for intelligent vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 18(10), 2584–2601.