

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Estimação Estéreo usando Técnicas de Deep Learning

José Pedro Almeida Moura da Fonseca

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Pedro Santos

Orientador Externo: Joana Santos

24 de junho de 2019

© José Pedro Almeida Moura da Fonseca, 2019

Resumo

A Follow Inspiration é uma empresa tecnológica que se dedica ao desenvolvimento de soluções robóticas móveis, tendo como objetivo a incorporação de sistemas inteligentes de navegação em equipamentos tradicionais. Nesta empresa, é feito o desenvolvimento de robôs que fazem tanto reconhecimento e seguimento de pessoas como navegação autónoma. Uma vez que a maioria das aplicações da Follow Inspiration é em ambientes interiores, a principal tecnologia utilizada até agora para obter visão de profundidade é à base de infravermelhos.

Tendo surgido um projeto para ambiente exterior e também por terem havido já algumas dificuldades em ambientes interiores com grande exposição solar, procurou-se uma alternativa para obtenção de visão de profundidade. Estimação estereoscópica é uma das possíveis abordagens para encontrar profundidade em imagens, sendo utilizada em sistemas críticos como o Mars Rover e carros autónomos. É uma solução robusta e facilmente aplicável, pois apenas necessita de duas câmaras.

Nesta dissertação, são apresentadas e utilizadas duas técnicas distintas para obter mapas de disparidade, que permitem calcular a profundidade dos diversos pontos de uma imagem. A primeira é um algoritmo com resultados comprovados, o método de Semi-Global Matching, recorrendo aos recursos disponibilizados pela ferramenta OpenCV. Este algoritmo recebe um par de imagens coloridas e retorna o mapa de disparidade correspondente. A segunda abordagem implementada foi feita utilizando redes neurais. É apresentada uma rede FeedForward, lendo imagens linha a linha com um camada escondida. É feita uma exploração extensiva dos parâmetros da rede neuronal, nomeadamente do número de epochs de treino e do número de neurónios na camada escondida.

A avaliação dos resultados obtidos por estes dois métodos é feita com recurso ao dataset Middlebury. É comparado o grau de similaridade entre a saída de cada uma das duas implementações individualmente com o ground truth do dataset utilizado.

Após ser feita a validação com os datasets, é aplicado o mesmo sistema em tempo real, utilizando o sistema de obtenção de imagem estereoscópica sob avaliação pela Follow Inspiration. Os mapas de disparidade do algoritmo de SGBM foram utilizados como ground truth para a saída da rede neuronal. O índice de similaridade entre o ground truth e a saída da NN produzida é de cerca de 0.7.

Abstract

Follow Inspiration is an innovative technology company that focuses on developing mobile robotic solutions, with the purpose of incorporating smart navigation systems on traditional equipment. This company builds robots with the ability to recognize and follow people, as well as the ability to navigate autonomously. As most applications of the Follow Inspiration robots are indoors, the main technology used so far to obtain depth has been infrared.

Due to a new project requiring outdoor navigation and since there were already some difficulties in certain indoor environments with great sun exposure, an alternative was sought to obtain depth image. Stereo matching is one of the possible approaches to find depth in an image, being used in critical systems like the Mars Rover and self-driving cars. Since it only needs two cameras, it is a robust and easily applicable solution.

Within this dissertation, two different techniques are presented and used to obtain disparity maps, which can be converted into depth maps. The first is a well-established algorithm, the Semi-Global Matching method, making use of the resources provided by Open CV. This is a traditional approach that has proven results, with that being one of the reasons why it is implemented in OpenCV. This algorithm receives a pair of colored images and returns its corresponding disparity map. The second approach was implemented using neural networks. A FeedForward neural network, reading images line-by-line and with one hidden layer, is presented. An extensive exploration of the parameter space of the NN, namely number of training epochs and number of neurons in the hidden layer, is reported.

The evaluation of the results obtained by the two methods is done using the Middlebury dataset. The similarity between the output of both implementations is compared individually with the ground truth provided by the dataset.

After validating both algorithms using datasets, the same system is applied in real-time, using the stereoscopic image capture system under evaluation by Follow Inspiration. The disparity maps of the SGBM were used as ground truth to the NN output. The similarity index between ground truth and the output of the produced NN is around 0.7.

Agradecimentos

Quero agradecer aos meus orientadores, Pedro Santos e Joana Santos, pelo apoio que me deram ao longo destes meses.

Agradeço também a toda a gente na Follow Inspiration pela simpatia demonstrada diariamente e por estarem sempre prontos a ajudar em tudo o que eu precisasse.

Tenho de agradecer aos amigos com quem partilhei as minhas dores de cabeça e que tiveram a paciência para me ouvir.

Por último, agradeço à minha família porque sempre me apoiou, sempre me motivou a ser melhor e se não fosse por ela eu não teria chegado aqui.

José Pedro Fonseca

*“Anyone who conducts an argument by appealing to authority is not using his intelligence;
He is just using his memory.”*

Leonardo da Vinci

Conteúdo

1	Introdução	1
1.1	Enquadramento	1
1.2	Motivação	1
1.3	Objetivos e Contribuições	2
1.4	Estrutura da Dissertação	3
2	Revisão Bibliográfica	5
2.1	Algoritmos de Block-Matching	5
2.2	Redes Neuronais	6
2.2.1	Conceito	6
2.2.2	Estado da Arte	7
2.3	Conclusões	8
3	Desenvolvimento e Setup Experimental	11
3.1	Técnicas de Desenvolvimento e Avaliação	11
3.1.1	Métodos de Block-Matching	11
3.1.2	Desenvolvimento de Redes Neuronais	12
3.1.3	Datasets	12
3.1.4	Avaliação dos Resultados	13
3.2	Setup Experimental da Follow Inspiration	13
4	Desenvolvimento dos Algoritmos de Estimação Estereoscópica	17
4.1	Descrição, Motivação e Arquitetura dos Algoritmos Selecionados	17
4.1.1	Semi-Global Block-Matching	17
4.1.2	Redes Neuronais	17
4.2	Aquisição de Imagem e Parametrização do Algoritmo de SGBM	18
4.3	Desenvolvimento da Rede neuronal	20
4.3.1	Pré-Processamento	20
4.3.2	Data Augmentation	22
4.3.3	Parâmetros da Rede Neuronal	23
4.4	Conclusão	26
5	Caracterização de Desempenho com Sequência de Imagens Díspares	29
5.1	Métrica de avaliação	29
5.2	Semi-Global Block-Matching	29
5.3	Rede Neuronal de Disparidade Direta	30
5.4	Conclusão	31

6	Implementação em Tempo Real no Setup de Vídeo Estereoscópico	33
6.1	Setup e Captura de Vídeo Estereoscópico	33
6.2	Esforço de Integração	34
6.3	Resultados	36
6.4	Conclusão	38
7	Conclusões e Trabalho Futuro	39
7.1	Satisfação dos Objectivos	39
7.2	Principais Dificuldades	39
7.3	Trabalho Futuro	40
7.3.1	Semi-Global Block-Matching	40
7.3.2	Rede Neuronal de Disparidade Direta	40
	Referências	43

Lista de Figuras

1.1	wGO Retail	2
2.1	Par de imagens stereo (esquerda em cima, direita em baixo), retirado de [1]	6
2.2	Rede neuronal com múltiplas camadas, retirada de [2]	7
2.3	Exemplo de BackPropagation, retirado de [3]	8
3.1	Par de Imagens do dataset KITTI	13
3.2	Conjunto de Imagens do dataset Middlebury utilizado neste projeto (Esquerda, Direita e Disparidade)	14
3.3	Câmara Astra	15
4.1	Resultados iniciais	18
4.2	Workflow com redes neurais	19
4.3	Função de ativação utilizada	20
4.4	Canais de Cor	21
4.5	Mapa de disparidade antes (esquerda) e depois (direita) da normalização	22
4.6	Exemplo de zero-padding em imagens	23
4.7	Técnica de Data Augmentation aplicada ao dataset KITTI (original, intensidade aumentada e intensidade reduzida, de cima para baixo)	24
4.8	Relação entre o número de neurónios na camada escondida e a qualidade dos resultados com imagem do dataset Middlebury	24
4.9	Relação entre o número de neurónios na camada escondida e a qualidade dos resultados com imagem das câmaras Astra	25
4.10	Comparação entre 2, 5, 10, 20 (cima), 40, 80 e 160 neurónios na camada escondida com o ground truth (baixo) com uma imagem obtida através das câmaras Astra	25
4.11	Relação entre o número de epochs e a qualidade dos resultados com imagem do dataset Middlebury	26
4.12	Relação entre o número de epochs e a qualidade dos resultados com imagem das câmaras Astra	26
4.13	Comparação entre 500, 1000, 2000 (cima), 5000 e 10000 epochs com o ground truth (baixo) com uma imagem do dataset Middlebury	27
5.1	Aloe, Aloe Cortado, Aloe Cortado e Filtrado e Ground Truth Cortado, respetivamente	30
5.2	Aloe, Aloe com filtro de mediana e GT sub-amostrado	31
6.1	Disposição das câmaras	34
6.2	Diagrama de obtenção de GT para treinar a rede neuronal	35
6.3	Integração experimental com o wGO Retail	35

6.4	Histograma com os resultados dos frames de treino	36
6.5	Histograma com os resultados dos frames de teste	37
6.6	Imagen RGB, Imagem de disparidade obtida através de SGBM e Resultado da Rede para uma imagem de treino (1)	37
6.7	Imagen RGB, Imagem de disparidade obtida através de SGBM e Resultado da Rede para uma imagem de treino (2)	37
6.8	Imagen RGB, Imagem de disparidade obtida através de SGBM e Resultado da Rede para uma imagem de teste	38

Lista de Tabelas

5.1	Resultados dos métodos tradicionais	30
5.2	Resultados da rede neuronal com 104 neurónios na camada escondida	31
5.3	Resultados da rede neuronal com 208 neurónios na camada escondida	32

Abreviaturas e Símbolos

BP	BackPropagation
FF	FeedForward
FI	Follow Inspiration
GPU	Graphics Processing Unit
GT	Ground Truth
NN	Neural Network
SGBM	Semi-Global Block-Matching
USB	Universal Serial Bus

Capítulo 1

Introdução

Neste capítulo apresentam-se as motivações e contexto do trabalho proposto, assim como os objetivos que se pretendem atingir. Será também feita uma breve apresentação à estrutura e organização do documento.

1.1 Enquadramento

Com a crescente automatização de equipamentos do dia-a-dia, torna-se cada vez mais essencial o desenvolvimento de sistemas de visão computacional robustos de baixo custo. A Follow Inspiration é uma empresa tecnológica que se dedica ao desenvolvimento de soluções robóticas móveis, tendo como objetivo a incorporação de sistemas inteligentes de navegação em equipamentos tradicionais. O wGO, observável na figura 1.1, é um carro de compras autónomo desenhado para seguir pessoas com mobilidade reduzida, acabando com a necessidade de empurrar ou sequer controlar o carro. Atualmente, cerca de 1% da população dos países desenvolvidos necessita de uma cadeira de rodas para se movimentar, o que estabelece logo à partida um público-alvo constituído por 10 milhões de pessoas. Esta é uma tecnologia flexível aplicável em diferentes contextos, nomeadamente na indústria, proporcionando diversas vantagens e maior conforto ao utilizador. Para funcionar corretamente, o wGO necessita de visão computacional de modo a saber o que o rodeia. Assim, a tecnologia wGO inclui dois módulos base de desenvolvimento: módulo de seguimento (forte componente de visão por computador) e módulo de navegação natural (caracterização do ambiente e capacidade de navegar autonomamente) [4] [5].

1.2 Motivação

O sistema de visão do sistema wGO recorre a câmaras para obtenção de imagem RGB-D. Acontece que os sensores presentes nesta câmara que permitem capturar a profundidade da imagem são à base de radiação InfraVermelha. Esta especificação tem muito bons resultados em ambientes indoor, mas não é aplicável em ambientes outdoor ou até ambientes indoor com grande



Figura 1.1: wGO Retail

exposição de raios solares, exatamente por a deteção de profundidade ser feita à base de InfraVermelhos. Foi então decidido adotar uma abordagem de reconstrução stereo, utilizando a informação de duas câmaras com diferentes pontos de vista para criar uma imagem de profundidade. Existem já diversos métodos para efetuar estimação stereo, cada um com as suas vantagens. É relevante referir que este sistema requer tempos de resposta muito curtos, para evitar a colisão do robô com eventuais obstáculos a velocidades elevadas. É assim que surge a problemática investigada nesta dissertação [6].

1.3 Objetivos e Contribuições

Com esta dissertação, pretende-se investigar métodos de reconstrução stereo que permitam obter uma imagem de profundidade em tempo real. Espera-se criar um módulo de estimação estereoscópica outdoor e em tempo real. Para chegar a este objetivo, é necessário fazer um estudo da tecnologia existente na área, do qual este documento faz parte, onde são identificadas as principais soluções de visão estereoscópica baseadas em redes neurais já existentes. Com base nas conclusões retiradas do passo anterior, é estabelecido um plano, passando pela implementação e subsequente teste das metodologias escolhidas, sujeito a avaliação segundo métricas de desempenho escolhidas.

São utilizadas duas técnicas distintas para atingir estes objetivos. A primeira é um algoritmo com resultados comprovados, o método de Semi-Global Matching, recorrendo aos recursos disponibilizados pela ferramenta OpenCV. Este algoritmo recebe um par de imagens coloridas e retorna o mapa de disparidade correspondente. A segunda abordagem implementada foi feita utilizando redes neuronais. É apresentada uma rede FeedForward, lendo imagens linha a linha com uma camada escondida. É feita uma exploração extensiva dos parâmetros da rede neuronal, nomeadamente do número de epochs de treino e do número de neurónios na camada escondida.

1.4 Estrutura da Dissertação

A estrutura deste documento é a seguinte:

- No capítulo 2 é feita a revisão bibliográfica e é descrito o estado da arte.
- No capítulo 3 é feita a apresentação das principais ferramentas utilizadas no desenvolvimento e validação dos algoritmos utilizados, assim como na posterior integração com o sistema físico.
- No capítulo 4 são descritos os métodos de estimação stereo utilizados no desenvolvimento deste projeto e os processos de desenvolvimento e afinação dos mesmos, utilizando os datasets disponíveis na literatura.
- No capítulo 5 são apresentados os resultados obtidos pelos dois métodos com a utilização de datasets.
- No capítulo 6 é descrita a integração com o sistema físico da Follow Inspiration e são apresentados alguns resultados obtidos.
- Por último, no capítulo 7, é feita a análise dos resultados obtidos e são apresentados alguns possíveis melhoramentos ao sistema.

Capítulo 2

Revisão Bibliográfica

Neste capítulo apresenta-se o estado da arte, que inclui ideias e conceitos essenciais à compreensão do assunto em questão. Como tal, é feito um levantamento das diferentes técnicas utilizadas na atualidade para responder ao mesmo problema. Começa-se pela análise de métodos tradicionais, passando-se posteriormente à utilização de redes neurais.

2.1 Algoritmos de Block-Matching

Uma das abordagens utilizadas para estimativa stereo utiliza uma característica presente em todas as imagens, chamada de aresta. A deteção de arestas não é nada mais nada menos do que a identificação dos pontos em que intensidade da imagem muda de forma brusca ou apresenta descontinuidades. Mais especificamente para visão estereoscópica, após ser feita segmentação por áreas das duas imagens provenientes dos dois diferentes pontos de vista, é feita a sua fusão, da qual resultam descontinuidades na intensidade resultantes da discrepância existente entre as duas imagens. Fazendo a deteção de arestas do resultado, obtém-se o mapa de profundidade [7] [8].

Um recurso muito utilizado para visão estereoscópica é o OpenCV. Um dos artigos analisados baseados em OpenCV utiliza Harris corner detection para chegar aos chamados pontos de feature. Tal como o nome indica, o Harris corner detection é um algoritmo para deteção de cantos nomeado a partir de um dos seus criadores e é muito utilizado em pré-processamento de imagem. Utilizando estes pontos de feature, faz-se a fusão das duas imagens utilizando o algoritmo *sparse point matching* [9] [10] [11]. A biblioteca do OpenCV já tem também implementadas algumas classes e funções eficientes para fazer a estimativa stereo de pares de imagens. Apresenta-se um exemplo de par de imagens na figura 2.1.

Uma delas chama-se StereoBM que, tal como o nome indica, faz o computação stereo utilizando block matching. Este algoritmo recebe imagens em tons de cinza e retorna um mapa de disparidade. É de notar que este algoritmo foi inicialmente criado para uma pequena plataforma em hardware e foi posteriormente criada uma implementação adaptada para o OpenCV, sendo por isso destinado a plataformas com poucos recursos [12].

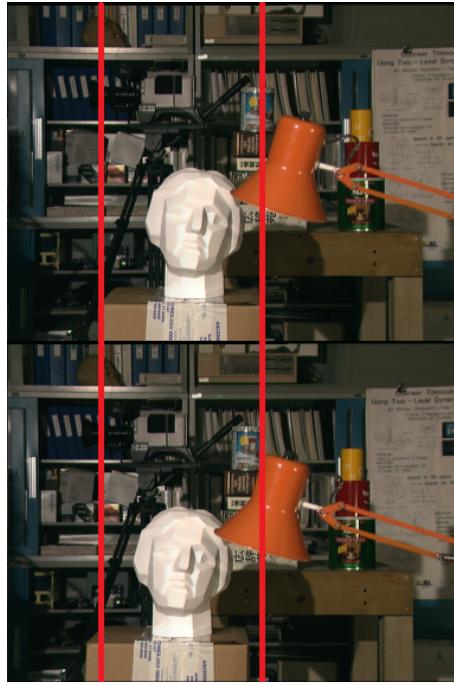


Figura 2.1: Par de imagens stereo (esquerda em cima, direita em baixo), retirado de [1]

Há outra implementação disponibilizada por esta ferramenta que é mais recente, mais complexa e que apresenta melhores resultados. É denominada de StereoSGBM, por recorrer à técnica de Semi-Global Matching de Heiko Hirschmuller aplicada a blocos. Há algumas diferenças entre as duas abordagens, tais como o matching em OpenCV ser feito em blocos em contraste com o matching de píxeis. No entanto, é possível estabelecer o tamanho dos blocos para apenas 1 píxel. Por defeito, o varrimento de píxeis/blocos no OpenCV é single-pass, feito apenas em 5 direções, sendo também possível alterar este parâmetro para 8 direções como na implementação de Heiko Hirschmuller, tendo um custo na rapidez e necessidade de memória do algoritmo [13].

2.2 Redes Neuronais

2.2.1 Conceito

Redes neurais artificiais são sistemas computacionais inspirados nas redes neurais biológicas, com a intenção de as simular. Estas estruturas são compostas por múltiplos nós que se encontram ligados de um modo específico, dependendo da rede. A ligação entre os nós e os pesos atribuídos a cada ligação é o que vai definir o funcionamento da rede na totalidade.

As redes neurais, por serem uma simulação de um cérebro humano, têm de passar por um processo de aprendizagem. Depois de concluído, esta estrutura permite velocidades de respostas bastante altas, o que é essencial para este projeto.

Duas características muito importantes a definir numa rede neuronal são o número de camadas de neurónios presentes na rede e o número de neurónios por camada [14] [15] [16]. Serão testados

vários valores e, utilizando a métrica de avaliação escolhida em conjunto com outros fatores (como a velocidade de processamento), chegar-se-á aos valores finais. Na figura 2.2 está representada uma rede neuronal com 4 camadas, sendo 2 delas escondidas.

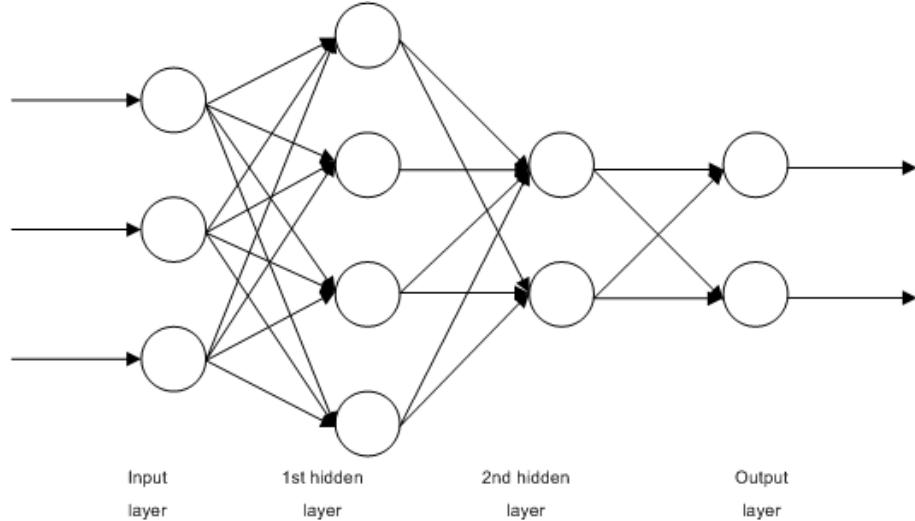


Figura 2.2: Rede neuronal com múltiplas camadas, retirada de [2]

2.2.2 Estado da Arte

São várias as abordagens tipicamente utilizadas para fazer estimação estereoscópica com recurso a redes neurais com o objetivo de maximizar o desempenho do sistema. A utilização de redes neurais convolucionais revelou-se vantajosa, pois não necessita de pós-processamento adicional, ao contrário de abordagens recorrendo a métodos tradicionais de stereo matching. As abordagens adotada nos artigos analisados são soluções end-to-end que necessitam de menos tempo de processamento por par de imagens do que os métodos tradicionais, resultando em sistemas de visão em tempo real [17] [18].

Foram identificados os seguintes métodos de estimação stereo que utilizam redes neurais:

- Redes neurais convolucionais
- Autómatos Celulares
- Redes do tipo Vitality Conservation em conjunto com redes do tipo BackPropagation
- Redes neurais em paralelo
- Conditional Random Fields

Já desde há mais de duas décadas que se recorre à utilização de redes neurais celulares para obtenção de um sistema de visão em tempo real. Estas redes combinam as características de uma rede neuronal convencional com as características dos autómatos celulares. Uma das principais

características dos autómatos celulares é a ligação de uma célula com as células vizinhas e a dependência da sua ativação. Esta abordagem permite tratar esta tarefa como um problema de otimização [19] [20].

Outro método utilizado recorre à combinação de dois tipos diferentes de redes neurais, sendo eles vitality conservation e backpropagation. Vitality conservation é a estimativa de frequência com que cada ligação é utilizada e utilização dessa informação para reduzir o tempo de aprendizagem. Posteriormente, é utilizada backpropagation para reduzir o erro existente na primeira [21]. Na figura 2.3 é exemplificado o processo de backpropagation.

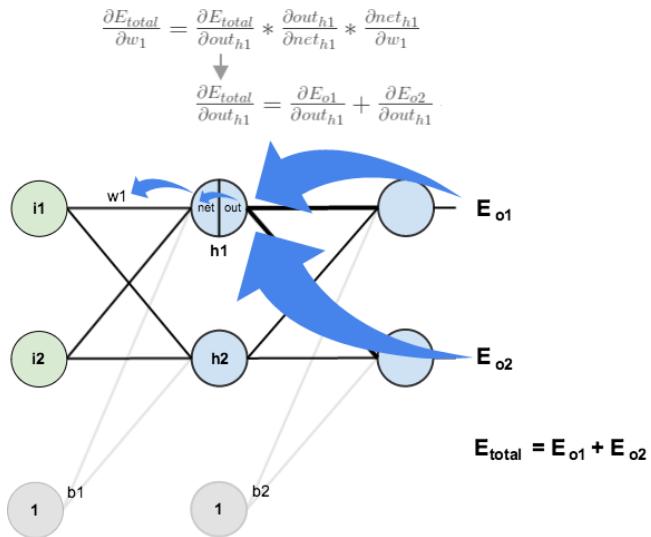


Figura 2.3: Exemplo de BackPropagation, retirado de [3]

Mais recentemente, algumas abordagens experimentais permitiram obter resultados promissores, nomeadamente a utilização de redes neurais em paralelo, tendo sido obtidas velocidades de processamento dez vezes superiores a outros sistemas que utilizam redes neurais [22].

Outra equipa recorreu à combinação de redes neurais convolucionais com conditional random fields. A inferência dos conditional random fields está formulada como uma rede neuronal recorrente, estabelecendo-se uma semelhança com os dois métodos analisados anteriores, na medida em que, de certa forma, associa dois tipos diferentes de redes neurais. Os resultados obtidos por este sistema foram melhores do que os resultados típicos de sistemas que utilizam conditional random fields, evidenciando um potencial benefício na utilização de redes neurais em conjunto com outras metodologias para estimação stereo [23].

2.3 Conclusões

Todas as alternativas analisadas revelam grande potencial. Será feita uma comparação utilizando uma métrica definida e descrita neste documento de modo a determinar qual dos métodos permite atingir o melhor compromisso entre velocidade de processamento e precisão, tendo em

conta que o sistema deve funcionar em ambiente exterior. Esta última característica é bastante relevante, pois não está presente em todos os métodos estudados. Pode-se desde já assumir que seria desejável a utilização de combinações de diferentes tipos de redes neurais associadas a outros métodos de estimativa estereoscópica.

Capítulo 3

Desenvolvimento e Setup Experimental

Neste capítulo são apresentadas as principais ferramentas utilizadas no desenvolvimento e validação dos algoritmos utilizados, assim como na posterior integração com o sistema físico.

3.1 Técnicas de Desenvolvimento e Avaliação

Aqui são descritas as ferramentas utilizadas para o desenvolvimento e validação dos algoritmos.

3.1.1 Métodos de Block-Matching

Para a aplicação do algoritmo utilizando block-matching, recorreu-se ao OpenCV. O OpenCV é uma biblioteca open source de funções orientadas a visão computacional. É composto por vários módulos, cada um dedicado a determinada área de visão. Os módulos utilizados foram os seguintes:

Core — É o módulo essencial a toda a funcionalidade do OpenCV onde estão as principais estruturas classes e estruturas, tais como a classe Mat que é utilizada para armazenar imagens;

Hightgui — É o módulo responsável por funções que facilitam a utilização da biblioteca, incluindo funções que permitem criar e manipular janelas para a visualização de imagens, assim como leitura de vídeo

Imgproc — Este é o módulo no qual estão presentes as funções mais simples de processamento de imagem, permitindo usar matemática morfológica, redimensionamento de imagens, aplicação de máscaras e outros filtros

Imgcodecs — Este módulo providencia opções mais variadas de leitura e escrita de imagens, controlando a resolução de cor e o número de canais de cor (imagem colorida ou escala de cinzas, por exemplo)

Videoio — Este módulo é utilizado para a leitura sucessiva de diferentes frames em tempo real, sendo também útil para a leitura de datasets e na saída de resultados (retornar um vídeo em vez de uma série de frames)

Calib3d — É este o módulo mais importante para este projeto, pois é aqui que estão presentes as classes e respectivas funções que são usadas na estimativa stereo. Também inclui funções para calibração de câmaras, que serão úteis na implementação deste sistema em ambiente real

Esta ferramenta foi utilizada utilizando a sua linguagem de programação nativa, C++.

3.1.2 Desenvolvimento de Redes Neuronais

A programação do algoritmo que utiliza redes neurais foi feita utilizando a linguagem de programação Python. Python é uma linguagem de programação de alto nível muito versátil.

O propósito inicial de utilizar esta linguagem era o de aproveitar as capacidades de aceleração de treino de redes neurais através do uso de GPU disponibilizadas pelo PyTorch. No entanto, este recurso acabou por não ser utilizado porque não havia compatibilidade com o computador onde foi criado todo o programa. Ainda assim, valeu a pena utilizar esta linguagem por ser tão intuitiva e estar em crescimento, tendo sido uma aprendizagem de grande valor acrescido.

3.1.3 Datasets

Numa fase inicial de desenvolvimento e posterior validação dos algoritmos, foram utilizados dois datasets stereo distintos: KITTI 2015 e Middlebury 2006.

O dataset KITTI é um dataset muito utilizado, que se destaca por ter imagens em ambiente real exterior, ao contrário do dataset Middlebury. É muito extenso, sendo útil para o treino de redes neurais. No entanto, o formato do ground truth deste dataset não é compatível com a arquitetura de rede utilizada, pelo que é necessário contornar esta situação. A parte do dataset utilizada é composta por 4200 frames, dos quais foram utilizados 1000. O tamanho das imagens ronda 1392x512 (não têm todas o mesmo tamanho). Na figura 3.1 pode observar-se um par de imagens proveniente deste dataset.

Foi utilizado o resultado do algoritmo de métodos tradicionais para criar um ground truth deste dataset compatível com a rede neuronal. Apesar de haver uma transmissão do ruído do primeiro algoritmo para o segundo, esta operação foi feita com sucesso [24] [25] [26] [27] [28].

O dataset Middlebury é um dos mais utilizados em visão computacional, providenciando várias imagens com um ground truth de alta qualidade. Infelizmente, é bastante limitado em quantidade de imagens e diversidade de cenários. A parte do dataset utilizada é composta por 21 frames, dos quais foram utilizados 10. O tamanho das imagens ronda 420x370 (não têm todas o mesmo tamanho). Será utilizado como dataset de avaliação [29] [30] [31] [32] [33]. Na figura 3.2 é dado um exemplo de par de imagens deste dataset em conjunto com a sua disparidade, também fornecida pelo dataset.

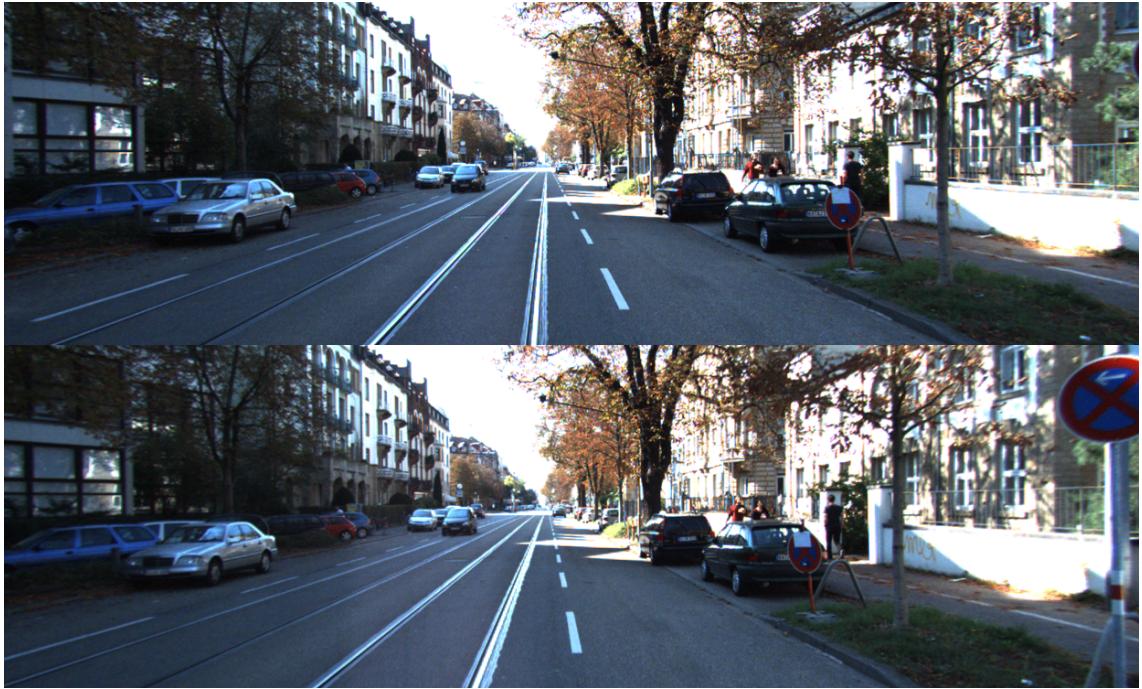


Figura 3.1: Par de Imagens do dataset KITTI

3.1.4 Avaliação dos Resultados

Por último, foi utilizado o Matlab para ser feita a avaliação de resultados. O Matlab possui um ambiente de desenvolvimento e programação com muita versatilidade, tendo inúmeras funções implementadas para todo o tipo de computação.

Uma grande vantagem na utilização desta ferramenta é a fácil leitura e processamento de imagem. É exatamente esta funcionalidade que é útil para este projeto, permitindo aplicar alguns filtros simples a imagens e ainda fazer comparação de imagens. Também tem implementadas algumas técnicas de estimação stereo, mas a utilização desta ferramenta limitou-se à comparação dos resultados com os seus ground truths correspondentes [34].

3.2 Setup Experimental da Follow Inspiration

Sem entrar em demasiados detalhes por motivos de confidencialidade, a imagem de profundidade é utilizada para reconhecimento de utilizadores e seguimento de pessoas pela Follow Inspiration.

Para a integração com o sistema da Follow Inspiration, foram utilizadas câmaras Astra, como aquela representada na figura 3.3. Estas câmaras não são as mais adequadas para este tipo de aplicação, pois possuem capacidades desnecessárias a este projeto, como por exemplo a deteção de profundidade utilizando radiação infravermelha, e também por terem uma qualidade limitada de imagem RGB e baixa resolução. Além disso, ainda envolvem um esforço adicional para a



Figura 3.2: Conjunto de Imagens do dataset Middlebury utilizado neste projeto (Esquerda, Direita e Disparidade)

instalação de drivers e bibliotecas para ser possível a sua utilização, o que não aconteceria com o uso de câmaras convencionais. No entanto, eram as únicas câmaras disponíveis em número par e



Figura 3.3: Câmara Astra

é de extrema importância que ambas as câmaras sejam iguais (para evitar fazer correções de cor e eventuais retificações).

Para este projeto, a funcionalidade de utilizar radiação infravermelha para deteção de profundidade não foi utilizada (é exatamente isso que se pretende substituir para poder haver visão com profundidade em ambiente exterior), mas é possível ser utilizada para a criação de datasets de treino. No entanto, esta capacidade não foi explorada por ser mais vantajoso utilizar o algoritmo de métodos tradicionais para este efeito (os sensores de profundidade não estão alinhados com as câmaras e apenas permitiriam a criação de datasets em ambiente interior) [35] [36].

Capítulo 4

Desenvolvimento dos Algoritmos de Estimação Estereoscópica

Neste capítulo são descritos os métodos de estimação stereo utilizados no desenvolvimento deste projeto e os processos de desenvolvimento e afinação dos mesmos, utilizando os datasets disponíveis na literatura.

4.1 Descrição, Motivação e Arquitetura dos Algoritmos Selecionados

Aqui é feita uma descrição geral do método de block-matching utilizado, seguido do método com recurso a redes neurais.

4.1.1 Semi-Global Block-Matching

A implementação selecionada de métodos tradicionais selecionada para este projeto é baseada numa técnica de semi-global block-matching. Entre outras, uma grande vantagem que esta técnica tem sobre a outra implementada em OpenCV que já foi referida na revisão bibliográfica é a utilização de três canais de cor, aproveitando mais informação da imagem para fazer a estimação stereo. Este é um método que faz matching semi-global de blocos, utilizando a função de custo Birchfield-Tomasi em vez da informação mútua para compensar as diferenças radiométricas nas funções de entrada [37]. Esta técnica de matching semi-global era um dos melhores algoritmos de estimação estereoscópica na altura em que foi criado [13], sendo que o desempenho da versão utilizada neste projeto é ligeiramente inferior de modo a promover a sua velocidade execução.

4.1.2 Redes Neuronais

Apesar de a maior parte das técnicas de estimação stereo com recurso a redes neurais analisadas na revisão bibliográfica serem feitas com recurso a redes neurais convolucionais, neste projeto foi tomada uma abordagem distinta. Inicialmente, o plano passava pela implementação de uma rede convolucional, mas após alguns testes utilizando redes de feedforward com resultados

aceitáveis, representados na figura 4.1, foi decidido investigar esta alternativa. Adicionalmente, esta abordagem é bastante robusta, pois funciona apenas à base de intensidades de píxeis, em oposição à detecção de features realizada em redes convolucionais.

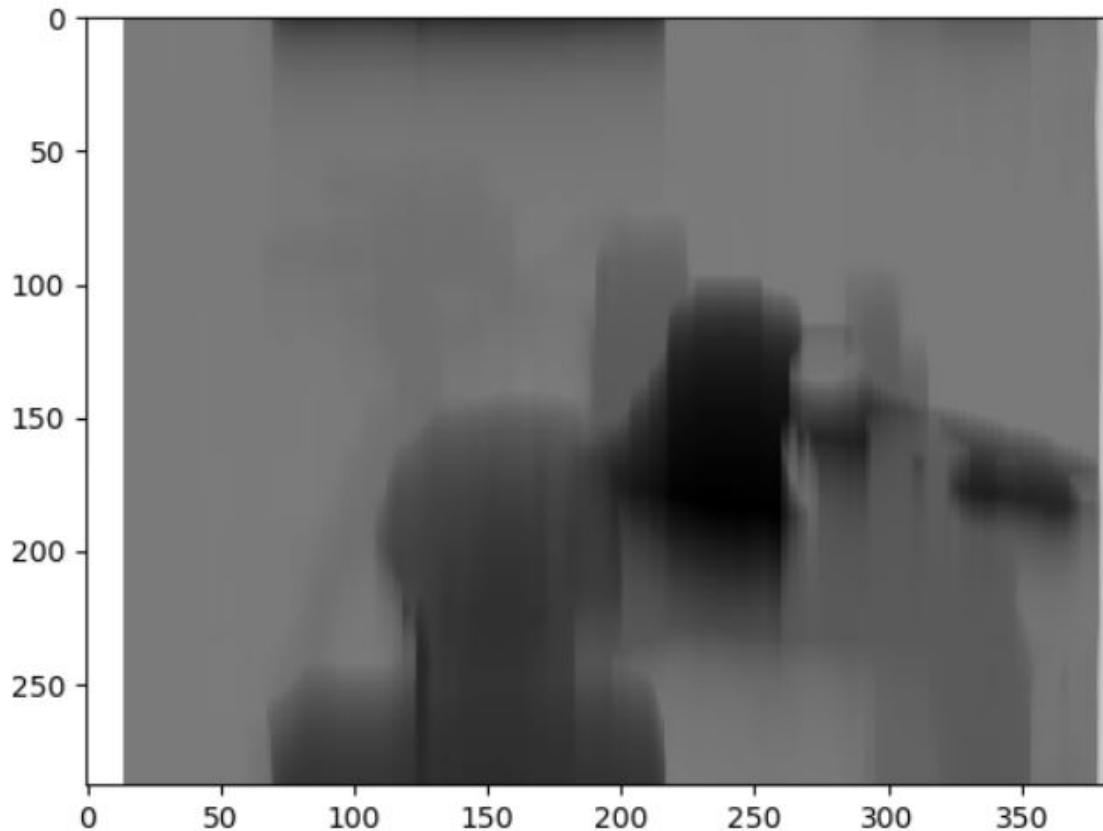


Figura 4.1: Resultados iniciais

A arquitetura (apresentada na figura 4.2) desta rede neuronal passa pela abstração do conceito de imagem, passando a considerar a imagem como uma sucessão de vetores de intensidade. Assim sendo, a rede recebe um par de linhas de píxeis (primeira linha da imagem esquerda e primeira linha da imagem direita, por exemplo) e retorna uma linha com a sequência de intensidades do mapa de disparidade. Mesmo no ser humano, a profundidade é detetada principalmente através de disparidades horizontais (perspetiva dos dois olhos), pelo que surgiu algum interesse em investigar esta arquitetura. A função de ativação utilizada é a sigmóide, representada na figura 4.3.

4.2 Aquisição de Imagem e Parametrização do Algoritmo de SGBM

A aquisição de imagem para a validação com datasets é feita através do armazenamento em dois vetores distintos do tipo Mat, um para imagens da esquerda, outro para imagens da direita. No entanto, o processamento das imagens em si é feito individual e sequencialmente.

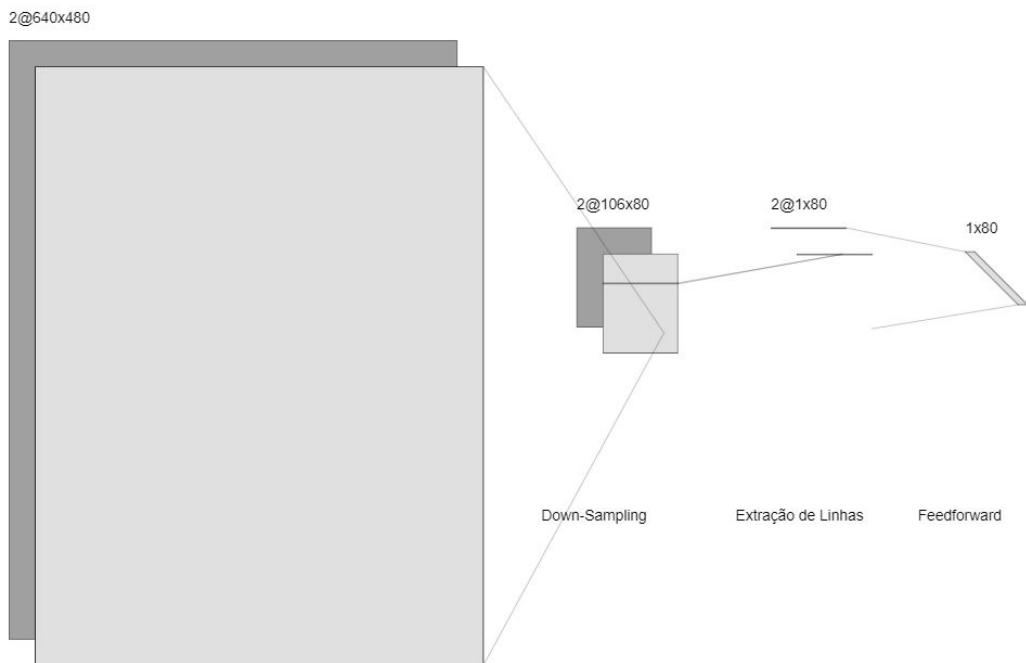


Figura 4.2: Workflow com redes neurais

Apesar de trazer uma complexidade acrescida e exigir maior poder de processamento, decidiu-se utilizar imagem BGR (na biblioteca OpenCV, a codificação de imagem é feita nesta ordem, como se observa na figura 4.4, porque este formato era popular entre produtores de câmaras na altura em que esta ferramenta começou a ser implementada, por volta do ano 2000) de modo a obter resultados com melhor qualidade.

Há alguns parâmetros que é necessário definir e afinar para se atingir os melhores resultados.

Estes são alguns dos parâmetros que é possível controlar:

Mode — Neste parâmetro é possível alterar o modo do algoritmo para corresponder à implementação inicial (8 direções em vez de 5). No entanto, optou-se por seguir a abordagem single-pass

BlockSize — Este parâmetro define o tamanho do bloco de matching. Por motivos computacionais, tem de ter sempre um valor ímpar igual ou superior a um. Como foi referido anteriormente, quando o valor deste parâmetro é um, o matching é feito píxel a píxel. Neste projeto, foi decidido utilizar este mesmo valor unitário

MinDisparity — Representa o valor mínimo de disparidade possível na imagem. Por defeito, este valor é zero e apenas se altera quando há distorções nas imagens causadas por determinados algoritmos. Na implementação deste projeto, o valor deste parâmetro é zero

P1 — Este é um de dois parâmetros que controlam a suavidade da disparidade. Tipicamente, o valor deste parâmetro é igual ao número de canais de cor das imagens em questão (um para

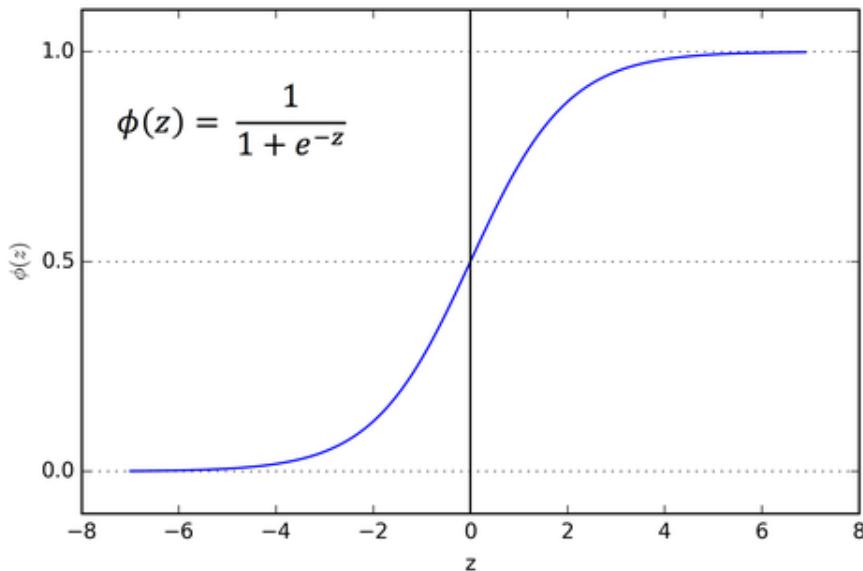


Figura 4.3: Função de ativação utilizada

imagens em tons de cinza, 3 para imagens a cor como é o caso), multiplicado pelo quadrado do tamanho do bloco de matching, multiplicando-se ainda este valor por oito

P2 — De forma semelhante ao parâmetro anterior, também serve para controlar a suavidade da disparidade e o seu valor típico é alcançado utilizando a mesma fórmula à exceção da constante, que em vez de ser oito é trinta e dois

Após terem sido executadas todas as operações anteriormente descritas, é necessário normalizar os valores do mapa de disparidade para um intervalo que permita analisar melhor as imagens, pois o mapa de disparidade criado tem, geralmente, valores muito pequenos e acaba por ser demasiado escuro para se distinguirem diferenças, como se pode observar na figura 4.5. É, então, feita a normalização para um intervalo de intensidades de 0 a 255, convertendo ao mesmo tempo o formato da imagem para CV_8U, em tons de cinza portanto.

4.3 Desenvolvimento da Rede neuronal

Aqui é descrito todo o processo de desenvolvimento da rede neuronal utilizada neste projeto. Inicialmente é feito algum pré-processamento, utilizando-se data augmentation por vários motivos a explicar, passando-se, finalmente, à exploração dos parâmetros da rede.

4.3.1 Pré-Processamento

São feitos vários passos de pré-processamento antes que as imagens possam ser utilizadas para treinar a rede neuronal.

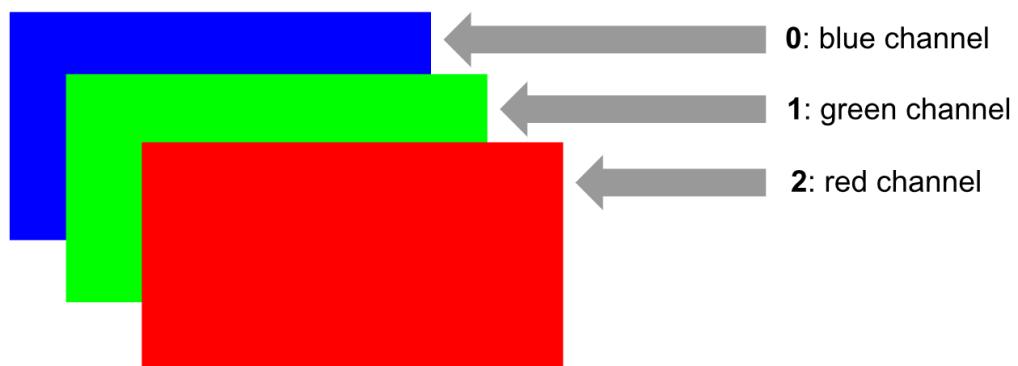


Figura 4.4: Canais de Cor

Inicialmente, é reduzida a resolução das imagens para diminuir o número de neurónios e, subsequentemente, aumentar a velocidade de treino da rede. São utilizados diferentes resoluções para diferentes imagens devido a diferentes *aspect ratios*. As imagens do dataset Middlebury foram reduzidas para uma resolução de 104x90 (a partir de 214x185), enquanto que as do dataset KITTI foram reduzidas para 300x90 (a partir de cerca de 1242x375, nem todas as imagens têm a mesma resolução). Quando se passou para a implementação com imagens obtidas com as câmaras Astra, como o treino com datasets havia sido muito demorado, optou-se por uma resolução não superior às dos datasets, escolhendo-se uma resolução de 80x106 (reduzida de 480x640).

Ainda seguindo a mesma filosofia, as imagens são convertidas para tons de cinza. Naturalmente, há alguma informação que se perde mas serve como uma prova de conceito a esta arquitetura de NN. É evidente que este passo não é necessário nas imagens de ground truth, pois já estão em escala de cinzas.

Outro passo que é necessário antes de treinar a rede é garantir que as imagens têm todas a mesma dimensão. Acontece que algumas imagens dos datasets utilizados, nomeadamente do KITTI, têm dimensões ligeiramente diferentes. São apenas alguns píxeis de diferença, mas isso é suficiente para ser incompatível com a NN quando se faz multiplicação de matrizes. Este detalhe é mitigado utilizando a técnica de zero-padding, exemplificado na figura 4.6. Verifica-se qual é o tamanho máximo de cada dimensão (altura e largura) de todas as imagens do dataset em questão e acrescentam-se zeros nas que não atingem essa dimensão. Mais uma vez, trata-se de uma questão de poucos píxeis, pelo que os resultados não são afetados por este processo.

Em último lugar, garante-se que nenhuma das imagens tem píxeis com valores 0 ou 1, pois estes valores interferem com a capacidade de aprendizagem da rede neuronal. De modo a evitar que isto aconteça, faz-se um varrimento de todos os píxeis e substitui-se o valor dos zeros por 0.01 e dos uns por 0.99, tendo um impacto mínimo no desempenho do algoritmo.

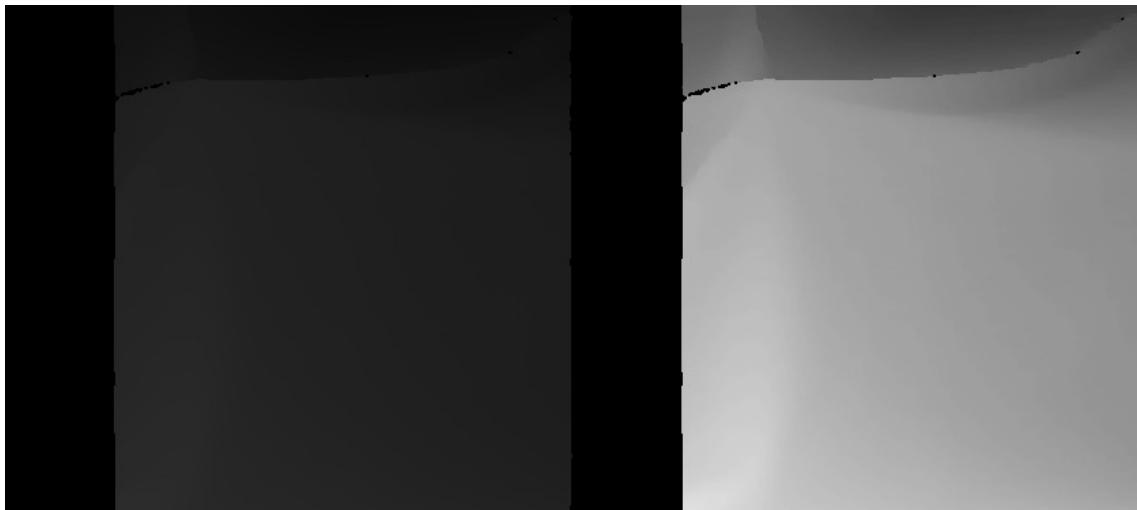


Figura 4.5: Mapa de disparidade antes (esquerda) e depois (direita) da normalização

4.3.2 Data Augmentation

Devido a algumas dificuldades encontradas na manipulação dos datasets e na leitura das imagens, foi necessário recorrer a data augmentation para aumentar o volume do material de treino da rede.

Data augmentation é uma técnica muito utilizada quando se pretende expandir o tamanho de um dataset de treino através da criação de versões modificadas das imagens presentes nesse mesmo dataset. Antes de serem apresentados diversos métodos para aplicar esta técnica, é de notar que, nesta situação concreta, qualquer alteração que seja aplicada numa imagem deve ser também aplicada à imagem correspondente na outra vista (alterações à imagem esquerda devem ser replicadas na imagem direita). Uma das possíveis formas de aplicar data augmentation é fazer translações nas imagens. Esta técnica pode ser aplicada em qualquer direção, mas, ainda assim, optou-se por não a utilizar devido a duas razões:

- Fazer translações verticais não teria qualquer alteração no treino desta rede devido à sua arquitetura, que será explicada um pouco mais à frente. De uma forma resumida, a rede processa imagens sem qualquer relação entre as diferentes linhas da imagem, por isso mover linhas de pixels para cima ou para baixo teria exatamente o mesmo resultado do ponto de vista da rede.
- Fazer translações horizontais produz um resultado pouco saudável nas imagens, que é a criação de colunas de pixels na imagem sem informação/vazios. Este tipo de situação introduz bastante ruído na rede.

Utilizar rotações para fazer data augmentation está fora de questão para este projeto porque estaria a afetar a relação entre as imagens da direita e da esquerda. Fazer flip às imagens também

0.14	0.2	0.3	0.4	0.5	0
0.53	0.6	0.6	0.5	0.7	0
0.8	0.4	0.9	0.7	0.10	0
0.2	0.3	0.12	0.6	0.6	0
0.2	0.3	0.15	0.8	0.8	0
0.6	0.7	0.18	0.9	0.11	0
0.9	0.9	0.21	0.10	0.7	0
0.3	0.4	0.24	0.11	0.9	0
0.3	0.4	0.27	0.12	0.12	0
0.7	0.8	0.30	0.13	0.8	0
0	0	0	0	0	0

Figura 4.6: Exemplo de zero-padding em imagens

não é uma boa abordagem, pois estar-se-ia a afetar a perspetiva dos obstáculos relativamente à câmara.

O método de data augmentation que foi selecionada é um que, para além de cumprir o objetivo geral desta técnica, ainda permite simular situações úteis para aprendizagem da rede. Mais concretamente, é aumentada ($x2$) e diminuída ($x0.5$) a intensidade dos píxeis de forma uniforme. Isto permite simular situações de maior ou menor luminosidade para o mesmo cenário. Na figura 4.7 é demonstrado um exemplo de data augmentation aplicado a uma imagem do dataset KITTI.

4.3.3 Parâmetros da Rede Neuronal

A rede tem tantos neurónios de entrada como o número de píxeis da largura da imagem, sendo também este o caso para a saída. Uma vez que o número de neurónios na camada de entrada está diretamente relacionado com a resolução da imagem, resta explorar o **número de neurónios na camada escondida** e o **número de epochs**.

Relativamente à camada escondida, foram testados vários valores para se determinar qual seria aquele que permitiria obter melhores resultados. No estudo apresentado em seguida, foram fixados todos os outros parâmetros da rede, variando apenas o número de neurónios da camada escondida. Como se pode observar, os resultados começam a estabilizar por volta de metade dos neurónios de entrada). É, então, feito um compromisso entre qualidade e velocidade de treino, optando-se por utilizar um valor igual ao número de neurónios de saída. A qualidade dos resultados é medida através da similaridade entre a saída da rede e o ground truth. Nas figuras 4.8 e 4.9 apresenta-se a evolução da qualidade dos resultados em função do número de neurónios. Adicionalmente, é dado algum contexto visual na figura 4.10.



Figura 4.7: Técnica de Data Augmentation aplicada ao dataset KITTI (original, intensidade aumentada e intensidade reduzida, de cima para baixo)

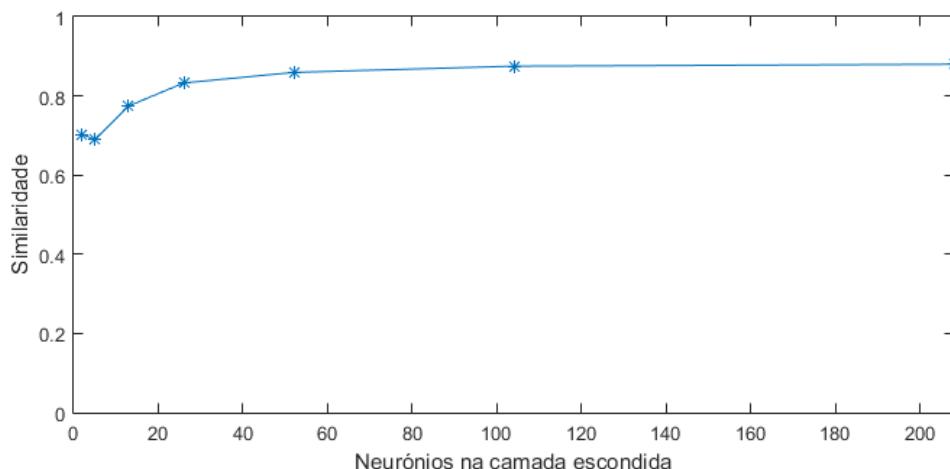


Figura 4.8: Relação entre o número de neurónios na camada escondida e a qualidade dos resultados com imagem do dataset Middlebury

O número de epochs para o treino da rede foi determinado utilizando uma abordagem semelhante: fixaram-se todos os outros parâmetros e variou-se este valor. Observa-se que, de uma

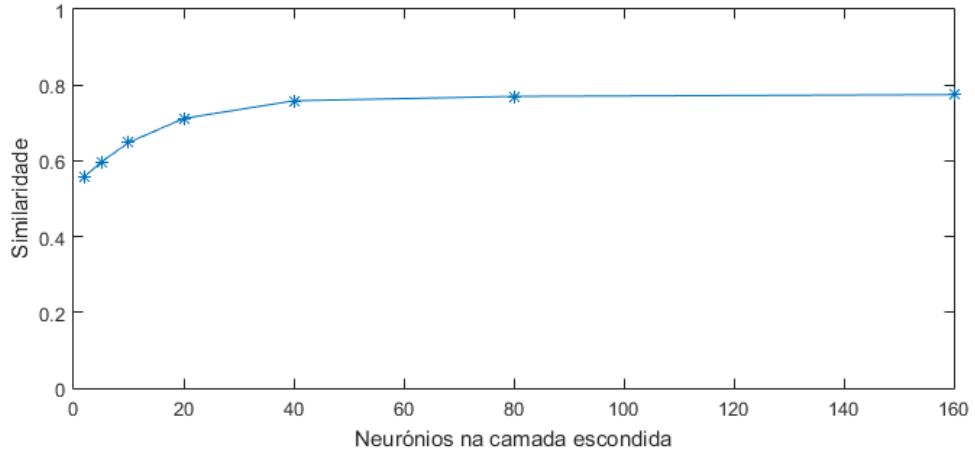


Figura 4.9: Relação entre o número de neurónios na camada escondida e a qualidade dos resultados com imagem das câmaras Astra

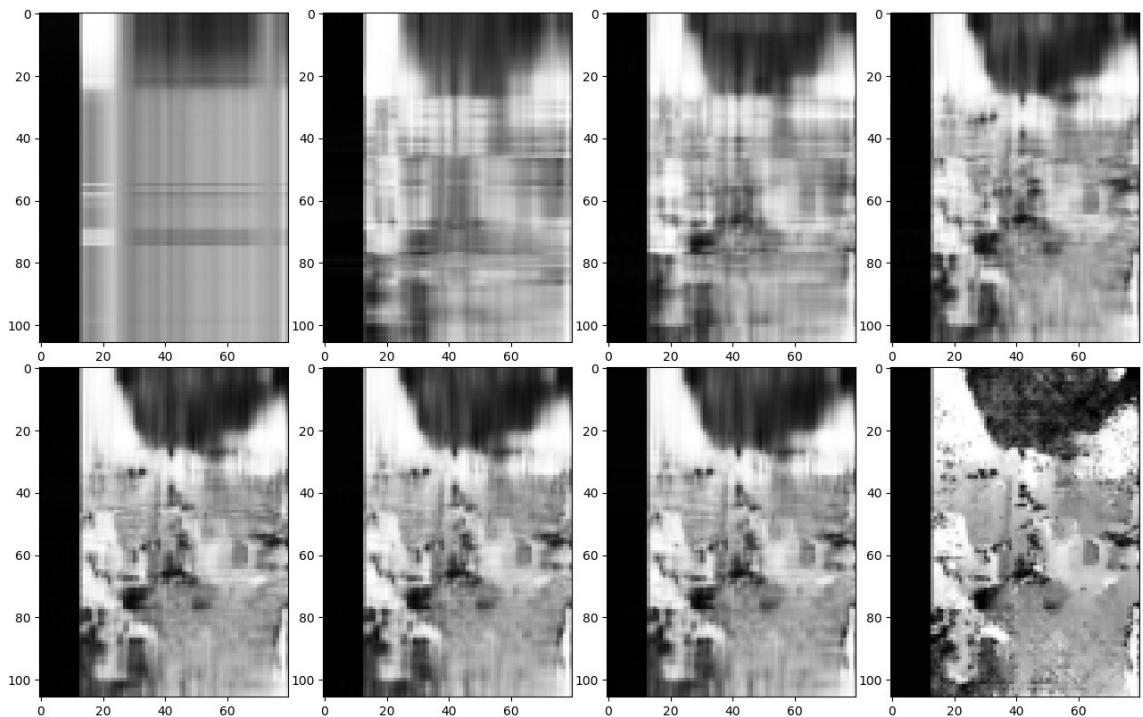


Figura 4.10: Comparação entre 2, 5, 10, 20 (cima), 40, 80 e 160 neurónios na camada escondida com o ground truth (baixo) com uma imagem obtida através das câmaras Astra

maneira geral, para se obterem os melhores resultados, deve-se aumentar o número de iterações. No entanto, não foram testados valores superiores a 10000 porque já este valor implica um tempo de treino muito longo, chegando a durar vários dias se se utilizarem demasiadas imagens. Esta evolução está representada nas figuras 4.11 e 4.12. Também é possível acompanhar esta evolução através das imagens na figura 4.13.

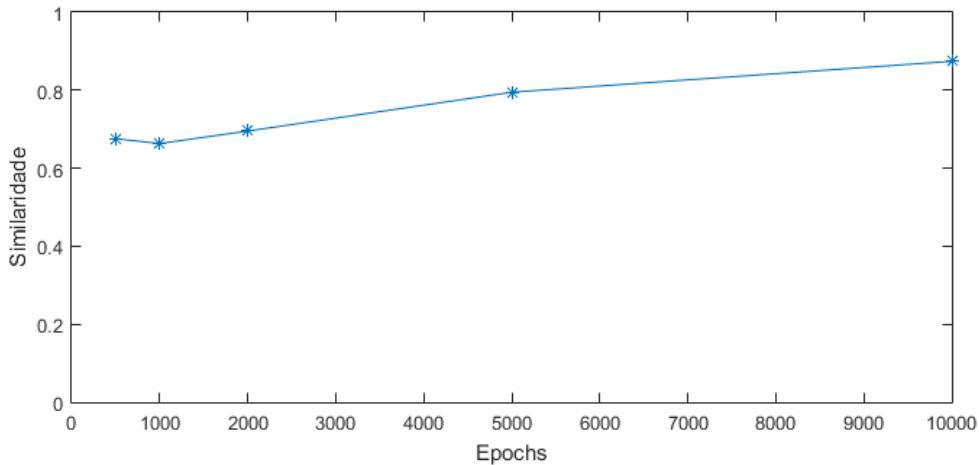


Figura 4.11: Relação entre o número de epochs e a qualidade dos resultados com imagem do dataset Middlebury

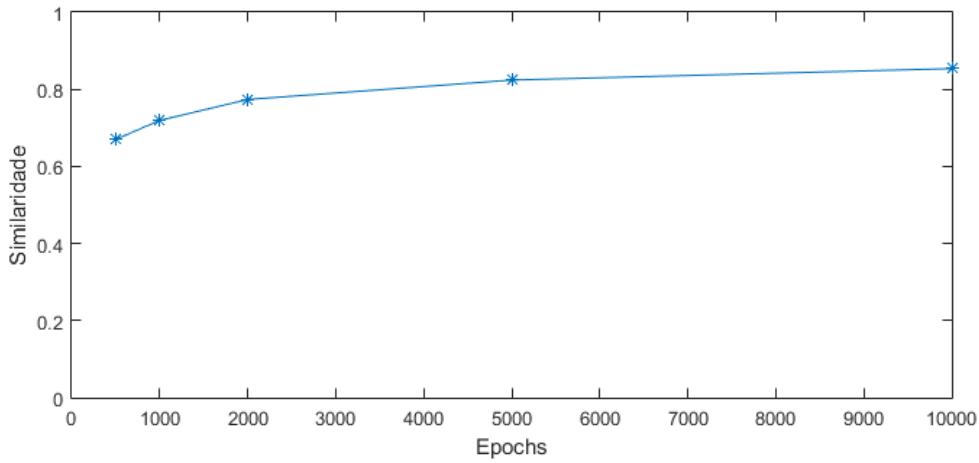


Figura 4.12: Relação entre o número de epochs e a qualidade dos resultados com imagem das câmeras Astra

A função de custo utilizada para fazer backpropagation é uma simples soma de quadrados para calcular o erro. É utilizado o método do gradiente para encontrar o mínimo desta função.

4.4 Conclusão

Tanto num método como no outro, tiveram de ser feitas várias escolhas com as suas implicações, nomeadamente balancear a qualidade dos resultados com a velocidade de processamento (no caso do método de Semi-Global Matching) ou com a velocidade de treino da rede neuronal. Dependendo da aplicação e do tempo de preparação, é possível que diferentes parâmetros para ambas as abordagens obtenham resultados mais vantajosos.

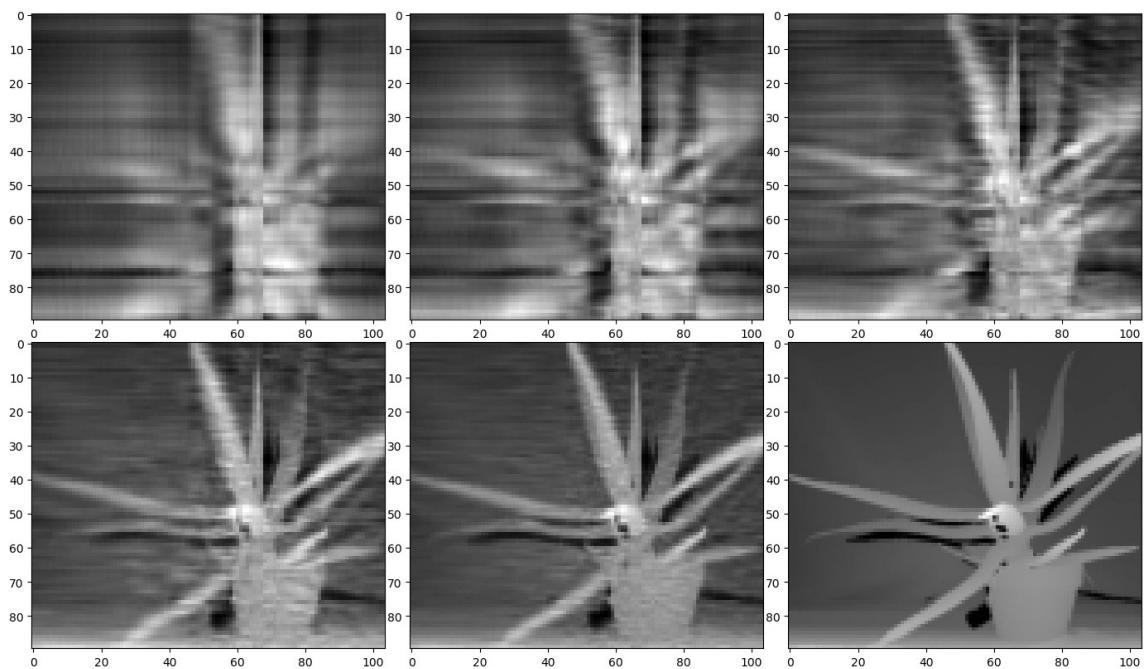


Figura 4.13: Comparação entre 500, 1000, 2000 (cima), 5000 e 10000 epochs com o ground truth (baixo) com uma imagem do dataset Middlebury

Mais uma vez, realça-se o requisito de funcionamento em tempo real que está associado a este sistema.

Capítulo 5

Caracterização de Desempenho com Sequência de Imagens Díspares

Neste capítulo são apresentados os resultados obtidos pelos dois métodos com a utilização de datasets. São utilizados os parâmetros de SGBM e da NN identificados no capítulo anterior. O objetivo deste capítulo é a caracterização do desempenho de ambos os métodos com uma sequência de imagens extremamente díspares entre si. A avaliação será feita, tanto para uma abordagem como para a outra, utilizando o dataset de Middlebury, pois é o dataset com ground truth mais compatível com os resultados produzidos neste projeto.

5.1 Métrica de avaliação

A métrica selecionada para avaliar os resultados dos métodos utilizados é o índice de similaridade estrutural. Esta técnica recorre à computação da luminância, do contraste e da estrutura para determinar a percentagem de similaridade entre duas imagens distintas [38]. Este método de avaliação está implementado na ferramenta do Matlab, tendo, portanto, sido utilizado através desta ferramenta.

5.2 Semi-Global Block-Matching

É relevante referir que aqui está a ser criado um mapa de disparidade, não um mapa de profundidade. Isto significa que algoritmos diferentes podem atribuir intensidades diferentes ao mesmo elemento da imagem, o que irá prejudicar os resultados quantitativos quando comparados ao ground truth.

Também se verificou uma situação que terá de ser retificada no futuro, que é o corte dos primeiros 80 píxeis de cada coluna das imagens. Também isto terá, naturalmente, uma influência negativa nos resultados.

Na figura 5.1 demonstram-se as alterações que foram feitas para a obtenção dos valores representados na tabela, utilizando como exemplo a primeira imagem do dataset em questão. Primeiro

foi cortada a porção da imagem que não é processada, para não influenciar os resultados da parte que importa. Posteriormente, é feita uma diminuição na intensidade (multiplica-se a intensidade de cada píxel por um fator inferior a 1, de forma uniforme) para a aproximar mais do ground truth. Os resultados estão apresentados na tabela 5.1. É possível visualizar resultados com o dataset KITTI neste vídeo: <https://youtu.be/GZ4cboP4NfY>.

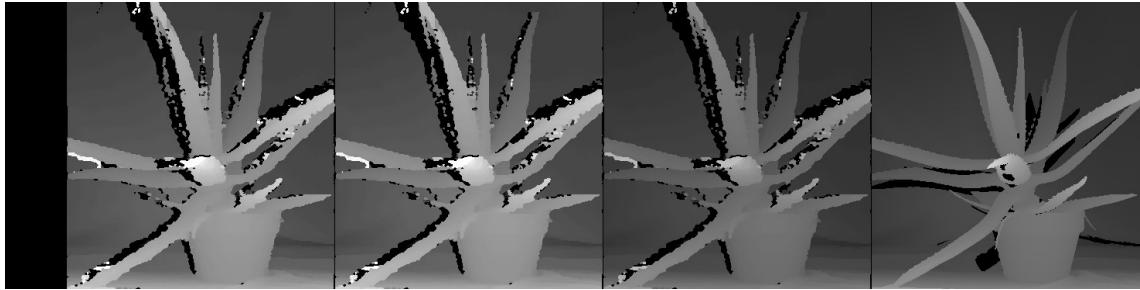


Figura 5.1: Aloe, Aloe Cortado, Aloe Cortado e Filtrado e Ground Truth Cortado, respetivamente

Tabela 5.1: Resultados dos métodos tradicionais

Imagen	Similaridade	Sim. Cortada	Sim. Cort. Filtrada
Aloe	0.5438	0.6742	0.7161
Baby1	0.6693	0.8364	0.8783
Baby2	0.6720	0.8354	0.8794
Baby3	0.6069	0.7490	0.7845
Bowling1	0.6462	0.8020	0.8419
Bowling2	0.6535	0.7939	0.8331
Cloth1	0.7455	0.9272	0.9685
Cloth2	0.6706	0.8260	0.8610
Cloth3	0.7213	0.8955	0.9378
Cloth4	0.6644	0.8197	0.8608
Média	0.6593	0.8159	0.8561

5.3 Rede Neuronal de Disparidade Direta

O método de avaliação utilizado foi, em semelhança ao dos métodos tradicionais, a similaridade entre imagens. A rede foi treinada com as imagens do dataset Middlebury utilizadas para a avaliação, com 10000 epochs. De seguida, foram alimentadas de novo as imagens à rede sem ser feita backpropagation e são esses os resultados que são utilizados para a avaliação. Para o treino desta rede, foram utilizados 104 neurónios na camada escondida (tantos como na camada de saída), tal como havia sido definido anteriormente.

Como se verifica um ruído considerável, é utilizado o filtro de mediana para melhorar um pouco os resultados. Adicionalmente, de forma a mitigar um pouco a resolução reduzida a que a rede processa imagens de forma a reduzir o tempo de treino, é feito um downsampling do ground

truth antes de ser feita a comparação com o resultado da rede, até porque o treino da rede é feito efetivamente com uma versão sub-amostrada do ground truth. Estas alterações podem ser visualizadas na figura 5.2. Os resultados obtidos com 104 neurónios na camada escondida estão apresentados na tabela 5.2. Também foram obtidos resultados com 208 neurónios para termo de comparação, apresentados na tabela 5.3.



Figura 5.2: Aloe, Aloe com filtro de mediana e GT sub-amostrado

Tabela 5.2: Resultados da rede neuronal com 104 neurónios na camada escondida

Imagen	Similaridade	Sim. Filtrada	Sim. Filt. Downsampled GT
Aloe	0.5006	0.7255	0.8184
Baby1	0.5116	0.7597	0.7927
Baby2	0.6130	0.8054	0.8388
Baby3	0.5540	0.7404	0.8066
Bowling1	0.5441	0.6762	0.7050
Bowling2	0.6256	0.7908	0.8449
Cloth1	0.4864	0.8486	0.8527
Cloth2	0.6350	0.8501	0.8954
Cloth3	0.5360	0.8214	0.8585
Cloth4	0.6432	0.8630	0.9009
Média	0.5655	0.7881	0.8314

5.4 Conclusão

Esta avaliação foi feita para validar a qualidade do método de Block-Matching e da rede neuronal. Se uma destas duas abordagens tivesse obtido resultados muito fracos, não seria possível avançar para o próximo passo deste projeto: se o método de block-matching não estivesse funcional, não seria possível criar um ground truth para treinar a rede com imagens obtidas com setup experimental, enquanto que se a rede neuronal não estivesse funcional utilizando datasets, muito provavelmente estaria ainda menos quando treinada com imagens que já contêm ruído proveniente do algoritmo com que foram criadas. Também é possível confirmar que a rede neuronal de disparidade direta tem a capacidade de aprender várias imagens com características bastante distintas.

Tabela 5.3: Resultados da rede neuronal com 208 neurónios na camada escondida

Imagen	Similaridade	Sim. Filtrada	Sim. Filt. Downsampled GT
Aloe	0.5379	0.7605	0.8559
Baby1	0.5108	0.7602	0.7909
Baby2	0.6070	0.8111	0.8464
Baby3	0.5805	0.7486	0.8176
Bowling1	0.5914	0.7141	0.7451
Bowling2	0.6428	0.7976	0.8470
Cloth1	0.5431	0.8530	0.8573
Cloth2	0.6649	0.8688	0.9145
Cloth3	0.6101	0.8733	0.9105
Cloth4	0.6940	0.8917	0.9278
Média	0.5982	0.8079	0.8513

Capítulo 6

Implementação em Tempo Real no Setup de Vídeo Estereoscópico

Neste capítulo é descrita a integração com o sistema físico da Follow Inspiration e são apresentados alguns resultados obtidos.

6.1 Setup e Captura de Vídeo Estereoscópico

No tempo disponível para desenvolvimento deste projeto, não foi possível fazer integração completa com uma plataforma robótica da Follow Inspiration. Para contornar esta situação, foram seguidas duas abordagens diferentes para poder ser feita a obtenção de imagem e aplicar as técnicas de estimativa estereoscópica. Algo que é comum a ambas é a forma como as câmaras estão dispostas e fixas (apresentada na figura 6.1).

Após serem recolhidas as imagens, é obtido um ground truth utilizando o algoritmo de Block-Matching usado neste projeto para ser feito o treino da rede neuronal de disparidade. Este processo pode ser visualizado na figura 6.2.

A aquisição de imagem foi feita de duas maneiras. Na situação de o robô estar à beira do computador onde foi programado todo o projeto, é possível ligar as câmaras a esse mesmo computador (mantendo as câmaras fixas no robô) e obter a imagem processada em tempo real recorrendo à ferramenta OpenNI para fazer a leitura de imagem das câmaras Astra. Claro está, esta abordagem tem a limitação de apenas ser possível obter imagens a partir de uma posição praticamente fixa, permitindo ao robô movimentar-se menos de 1 metro.

A segunda abordagem consiste na ligação das duas câmaras a um wGO Retail (foi escolhido este modelo de robô porque já está preparado para ter a ligação de mais de uma câmara) e posterior gravação da imagem RGB proveniente dessas mesmas duas câmaras. Este setup apenas permite capturar vídeo, sendo todo o processamento, quer por SGBM quer por NN, feito posteriormente no computador onde foi desenvolvido todo este projeto. Pode ver-se a montagem experimental na plataforma robótica da Follow Inspiration na figura 6.3. Esta gravação foi feita utilizando scripts



Figura 6.1: Disposição das câmaras

disponibilizados pela Follow Inspiration que utilizam a ferramenta ROS para criar datasets de imagem. Naturalmente, este método tem a desvantagem de não ser feito o processamento em tempo real, tornando também difícil perceber se há necessidade de ajustar as câmaras. Há ainda outro problema associado, que é a perda de frames por parte de uma das câmaras. Não foi apurada a sua origem, mas resulta na não correspondência direta entre os frames provenientes dos dois feeds, dificultando a sua associação para ser feita a estimativa estereoscópica. Esta correspondência teve de ser feita manualmente.

6.2 Esforço de Integração

As câmaras Astra estão ligadas por USB e é necessário definir o ID que lhes é associado cada vez que se liga/desliga essa câmara ou cada vez que se inicia o computador ao qual estão ligadas.

Pequenas variações na posição de uma câmara relativamente à outra têm consequências drásticas, pelo que teve de ser encontrada empiricamente a posição ideal e nela fixar as câmaras o melhor possível para os resultados não se deteriorarem demasiado. Ainda assim, associado a esta montagem rudimentar vem bastante ruído introduzido no algoritmo de Block-Matching e, por consequência, na rede neuronal de disparidade.

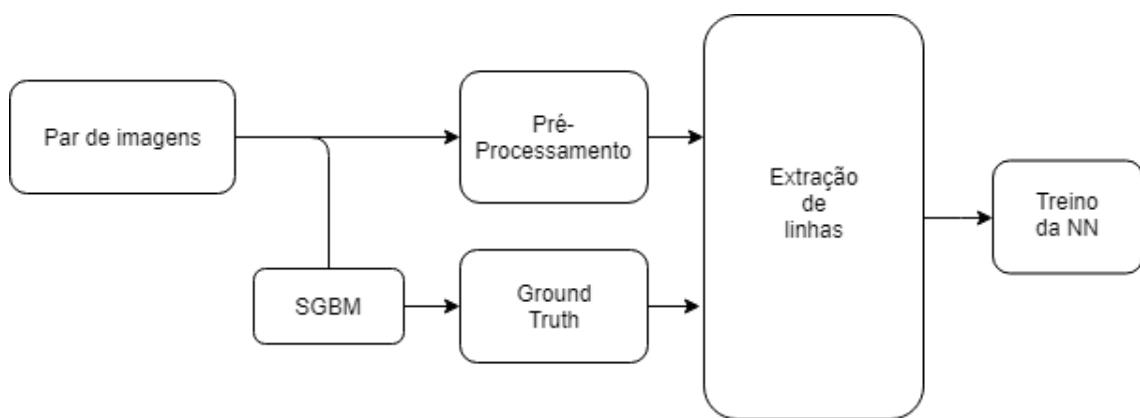


Figura 6.2: Diagrama de obtenção de GT para treinar a rede neuronal



Figura 6.3: Integração experimental com o wGO Retail

Antes de executar, é necessário garantir que as imagens estão no formato apropriado e com as dimensões corretas. Neste caso, o tipo de dados utilizado é CV_8UC3, o que significa que as imagens têm 3 canais de cor, sendo que cada um tem uma profundidade de cor de 8 bits do tipo inteiro unsigned.

Adicionalmente, devido à forma como as câmeras estão dispostas, é necessário fazer uma rotação de 90 graus a cada frame, para a direita ou para a esquerda dependendo da câmera. As

câmaras estão dispostas desta forma com o objetivos de facilitar a fixação das mesmas uma à outra de modo a que estas estejam sempre no mesmo plano e à mesma altura e de reduzir a distância entre as duas lentes. Esta estratégia melhorou exponencialmente a qualidade dos resultados obtidos.

6.3 Resultados

Utilizando o setup anteriormente descrito, foram gravados milhares de frames, dos quais foram selecionados cerca de 120 (123, mais especificamente) para testar a rede neuronal. O número de frames foi escolhido com base no tempo de treino e no tempo disponível para obter resultados. Destes 123, 85 foram utilizados para treinar a rede e 38 para a testar. O treino foi feito com 10000 epochs e 80 neurónios na camada escondida (tantos neurónios como na camada de saída). A avaliação foi, mais uma vez, feita utilizando o índice de similaridade. Todos os resultados apresentados neste capítulo são obtidos diretamente da saída da rede (não é feito pós-processamento). Estes resultados podem ser observados nas figuras 6.4 e 6.5. Adicionalmente, são apresentados alguns resultados visuais para imagens de treino nas figuras 6.6 e 6.7, assim como os resultados para uma imagem de treino na figura 6.8. É possível visualizar alguns resultados neste vídeo: <https://youtu.be/8IKd2PBHNQU>.

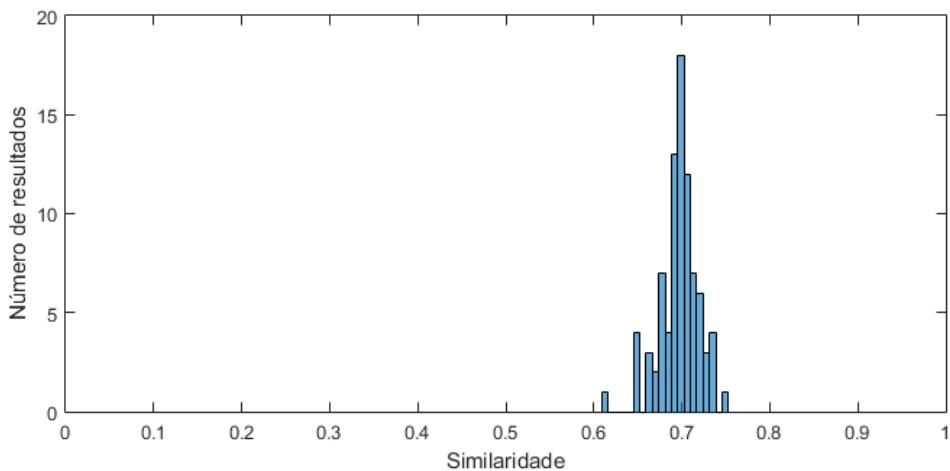


Figura 6.4: Histograma com os resultados dos frames de treino

Usando o dataset de treino, observa-se uma distribuição de valores de similaridade concentrada em 0.7. Nos resultados do dataset de teste observam-se dois grupos de resultados, correspondentes a duas diferentes cenas presentes no dataset. Numa das cenas, o dataset de teste é bastante semelhante ao dataset de treino (mas, ainda assim, diferente) e, por isso, obtém-se resultados também a rondar o índice de similaridade de 0.7. O outro grupo corresponde a uma cena com elementos diferentes dos elementos presentes no dataset de treino, obtendo-se resultados com um índice de similaridade mais baixo perto de 0.6.

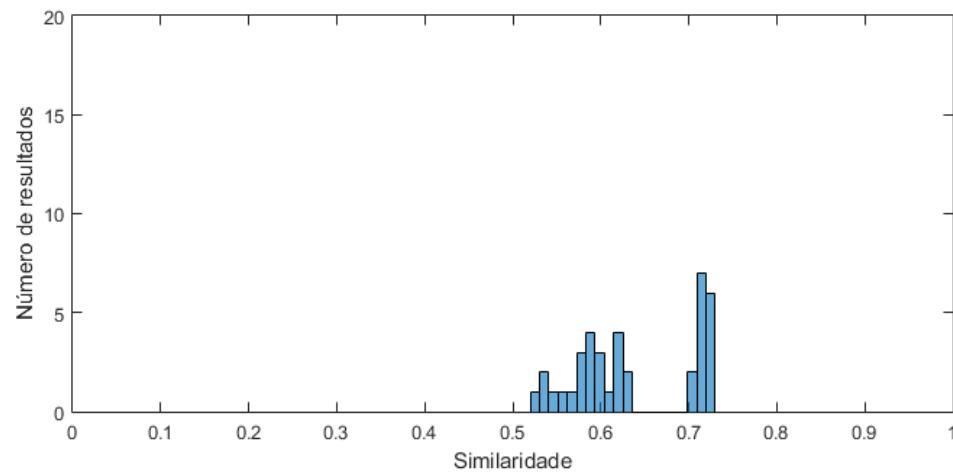


Figura 6.5: Histograma com os resultados dos frames de teste

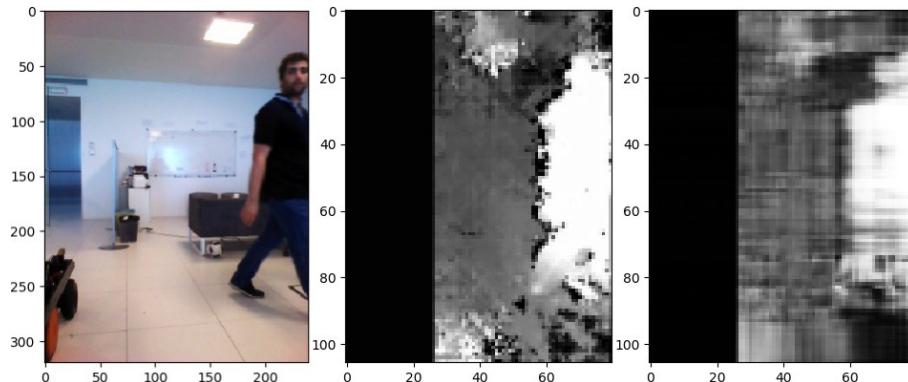


Figura 6.6: Imagem RGB, Imagem de disparidade obtida através de SGBM e Resultado da Rede para uma imagem de treino (1)

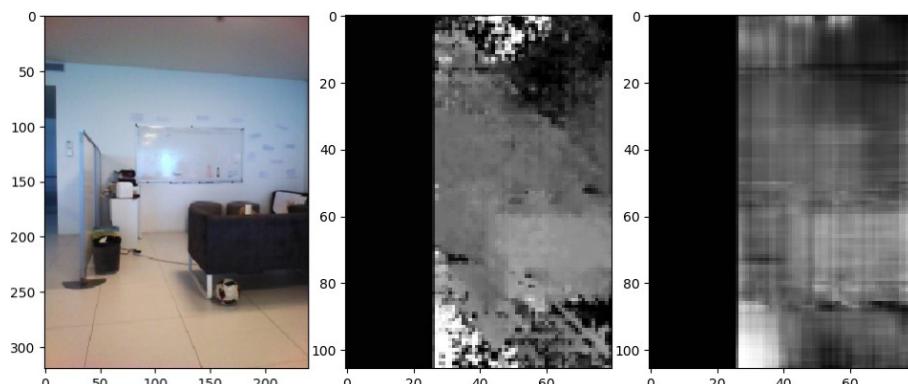


Figura 6.7: Imagem RGB, Imagem de disparidade obtida através de SGBM e Resultado da Rede para uma imagem de treino (2)

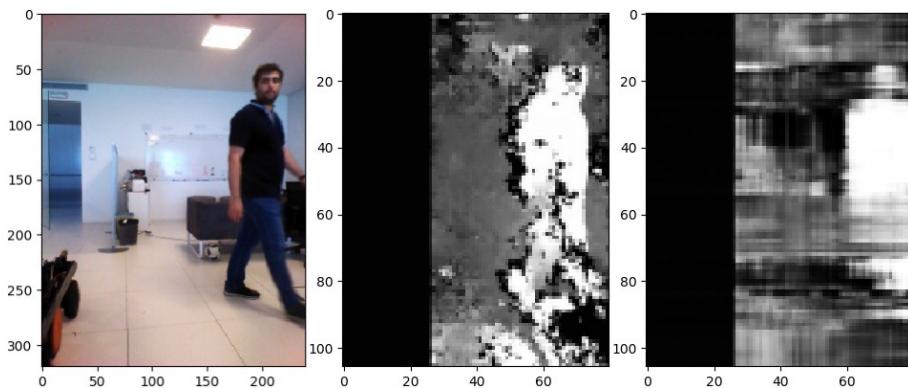


Figura 6.8: Imagem RGB, Imagem de disparidade obtida através de SGBM e Resultado da Rede para uma imagem de teste

6.4 Conclusão

Aceitando como válido o ground truth criado utilizando o algoritmo de block-matching, foi possível analisar o comportamento da rede neuronal de disparidade em ambiente real. Ainda assim, é necessário realçar a dificuldade de manter as câmaras numa posição que permitisse obter resultados minimamente bons. Esta dificuldade torna-se ainda mais significativa quando o robô se encontra em movimento, nomeadamente nos arranques e nas travagens onde se dá uma variação brusca de velocidade.

Relativamente aos resultados da rede neuronal, vale a pena notar que a qualidade dos resultados obtidos com as imagens de teste não são muito piores do que os resultados das imagens de treino. Como seria de esperar, os resultados das imagens de teste são tão bons quanto mais parecida a imagem for com imagens de treino. Daqui se pode concluir que se houvesse a possibilidade de treinar a rede com um volume muito maior de imagens, os resultados das imagens de teste aproximariam-se ainda mais das das imagens de treino.

Capítulo 7

Conclusões e Trabalho Futuro

O último capítulo é dedicado à explicitação das conclusões do projeto, assim como à definição de próximos passos a realizar no desenvolvimento do mesmo.

7.1 Satisfação dos Objectivos

Foi implementada com sucesso uma rede neuronal capaz criar mapas de disparidade, tendo como termo de comparação uma abordagem que utiliza block-matching. Como é possível ver pelas tabelas de resultados, a rede neuronal consegue aproximar-se da qualidade do método tradicional utilizado, apesar de ser um sucesso condicional: de um ponto de vista prático, a implementação do semi-global matching é melhor, pois a rede neuronal irá sempre necessitar de outro método que produza um ground truth para ser treinada, nunca será verdadeiramente independente. Também se verifica que o tempo de treino da rede é muito longo, demorando vários dias para treinar em algumas situações. Esta situação poderia ser mitigada reduzindo o número de neurónios na rede, afetando o seu desempenho, ou também utilizando GPU para acelerar este processo.

7.2 Principais Dificuldades

Foram utilizadas muitas ferramentas neste projeto, como já foi apresentado no início deste documento. Essas ferramentas têm o propósito de ajudar no desenvolvimento de projetos, mas também exigem algum tempo de estudo e de treino até haver alguma familiarização com o seu potencial. A utilização de todas estas ferramentas teve, portanto, o seu custo no tempo disponível para a realização do trabalho em si.

Outra dificuldade encontrada nesta dissertação foi a integração em ambiente empresarial, sem qualquer crítica dirigida à Follow Inspiration, até porque todos os seus colaboradores foram extremamente simpáticos. É um ambiente diferente de tudo aquilo que já havia experienciado e exigiu algum tempo de adaptação aos métodos de trabalho.

Outra dificuldade encontrada foi o trabalho com redes neuronais. Até à data, o contacto com este tipo de sistema tinha sido mínimo, de modo que foi um pouco complicada a aprendizagem e posterior implementação num intervalo de tempo tão curto.

7.3 Trabalho Futuro

7.3.1 Semi-Global Block-Matching

Aqui são apresentadas algumas sugestões de trabalho a realizar como continuação deste projeto para o algoritmo implementado com recurso à biblioteca OpenCV.

7.3.1.1 Disparidade para Profundidade

O algoritmo implementado retorna um mapa de disparidade. No entanto, para a sua informação ser útil para a visão computacional de um robô real, é necessário que este mapa de disparidade seja convertido num mapa de profundidade. Naturalmente, esta conversão depende das câmaras utilizadas e da sua disposição, sendo um processo sujeito a calibração.

7.3.1.2 Regras das câmaras

Atualmente, é necessário definir manualmente o ID das câmaras Astra de cada vez que elas são ligadas/desligadas ao computador e de cada vez que o computador é reiniciado. Uma forma de contornar isto seria automatizar o processo de deteção das câmaras quando elas são ligadas à porta USB, criar um script que permite ajudar o utilizador neste processo ou até mesmo criar uma interface.

7.3.1.3 Pós-Processamento

Como foi possível observar, pós-processamento pode ter uma influência considerável nos resultados, pelo que a investigação de técnicas apropriadas a este projeto pode ser muito benéfica. Este trabalho também pode passar pela configuração de alguns parâmetros que não foram utilizados.

7.3.2 Rede Neuronal de Disparidade Direta

Nesta secção são apresentados possíveis melhoramentos à estimação stereo com recurso a redes neuronais.

7.3.2.1 Pré-Processamento

Foi possível verificar que a rede consegue criar um mapa de disparidade razoável quando recebe imagens com luminosidade muito diferente do que aquela para que foi treinada. No entanto, esta alteração na imagem traz na mesma algum ruído associado, pelo que talvez fosse benéfico para

a rede filtrar a luminosidade média de todas as imagens que lhe são fornecidas e, posteriormente em ambiente real, filtrar também a imagem das câmaras antes de as processar através da rede.

7.3.2.2 Diferente Arquitetura

Na revisão bibliográfica foi possível observar que a fusão de redes neurais com outras técnicas pode ter resultados muito bons. É uma abordagem que vale a pena investigar com detalhe. Certamente seria interessante também experimentar o uso de redes convolucionais e comparar com os resultados obtidos neste projeto.

7.3.2.3 Utilizar RGB

A rede faz todo o seu treino e processamento utilizando apenas imagens em escalas de cinza. Há aí uma percentagem de informação que se perde, pelo que seria certamente benéfico criar uma implementação que utilizasse todos os 3 canais de cor, seja a mesma rede a receber e processar simultaneamente esses 3 canais, sejam 3 redes distintas a processar cada uma um canal de cor e haver posterior fusão da informação.

Referências

- [1] Imagens Tsukuba. URL: <http://vision.middlebury.edu/stereo/data/scenes2001/>.
- [2] Multilayer Neural Network. URL: https://commons.wikimedia.org/wiki/File:Multilayer_Neural_Network.png.
- [3] Exemplo de backpropagation. URL: <https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/>.
- [4] Tom Bonkenburg. URL: http://www.dhl.com/content/dam/downloads/g0/about_us/logistics_insights/dhl_trendreport_robotics.pdf.
- [5] Wheelchair Foundation. URL: <https://www.wheelchairfoundation.org/programs/from-the-heart-schools-program/materials-and-supplies/analysis-of-wheelchair-need/>.
- [6] Joana Santos, Daniel Campos, Fábio Duarte, Filipe Pereira, Inês Domingues, Joana Santos, João Leão, José Xavier, Luís de Matos, Manuel Camarneiro, Marcelo Penas, Maria Miranda, Ricardo Morais, Ricardo Silva, e Tiago Esteves. A personal robot as an improvement to the customers' in-store experience. Em Antonio J. R. Neves, editor, *Service Robots*, chapter 1. IntechOpen, Rijeka, 2018. URL: <https://doi.org/10.5772/intechopen.70277>, doi:10.5772/intechopen.70277.
- [7] R. A. Hamzah e K. A. A. Aziz. Region of interest in disparity mapping for distance estimation on stereo vision application. Em *Fourth International Conference on Digital Image Processing (ICDIP 2012)*, volume 8334 de *\prospie*, página 83340Q, 2012. doi:10.1111/12.946057.
- [8] L. Cui. Research on english translation distortion detection based on image evolution. *Eurasip Journal on Image and Video Processing*, 2019(1), 2019. URL: www.scopus.com.
- [9] OpenCV. URL: <https://opencv.org/>.
- [10] C Lü, X Wang, e Y Shen. A stereo vision measurement system Based on OpenCV. Em *2013 6th International Congress on Image and Signal Processing (CISP)*, volume 2, páginas 718–722, 2013. doi:10.1109/CISP.2013.6745259.
- [11] Chris Harris e Mike Stephens. A combined corner and edge detector. Em *Alvey vision conference*, volume 15, páginas 10–5244, 1988.
- [12] Kurt Konolige. Small vision systems: Hardware and implementation. *Proceedings of Eighth International Symposium Robotics Research*, 8, 07 1998. doi:10.1007/978-1-4471-1580-9_19.

- [13] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, Feb 2008. doi:10.1109/TPAMI.2007.1166.
- [14] S. Haykin. *Neural Networks. A Comprehensive Foundation*. 1994.
- [15] Eiji Mizutani Jyh-Shing Roger Jang, CXhuen-Tsai Sun. *Neuro-fuzzy and soft computing*. 1995.
- [16] D.L. Elliot O. Omidvar. *Neural Systems for Control*. 1997.
- [17] X. Zeng, Y. Li, Z. Chen, e L. Zhu. A hybrid 2d and 3d convolution neural network for stereo matching. Em *Proceedings - 21st IEEE International Conference on Computational Science and Engineering, CSE 2018*, páginas 152–156, 2018. URL: www.scopus.com.
- [18] M. Malița, O. Nedescu, A. Negoiță, e G. M. Ștefan. Deep learning in low-power stereo vision accelerator for automotive. Em *2018 IEEE International Conference on Consumer Electronics (ICCE)*, páginas 1–6, Jan 2018. doi:10.1109/ICCE.2018.8326285.
- [19] Andrea Zanella e Sergio Taraglio. Sensing the third dimension in stereo vision systems: a cellular neural networks approach. *Engineering Applications of Artificial Intelligence*, 11(2):203–213, 1998. doi:[https://doi.org/10.1016/S0952-1976\(97\)00076-6](https://doi.org/10.1016/S0952-1976(97)00076-6).
- [20] A Zanella e S Taraglio. A cellular neural network stereo vision system for autonomous robot navigation. Em *Proceedings of the 2000 6th IEEE International Workshop on Cellular Neural Networks and their Applications (CNNA 2000) (Cat. No.00TH8509)*, páginas 117–122, 2000. doi:10.1109/CNNA.2000.876831.
- [21] Jung-Hua Wang e Chih-Ping Hsiao. On Disparity Matching in Stereo Vision via a Neural Network Framework. 23, 2008.
- [22] Y Chou, D Lee, D Zhang, e K Hill. A parallel convolutional neural network architecture for stereo vision estimation. Em *2017 IEEE International Conference on Image Processing (ICIP)*, páginas 2508–2512, 2017. doi:10.1109/ICIP.2017.8296734.
- [23] Zhi Wang, Shiqiang Zhu, Yuehua Li, e Zhengzhe Cui. Convolutional neural network based deep conditional random fields for stereo matching. *Journal of Visual Communication and Image Representation*, 40:739–750, 2016. doi:<https://doi.org/10.1016/j.jvcir.2016.08.022>.
- [24] KIT. URL: <http://www.cvlabs.net/datasets/kitti/>.
- [25] Andreas Geiger, Philip Lenz, Christoph Stiller, e Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.
- [26] Moritz Menze e Andreas Geiger. Object scene flow for autonomous vehicles. Em *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [27] Andreas Geiger, Philip Lenz, e Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. Em *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [28] Jannik Fritsch, Tobias Kuehnl, e Andreas Geiger. A new performance measure and evaluation benchmark for road detection algorithms. Em *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.

- [29] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, e Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. volume 8753, páginas 31–42, 09 2014. doi:10.1007/978-3-319-11752-2_3.
- [30] Heiko Hirschmüller e Daniel Scharstein. Evaluation of cost functions for stereo matching. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, páginas 1–8, 2007.
- [31] D. Scharstein e C. Pal. Learning conditional random fields for stereo. Em *2007 IEEE Conference on Computer Vision and Pattern Recognition*, páginas 1–8, June 2007. doi:10.1109/CVPR.2007.383191.
- [32] Daniel Scharstein e Richard Szeliski. High-accuracy stereo depth maps using structured light. Em *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR’03, páginas 195–202, Washington, DC, USA, 2003. IEEE Computer Society. URL: <http://dl.acm.org/citation.cfm?id=1965841.1965865>.
- [33] D. Scharstein, R. Szeliski, e R. Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Em *Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)*, páginas 131–140, Dec 2001. doi:10.1109/SMBV.2001.988771.
- [34] MathWorks. URL: <https://www.mathworks.com/products/matlab.html>.
- [35] Orbbec. URL: <https://orbbec3d.com/product-astra-pro/>.
- [36] Occipital. URL: <https://structure.io/openni>.
- [37] Stan Birchfield e Carlo Tomasi. Depth discontinuities by pixel-to-pixel stereo. *International Journal of Computer Vision*, 35(3):269–293, 1999.
- [38] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.