UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Vinicius Rodrigues Camargo

Uso de Visão Computacional Estéreo para Estimativa da Distância em Veículos Autônomos

Porto Alegre

2021

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Vinicius Rodrigues Camargo

Uso de Visão Computacional Estéreo para Estimativa da Distância em Veículos Autônomos

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como requisito parcial para Graduação em Engenharia Elétrica.

Orientador: Prof. Dr. Ronaldo Husemann

Porto Alegre
2021

CIP - Catalogação na Publicação

Camargo, Vinicius Rodrigues
Uso de visão computacional estéreo para estimativa da distância em veículos autônomos / Vinicius Rodrigues Camargo. -- 2021.
48 f.
Orientador: Ronaldo Husemann.

Trabalho de conclusão de curso (Graduação) -- Universidade Federal do Rio Grande do Sul, Escola de Engenharia, Curso de Engenharia Elétrica, Porto Alegre, BR-RS, 2021.

1. visão computacional. 2. visão computacional estéreo. 3. mapa de disparidade. 4. veículos autônomos. I. Husemann, Ronaldo, orient. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da UFRGS com os dados fornecidos pelo(a) autor(a).

VINICIUS RODRIGUES CAMARGO

Uso de Visão Computacional Estéreo para Estimativa da Distância em Veículos Autônomos

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como requisito parcial para Graduação em Engenharia Elétrica.

Prof. Dr. Ronaldo Husemann
Orientador - UFRGS

Prof. Dr. Roberto Petry Homrich
Chefe do Departamento de Engenharia Elétrica (DELET) - UFRGS

Aprovado em ___ de ___ de ___.

BANCA EXAMINADORA

Prof. Dr. Ronaldo Husemann
UFRGS

Prof. Dr. Altamiro Amadeu Suzin
UFRGS

Prof. Dr. Raphael Martins Brum
UFRGS

Agradecimentos

À minha avó e minhas tias, que me incentivaram e proporcionaram a melhor educação que lhes tinha alcance apesar de todos os desafios, e compreenderam a minha ausência enquanto eu me dedicava à vida acadêmica.

Aos meus sogros, que se tornaram parte da minha família de maneira tão rápida, e que nunca deixaram de ficar felizes pelas minhas conquistas, por menores que fossem.

Aos amigos peludos, Skyp, Pascoal, Lucky, Lupin e Ginger, fontes ambulantes de alegria imediata.

Ao meu amigo Giancarlo, companheiro na luta árdua que foi a graduação, estando sempre presente nas vitórias mas também nas derrotas (e não só do *Towerfall*), uma amizade que com certeza perdurará para outras batalhas.

Ao meu amigo Bruno, pela amizade incondicional e pelo apoio demonstrado ao longo de todos esses anos, tanto nos inúmeros almoços no RU quanto nas jantas aleatórias, sempre sendo uma companhia alegre e necessária.

À Patrulha, que desde a 5^a série viraram pessoas indispensáveis na minha vida, sendo desde o ponto de escape no recreio da escola até à parceria nas jantas, festas e Rock n Biras da faculdade (até mesmo quando o sono bate em cima da caixa de som).

Ao meu orientador, Professor Doutor Ronaldo Husemann, por ser um ponto de inspiração durante a trajetória no curso e pela grande atenção que se tornou essencial para que o projeto fosse concluído.

À Universidade Federal do Rio Grande do Sul e a todos os professores da Engenharia pela elevada qualidade do ensino oferecido.

E principalmente, à minha namorada, amiga e companheira, Bru, que sempre esteve ao meu lado nos bons e maus momentos com seu amor incondicional, por compreender o mau humor nos finais de semestre, sempre estando lá para ouvir minhas lamentações e ajudar a resolver tudo indo em um café na *CB*.

Somos apenas uma espécie avançada de macacos em um planeta menor de uma estrela muito comum. Mas podemos entender o universo. Isto nos torna muito especiais.

Stephen Hawking, 1988

Resumo

Este trabalho apresenta a proposta de um sistema de estimativa de distâncias em utilizando visão computacional estéreo. Para tanto, foram descritos assuntos teóricos como visão humana, visão computacional estéreo, calibração, retificação, correspondência e percepção de profundidade. O sistema de aquisição proposto foi montado em uma superfície de testes onde a metodologia descrita foi posta em prática, de maneira a se obter diversas comparações e aquisições de distâncias de objetos em cena. Posteriormente, o sistema foi testado em um veículo automotor em movimento, onde a eficácia do sistema foi comprovada em uma situação similar ao uso real. Os resultados foram satisfatórios, tendo-se obtido bons mapas de disparidade, com eficiência na identificação de objetos em cena. As respectivas distâncias entre os objetos e o sistema de aquisição puderam ser inferidas pela mudança de tonalidade com boa precisão.

Palavras-chave: visão computacional, visão computacional estéreo, mapa de disparidade, veículos autônomos.

Abstract

This thesis proposes a distance estimation system using computer stereo vision. Therefore, theoretical subjects such as human vision, computer stereo vision, calibration, rectification, matching and depth perception were described. The distance acquisition system was built on a testing surface where the proposed methodology was put into practice, in order to obtain several comparisons and acquisitions of object distances in the scene. Subsequently, the system was tested in a moving vehicle, where the effectiveness of the system was proven in a situation similar to real use. The results were satisfactory, the system generated good disparity maps with great identification of objects in the scene. The different distances between the objects and the acquisition system could be inferred by the change in color hue with good precision.

Keywords: computer vision, compouter stereo vision, disparity maps, autonomous vehicles.

Lista de Figuras

Sumário

1 Introdução

1.1 O automóvel e sua história

Segundo a Encyclopædia Britannica, o automóvel foi uma das maiores invenções da humanidade (CROMER; CROMER, 1998). Sua criação possibilitou grandes mudanças na sociedade, facilitando a locomoção humana e possibilitando maior autonomia às pessoas em seu dia a dia, fazendo com que a distância do trajeto já não fosse um problema tão grande, permitindo que as pessoas não morassem mais tão próximas de seus locais de trabalho, por exemplo.

No entanto, nos seus dias iniciais, o automóvel era visto por alguns como um grande problema, e não uma solução. Um exemplo disso foi a Lei da Bandeira Vermelha, imposta no Reino Unido em 1865, que obrigava os veículos a serem precedidos por um homem a pé acenando uma bandeira vermelha e soprando uma corneta. Tal lei permaneceu ativa até 1896 (ARCHIVES, 1865).

Diversas melhorias foram feitas nos automóveis desde então, de forma a aumentar a segurança dos seus ocupantes e dos pedestres. Tais melhorias, aliadas com o rápido crescimento industrial provocado com o início da primeira guerra mundial, fizeram com que o automóvel fosse cada vez mais ganhando espaço no trânsito, tornando-se um acessório amplamente utilizado nas grandes cidades.

1.2 Veículos autônomos e segurança no trânsito

Segundo a OMS, o Brasil está na quarta posição entre os países com mais mortes em acidentes de trânsito no mundo, com uma morte a cada quinze minutos, e a cada dois minutos ocorre um acidente com sequelas permanentes (WHO, 2018). Tais dados são alarmantes, e a discussão em relação aos acidentes de trânsito causados por erro humano tem recentemente ganhado espaço na mídia, principalmente com o advento do uso de inteligência artificial na direção dos automóveis, nos chamados veículos autônomos, como forma de reduzir os acidentes de trânsito.

Outro benefício de tal tecnologia, seria tornar o ato de dirigir o automóvel algo que não mais requer a atenção do motorista, em que este poderia utilizar o tempo de locomoção com outras atividades mais importantes, tendo em vista que nos grandes centros urbanos o tempo de comutação está cada vez maior.

Porém, os veículos autônomos também estão enfrentando certos movimentos de desconfiança por parte da população e governos no geral, assim como os primeiros veículos não autônomos enfrentaram em seus primórdios.

1.3 Motivação para este trabalho

Apesar de o automóvel ter sido uma grande invenção, um veículo autônomo seria muito superior ao motorista humano, tendo como reagir mais rapidamente a problemas inesperados na pista, e também sempre iria dirigir seguindo as leis de trânsito, evitando assim excessos de velocidade e problemas causados por embriaguez ao volante, por exemplo. Em uma sociedade em que todos os veículos são autônomos, a quantidade de acidentes fatais ou com sequelas seria extremamente menor, ao contrário do que algumas pessoas ainda acreditam.

A motivação inicial deste trabalho seria tentar demonstrar que um veículo autônomo teria capacidade de se posicionar no mundo à sua volta com ótima eficiência, podendo, em um futuro próximo, ser uma tecnologia extremamente confiável ao substituir motoristas humanos.

1.4 Objetivos deste trabalho

Primeiramente, este trabalho visará desenvolver um sistema que utilize processamento de imagens e visão computacional para determinar a distância entre o sistema e os seus arredores, utilizando um conjunto de câmeras estéreo (câmeras posicionadas a uma pequena distância uma da outra), de forma a imitar a visão humana e possibilitar a criação de um mapa de profundidade utilizando a disparidade entre as imagens capturadas pelas câmeras, de forma a utilizar tal mapa para calcular a distância entre o veículo e seus arredores.

Posteriormente, será feita uma avaliação do sistema projetado e sua possibilidade de uso em tempo real.

2 Revisão teórica e bibliográfica

2.1 Visão humana

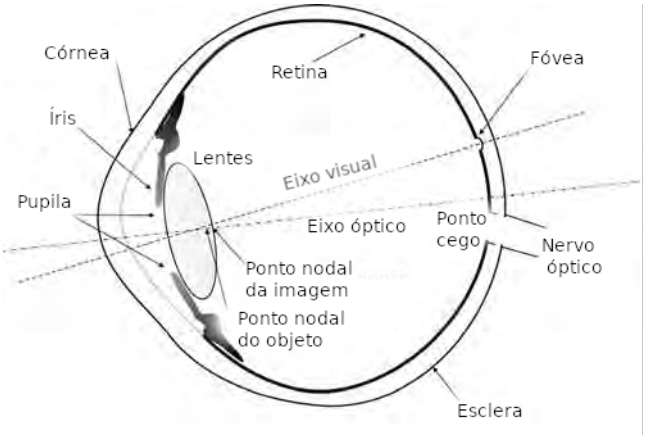
O sistema visual compreende o órgão sensorial (o olho) e partes do sistema nervoso central (a retina, o nervo óptico, o trato óptico e o córtex visual) e nos proporciona o sentido da visão (a capacidade de detectar e processar luz visível). Ele detecta e interpreta informações do espectro de luz visível para "construir uma representação" do ambiente circundante. O sistema realiza uma série de tarefas complexas, incluindo a recepção de luz e a formação de representações neurais monoculares, percepção de cores, estereopsia e avaliação de distâncias para e entre objetos, identificação de objetos, percepção de movimento, análise e integração de informações visuais, reconhecimento de padrões e muito mais.

Neste capítulo, serão descritos diversos conceitos necessários para o entendimento do leitor, tendo como objetivo final que o mesmo tenha um "conhecimento" claro e conciso da base teórica que motivou o estudo dos paralelos entre a visão humana e a visão computacional.

2.1.1 O olho humano

De maneira simplificada, o olho humano é uma câmera esférica com uma lente de distância focal de 20 mm no lado externo, focalizando a imagem na retina que está oposta à lente e fixada no lado interno da superfície da esfera, conforme ilustrado na Figura 1. A íris controla a quantidade de luz que passa pela "lente", controlando o tamanho da pupila. Cada olho tem aproximadamente cem milhões de células receptoras. Além disso, a retina é povoada de forma desigual com células sensoras. Uma área próxima ao centro da retina, chamada fóvea, tem uma concentração muito densa de receptores de cor, chamados cones. Longe do centro, a densidade dos cones diminui enquanto a densidade dos receptores de preto e branco, os bastonetes, aumenta (SHAPIRO, 1992).

Figura 1 – O olho humano.



Fonte: (CYGANEK; SIEBERT, 2011) (Adaptado).

O olho humano percebe três intensidades separadas para as três cores constituintes de uma única imagem de ponto de superfície na fóvea, porque a luz recebida desse ponto incide em 3 tipos diferentes de cones. Cada tipo de cone possui um pigmento especial que é sensível aos comprimentos de onda da luz em uma determinada faixa.

Uma parte significativa do cérebro humano está envolvida no processamento de dados visuais. Uma das propriedades mais intrigantes do cérebro humano é sua capacidade de perceber suavemente um mundo estável e uniforme, embora os olhos estejam em constante movimento.

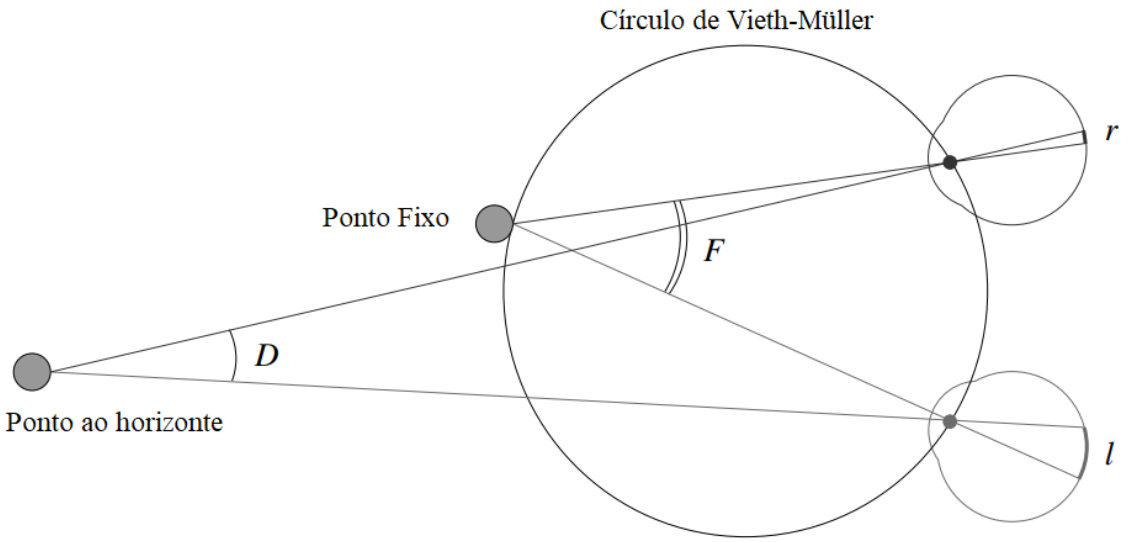
No contexto da visão binocular e da percepção estereoscópica de profundidade, deve-se notar que ao contrário das câmeras rigidamente fixadas a um equipamento estéreo passivo, os dois olhos de uma pessoa podem girar em seus encaixes. A cada instante, eles se fixam em um determinado ponto do espaço (ou seja, eles giram de modo que as imagens correspondentes se formem no centro de suas fóveas).

Uma interpretação mais simples deste fenômeno pode ser demonstrada se for reduzido o escopo de análise para o caso bidimensional: se l e r denotam os ângulos (sentido anti-horário) entre os planos verticais de simetria de dois olhos e dois raios que passam pelo mesmo ponto de cena, define-se a disparidade correspondente como $d = r - l$. É um exercício elementar de trigonometria mostrar que $d = D - F$, onde D denota o ângulo entre esses raios e F é o ângulo entre os dois raios que passam pelo ponto fixado (FORSYTH; PONCE, 2015).

Os pontos com disparidade zero encontram-se no círculo Vieth-Müller, que passa pelo ponto fixado e pelos centros ópticos dos olhos. Os pontos dentro deste círculo têm uma disparidade positiva, os pontos fora dele têm, como na Figura 2, uma disparidade negativa, e o locus de todos os pontos tendo uma dada disparidade d forma, conforme d varia, a família de todos os círculos que passam pelo centros ópticos de dois olhos. Esta propriedade

é claramente suficiente para classificar pontos pela ordem em que estão próximos do ponto de fixação de acordo com sua profundidade. Porém, também está claro que os ângulos de vergência entre o plano vertical mediano de simetria da cabeça e os dois raios de fixação devem ser conhecidos para reconstruir a posição absoluta dos pontos de cena.

Figura 2 – Disparidade retiniana em um caso bidimensional simples



Neste diagrama, o ponto próximo é fixado pelos olhos e se projeta no centro de suas fóveas sem disparidade. As duas imagens do ponto distante desviam-se desta posição central em quantidades diferentes, indicando uma profundidade diferente. Fonte: (FORSYTH; PONCE, 2015) (Adaptado).

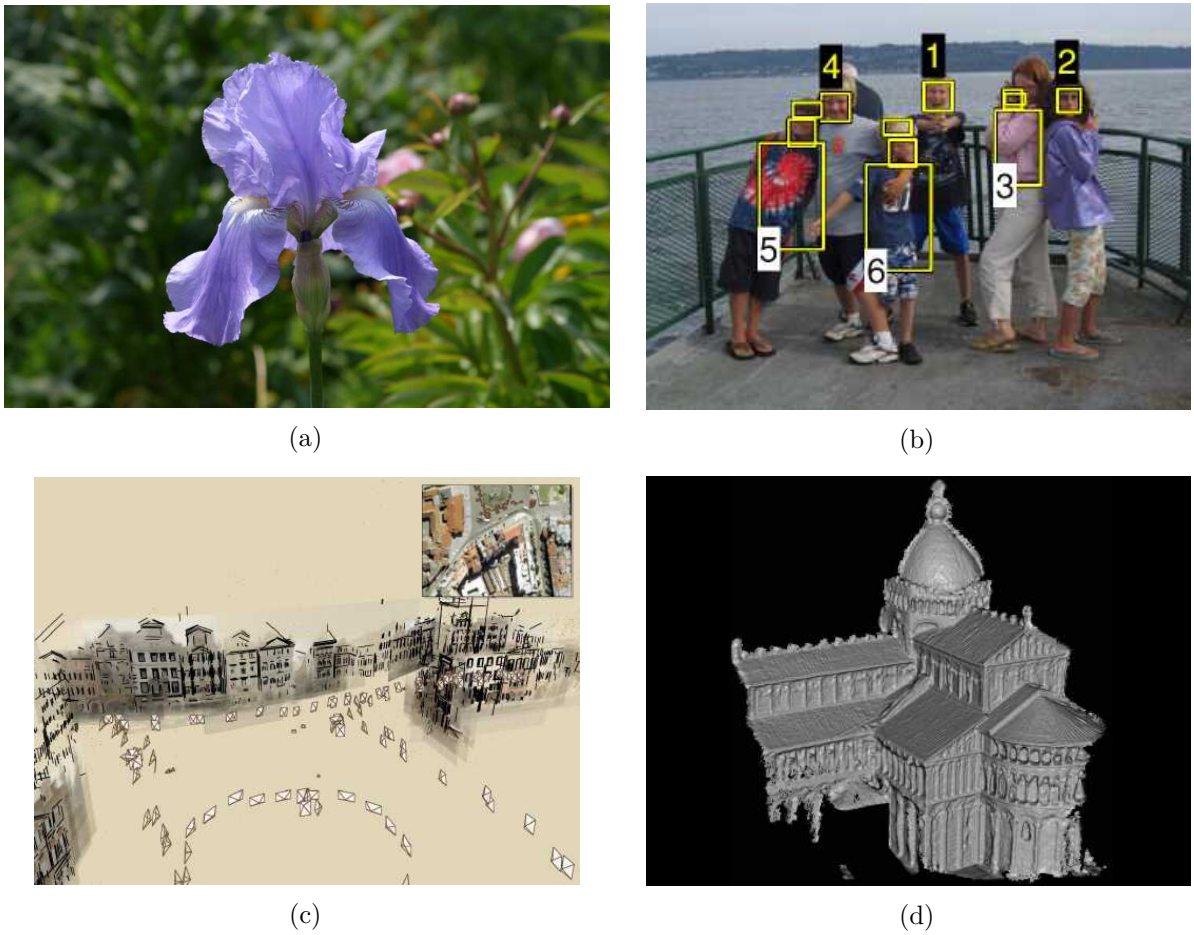
O caso tridimensional é naturalmente mais complicado, com o locus dos pontos de disparidade zero tornando-se uma superfície, o horóptero, mas a conclusão geral é a mesma, e o posicionamento absoluto requer os ângulos de vergência. Existem algumas evidências de que esses ângulos não podem ser medidos com precisão por nosso sistema nervoso. No entanto, a profundidade relativa, ou classificação dos pontos ao longo da linha de visão, pode ser avaliada com bastante precisão para disparidades de alguns segundos de arco. Para tanto, o sistema visual utiliza diversas *pistas visuais*, que serão estudadas com maior detalhe no capítulo 2.1.3.

2.1.2 Percepção visual

O ser humano é capaz de perceber a estrutura tridimensional do mundo ao seu redor com grande facilidade. Por exemplo, ao olhar para um vaso de flores, pode-se perceber a translucidez de cada pétala por meio dos padrões sutis de luz e sombreamento que atuam em sua superfície, e segmentar sem esforço cada flor do fundo da cena. Ao olhar para o retrato de grupo emoldurado, é possível contar e nomear todas as pessoas na foto e

até mesmo adivinhar suas emoções a partir de suas expressões faciais. Exemplos de tais "façanhas" estão dispostos na Figura 3.

Figura 3 – Exemplos da capacidade de percepção da visão humana



(a) (b)

(c) (d)

(a) Uma flor, onde é possível facilmente destacar as pétalas e extrair informações relevantes sobre a mesma. (b) Um retrato de um grupo de amigos, onde é possível contar o número de participantes, identificar suas emoções, etc. (c) Uma reconstrução 3D de uma peça, feita utilizando milhares de fotos sobrepostas. É possível identificar as construções e até mesmo uma fonte. (d) Uma reconstrução de um objeto 3D, utilizando *point-cloud stereoscopy*. **Fonte:** (SZELISKI, 2010)

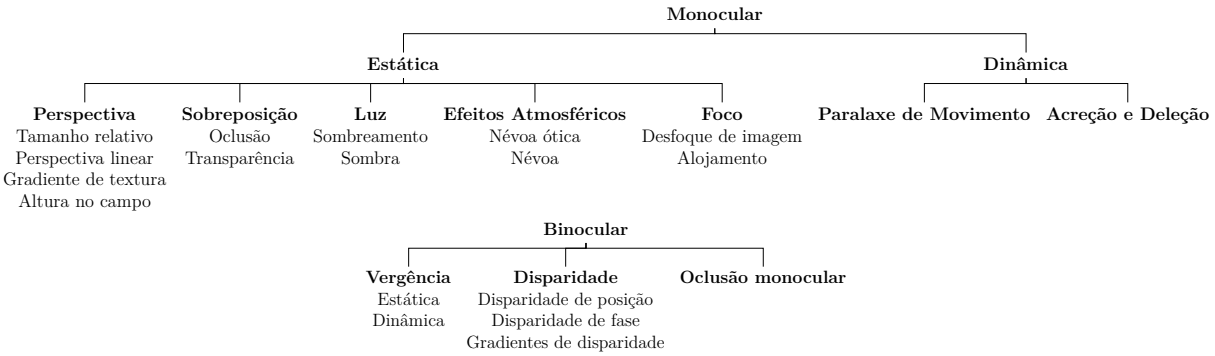
Psicólogos especializados em percepção humana passaram décadas tentando entender como o sistema visual funciona, e uma solução completa para esse quebra-cabeça permanece indefinida. Uma excelente (e extremamente detalhada) leitura sobre a parte teórica da visão e percepção do mundo ao nosso redor pode ser explorada com maior profundidade pelo leitor na obra *"Perceiving in Depth"*, de Ian P. Howard, um dos maiores nomes da pesquisa sobre percepção visual humana.

2.1.3 Percepção de profundidade na visão humana

Percepção de profundidade é a habilidade visual de perceber o mundo em três dimensões e possibilitar a estimativa da distância de um objeto. A percepção de profundi-

dade feita pelo ser humano não usa apenas a disparidade vista na seção 2.1.1, pois como visto, não são realizados "cálculos mentais" com os ângulos formados pela imagem com a fóvea para interpretar distâncias. Em vez disso, o ser humano utiliza estes dados de maneira intrínseca, associados a uma variedade de pistas de profundidade (Figura 4) que são processadas e analisadas pelo cérebro sem a necessidade de se interpretar o que está sendo visualizado. Tais pistas são normalmente classificadas em pistas monoculares, que podem ser representadas em apenas duas dimensões e observadas com apenas um olho, e pistas binoculares, que são baseadas no recebimento de informações sensoriais em três dimensões de ambos os olhos. Uma breve e sucinta explicação das pistas mais importantes será disposta a seguir.

Figura 4 – Diagrama das diversas pistas visuais utilizadas para percepção de distância



Fonte: (HOWARD; ROGERS, 2012) (Adaptado).

As pistas monoculares incluem:

- **Tamanho relativo:** Objetos distantes geram ângulos visuais menores do que objetos próximos;
- **Perspectiva linear:** Linhas paralelas encontrando-se no horizonte infinito;
- **Gradiente de textura:** Objetos próximos tendem a ter melhor percepção de detalhes do que objetos distantes;
- **Oclusão:** Objetos mais próximos serão vistos à frente de objetos distantes;
- **Transparência:** Objetos que possuem transparência possibilitarão que objetos opacos sejam vistos através de si;
- **Diferenças de contraste:** Devido ao espalhamento da luz na atmosfera, objetos distantes possuem menos contraste de cor;
- **Paralaxe de movimento:** Objetos distantes parecem se movimentar mais lentamente que objetos próximos.

As pistas binoculares incluem:

- **Disparidade retiniana (estereopsia):** Como os olhos humanos estão dispostos frontalmente e com uma pequena distância colocados frontalmente também podem usar informações derivadas de diferentes projeções de objetos em cada retina para julgar a profundidade. Usando duas imagens da mesma cena obtidas de ângulos ligeiramente diferentes, é possível triangular a distância até um objeto com um alto grau de precisão;
- **Vergência:** Sensações musculares causadas pela focalização dos olhos em objetos, derivada da disparidade retiniana.

Para o contexto deste trabalho, em que se propõe utilizar um sistema de câmeras estéreo de forma a reproduzir a visão humana, apenas as pistas binoculares poderiam servir de inspiração para o desenvolvimento de um sistema de detecção da profundidade. Destas, a única pista que poderá produzir bons resultados é a utilização da disparidade entre as imagens capturadas pelas câmeras para simular a estereopsia, já que não há contatos mecânicos entre as câmeras que possibilitem o cálculo do estresse muscular causado pela focalização em objetos da cena.

2.2 Visão computacional

O interesse em desvendar as diversas singularidades da capacidade humana de percepção do ambiente ao seu redor também é um campo de estudo muito importante para a visão computacional, onde os pesquisadores vêm desenvolvendo técnicas matemáticas para recuperar a forma tridimensional e a aparência de objetos em imagens. No campo científico, o progresso nas últimas duas décadas foi muito rápido, tendo-se desenvolvido técnicas confiáveis para calcular com precisão um modelo 3D de um ambiente a partir de milhares de fotografias parcialmente sobrepostas, modelos 3D densos e precisos de superfícies de objetos usando correspondência estéreo, ou ainda delinear pessoas e objetos em uma fotografia, identificando suas emoções e ações (SZELISKI, 2010). Exemplos estão dispostos na Figura 3. No entanto, apesar de todos esses avanços, o sonho de ter um computador explicando uma imagem com o mesmo nível de detalhe e causalidade de uma criança de dois anos permanece indefinido.

Visão computacional é, por fim, um campo interdisciplinar cujo objetivo final é desenvolver métodos para compreensão e extração do conteúdo presente em imagens e vídeos, onde tenta-se descrever o mundo que é visto através de uma ou mais imagens e reconstruir suas propriedades, como forma, iluminação e distribuição de cores. Serve ainda como base na criação de algoritmos para tomada de decisão sobre objetos reais e cenas baseadas em imagens capturadas (SHAPIRO, 1992).

2.2.1 Transformação de coordenadas

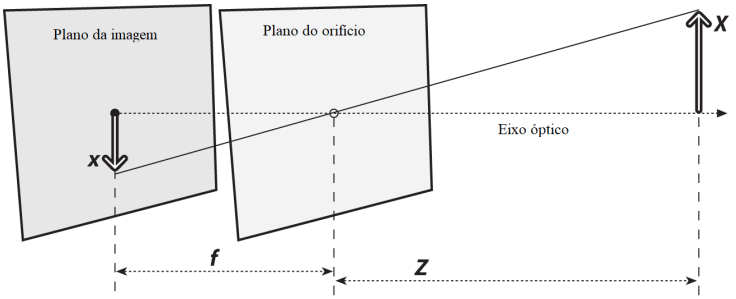
Todo sistema de aquisição de imagem, seja o sistema visual humano ou de máquina, por sua natureza realiza algum tipo de transformação do espaço 3D real em espaço 2D imaginário. Encontrar os parâmetros de tal transformação é fundamental para descrever o sistema de aquisição. Um modelo de câmera muito utilizado para descrever o sistema e também para transformar o sistema de coordenadas, é o modelo da câmera estenopeica, mais conhecido como câmera *pinhole*.

2.2.2 O modelo da câmera *pinhole*

No modelo mais simples de câmera, o *pinhole*, idealiza-se a luz como se estivesse sendo refletida da cena ou emitida por um objeto distante, mas apenas um único raio entra no orifício de qualquer ponto específico daquela cena. Em uma câmera *pinhole* física, esse ponto é então "projetado" em um *plano de imagem* (KAEHLER; BRADSKI, 2016).

Como resultado, a imagem neste *plano de imagem* está sempre em foco, e o tamanho da imagem em relação ao objeto distante é dado por um único parâmetro da câmera: seu comprimento focal. Para a câmera *pinhole* idealizada, a distância da abertura do orifício à tela é precisamente a distância focal. Isso é mostrado na Figura 5, onde f é a distância focal da câmera, Z é a distância da câmera ao objeto, X é o comprimento do objeto e x é a imagem do objeto no plano de imagem. Na figura, pode-se ver por semelhança de triângulos que $\frac{-x}{f} = \frac{X}{Z}$, ou ainda, $-x = f \cdot \frac{X}{Z}$.

Figura 5 – Modelo idealizado de uma câmera *pinhole*

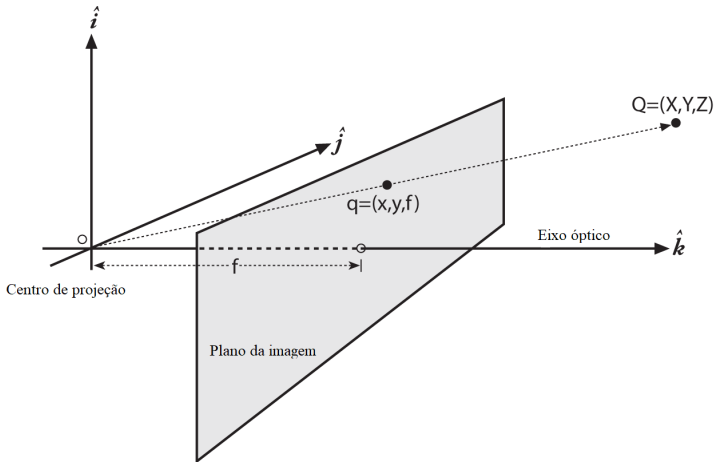


Fonte: (KAEHLER; BRADSKI, 2016) (Adaptado).

É possível, porém, reorganizar o modelo idealizado na figura 5 para um modelo em que que se pode extrair as equações matemáticas necessárias de maneira mais simples. Na Figura 6, trocou-se o orifício e o plano da imagem. A principal diferença é que o objeto agora aparece com o lado direito para cima. O ponto no orifício é reinterpretado como o centro da projeção. Nessa forma de ver as coisas, cada raio deixa um ponto no objeto distante e se dirige ao centro de projeção. O ponto na intersecção do plano da imagem e do eixo óptico é referido como o ponto principal.

Nesse novo plano de imagem frontal, a imagem do objeto distante tem exatamente o mesmo tamanho que estava no plano de imagem na Figura 5. A imagem é gerada pela intersecção desses raios com o plano da imagem, que passa a estar exatamente a uma distância f do centro de projeção. Isso torna a relação de triângulos semelhantes $\frac{-x}{f} = \frac{X}{Z}$ mais diretamente evidente do que antes. O sinal negativo desaparece porque a imagem do objeto não está mais invertida.

Figura 6 – Modelo reorganizado de uma câmera *pinhole* de forma a simplificar os cálculos



Fonte: (KAEHLER; BRADSKI, 2016) (Adaptado).

Devido a imperfeições no processo de manufatura ou outras aleatoriedades, o centro do chip de uma câmera geralmente não está no eixo óptico. Assim, necessitam-se dois novos parâmetros, c_x e c_y, para modelar um possível deslocamento (para longe do eixo óptico) do centro de coordenadas na tela de projeção. O resultado é que um modelo relativamente simples no qual um ponto Q no mundo físico, cujas coordenadas são (X, Y, Z), é projetado no plano em alguma localização de *pixel* dada por (x_{tela}, y_{tela}) de acordo com as Equações 1 e 2 (KAEHLER; BRADSKI, 2016):

$$x_{tela} = f_x \cdot \frac{X}{Z} + c_x \tag{1}$$

$$y_{tela} = f_y \cdot \frac{Y}{Z} + c_y \tag{2}$$

Observe que foram introduzidas duas distâncias focais diferentes; a razão para isso é que os *pixels* individuais em uma câmera de baixo custo são retangulares em vez de quadrados. A distância focal f_x, por exemplo, é na verdade o produto da distância focal física da lente e o tamanho s_x dos elementos individuais do gerador de imagens. É importante ter em mente, entretanto, que s_x e s_y não podem ser medidos diretamente por meio de nenhum processo de calibração de câmera, e nem a distância focal física f

pode ser diretamente mensurável. É possível derivar apenas as combinações $f_x = F \cdot s_x$ e $f_y = F \cdot s_y$ sem realmente desmontar a câmera e medir seus componentes diretamente.

2.2.3 Parâmetros intrínsecos da câmera

Depois de projetar um ponto 3D através de um orifício ideal usando uma matriz de projeção, ainda é necessário transformar as coordenadas resultantes de acordo com o espaçamento do sensor e a posição relativa do plano do sensor em relação à origem.

Os sensores de imagem retornam valores de *pixel* indexados por coordenadas inteiras de *pixel* $(x_s; y_s)$, geralmente com as coordenadas começando no canto superior esquerdo da imagem e movendo-se para baixo e para a direita. Para mapear centros de *pixel* para coordenadas 3D, primeiro escala-se os valores $(x_s; y_s)$ pelos tamanhos de *pixels* $(s_x; s_y)$ e, em seguida, descreve a orientação da matriz do sensor em relação ao centro de projeção da câmera $\mathbf{O}_c$ com uma origem $\mathbf{c}_s$ e uma rotação 3D $\mathbf{R}_s$.

A matriz de projeção 3D para 2D pode ser finalmente definida conforme a Equação 3 (SZELISKI, 2010):

$$\mathbf{p} = \begin{bmatrix} \mathbf{R}_s & \mathbf{c}_s \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \mathbf{M}_s \overline{\mathbf{x}}_s \tag{3}$$

As primeiras duas colunas da matriz 3×3 $\mathbf{M}_s$ são os vetores 3D correspondentes a espaços unitários na matriz de *pixels* da imagem ao longo das direções x_s e y_s, enquanto a terceira coluna é a origem da matriz de imagens 3D $\mathbf{c}_s$.

A matriz $\mathbf{M}_s$ é parametrizada por oito incógnitas: os três parâmetros que descrevem a rotação $\mathbf{R}_s$, os três parâmetros que descrevem a translação $\mathbf{c}_s$ e os dois fatores de escala $(s_x; s_y)$. Observe que ignora-se aqui a possibilidade de inclinação entre os dois eixos no plano da imagem, uma vez que as técnicas de manufatura de estado sólido tornam isso desprezível. Na prática, a menos que se tenha conhecimento externo preciso do espaçamento do sensor ou orientação do sensor, existem apenas sete graus de liberdade, uma vez que a distância do sensor da origem não pode ser separada do espaçamento do sensor, com base apenas na medição de imagem externa. No entanto, estimar um modelo de câmera M_s com os sete graus de liberdade necessários é impraticável, então a maioria dos profissionais assume uma forma de matriz homogênea 3×3 geral.

A relação entre o centro do *pixel* 3D $\mathbf{p}$ e o ponto central da câmera 3D $\mathbf{p}_c$ é dada por uma escala desconhecida s, $\mathbf{p} = s \cdot \mathbf{p}_c$. É necessário, portanto, escrever a projeção

completa entre o $\mathbf{p}_c$ e uma versão homogênea do endereço de *pixel* $\tilde{x}_s$ conforme a Equação 4 (SZELISKI, 2010):

$$\tilde{x}_s = \alpha \mathbf{M}_s^{-1} \ \mathbf{p}_c = K\mathbf{p}_c \tag{4}$$

A matriz 3×3 $\mathbf{K}$ é chamada de matriz de calibração e descreve os intrínsecos da câmera (em oposição à orientação da câmera no espaço, que são chamados de extrínsecos). Ela é geralmente representada pela Equação 5 (SZELISKI, 2010):

$$\mathbf{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

Frequentemente, definir a origem aproximadamente no centro da imagem, por exemplo, $(c_x;\ c_y) = (W/2;\ H/2)$ (onde W e H são a largura e a altura da imagem, respectivamente), pode resultar em um uso modelo de câmera "perfeito" com uma única incógnita, a distância focal f.

Pode-se finalmente, ao combinar os parâmetros extrínsecos e intrínsecos com a matriz de calibração K, obter a matriz da câmera, demonstrada na Equação 6 (SZELISKI, 2010):

$$\mathbf{P} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \tag{6}$$

Tal matriz pode ser utilizada para converter com precisão coordenadas 3D no mundo real para coordenadas 2D do mundo projetado na imagem, e vice-versa. Com uma câmera *pinhole* ideal, tem-se um modelo útil para parte da geometria tridimensional da visão. No entanto, muito pouca luz passa pelo orifício. Na prática, tal arranjo resultaria em imagens com tempo de captura muito elevado, enquanto espera-se o acúmulo de luz suficiente. Para uma câmera formar imagens de forma mais rápida, deve-se reunir muita luz em uma área mais ampla e focar essa luz para convergir no ponto de projeção. para isso, utiliza-se uma lente. Uma lente pode focar uma grande quantidade de luz em um ponto para fornecer imagens rápidas, mas isso vem ao custo de introduzir distorções.

2.2.4 Distorção de lentes

Em teoria, é possível construir uma lente que não apresentará distorções. Na prática, entretanto, nenhuma lente é perfeita. Isso ocorre principalmente por motivos de fabricação; é muito mais fácil fazer uma lente "esférica" do que uma lente "parabólica" matematicamente ideal. Também é difícil alinhar mecanicamente a lente e o gerador de imagens com exatidão. Aqui, descrevem-se as duas distorções principais da lente e como

modelá-las. Existem muitos outros tipos de distorções que ocorrem em sistemas de imagem, mas eles normalmente têm efeitos menores do que as distorções radiais e tangenciais e são negligenciáveis.

As **distorções radiais** surgem como resultado do formato da lente, distorcendo visivelmente a localização dos *pixels* perto das bordas do sensor de imagens. Este fenômeno de protuberância é a fonte do efeito "barril", "almofada" ou "olho de peixe", exemplificados na Figura 7.

Figura 7 – Exemplos de distorção de lentes

(a) (b) (c)

Distorções da lente radial: (a) cilindro, (b) almofada e (c) olho de peixe. A imagem olho de peixe mede quase 180º de lado a lado. **Fonte:** (SZELISKI, 2010)

Percebe-se que a distorção é nula no centro (óptico) do sensor de imagens e aumenta à medida em que se move em direção à periferia. A correção se dá pelas Equações 7 e 8 (KAEHLER; BRADSKI, 2016):

$$x_{corrigido} = x \cdot (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{7}$$

$$y_{corrigido} = y \cdot (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{8}$$

A segunda maior distorção comum é a **distorção tangencial**. Essa distorção se deve a defeitos de fabricação resultantes da lente não ser exatamente paralela ao plano de imagem. A correção da mesma pode ser efetuada pela aplicação das Equações 9 e 10 (KAEHLER; BRADSKI, 2016):

$$x_{corrigido} = x + [2p_1 xy + p_2(r^2 + 2x^2)] \tag{9}$$

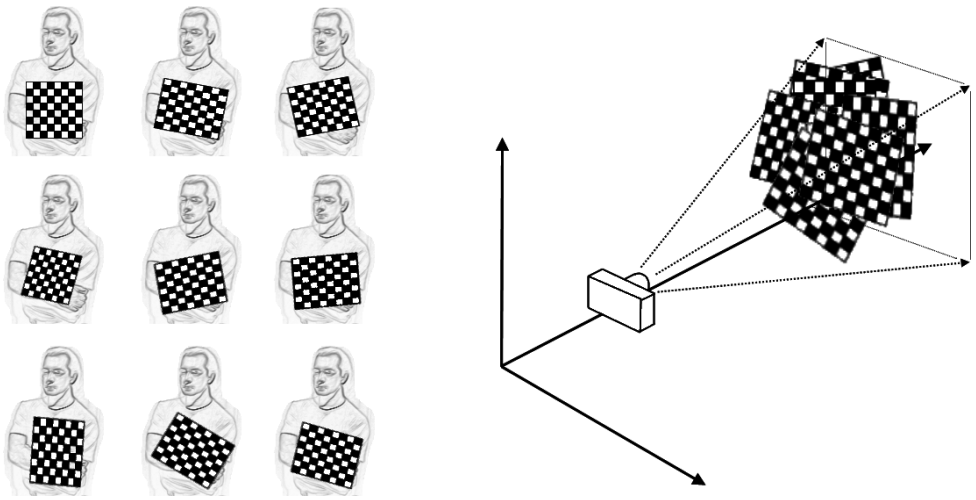
$$y_{corrigido} = y + [p_1(r^2 + 2x^2) + 2p_2xy] \tag{10}$$

2.2.5 Calibração

Agora que é conhecido como descrever matematicamente as propriedades intrínsecas e de distorção de uma câmera, é possível utilizar métodos de calibração para determinar tais matrizes. Métodos de calibração envolvem direcionar a câmera para uma estrutura conhecida que possui muitos pontos individuais e identificáveis.

Ao visualizar essa estrutura por uma variedade de ângulos, pode-se calcular a localização relativa e a orientação da câmera no momento de cada imagem, bem como os parâmetros intrínsecos da câmera. Para isso, é necessário **rotacionar** e **transladar** o objeto de maneira a cobrir toda a área da câmera. O método mais usual utiliza um padrão de "*tabuleiro de xadrez*", demonstrado na Figura 8. Nela, um pesquisador hipotético segura o tabuleiro de calibração em frente à câmera, e faz diversos movimentos de translação e rotação do mesmo, de forma a mapear toda a extensão da área de captura da câmera.

Figura 8 – O método de calibração por tabuleiro de xadrez

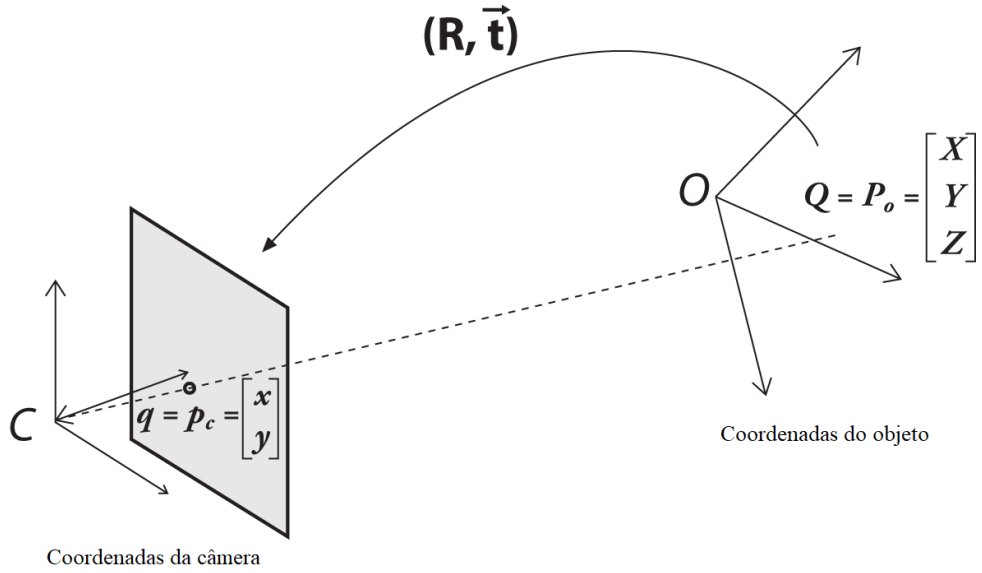


Fonte: (KAEHLER; BRADSKI, 2016)

O uso de um padrão de quadrados pretos e brancos alternados garante que não haja tendência para um lado ou outro na medição. Além disso, os cantos da grade resultantes se prestam naturalmente à função de localização de subpixel. Alguns métodos de calibração na literatura dependem de objetos tridimensionais (por exemplo, uma caixa coberta com marcadores), mas os padrões de tabuleiro de xadrez são muito mais fáceis de lidar, além de que é bastante difícil confeccionar objetos de calibração tridimensionais precisos.

Para cada imagem que a câmera tira de um objeto específico, é possível descrever a pose do objeto em relação ao sistema de coordenadas da câmera em termos de rotação e translação, como demonstrado na Figura 9.

Figura 9 – Conversão de um objeto em diferentes sistemas de coordenadas.



Fonte: (KAEHLER; BRADSKI, 2016) (Adaptado).

Em geral, uma rotação em qualquer número de dimensões pode ser descrita em termos de multiplicação de um vetor de coordenadas por uma matriz quadrada de tamanho apropriado. Em última análise, uma rotação é equivalente a introduzir uma nova descrição da localização de um ponto em um sistema de coordenadas diferente. Girar o sistema de coordenadas por um ângulo ϕ é equivalente a girar o ponto alvo em torno da origem desse sistema de coordenadas pelo mesmo ângulo ϕ. A representação de uma rotação bidimensional como multiplicação de matriz é mostrada na Equação 11. A rotação em três dimensões pode ser decomposta em uma rotação bidimensional em torno de cada eixo em que as medições do eixo do pivô permanecem constantes. A rotação ocorrer em torno dos eixos x, y e z em sequência com respectivos ângulos de rotação ψ, ϕ e θ, o resultado é uma matriz de rotação total R que é dada pelo produto das três matrizes $R_x(\psi), R_y(\phi), R_z(\theta)$, onde, conforme as matrizes abaixo, tem-se $R = R_x(\psi) \cdot R_y(\phi) \cdot R_z(\theta)$ (KAEHLER; BRADSKI, 2016).

$$R_x(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\psi & sin\psi \\ 0 & -sin\psi & cos\psi \end{bmatrix}, R_y(\phi) = \begin{bmatrix} cos\phi & 0 & -sin\phi \\ 0 & 1 & 0 \\ sin\psi & 0 & cos\psi \end{bmatrix}, R_z(\theta) = \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

A matriz de rotação R tem a propriedade de que seu inverso é sua transposta; portanto, tem-se $R^T \cdot R = R \cdot R^T = I_3$, onde I_3 é a matriz identidade 3×3, com sua diagonal principal tendo apenas o elemento 1 e o restante dos elementos são formados por zeros.

O vetor de translação é como se representa uma mudança de um sistema de coordenadas para outro sistema cuja origem é deslocada para outro local; em outras palavras, o vetor de translação é apenas o deslocamento da origem do primeiro sistema de coordenadas para a origem do segundo sistema de coordenadas. Assim, para mudar de um sistema de coordenadas centrado em um objeto para um centralizado na câmera, o vetor de translação apropriado é simplesmente $\overrightarrow{T} = origem_{objeto} - origem_{câmera}$. Sabe-se então, conforme a Figura 9, que um ponto no quadro de coordenadas do objeto $\overrightarrow{P}_o$ tem coordenadas $\overrightarrow{P}_c$ nas coordenadas da câmera, conforme a Equação 12 (KAEHLER; BRADSKI, 2016):

$$\overrightarrow{P}_c = R \cdot (\overrightarrow{P}_o - \overrightarrow{T}) \tag{12}$$

Combinar esta equação para $\overrightarrow{P}_c$ com as correções intrínsecas da câmera formará um sistema básico de equações. A solução para essas equações conterá os parâmetros de calibração da câmera buscada.

2.2.5.1 Calibração estéreo

Calibração estéreo consiste no ato de combinar as duas calibrações realizadas para cada câmera individualmente, de forma a mapear a relação geométrica entre ambas as câmeras utilizadas no sistema de visão computacional estéreo. As Equações 13 e 14 denotam como utilizar a matriz de rotação e vetor de translação encontrados para cada câmera, de forma a se obter a matriz de rotação estéreo e o vetor de translação estéreo.

$$R = R_r \cdot R_l^T \tag{13}$$

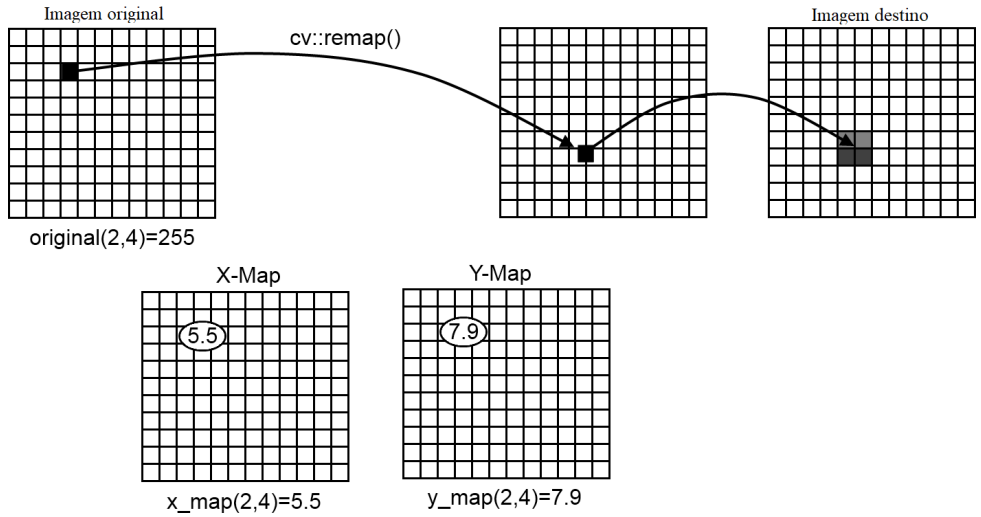
$$\overrightarrow{T} = \overrightarrow{T}_r - R \cdot \overrightarrow{T}_l \tag{14}$$

2.2.6 Retificação

Ao realizar a remoção de distorções em uma imagem, processo conhecido como retificação, é necessário especificar onde cada *pixel* da imagem de entrada deve ser movido na imagem de saída. Tal especificação é chamada de mapa de retificação (ou às vezes apenas mapa de distorção). Existem várias representações disponíveis para esses mapas, mas a representação flutuante de dois canais é a mais usual.

Nesta representação, um remapeamento para uma imagem N × M é representado por uma matriz N × M de números de ponto flutuante de dois canais, conforme mostrado na Figura 10.

Figura 10 – Representação flutuante de dois canais para remoção de distorções.



Fonte: (KAEHLER; BRADSKI, 2016) (Adaptado).

Para qualquer entrada (i, j) na imagem, o valor dessa entrada será um par de números (i^*, j^*) indicando o local para o qual o *pixel* (i, j) da imagem de entrada deve ser realocado. Obviamente, como (i^*, j^*) são números de ponto flutuante, a interpolação na imagem de destino está implícita.

2.3 Técnicas de correspondência em imagens estéreo

Existem diversos tipos de técnicas de correspondência de imagens, e cada técnica possui diferentes algoritmos otimizados para um determinado tipo de utilização ou resultado esperado. Para o escopo deste trabalho, focar-se-á na descrição de técnicas relevantes para a visão computacional estéreo.

2.3.1 Restrição epipolar

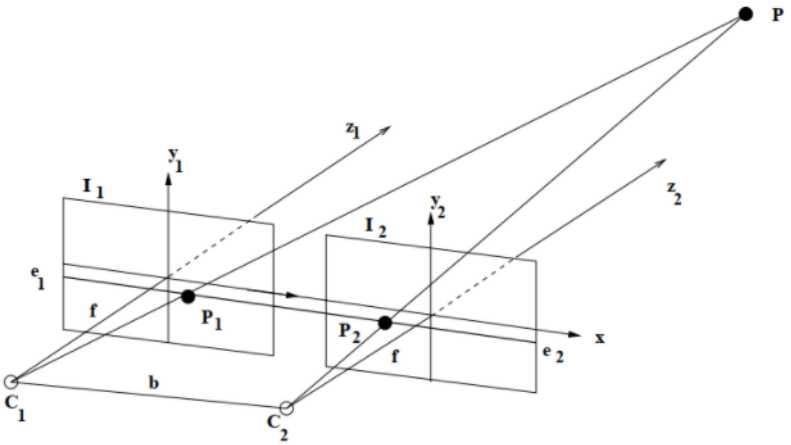
A correspondência estéreo pode ser bastante simplificada se a orientação relativa das câmeras for conhecida, pois o espaço de busca bidimensional para o ponto em uma imagem que corresponde a um determinado ponto em uma segunda imagem é reduzido a uma busca unidimensional, devido à geometria epipolar do par de imagens.

Segundo (SHAPIRO, 1992), o plano epipolar é o plano que contém o ponto 3D P, as duas câmeras C_1 e C_2, e os pontos correspondentes P_1 e P_2 aos quais P é projetado.

As linhas e_1 e e_2 resultantes da interseção do plano epipolar com as imagens I_1 e I_2 são chamadas de linhas epipolares, e o ponto no qual elas intersectam é chamado de epipolo.

A Figura 11 mostra a geometria epipolar no caso simples em que os dois planos de imagem são idênticos e paralelos à linha de base.

Figura 11 – Restrição epipolar



Fonte: (HARTLEY; ZISSERMAN, 2003)

Neste caso, dado um ponto $P_1 = (x_1,y_1)$ na imagem I_1, o ponto correspondente $P_2 = (x_2,y_2)$ na imagem I_2 é conhecido por estar na mesma linha de varredura; ou seja, $y_1 = y_2$. Tal configuração é conhecida como par de imagens normal. Embora o par normal torne a geometria simples, nem sempre é possível colocar câmeras nesta posição e pode não levar a disparidades grandes o suficiente para calcular informações de profundidade precisas. A configuração geral do estéreo possui posições e orientações arbitrárias das câmeras, cada uma das quais deve visualizar um subvolume significativo do objeto. Dado o ponto P_1 na linha epipolar e_1 na imagem I_1 e conhecendo as orientações relativas das câmeras, é possível encontrar a linha epipolar correspondente e_2 na imagem I_2 na qual o ponto P_2 correspondente deve estar.

2.3.2 Correspondência por blocos

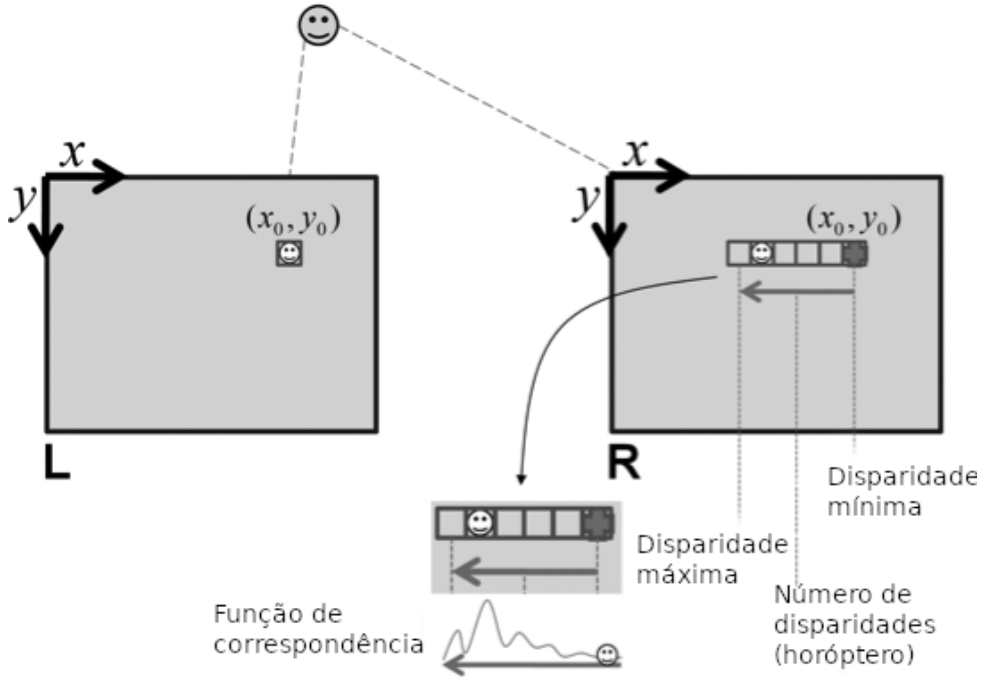
Este tipo de correspondência parte do pressuposto que o usuário já realizou a calibração das câmeras e que a distorção das lentes foi removida. Com estas ressalvas, é possível aplicar uma restrição epipolar de forma a se reduzir a procura de correspondências para apenas uma linha de *pixels* por vez, pois têm-se a garantia de que as imagens são correspondentes.

Primeiramente, realiza-se uma pré-filtragem das imagens capturadas de forma a se reduzir diferenças de iluminação e também salientar a textura dos objetos da cena. Em seguida, roda-se uma pesquisa de correspondências ao longo de uma das linhas de *pixels*

da imagem. Cada *pixel* analisado corresponde a uma disparidade. A janela de pesquisa começa na coordenada exata correspondente da imagem esquerda para a imagem direita (conhecida como disparidade mínima), e move-se para da direita para a esquerda. O tamanho da janela corresponde à disparidade máxima.

Definir o número de disparidades a serem pesquisadas estabelece o horóptero, o volume tridimensional que é coberto pela faixa de pesquisa do algoritmo estéreo. A Figura 12 mostra os limites de pesquisa de disparidade de cinco *pixels*. Cada limite de disparidade define um plano a uma profundidade fixa das câmeras. Fora desta faixa, a profundidade não será encontrada e representará um "buraco" no mapa de profundidade onde a profundidade não é conhecida. Notavelmente, é possível tornar os horópteros maiores diminuindo a distância entre as câmeras, tornando a distância focal menor, aumentando a faixa de pesquisa de disparidade estéreo ou aumentando a largura do *pixel*.

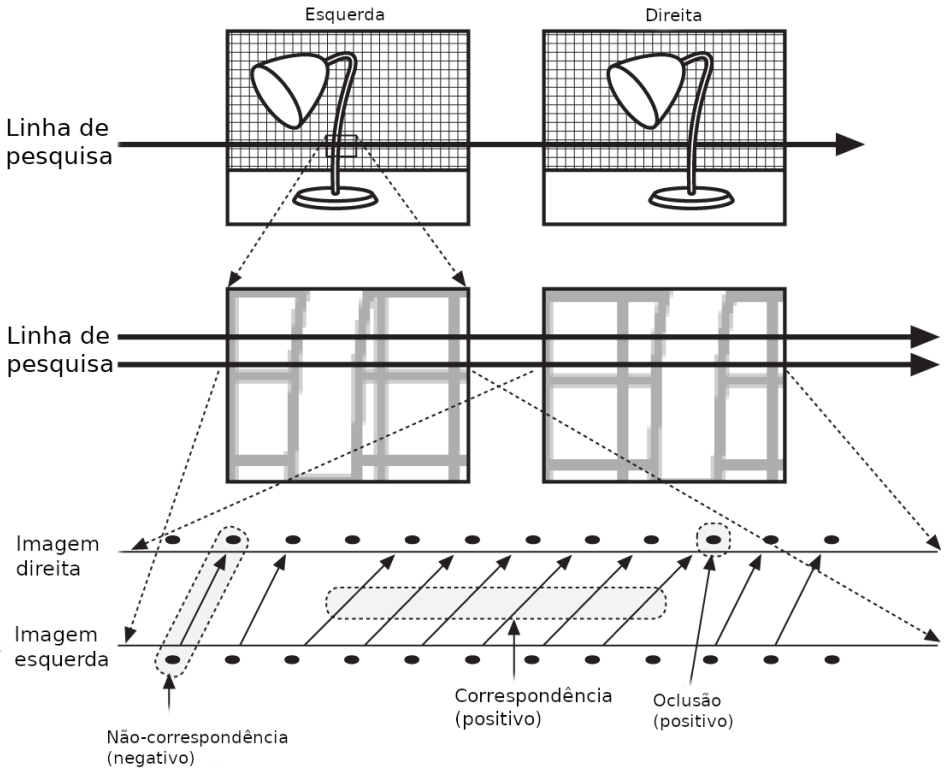
Figura 12 – Algoritmo de correspondência por blocos



Fonte: (KAEHLER; BRADSKI, 2016) (Adaptado).

A correspondência dentro do horóptero tem uma restrição embutida, chamada de restrição de ordem, que simplesmente afirma que a ordem de cada *feature* não pode mudar da vista da esquerda para a direita. Podem haver *features* ausentes - onde, devido à oclusão e ruído, algumas *features* encontrados à esquerda não podem ser encontrados à direita - mas a ordem das *features* encontrados permanece a mesma. O procedimento é ilustrado na Figura 13. Correspondências diretas e oclusões são contadas como positivas para o cálculo da correspondência.

Figura 13 – Correspondência de *features*



Fonte: (KAEHLER; BRADSKI, 2016) (Adaptado).

2.3.3 Correspondência por blocos semi-global

A correspondência por blocos semi-global consiste em um algoritmo superior à correspondência por blocos tradicional, pois utiliza técnicas de correspondência local e a aplicação de restrições de consistência ao longo de direções diferentes da linha horizontal (epipolar). Em um nível mais alto, os efeitos dessas adições fornecem robustez muito maior à iluminação e outras variações entre as imagens esquerda e direita, e para ajudar a eliminar erros ao impor restrições geométricas mais fortes em toda a imagem. Assim como a correspondência por blocos, a correspondência semi-global opera em pares de imagens estéreo retificadas e não distorcidas.

O elemento-chave do algoritmo é atribuir um custo a cada *pixel* para qualquer disparidade possível. Essencialmente, isso é análogo ao que é feito na correspondência por blocos, mas existem algumas novidades. A primeira é que utiliza-se algumas métricas de subpixel mais otimizadas para comparar *pixels*, em vez da simples diferença absoluta. A segunda é que uma suposição de continuidade de disparidade muito é inserida (*pixels* vizinhos provavelmente têm a mesma disparidade) e, ao mesmo tempo, utiliza-se um tamanho de bloco muito menor (às vezes 3x3 ou 5×5) em vez de usar janelas maiores e totalmente independentes (na correspondência por blocos, utiliza-se janelas 11×11 ou

maiores). Isso possibilita que o algoritmo saiba se comportar de maneira muito mais eficiente e precisa em situações de descontinuidade de bordas de objetos.

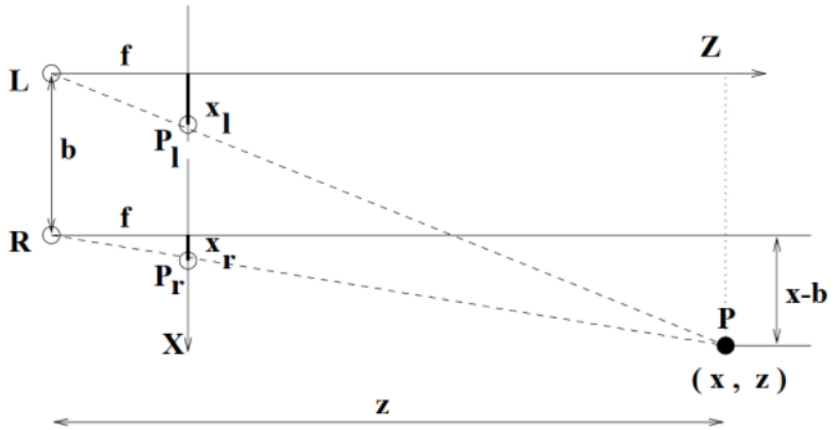
2.4 Visão computacional estéreo

Ao combinar as técnicas descritas até aqui, pode-se finalmente entender o conceito de visão computacional estéreo. Na visão estéreo tradicional, duas câmeras, deslocadas horizontalmente uma da outra, são usadas para obter duas imagens ligeiramente diferentes em uma cena, de maneira semelhante à visão binocular humana. Após o processo de calibração e retificação, coletam-se capturas de imagem de cada câmera ao mesmo instante. Ao comparar essas duas imagens utilizando as técnicas de correspondência anteriormente descritas, as informações de profundidade relativa podem ser obtidas na forma de um mapa de disparidade, que codifica a diferença nas coordenadas horizontais dos pontos da imagem correspondentes. Os valores neste mapa de disparidade são inversamente proporcionais à profundidade da cena na localização do *pixel* correspondente.

2.4.1 Percepção de profundidade utilizando câmeras estéreo

Apenas geometria e álgebra simples são necessárias para entender como os pontos 3D podem ser localizados no espaço usando um sensor estéreo, conforme mostrado na Figura 14. Suponha que duas câmeras, ou olhos, estejam cuidadosamente alinhados de modo que seus eixos X sejam colineares e seus eixos Y e Z sejam paralelos. O eixo Y é perpendicular à página e não é utilizado nas derivações. A origem (ou centro de projeção) da câmera direita é deslocada por b, que é a linha de base do sistema estéreo. O sistema observa algum ponto do objeto P no ponto esquerdo da imagem P_L e no ponto direito da imagem P_R. Geometricamente, o ponto P deve estar localizado na intersecção do raio LPL e do raio RPR.

Figura 14 – Geometria utilizada para cálculo da profundidade



Fonte: (SHAPIRO, 1992)

Utilizando semelhança de triângulos, obtém-se as Equações 15, 16 e 17:

$$\frac{z}{f} = \frac{x}{x_l} \tag{15}$$

$$\frac{z}{f} = \frac{x-b}{x_r} \tag{16}$$

$$\frac{z}{f} = \frac{y}{y_l} = \frac{y}{y_r} \tag{17}$$

Como $y_r = y_l$ devido ao fato que as câmeras estão posicionadas no mesmo plano horizontal, é simples obter as coordenadas do ponto de interesse no mundo real, utilizando as Equações 18, 19, 20, onde d é a disparidade, f é a distância focal obtida na resolução da matriz de projeção, b é a distância entre as câmeras, x_l, y_l, x_r, y_r são as coordenadas do ponto nas imagens capturadas pela câmera esquerda e direita, respectivamente, e z, x, e y são as coordenadas do ponto no mundo real, em metros:

$$z = \frac{fb}{x_l - x_r} = \frac{fb}{d} \tag{18}$$

$$x = x_l\frac{z}{f} = b + x_r\frac{z}{f} \tag{19}$$

$$y = y_l\frac{z}{f} = y_r\frac{z}{f} \tag{20}$$

Ao resolver as equações de x, y e z para o ponto P, reintroduz-se o conceito de disparidade (anteriormente citado para explicar a estereopsia humana), aqui definida como d, que é a diferença entre as coordenadas de imagem x_l e x_r nas imagens esquerda e direita, respectivamente, conforme a Equação 21. A solução de tais equações acaba por conseguir localizar completamente o ponto P no espaço 3D.

$$d = X_r - X_l \tag{21}$$

Segundo (SHAPIRO, 1992), disparidade refere-se à diferença da localização na imagem de um mesmo ponto 3D quando projetado sobre perspectiva de duas câmeras diferentes.

É possível notar que conforme a disparidade tende à zero, a distância tende a infinito, pois a câmera esquerda estaria localizada exatamente sobre a câmera direita, e este voltaria a ser o caso de apenas uma câmera, onde não é possível calcular a distância utilizando a disparidade.

A parte mais difícil de um sistema de visão estéreo não é o cálculo da distância, mas sim a determinação das correspondências entre os pontos utilizados neste cálculo. Na Figura

14, foi utilizado um mesmo ponto P para ilustração da matemática envolvida no cálculo da disparidade. Em uma imagem real, pode-se tornar extremamente difícil identificar o mesmo ponto P nas duas imagens geradas, pois um mesmo ponto possui diversas possíveis correspondências. Se alguma correspondência estiver incorreta, ela produzirá distâncias incorretas, que podem estar um pouco erradas ou totalmente erradas.

Na Figura 15, tal conceito torna-se mais claro. Seria consideravelmente difícil identificar um mesmo ponto exato em uma captura de uma câmera em posição diferente, tendo em vista as inúmeras semelhanças entre as texturas do campo de milho.

Figura 15 – Exemplo de uma imagem onde a correspondência de semelhanças seria difícil

Fonte: (SHAPIRO, 1992)

3 Metodologia

Conforme visto anteriormente, o problema da estimativa de profundidade utilizando câmeras estéreo é um problema relativamente complexo, onde diversos paradigmas devem ser considerados. São eles:

1. Calibração das câmeras utilizadas, conforme descrito na fundamentação teórica, de forma a se obter a matriz de projeção do sistema de coordenadas real para o sistema de coordenadas da imagem;

2. Remoção das distorções das lentes, de forma a possibilitar o uso de algoritmos de correspondência que utilizem restrição epipolar;

3. Realizar a correspondência entre os pontos de interesse da imagem obtida pela câmera esquerda e a imagem obtida pela câmera direita;

4. Utilizar a matriz de projeção e a correspondência entre os pontos para encontrar a disparidade do ponto de interesse, e com ela, identificar a distância entre o plano das câmeras e o ponto de interesse.

3.1 OpenCV

OpenCV é uma biblioteca de visão computacional de código aberto (BRADSKI, 2000), inicialmente criada por Gary Bradski da *Intel Corporation*, mas que atualmente recebe contribuições de centenas de desenvolvedores ao redor do mundo. A biblioteca é escrita em *C* e *C++* e possui compatibilidade para *Python* e outras linguagens. Um dos objetivos do *OpenCV* é fornecer uma infraestrutura de visão computacional simples de usar e que ajude as pessoas a criar aplicativos de visão computacional de maneira rápida e simples. A biblioteca *OpenCV* contém mais de 500 funções que abrangem muitas áreas da visão computacional, incluindo inspeção de produtos de fábrica, imagens médicas, segurança, interface do usuário, calibração de câmeras, visão estéreo e robótica.

Por tais motivos, a biblioteca foi escolhida como principal meio de aquisição de dados, calibração, retificação e cálculo da disparidade de cada imagem, reduzindo o escopo para a fundamentação e aplicação prática dos algoritmos discutidos na seção 2, removendo a necessidade da implementação matemática dos inúmeros cálculos descritos.

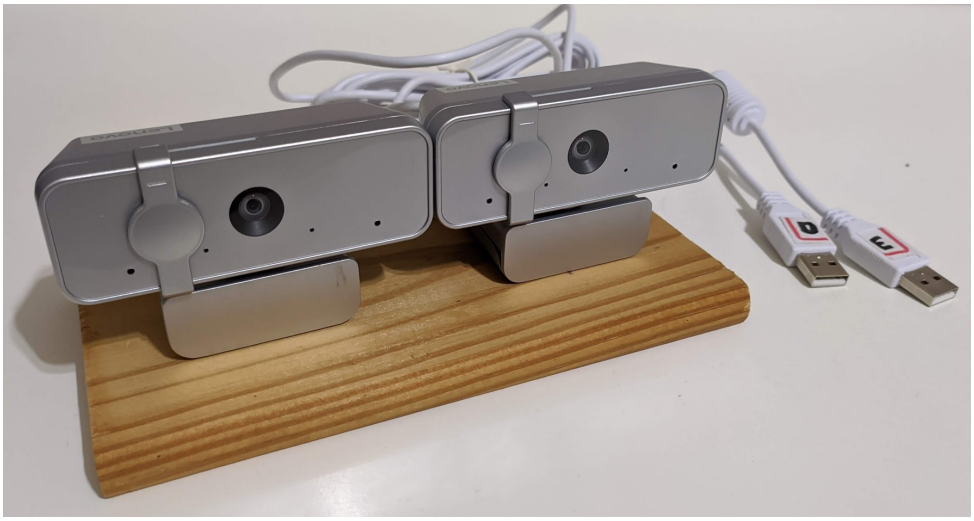
Assim, desenvolveu-se um código em *Python* 3.9, utilizando a biblioteca *OpenCV* 4.5.4.58.

3.2 Sistema de câmeras estéreo

O sistema de câmeras estéreo foi construído utilizando duas câmeras *Lenovo* modelo FHD300 (LENOVO, 2019), idênticas. Cada câmera possui um sensor CMOS de 2 megapixels, resolução 1920 x 1080, campo de visão de 95 graus. As dimensões físicas do produto são: 9 cm de largura, 6.2 cm de altura, e 4.6 cm de profundidade.

As câmeras foram fixadas utilizando uma superfície de madeira, através de um parafuso formato universal de tripés de câmera, já que o fabricante incluiu tal suporte. Como as câmeras foram encostadas uma à outra, e a lente está localizada exatamente ao centro do produto, a distância entre cada lente no sistema estéreo projetado ficou permanentemente definida como 9cm. O sistema utilizado está disposto na Figura 16.

Figura 16 – Sistema de câmeras estéreo utilizado.

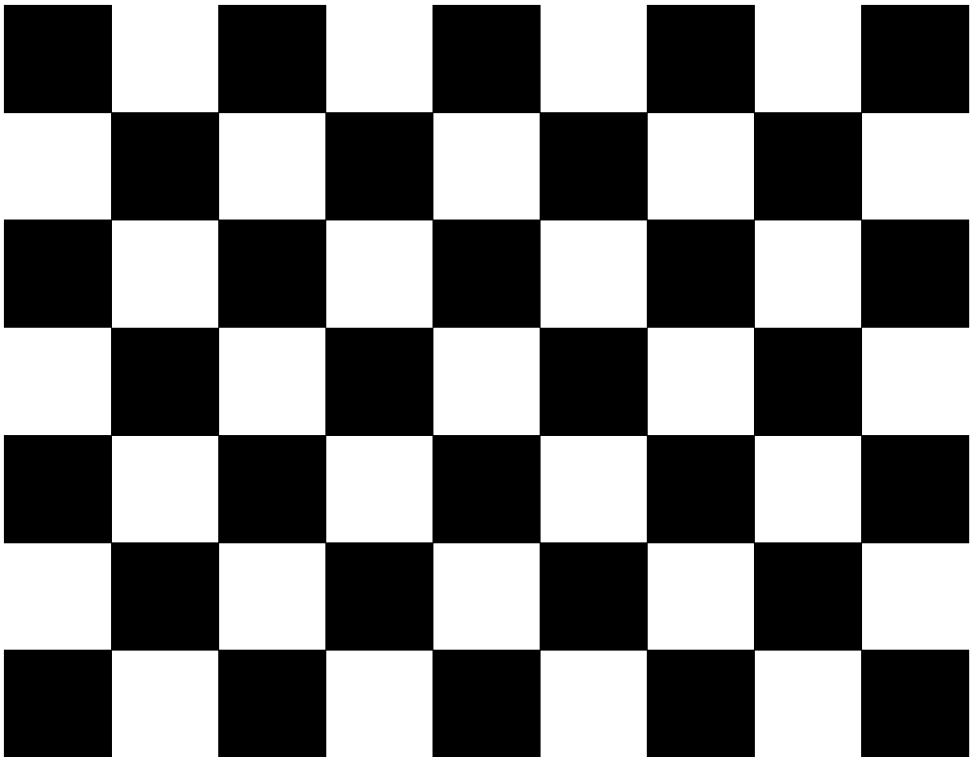


Fonte: O autor (2021).

3.3 Calibração individual

Para a etapa de calibração, pôs-se em prática a descrição teórica do capítulo 2.2.5. Para isso, foi escolhido como objeto um *tabuleiro de xadrez* com nove colunas e sete linhas, e quadrados de 25mm. O mesmo está disposto na Figura 17. Foram feitas 250 capturas em cada câmera, com o tabuleiro sendo movido entre cada captura de forma a se obter o mapeamento em todos os *pixels* da imagem, totalizando assim 500 capturas.

Figura 17 – Tabuleiro de calibração utilizado.



Fonte: O autor (2021).

A biblioteca *OpenCV* possui o conveniente método *findChessboardCorners*, demonstrado abaixo. Este método aceita quatro argumentos. O primeiro consiste em uma imagem contendo um tabuleiro de xadrez. O segundo argumento, indica quantos cantos internos existem em cada linha e coluna do tabuleiro. O próximo argumento é a lista passada por referência onde as localizações dos cantos serão registradas. O argumento final pode ser opcionalmente usado para implementar uma ou mais etapas de filtragem adicionais para ajudar a encontrar os cantos no tabuleiro de xadrez. A implementação utilizada está disposta abaixo:

```
1    # Encontra os cantos do tabuleiro. GrayR e GrayL são os frames
     ↪ capturados.
2    retR, cornersR = cv2.findChessboardCorners(grayR, (patternX,
     ↪ patternY), None, chess_flags)
3    retL, cornersL = cv2.findChessboardCorners(grayL, (patternX,
     ↪ patternY), None, chess_flags)
4
5    # Refinar os cantos para precisão subpixel
```

```
6       corners_sp_L = cv2.cornerSubPix(grayL, cornersL, (11, 11), (-1, -1),
        ↪  termination_criteria_subpix)
7       corners_sp_R = cv2.cornerSubPix(grayR, cornersR, (11, 11), (-1, -1),
        ↪  termination_criteria_subpix)
```

Após identificados os pontos correspondentes aos cantos dos quadrados do tabuleiro, utilizou-se o método *calibrateCamera* disposto abaixo para o cálculo da matriz de rotação e vetor de translação da câmera. Este método tem como argumentos, em ordem, os pontos dos cantos previamente encontrados, o tamanho da imagem, e o critério de parada do algoritmo. A função retorna, em ordem, o valor eficaz da calibração, a matriz de calibração, os coeficientes da câmera, a matriz de rotação e o vetor de translação.

```
1       # Calibra câmera esquerda
2       rms_int_L, mtxL, distL, rvecsL, tvecsL = cv2.calibrateCamera(
        ↪  objpoints_left_only, imgpoints_left_only, grayL.shape[::-1], None,
        ↪  None, criteria=termination_criteria_intrinsic)
3
4       # Calibra câmera direita
5       rms_int_R, mtxR, distR, rvecsR, tvecsR = cv2.calibrateCamera(
        ↪  objpoints_right_only, imgpoints_right_only, grayR.shape[::-1],
        ↪  None, None, criteria=termination_criteria_intrinsic)
```

3.4 Retificação individual

Na retificação, utilizou-se a teoria descrita no capítulo 2.2.6 em combinação com o método *undistort* da biblioteca *OpenCV*. O método possui como retorno a imagem retificada, e como parâmetros a imagem a ser retificada, a matriz de calibração e os coeficientes de distorção calculados previamente.

```
1       # Remove distorções da captura de cada câmera utilizando a matriz de
        ↪  calibração
2       undistortedL = cv2.undistort(frameL, mtxL, distL, None, None)
3       undistortedR = cv2.undistort(frameR, mtxR, distR, None, None)
```

3.5 Calibração estéreo

Em seguida, a calibração estéreo foi realizada com base no capítulo 2.2.5.1, e utilizando o método *stereoCalibrate*. Este método combina as matrizes de ambas as câmeras para calcular os parâmetros extrínsecos das câmeras, e finalmente obter-se as posições relativas entre a imagem capturada pela câmera esquerda e a câmera direita. Como

argumentos, o método aceita os pontos da calibração realizada com o objeto tabuleiro de xadrez, as matrizes de calibração e coeficientes de distorção da câmera esquerda e direita, e o tamanho da imagem. Como retorno, o método disponibiliza o valor eficaz da calibração, a matriz de calibração extrínseca de cada câmera, os coeficientes de cada câmera, e os vetores de rotação e translação.

```
1     # Realiza a calibração estéreo e cálculo dos parâmetros extrínsecos
2     rms_stereo, camera_matrix_l, dist_coeffs_l, camera_matrix_r,
      ↪ dist_coeffs_r, R, T, E, F) = cv2.stereoCalibrate(objpoints_pairs,
      ↪ imgpoints_left_paired, imgpoints_right_paired, mtxL, distL, mtxR,
      ↪ distR, grayL.shape[::-1],
      ↪ criteria=termination_criteria_extrinsics, flags=0
```

3.6 Retificação estéreo

Em seguida, pode-se realizar a retificação e mapeamento das imagens capturadas e já retificadas das câmeras esquerda e direita, de forma a alinhar as capturas e deixá-las prontas para a utilização de algoritmos de correspondência que se beneficiem da restrição epipolar, conforme capítulo 2.3.1. Para tanto, foram utilizados os métodos *stereoRectify*, que irá computar as matrizes de retificação que ainda necessitam ser calculadas, e *initUndistortRectifyMap*, que finalmente irá realizar o mapeamento da imagem esquerda para a imagem direita.

```
1     # Retifica esquerda -> direita
2     RL, RR, PL, PR, Q, _, _ = cv2.stereoRectify(camera_matrix_l,
      ↪ dist_coeffs_l, camera_matrix_r, dist_coeffs_r, grayL.shape[::-1],
      ↪ R, T, alpha=-1)
3
4     # Produz o mapa de remoção de distorções para as câmeras
5     mapL1, mapL2 = cv2.initUndistortRectifyMap(camera_matrix_l,
      ↪ dist_coeffs_l, RL, PL, grayL.shape[::-1], cv2.CV_32FC1)
6     mapR1, mapR2 = cv2.initUndistortRectifyMap( camera_matrix_r,
      ↪ dist_coeffs_r, RR, PR, grayR.shape[::-1], cv2.CV_32FC1)
7
8     # Utiliza o mapa criado para alinhar as imagens
9     undistorted_rectifiedL = cv2.remap(grayL, mapL1, mapL2,
      ↪ cv2.INTER_LINEAR)
10    undistorted_rectifiedR = cv2.remap(grayR, mapR1, mapR2,
      ↪ cv2.INTER_LINEAR)
```

3.7 Correspondência

Para a realização da correspondência de objetos entre a imagem esquerda e a imagem direita, escolheu-se o método de correspondência por blocos semi-global, descrito no capítulo 2.3.3. Para tanto, utilizou-se o método *StereoSGBM_create*, aliado à um filtro de mínimos quadrados ponderados, para se obter um mapa de disparidades cujos contornos em regiões de translação de disparidades não sofressem de aberrações. Os parâmetros de configuração demonstrados abaixo são resultados de diversas tentativas de calibração dos parâmetros, e foram os escolhidos como finais pois apresentaram os melhores resultados. Tal metodologia "erro/acerto" é a recomendada pelo *OpenCV* para determinação dos melhores parâmetros para cada tipo de aplicação.

```
1    # Cria o correspondedor estéreo por blocos semi-global
2    left_matcher = cv2.StereoSGBM_create(minDisparity=0,
     ↪  numDisparities=6*16, blockSize=7, P1=8*3*7**2, P2=32 * 3 * 7**2,
     ↪  disp12MaxDiff=1, uniquenessRatio=16, speckleWindowSize=50,
     ↪  speckleRange=2, preFilterCap=63,
     ↪  mode=cv2.STEREO_SGBM_MODE_SGBM_3WAY)
3    right_matcher = cv2.ximgproc.createRightMatcher(left_matcher)
4
5    # Cria o filtro de minimos quadrados ponderados
6    wls_filter =
     ↪  cv2.ximgproc.createDisparityWLSFilter(matcher_left=left_matcher)
7    wls_filter.setLambda(80000)
8    wls_filter.setSigmaColor(1.2)
```

3.8 Mapa de disparidade

Finalmente, é possível obter o mapa de disparidade utilizando o método *compute*, que ordenará o correspondedor semi-global criado anteriormente a calcular a disparidade para todos os pontos da imagem de destino. Em seguida, o mapa passa pelo filtro de mínimos quadrados ponderados para a remoção de regiões onde a transição de profundidade resulta em aberrações.

```
1    # Computa a disparidade a partir das imagens retificadas
2    disparity_l = left_matcher.compute(undistorted_rectifiedL,
     ↪  undistorted_rectifiedR)
3    disparity_r = right_matcher.compute(undistorted_rectifiedR,
     ↪  undistorted_rectifiedL)
```

```
4       # Pondera o filtro
5       disparity = wls_filter.filter(displ, undistorted_rectifiedL, None,
        ↪  dispr)
```

A disparidade aqui resultante é composta por um vetor de pontos, onde cada ponto corresponde a disparidade de um pixel. Este vetor pode ser diretamente interpretado como uma imagem em escala de cinza, onde tem-se o mapa de disparidades finalmente calculado. Para facilitar ainda mais a interpretação do usuário, é possível aplicar uma coloração da saída deste mapa, de forma que objetos próximos (com alta disparidade) terão cor vermelha e objetos afastados (com baixa disparidade) terão cor azulada, com o seguinte método:

```
1 disparity_colour_mapped = cv2.applyColorMap((disparity_scaled * (256. /
  ↪  max_disparity)).astype(np.uint8), cv2.COLORMAP_JET)
```

4 Resultados e Discussões

Seguindo a ordem de implementação descrita no capítulo 3, foram desenvolvidos os métodos responsáveis pela calibração, retificação, e cálculo do mapa de disparidades. Os resultados obtidos em tais operações serão aqui descritos.

4.1 Matrizes de calibração e retificação

A calibração das câmeras conforme a seção 2.2.5, resultou nas matrizes descritas abaixo. A Equação 22 corresponde à matriz de calibração intrínseca K para a câmera esquerda, e a Equação 23 corresponde à matriz de calibração intrínseca K para a câmera direita.

$$K_l = \begin{bmatrix} 349.89692 & 0 & 295.88529 \\ 0 & 348.55842 & 268.84223 \\ 0 & 0 & 1 \end{bmatrix} \tag{22}$$

$$K_r = \begin{bmatrix} 354.00376 & 0 & 342.74042 \\ 0 & 352.00299 & 252.45960 \\ 0 & 0 & 1 \end{bmatrix} \tag{23}$$

O tempo de processamento do algoritmo para as 500 capturas foi de 120 minutos. O valor RMS da calibração individual de cada câmera foi 0.567167 para a câmera direita e 0.83194 para a câmera esquerda, e o erro médio em *pixels* calculado para a retificação obtida de cada câmera foi 0.13981 para a câmera esquerda e 0.16585 para a câmera direita. Tais valores indicam que o processo de calibração individual foi satisfatório.

Após a retificação das imagens capturadas utilizando tais matrizes, foi feita a calibração estéreo do sistema, de maneira a mapear a rotação e translação entre a câmera esquerda e a câmera direita, conforme a seção 2.2.5.1. Com isso, foi obtido a matriz de rotação R descrita na Equação 24 e o vetor de translação T descrito na Equação 25.

$$R = \begin{bmatrix} 998.318 \cdot 10^{-3} & 15.331 \cdot 10^{-3} & -55.899 \cdot 10^{-3} \\ -15.370 \cdot 10^{-3} & 999.881 \cdot 10^{-3} & -0.260 \cdot 10^{-3} \\ 55.889 \cdot 10^{-3} & 1.119 \cdot 10^{-3} & 998.436 \cdot 10^{-3} \end{bmatrix} \tag{24}$$

$$T = \begin{bmatrix} -89.79588591 \\ 2.4717301 \\ 1.9620371 \end{bmatrix} \tag{25}$$

Os coeficientes de distorção obtidos para a câmera esquerda estão dispostos na Equação 26, e para a câmera direita na Equação 27.

$$D_l = \Big[-0.14142 0.03664 - 0.00038 0.00211 - 0.01014\Big] \tag{26}$$

$$D_r = \Big[-0.13640 0.02080 0.00026 - 0.00089 - 0.00123\Big] \tag{27}$$

O valor RMS obtido para a calibração estéreo foi 0.78944, indicando que a calibração total do sistema foi satisfatória.

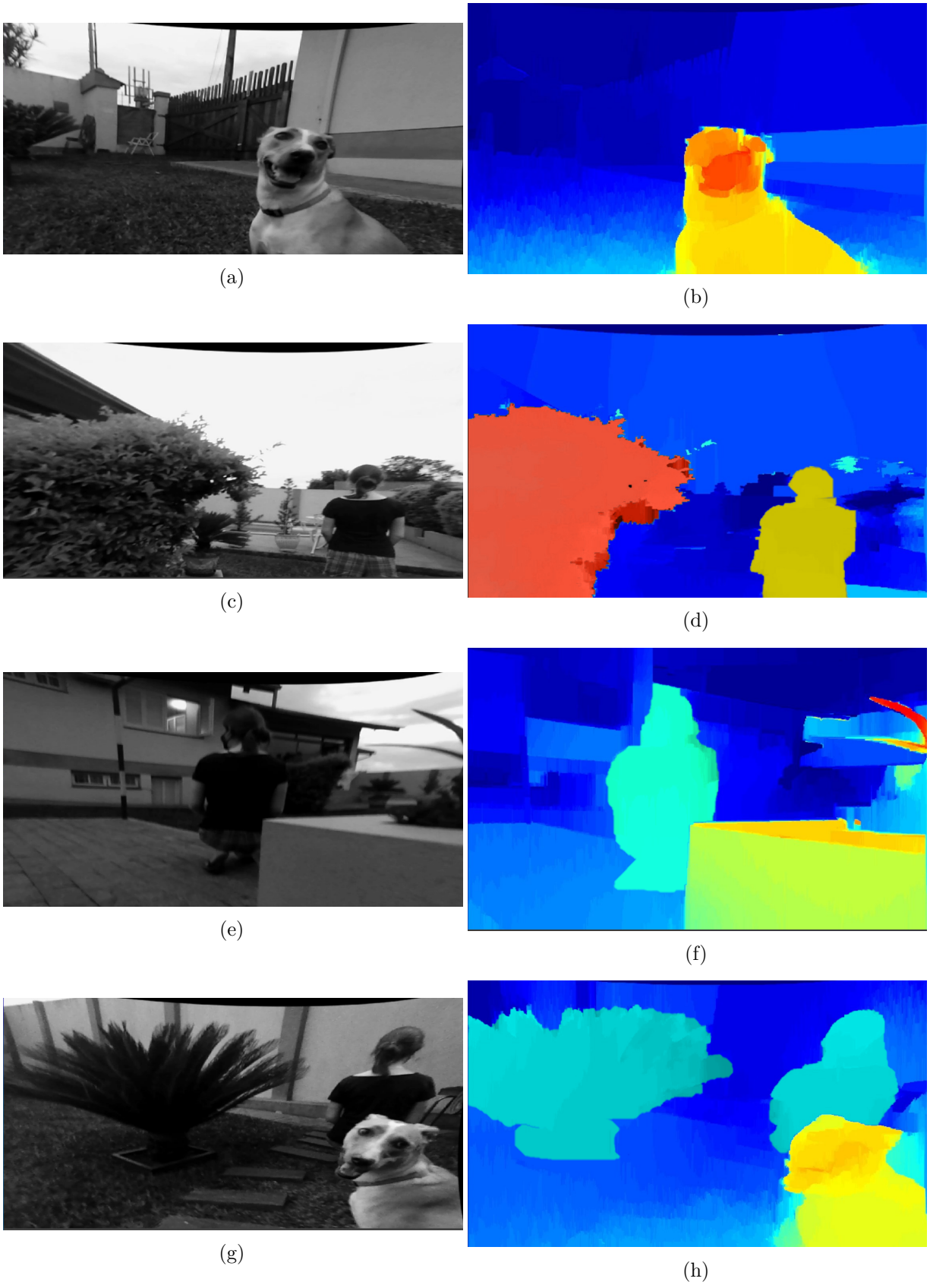
4.2 Mapa de disparidades

Através da metodologia apresentada no capítulo 3.8, foi possível gerar o mapa de disparidades do sistema em tempo real. Na Figura 18 estão dispostas cinco capturas de tela do mapa de disparidade gerado, em diferentes situações de interesse.

O esquema de cor adotado permite identificar e diferenciar as distâncias dos objetos presentes na cena, onde objetos próximos terão tonalidade vermelha, passando então para uma tonalidade amarela em casos que os objetos estejam à uma distância média, e azul para casos onde os objetos estejam longe do sistema. É possível identificar com grande precisão visual a mudança de distância, inclusive para distâncias pequenas entre dois objetos, pois a gama de tonalidades aplicada possui alta sensibilidade para o escopo de disparidades resultante.

Na Figura 18, as imagens à esquerda *(a, c, e)* representam a entrada de uma das câmeras, em escala de cinza. Já as imagens à direita representam a saída do mapa de disparidade gerado por cada imagem correspondente *(b, d, f)*. No par *ab*, é possível notar claramente que o cachorro estava próximo à câmera, e seu focinho ganhou destaque vermelho pela proximidade maior que o resto do seu corpo. Nos pares *cd* e *ef*, é possível notar que a pessoa está mais distante que o arbusto e o vaso, respectivamente. Por fim, no par *gh*, fica claro que o cachorro estava à frente da pessoa e da planta, enquanto estas tinham a mesma distância do sistema, ficando assim com a mesma cor.

Figura 18 – Mapas de disparidade gerados utilizando o sistema proposto



(a) (b) (c) (d) (e) (f) (g) (h)

Fonte: O autor.

4.3 *Performance* do sistema proposto

Para o processamento do mapa de disparidades foi utilizando um computador pessoal com as seguintes especificações: CPU Intel Core i7-7700HQ @ 2.8GHz, 16GB RAM. Não foi utilizado processamento por placa de vídeo dedicada.

O tempo de processamento do mapa de disparidades foi inferido através do cálculo da quantidade de quadros por segundo da saída do sistema. Ao utilizar as metodologias descritas no capítulo 3, obteve-se 32 quadros por segundo na saída do sistema de visualização do mapa de disparidade.

Segundo (JAVADI; DAHL; PETTERSSON, 2019), para que um veículo tenha tempo de processar os dados e tomar decisões baseadas na entrada (sejam elas referentes ao controle do freio, acelerador ou voltante), uma câmera operando a 30 quadros por segundo geraria entradas suficientes para que sejam detectados objetos com velocidade de até 70km/h com erro de 1.77%. Logo, é possível notar que o sistema proposto teria precisão suficiente para operação em vias urbanas, e que um sistema com mais quadros por segundo lhe concederia capacidade de utilização em vias interurbanas.

4.4 Avaliação do sistema em um veículo em movimento

Tendo em vista que a aplicação inicial do sistema provou-se satisfatória durante os testes realizados no ambiente de desenvolvimento, obtendo bons resultados na aquisição e discernimento do mapa de disparidade em objetos de interesse utilizados como exemplo, o sistema proposto foi então utilizado em um veículo (não autônomo) real.

Para tanto, as câmeras (montadas em seu suporte de madeira - vide Figura 19) foram fixadas sobre o painel do veiculo, e o sistema de aquisição do mapa de disparidades foi então acionado em duas circunstâncias: Primeiro, com o veículo parado em um estacionamento, e em seguida, com o veículo em movimento. Notou-se que, no estado em que o veículo encontra-se parado no estacionamento, o sistema comporta-se conforme o esperado, detectando obstáculos presentes à sua frente.

Figura 19 – Sistema de câmeras estéreo montado sobre o painel de um veículo.

Fonte: O autor (2021).

Na Figura 20, é possível discernir uma pessoa andando na calçada, além da silhueta de uma casa atrás dela. Nota-se claramente a diferença na distância de cada objeto e seus arredores: O veículo estava estacionado em frente à faixa de segurança, e a pessoa, como está próxima, tem cor vermelha.

Figura 20 – Uso do sistema em um veículo: Pedestre na faixa

Fonte: O autor (2021).

A Figura 21 ilustra um caso peculiar que ocorreu durante a aquisição dos dados de exemplo, em que um pássaro passou na frente das câmeras, e mesmo com o veículo em movimento e o pássaro ser um objeto de tamanho relativamente pequeno, o sistema soube classificar a sua distância em relação aos arredores.

Figura 21 – Uso do sistema em um veículo: Pássaro

Fonte: O autor (2021).

Na Figura 22, nota-se a eficiência do método em identificar o veículo estacionado à frente do sistema *(a)*, e a respectiva mudança de tonalidade conforme o sistema se aproximava do mesmo *(b)* e *(c)*.

Figura 22 – Uso do sistema em um veículo: Estacionamento

(a)

(b)

(c)

Fonte: O autor (2021).

Para a situação em que o veículo encontra-se em movimento na rodovia, o sistema também demonstrou ser eficiente no discernimento das distâncias dos objetos ao redor do veículo, tendo sido capaz de distinguir outros veículos, e mudar a resposta visual do sistema conforme a distância dos mesmos ao sistema se reduzia. Exemplo de tal situação pode ser verificado na Figura 23, onde o veículo que estava na pista contrária foi identificado e sua cor mudou conforme o mesmo se aproximava do sistema.

Figura 23 – Uso do sistema em um veículo: Trânsito

(a)

(b)

Fonte: O autor (2021).

Notou-se, porém, que o sistema apresentava erros na geração do mapa de disparidade na região referente ao céu, já que estava nublado, e conforme descrito anteriormente, as nuvens constituem um padrão de textura de difícil interpretação para o algoritmo de correspondência. Esta limitação, portanto, já era esperada, e foi solucionada limitando-se a área de captura das câmeras para não incluir o céu (já que de qualquer forma, esta área é irrelevante para um veículo terrestre).

5 Conclusão

O automóvel foi uma das maiores conquistas tecnológicas da humanidade, e é natural que seja necessário desenvolver sistemas que proporcionem conforto, facilitem seu uso e aumentem a segurança dos ocupantes e também dos pedestres.

Neste trabalho, avaliou-se a implementação de uma técnica simples de visão computacional, que ao ser combinada com um sistema que interprete os dados da saída da técnica proposta, poderia ser útil na tomada de decisões de um sistema de controle veicular autônomo.

A técnica proposta mostrou-se satisfatória, tendo sido eficaz na predição das distâncias entre o sistema de captura estéreo e seus arredores ou entre este e objetos de interesse através do uso da disparidade entre as imagens capturadas.

O sistema demonstrou baixas taxas de erro entre as capturas retificadas, indicando precisão nos dados coletados, e também apresentou imagens de saída extremamente satisfatórias, contendo o mapa de disparidade em cores com ótimo discernimento dos objetos e suas respectivas distâncias ao sistema de captura, com baixo tempo de processamento dos dados.

Os testes em um veículo real também foram satisfatórios, tendo-se obtido bons resultados nas capturas realizadas em tempo do sistema, onde o mesmo foi capaz de inferir a distância tanto com o veículo parado quanto com o veículo em movimento.

Apesar de o escopo do projeto se limitar à geração do sistema de cálculo e exibição do mapa de disparidade utilizando câmeras estéreo, o sistema se mostrou preciso e suficientemente rápido para aplicação do mesmo em um veículo real, necessitando apenas a criação do sistema de controle e combinação desta técnica com outras técnicas de redundância (fora do escopo do projeto).

Como trabalhos futuros, recomenda-se a combinação do sistema proposto com uma rede de identificação de objetos, permitindo-se assim que a distância exata em metros seja descoberta pelo algoritmo. Recomenda-se também a implementação de um sistema que utilize o sistema de câmeras estéreo combinado com o treinamento de uma inteligência artificial para reconhecimento da disparidade, podendo-se assim ter um possível ganho de performance, e também a remoção de eventuais aberrações causadas por disparidades calculadas erroneamente devido à composição da superfície sendo analisada.

Referências Bibliográficas

ARCHIVES, T. N. *The Locomotive Act of 1865*. 1865. Disponível em: <https://archive.org/details/statutesunitedk30britgoog/page/n246/mode/2up?view=theater>.

BRADSKI, G. The opencv library. *Dr. Dobb's Journal of Software Tools*, 2000. Disponível em: <https://opencv.org/>.

CROMER, G.; CROMER, O. *Automobile*. Encyclopædia Britannica, inc., 1998. Disponível em: <https://www.britannica.com/technology/automobile>.

CYGANEK, B.; SIEBERT, J. *An Introduction to 3D Computer Vision Techniques and Algorithms*. [S.l.]: Wiley, 2011. ISBN 9781119964476.

FORSYTH, D.; PONCE, J. *Computer Vision: A Modern Approach*. [S.l.]: Pearson India Education Services Pvt. Limited, 2015. ISBN 9789332550117.

HARTLEY, R.; ZISSERMAN, A. *Multiple View Geometry in Computer Vision*. [S.l.]: Cambridge University Press, 2003. (Cambridge books online). ISBN 9780521540513.

HOWARD, I.; ROGERS, B. *Perceiving in Depth, Volume 3: Other Mechanisms of Depth Perception*. [S.l.]: Oxford University Press, USA, 2012. (Oxford Psychology Series). ISBN 9780199764167.

JAVADI, S.; DAHL, M.; PETTERSSON, M. I. Vehicle speed measurement model for video-based systems. *Computers Electrical Engineering*, v. 76, p. 238–248, 2019. ISSN 0045-7906.

KAEHLER, A.; BRADSKI, G. *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*. [S.l.]: O'Reilly Media, 2016. ISBN 9781491937969.

LENOVO. *Webcam Lenovo 300 FHD*. 2019. Disponível em: <https://www.lenovo.com/br/pt/accessories-and-monitors/webcams-and-video/webcams/NET-BO-300-FHD-Webcam/p/GXC1B34793>.

SHAPIRO, L. *Computer Vision and Image Processing*. [S.l.]: Elsevier Science, 1992. ISBN 9780323141567.

SZELISKI, R. *Computer Vision: Algorithms and Applications*. [S.l.]: Springer London, 2010. (Texts in Computer Science). ISBN 9781848829350.

WHO, W. H. O. *Global status report on road safety*. 2018. Disponível em: <https://www.who.int/publications/i/item/9789241565684>.