

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE COMPUTAÇÃO**

Kaio Anselmo Pereira

**VISÃO ESTEREOSCÓPICA: OBTENÇÃO DO MAPA DE
PROFUNDIDADE DO AMBIENTE**

Araranguá

2017

Kaio Anselmo Pereira

**VISÃO ESTEREOSCÓPICA: OBTENÇÃO DO MAPA DE
PROFUNDIDADE DO AMBIENTE**

**Trabalho de Conclusão de
Curso submetido à Universi-
dade Federal de Santa Cata-
rina, como parte dos requisitos
necessários para a obtenção do
Grau de Bacharel em Engenha-
ria de Computação.**

**Orientador: Prof. Fábio Ro-
drigues de la Rocha, Dr.**

Araranguá, dezembro de 2017.

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Pereira, Kaio Anselmo

Visão estereoscópica: obtenção do mapa de profundidade do ambiente / Kaio Anselmo Pereira ; orientador, Fábio Rodrigues de La Rocha, 2017.
78 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Campus Araranguá,
Graduação em Engenharia de Computação, Araranguá, 2017.

Inclui referências.

1. Engenharia de Computação. I. La Rocha, Fábio Rodrigues de. II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Computação. III. Título.

Kaio Anselmo Pereira

**VISÃO ESTEREOSCÓPICA: OBTENÇÃO DO MAPA DE
PROFUNDIDADE DO AMBIENTE**

Este Trabalho de Conclusão de Curso foi julgado aprovado para a obtenção do Título de “Bacharel em Engenharia de Computação”, e aprovado em sua forma final pela Universidade Federal de Santa Catarina.

Araranguá, 07 de dezembro 2017.



Prof.^a. Dr.^a. Eliane Pozzebon
Coordenadora

Universidade Federal de Santa Catarina

Banca Examinadora:

Prof. Dr. Fábio Rodrigues de La Rocha
Universidade Federal de Santa Catarina

Prof. Dr. Anderson Luiz Fernandes Perez
Universidade Federal de Santa Catarina

Prof. Dr. Gustavo Mello Machado
Universidade Federal de Santa Catarina

Dedico este trabalho aos meus pais e irmãs. Vocês são fonte de inspiração para a realização de meus sonhos.

AGRADECIMENTOS

Gostaria de agradecer formalmente ao meu orientador, Prof. Dr. Fábio Rodrigues de la Rocha, por ter me orientado nesse trabalho com paciência, por passar seus conhecimentos a mim e ter me ajudado a chegar ao meu objetivo de concluir este trabalho, meu muito obrigado! Gostaria também de agradecer o Prof. Dr. Anderson Luiz Fernandes Perez pelas suas aulas, no qual sempre buscou extrair o melhor de seus alunos, principalmente durante a disciplina de Sistemas Operacionais Embarcados. Também agradecer a ele por ter me incentivado e auxiliado na abertura da EJEC - Empresa Júnior da Engenharia de Computação. Sem dúvidas foi a melhor experiência que pude ter durante meu período de graduação.

Sou grato a todos meus amigos que me acompanharam durante a graduação e aos que me ajudaram a abrir a EJEC e aos que passaram por ela. Ao Jônatas Negri Grandini, obrigado por tudo! Foram 5 anos de muita colaboração para atingirmos nosso objetivo. Ao Gabriel Bitencourt de Souza, obrigado por me acompanhar nesta jornada e pelos anos de amizade. Aos meus amigos Thiago Raulino Dal Pont e Alan Kunz Cechinel, por me ajudarem nos últimos semestres da graduação.

A minha família, pois sem eles não chegaria tão longe. Vocês são o motivo de eu ter atingido meus objetivos. Me incentivaram e me apoiaram em meus momentos mais difíceis. A minha irmã, Beatriz Anselmo Pereira por ter me auxiliado na conclusão deste trabalho.

E por fim, a todas as pessoas que direta e indiretamente fizeram parte da minha formação.

Não são as circunstâncias que causam resultados – são as pessoas

Jim Collins

RESUMO

Esta pesquisa apresenta um método para obtenção do mapa de profundidade de um ambiente. Por exemplo, para que um robô consiga se movimentar em um ambiente desconhecido de forma autônoma ele necessita saber informações do ambiente para realizar a tomada de decisão. Assim, foi desenvolvido um software capaz de obter o mapa de profundidade do ambiente em tempo hábil para aplicações em tempo real. O método utilizado para realizar isso envolve técnicas de visão estereoscópica, calibração de câmera, retificação de imagens, obtenção do mapa de disparidade e, por fim, a obtenção do mapa de profundidade. Os resultados apresentados são satisfatórios apresentando um erro médio de 1,1 centímetros à 2,50 metros de distância e 20,3 centímetros à 4,0 metros de distância. A pesquisa também propõe um conceito de algoritmo para obter o mapa de disparidade.

Palavras-chave: Visão Computacional, Visão Estereoscópica, Mapa de Disparidade, Mapa de Profundidade, OpenCV

ABSTRACT

This research presents one method to obtain depth maps of environments. For a robot to be able to move around an unknown environment autonomously, it needs the information from the environment to make decisions. Thus, in this research, we developed one software that maps the depth of an environment in real-time. The method used for this task involves technics of stereoscopic vision, using camera calibration, image rectification, obtaining the disparity map and, finally, obtaining the depth map. The presented results are satisfactory, with a average error of 1,1 centimeter to 2,50 meters of distance and 20,3 centimeters to 4,0 meters of distance. We also proposes an algorithm concept for obtaining the disparity map.

Keywords: Computer vision, Stereo Vision, Disparity Map, Depth Map, OpenCV

LISTA DE FIGURAS

Figura 1	Éstoque mundial de robôs industriais operacionais	26
Figura 2	Unidades vendidas de robôs de serviço 2016-2019	26
Figura 3	Sistema de Câmeras	31
Figura 4	Logitech HD Webcam C270	31
Figura 5	Distorção radial	33
Figura 6	Grade de círculos	34
Figura 7	Tabuleiro de Xadrez, ArUco e ChArUco	34
Figura 8	Demonstração da Rotação e Translação das Câmeras . .	35
Figura 9	Geometria Epipolar	36
Figura 10	Passos da Calibração e Retificação	36
Figura 11	Disparidade	37
Figura 12	Algoritmo SAD	38
Figura 13	Relação entre distância e disparidade	39
Figura 14	Área de captura	40
Figura 15	Padrão de Tabuleiro de Xadrez utilizado	42
Figura 16	Orientação do padrão para a obtenção da sequência de imagens	42
Figura 17	Exemplo de Aplicação da Equalização do Histograma . .	44
Figura 18	Exemplo de Aplicação do Alisamento Gaussiano	45
Figura 19	Gráfico Distância x Disparidade	47
Figura 20	Disparidade com Bloco 3x3 Sem Filtro	48
Figura 21	Disparidade com Bloco 51x51 Sem Filtro	48
Figura 22	Exemplificação do Algoritmo SAD	49
Figura 23	Conceito do novo algoritmo Imagem Esquerda	50
Figura 24	Conceito do novo algoritmo Imagem Direita	50
Figura 25	Cena para captura de dados	51
Figura 27	Matrizes de Resultados	52
Figura 26	Resultado da Aplicação da Calibração e Retificação . . .	53
Figura 28	Mapa de Disparidade apenas com Pré-Filtro	55
Figura 29	Mapa de Disparidade Com Pós-Filtro	55
Figura 30	Mapa de Profundidade	56
Figura 31	Mapa de Profundidade Colorido	57

Figura 32	Exemplificação da Aquisição de Dados.....	58
Figura 33	Disparidade Média Captura Ao Vivo	60
Figura 34	Disparidade Média Template Captura Ao Vivo	60
Figura 35	Distância Média Captura Ao Vivo.....	61
Figura 36	Disparidade Média Captura Estática.....	62
Figura 37	Disparidade Média Template Captura Estática	63
Figura 38	Distância Média Captura Estática.....	63

LISTA DE TABELAS

Tabela 1	Resultados Obtidos para Captura Ao Vivo.....	59
Tabela 2	Resultados obtidos para Captura Estática	62
Tabela 3	Resultados da disparidade do Algoritmo Proposto	64
Tabela 4	Resultados da distância do Algoritmo Proposto.....	64

LISTA DE QUADROS

1	Especificações da Câmera	32
2	Especificações do Notebook utilizado.....	32
3	Configurações da classe StereoBM	45
4	Configurações do Filtro DisparityWLSFilter	46
5	Especificações das Bibliotecas Utilizadas	78

LISTA DE ABREVIATURAS E SIGLAS

IRF	<i>International Federation of Robotics</i>	25
SONAR	<i>Sound Navigation And Ranging</i>	27
RADAR	<i>Radio Detection And Ranging</i>	27
LIDAR	<i>Light Detection And Ranging</i>	27
SAD	<i>Soma Absoluta das Diferenças</i>	31
Pixel	<i>Picture Element</i>	43
CPU	<i>Central Processing Unit</i>	45
GPU	<i>Graphics Processing Unit</i>	45
Hz	<i>Hertz</i>	56
SLAM	<i>Simultaneous localization and mapping</i>	69
OpenCV	<i>Open Source Computer Vision Library</i>	77
BSD	<i>Berkeley Software Distribution</i>	77
CUDA	<i>Compute Unified Device Architecture</i>	77

SUMÁRIO

1	INTRODUÇÃO	25
1.1	CONTEXTUALIZAÇÃO.....	25
1.2	PROBLEMÁTICA	27
1.3	JUSTIFICATIVA	28
1.4	OBJETIVOS	28
1.4.1	Geral	29
1.4.2	Específicos	29
1.5	METODOLOGIA	29
2	SISTEMA DE VISÃO ESTÉREO	31
2.1	O SISTEMA.....	31
2.2	CALIBRAÇÃO DE CÂMERA	32
2.2.1	Calibração Estéreo das Câmeras	34
2.3	RETIFICAÇÃO.....	35
2.4	DISPARIDADE	37
2.5	DISTÂNCIA.....	39
3	IMPLEMENTAÇÃO DO SISTEMA DE VISÃO ESTEREOSCÓPICA	41
3.1	CALIBRAÇÃO DAS CÂMERAS.....	41
3.1.1	Calibração de Câmera Estéreo e Retificação.....	43
3.2	MAPA DE DISPARIDADE.....	43
3.2.1	Pré-filtros	43
3.2.1.1	Equalização de Histograma	43
3.2.1.2	Alisamento Gaussiano.....	44
3.2.2	Obtenção do Mapa de Disparidade	45
3.2.3	Pós-Filtro	46
3.3	MAPA DE PROFUNDIDADE.....	46
3.4	ALGORITMO PROPOSTO	47
3.4.1	Descrição do Algoritmo Proposto	49
4	RESULTADOS	51
4.1	PRÉ-PROCESSAMENTO	52
4.2	PROCESSAMENTO	54
4.3	PÓS-PROCESSAMENTO.....	56
4.4	OBTENÇÃO DOS DADOS	57
4.5	ANÁLISE DOS RESULTADOS	58
4.5.1	Captura em Tempo Real	58
4.5.2	Captura Estática	61
4.6	RESULTADOS DO ALGORITMO PROPOSTO	64

5	CONSIDERAÇÕES FINAIS	67
5.1	DIFICULDADES	68
5.2	PROPOSTAS PARA TRABALHOS FUTUROS	68
	REFERÊNCIAS	71
	APÊNDICE A – Biblioteca <i>OpenCV</i>	77

1 INTRODUÇÃO

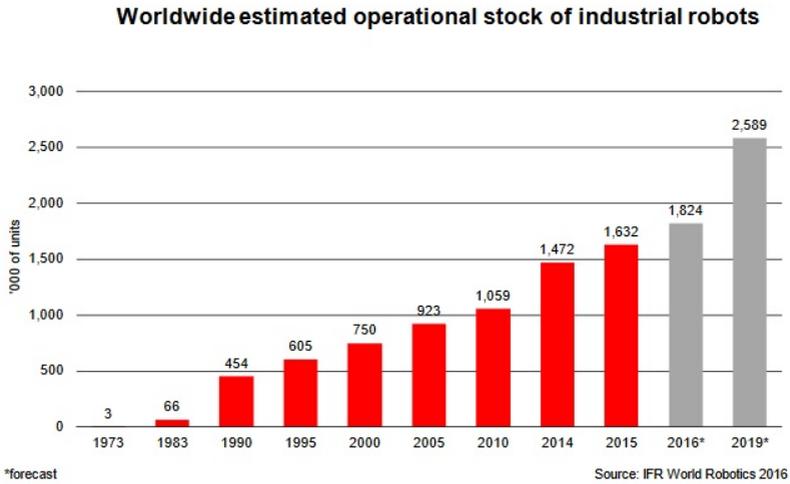
1.1 CONTEXTUALIZAÇÃO

A robótica móvel está e estará cada vez mais inserida na vida das pessoas, tanto de forma indireta, aplicada em serviços e produtos, quanto de forma direta, de uso pessoal e doméstico. Nos últimos anos, houve um crescimento no potencial dos sistemas robóticos. Silva (apud PEREIRA, 2003) classificou os robôs móveis presente no mercado mundial em cinco grandes áreas:

- Industrial: robôs que ajudam a locomover grandes quantidades de peças ou componentes pelo chão de fábrica.
- Serviços: área da robótica móvel autônoma como transporte, manipulação, limpeza, vigilância, etc.
- Pesquisa: robôs utilizados para área acadêmica, voltado para o desenvolvimento de novas tecnologias.
- Campo: exploração de ambientes desconhecidos, desestruturados e perigosos.
- Entretenimento: brinquedos, humanoides, etc.

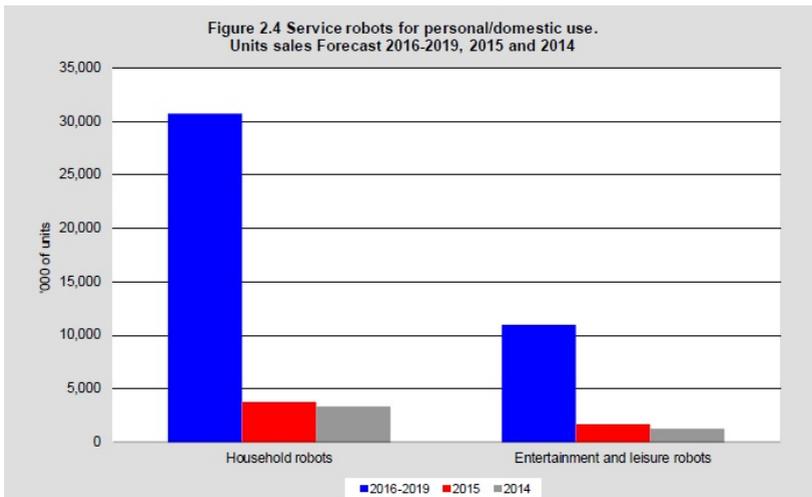
Segundo a *International Federation of Robotics* (2016), estima-se que o estoque mundial de robôs industriais operacionais aumentará perto de 1.631.600 unidades no final de 2015 para 2.589.000 unidades no final de 2019, representando uma taxa média de crescimento anual de 12% entre 2016 e 2019 (Figura 1). Entretanto, para robôs de uso pessoal e doméstico, estima-se que serão vendidas aproximadamente 42 milhões de unidades (Figura 2).

Figura 1: Estoque mundial de robôs industriais operacionais



Fonte: Adaptado de International Federation of Robotics (2016a)

Figura 2: Unidades vendidas de robôs de serviço 2016-2019



Fonte: Adaptado de International Federation of Robotics (2016b)

Destas áreas apresentadas, a área da robótica autônoma é con-

siderada importante para futuras aplicações, vindo que o surgimento de novas tecnologias perante a locomoção dos robôs vêm crescendo e aperfeiçoando-se.

Para que estes robôs possam locomover-se por ambientes desconhecidos, reconhecer objetos ou ações, precisam, primeiramente, receber informações do meio para conseguir tomar decisões. Essas informações podem ser adquiridas por sensores lasers, sensores ultrassônicos, sensores de rádio frequência e por câmeras, através da visão computacional.

A visão computacional é o processo de modelagem e replicação da visão humana usando software e hardware. É o processamento de dados a partir de imagens para a obtenção de informações que trazem dados relacionados a reconhecimento de objetos, segmentação, modelagem 3D, etc. Comparando a visão computacional com a visão humana, perante as funcionalidades, destaca-se as seguintes características: adaptabilidade, tomada de decisão, qualidade, velocidade, espectro e percepção tridimensional (CONCI, 2007).

1.2 PROBLEMÁTICA

O problema de um robô autônomo que precisa explorar um caminho desconhecido é bastante difícil. O fato de precisar caminhar de forma dinâmica, adquirindo dados do ambiente, evitando obstáculos, é desafiador para pesquisadores na área da robótica móvel. A medição da distância é uma das características importantes para este tipo de aplicação (BAEK; CHOI; LEE, 2010).

Olhando para a forma como os robôs autônomos adquirem as informações dos ambientes, tanto como distâncias ou características, que são feitas através das tecnologias de: SONAR - *Sound Navigation and Ranging*, RADAR - *Radio Detection And Ranging*, LIDAR - *Light Detection And Ranging* ou sensores infravermelhos possuem certas limitações.

Sensores como os infravermelhos ou lasers possuem interferência em suas medidas em áreas externas, causadas principalmente pela luz solar. A frequência da aquisição e a resolução dos dados com lasers variam entre os modelos disponibilizados no mercado. Porém destaca-se que este tipo de sensores adquire apenas um plano horizontal de pontos por varredura, sendo assim, para adquirir mais pontos no plano vertical, precisam realizar uma nova varredura. Usando como exemplo o modelo de sensor laser 2D UST-10LX, na realização de uma varredura

horizontal, ele demora 25 ms para a aquisição de 1.080 pontos com um ângulo de visão de 270° .

Portanto, o problema é como melhorar a forma de aquisição dessas informações? Para extrair dados parecidos, pode-se utilizar câmeras digitais, que hoje, estão mais evoluídas e difundidas pelo mundo, podendo ter a vantagem de extrair não somente a distância dos objetos ou do ambiente, mas também extrair informações como reconhecimento de objetos, movimentos ou ações. A forma de adquirir a distância entre o observador e os objetos no ambiente podem ser feitas através da triangulação entre os pontos de interesse. Efeito conhecido como paralaxe, utilizado na astronomia para medir as distâncias entre os astros. Contudo, para realizar este cálculo, necessita-se de dois pontos de observação, conseqüentemente, a adição de uma nova câmera.

Segundo Chang, Jeong e Hwang (2014), os sistemas de visão estéreo têm sido pesquisados há décadas e constituem o método tradicional para extrair a informação de profundidade em tempo real a partir de imagens tridimensionais (3D) usando sensores. No entanto, sistemas de visão estéreo passivos, que utilizam apenas as imagens normais capturadas pelas câmeras, mostram um erro significativo no processamento de regiões não texturizadas, que são frequentes em ambientes internos. Além disso, as modificações para processamento em tempo real produzem um erro usando algoritmos aproximados.

1.3 JUSTIFICATIVA

Observando toda a evolução dos robôs na sociedade, como eles trouxeram avanços para áreas de pesquisa, exploração, indústrias e serviços, tendo em vista que futuramente existirão robôs autônomos a serviço da sociedade, este trabalho contribuirá para realização deste futuro, trazendo uma nova forma de adquirir os dados de um ambiente, ajudando assim a sua locomoção, localização e facilitação para futuras aplicações em reconhecimento de objetos, em razão de que utilizarão os mesmos sensores.

1.4 OBJETIVOS

Nesta seção são descritos o objetivo geral e os objetivos específicos deste trabalho de conclusão de curso.

1.4.1 Geral

Desenvolver um software capaz de obter o mapa de profundidade de um ambiente através de imagens estereoscópicas em tempo real.

1.4.2 Específicos

- Aplicar a calibração e retificação das câmeras.
- Calcular a disparidade entre as imagens obtidas.
- Calcular a distância de objetos em uma determinada cena.
- Analisar o mapa de profundidade obtido.
- Propor um novo algoritmo para *Stereo Matching*¹

1.5 METODOLOGIA

Este trabalho é uma pesquisa experimental aplicada com revisão bibliográfica e projeto de protótipo experimental.

- Etapa 1: Adquirir o conhecimento necessários sobre os algoritmos de calibração de câmeras, retificação de imagens e *Stereo Matching* com uma revisão bibliográfica e levantamento de trabalhos relacionados ao tema;
- Etapa 2: Desenvolver um protótipo para aquisição das imagens;
- Etapa 3: Realizar a codificação para a calibração das câmeras, retificação e *Stereo Matching*;
- Etapa 4: Realizar a aquisição dos dados de um ambiente controlado;
- Etapa 5: Análise dos resultados.

¹Traduzido como: correspondência estéreo

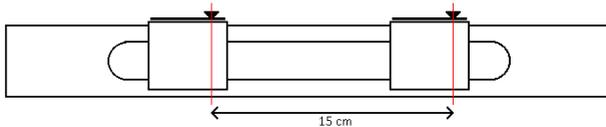
2 SISTEMA DE VISÃO ESTÉREO

Este capítulo apresenta um pouco sobre todo o sistema de visão estereoscópica em sua parte teórica, passando pela calibração das câmeras, retificação das imagens, disparidade, algoritmo SAD - Soma Absoluta das Diferenças - e o cálculo da distância. A biblioteca utilizada é comentada no Apêndice A.

2.1 O SISTEMA

O sistema é composto por uma dupla de câmeras Logitech® HD Webcam C270 dispostas paralelamente a uma distância de 15 cm (Figura 3). As câmeras foram colocadas o mais paralelo possível, para que os erros nas próximas etapas sejam mínimos. Elas também foram retiradas de suas capas para melhor fixação ao sistema. Algumas especificações importantes da câmera estão descritas no Quadro 1.

Figura 3: Sistema de Câmeras



Fonte: Elaborado pelo Autor.

Figura 4: Logitech HD Webcam C270



Fonte: Adaptado de Logitech (2016).

Quadro 1: Especificações da Câmera

Especificações da Câmera	
Lentes e Tipo do Sensor	Plástico
<i>Field Of View (FOV)</i>	60°
Distância Focal	4.0 mm
Tipo de Foco	Fixo
<i>Frame Rate (Máx)</i>	30fps @ 640x480

Fonte: Adaptado de Logitech (2014).

As próximas seções descrevem as etapas seguintes aplicadas ao sistema de uma maneira mais teórica. Sendo elas descritas por: Calibração de Câmera, Retificação, Disparidade e Distância.

Quadro 2: Especificações do Notebook utilizado

Samsung Expert X23 NP300E5K-X01BR	
Processador	Intel Core i5-5200U @ 2.20GHz
Memória	8,0 GB DDR3
Placa de Vídeo	NVIDIA GeForce 910M
Disco Rígido	WD10JPVX
Sistema Operacional	Windows 10 Home 64 bits

Fonte: Elaborada pelo Autor.

2.2 CALIBRAÇÃO DE CÂMERA

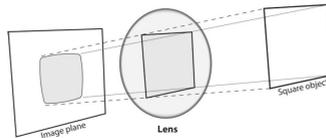
As câmeras digitais estão bem difundidas na população atualmente devido ao fato de que seu custo de fabricação foi reduzido, facilitando assim a sua aquisição e inserção em produtos que a utilizam, como os *smartphones*. Porém esse preço baixo não garante qualidade na fabricação, trazendo distorções às imagens capturadas.

Um dos problemas enfrentados na visão computacional é a fidelidade da imagem adquirida através das câmeras. O efeito de esfericidade, causado pelas lentes (demonstrado pela Figura 5), obtidas nas imagens, por mais que de modo geral seja imperceptível para os olhos humanos, carrega um erro em relação ao formato dos objetos. Caso o processamento das imagens seja para realizar cálculos de formato ou medição de distâncias, este efeito perturba o resultado final.

Para reduzir este efeito, pode-se fazer um processo chamado calibração de câmera. Na qual é obtido os parâmetros intrínseco e extrínseco da câmera e da lente. São estes os parâmetros:

- Extrínsecos:
 - Translação.
 - Rotação.
- Intrínsecos:
 - Distância focal.
 - Deslocamento do ponto principal.
 - Distorção da lente.

Figura 5: Distorção radial



Fonte: Adaptado de Kaehler e Bradski (2016)

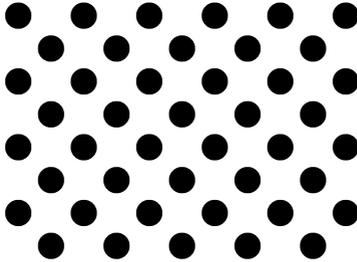
A biblioteca *OpenCV* trás a implementação do trabalho realizado por Zhang (2000), responsável pelo cálculo dos parâmetros da distância focal e do deslocamento do ponto principal, e do trabalho de Brown (apud KAEHLER; BRADSKI, 2016), responsável pelo cálculo dos parâmetros de distorção da lente. É utilizado o trabalho de Zhang (2000) pelo fato de ele ser flexível e de fácil aplicação. Esse é um dos pontos citados em seu trabalho, para que qualquer um pudesse realizar a calibração, com um simples padrão e poucas fotos do mesmo.

A calibração da câmera é feita utilizando um padrão conhecido em tamanho, dimensão e formato. Este padrão geralmente é um tabuleiro de xadrez de formato retangular e com um número par de quadradinhos. Existem outros padrões aceitos pela biblioteca, como por exemplo:

- Grade de círculos
- ArUco
- ChArUco
- Padrão randômico conhecido

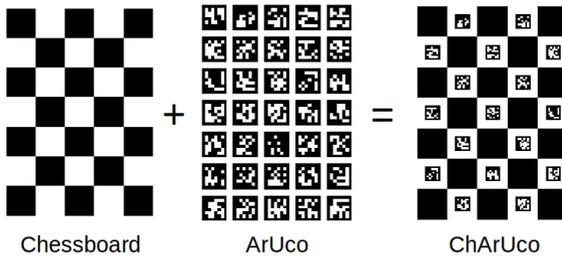
Os três últimos listados são suportados pela biblioteca *OpenCV Contrib*, que é uma biblioteca versão "beta" na qual as contribuições inseridas nela ainda não foram validadas para serem atribuídas à biblioteca principal.

Figura 6: Grade de círculos



Fonte: Adaptado de Robocraft (2011)

Figura 7: Tabuleiro de Xadrez, ArUco e ChArUco



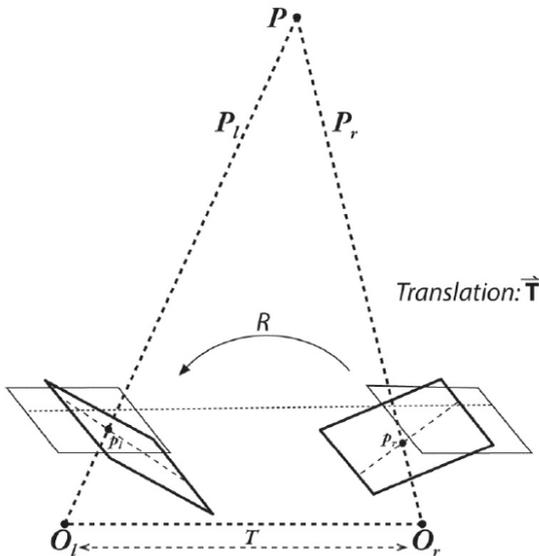
Fonte: Adaptado de OpenCV (2015)

2.2.1 Calibração Estéreo das Câmeras

Os parâmetros extrínsecos necessários para este trabalho precisam ser definidos para as duas câmeras utilizadas, e estes parâmetros são importantes para a próxima etapa, a retificação. A calibração individual das câmeras fornece dados extrínsecos, mas em relação a câmera e o padrão. Após a calibração individual das câmeras, é utilizada uma função fornecida pela biblioteca que calcula a rotação e a translação entre as câmeras. Esta etapa é importante pelo fato de que as imagens

obtidas pelo par de câmeras devem estar paralelas e alinhadas.

Figura 8: Demonstração da Rotação e Translação das Câmeras

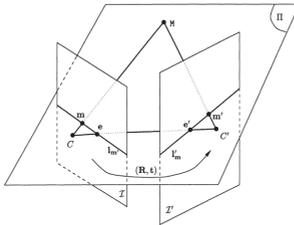


Fonte: Adaptado de Kaehler e Bradski (2016)

2.3 RETIFICAÇÃO

A geometria epipolar é a geometria básica de um sistema de visão estéreo (KAHLER; BRADSKI, 2016). Fundamentalmente, a geometria epipolar entre duas câmeras existe quando uma reta pode ser traçada entre pontos iguais em duas imagens (cenas) diferentes, obtidas ou não por uma mesma câmera. Neste caso, obtidas por duas câmeras iguais porém deslocadas lateralmente. Os livros de Kaehler e Bradski (2016) e Xu (1996) descrevem bem toda a parte matemática da geometria epipolar, sendo interessante para esse trabalho apenas o conceito da geometria epipolar aplicada pela retificação, que é o alinhamento entre as imagens. Este alinhamento garante que os objetos encontrados na imagem esquerda em uma determinada linha estejam na mesma linha nas imagens da câmera da direita.

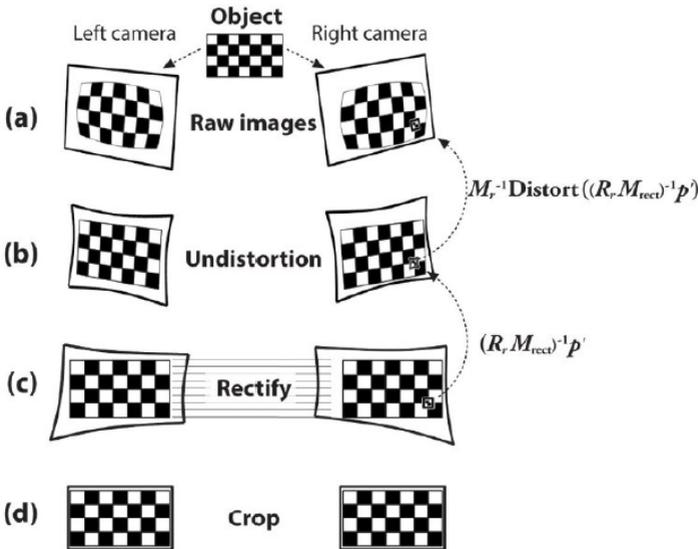
Figura 9: Geometria Epipolar



Fonte: Adaptado de Xu (1996)

A retificação também se responsabiliza pela maximização da área utilizável nas imagens, pois após feito a calibração e o alinhamento pela geometria epipolar, as imagens não serão mais as mesmas e possuirão áreas desnecessárias para o sistema. Essa maximização é representada na Figura 10.

Figura 10: Passos da Calibração e Retificação



Fonte: Adaptado de Kaehler e Bradski (2016)

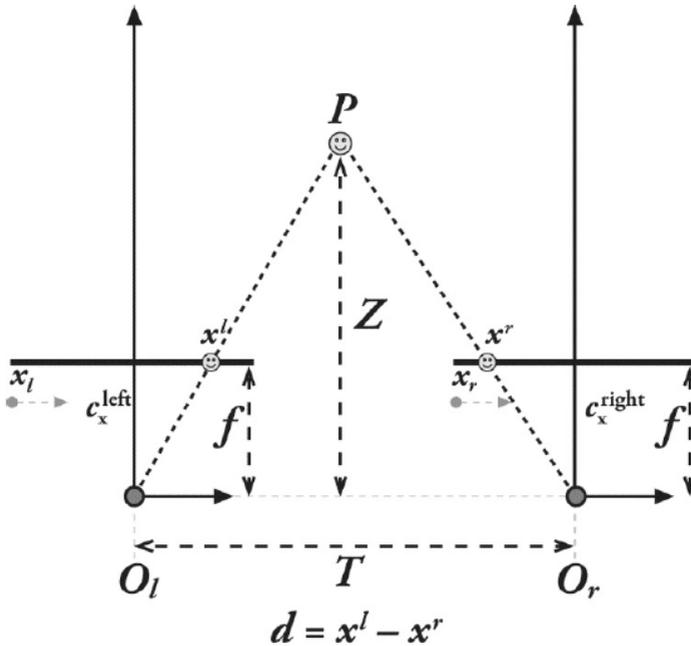
2.4 DISPARIDADE

A disparidade é a diferença entre os mesmos pontos em imagens diferentes. Este efeito da disparidade pode ser percebido quando observa-se objetos muito próximos aos nossos olhos e ficamos alternando entre eles, fechando um e abrindo o outro e vice-versa. Esse efeito de “deslocamento” é a disparidade e quanto mais longe o objeto estiver, menor será esse efeito.

O cálculo da disparidade é feito pela diferença entre o deslocamento no eixo x da imagem esquerda pelo da direita. A Figura 11 representa esse cálculo.

$$d = x^l - x^r \quad (1)$$

Figura 11: Disparidade



Fonte: Adaptado de Kaehler e Bradski (2016)

O algoritmo usado neste trabalho é o SAD. Este algoritmo fun-

ciona da seguinte maneira: tendo um ponto inicial (x_0, y_0) , cria-se uma janela de um tamanho m , sendo m um número ímpar e o ponto inicial no centro desta janela, na imagem da esquerda (representada pela letra T). Na imagem da direita (I) usa-se o mesmo ponto inicial (x_0, y_0) e o mesmo tamanho m de janela. Possuindo os dois blocos, então executa-se a seguinte equação:

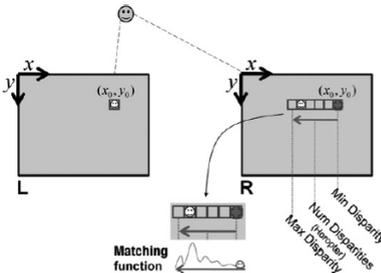
$$\sum_{x', y'}^{m, m} |(T(x', y') - I(x', y'))| \quad (2)$$

Esta janela na imagem da direita então é deslocada lateralmente e se realiza o mesmo calculo dado pela Equação 2. Este processo de deslocamento é repetido até um valor máximo de disparidade. Esse deslocamento é representado pela letra d .

$$\sum_{x', y'}^{m, m} |(T(x', y') - I(x' + d, y'))| \quad (3)$$

A Figura 12 representa o funcionamento deste algoritmo. Geralmente este deslocamento d é inicializado com o valor 0, mas pode assumir valores negativos ou até já começar com valores acima de 0. A disparidade entre os pontos será aquela na qual o resultado da Equação 3 for o menor positivo. Ou seja, quanto mais parecidas as imagens, mais próximo de zero a Equação 3 estará, porém isto pode gerar falsos positivos.

Figura 12: Algoritmo SAD



Fonte: Adaptado de Kaehler e Bradski (2016)

Todos os processos feitos anteriormente (calibração e retificação) são para garantir que as imagens estejam alinhadas pela linha, assim este e outros algoritmos podem ser executados com maior eficiência e

rapidez.

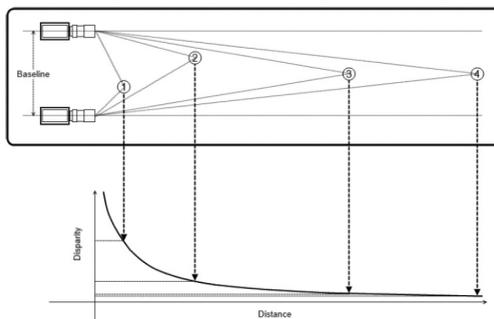
2.5 DISTÂNCIA

A distância ilustrada na Figura 11 pode ser obtida usando relação de triângulos. Para esse caso a distância pode ser expressa pela seguinte equação:

$$Z = \frac{f \times T}{x_l - x_r} \quad (4)$$

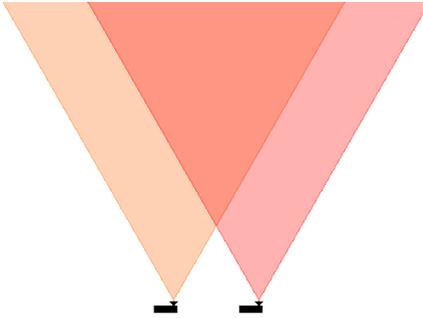
Onde Z é a distância, f a distância focal da câmera, T o *baseline* entre as câmeras (distância entre elas) e $x_l - x_r$ a disparidade, na qual eles representam o ponto central do bloco de interesse das imagens esquerda e direita. A distância possui uma relação inversa a disparidade. Quando a disparidade está próxima de zero, a distância possui um grande valor. E quanto maior a disparidade, menor a distância. Essa relação sempre existirá e está representada na Figura 13.

Figura 13: Relação entre distância e disparidade



Fonte: Adaptado de Kaehler e Bradski (2016)

Figura 14: Área de captura



Fonte: Elaborado pelo Autor.

A distância entre as câmeras também influencia na captura dos dados. Para capturas que necessitam de pequenas distâncias, quanto mais próximas as câmeras, melhores os resultados e, conseqüentemente, para grandes distâncias, mais afastadas as câmeras devem estar. Esse espaçamento influencia na área de captura, sendo essa área onde o campo de visão das câmeras se interceptam (Figura 14).

3 IMPLEMENTAÇÃO DO SISTEMA DE VISÃO ESTEREOSCÓPICA

Este capítulo apresenta todo o sistema de visão estereoscópica implementado, passando pelo processo de calibração das câmeras, retificação das imagens, obtenção do mapa de disparidade e do mapa de profundidade.

3.1 CALIBRAÇÃO DAS CÂMERAS

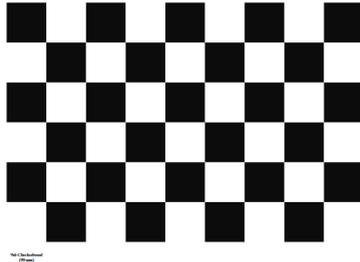
Na calibração das câmeras foram utilizados os algoritmos disponíveis na biblioteca *OpenCV*. Esses algoritmos são baseados nos trabalhos de Zhang (2000), responsável pelo cálculo da distância focal e deslocamento, e Brown (apud KAEHLER; BRADSKI, 2016), responsável pelo cálculo dos parâmetros de distorção.

Apesar de existirem vários outros métodos e algoritmos para fazer a calibração da câmera (SONG et al., 2013; HIERONYMUS, 2012), o método utilizado pela biblioteca *OpenCV* é mais fácil e flexível para a utilização, como citado no trabalho de Zhang (2000), sendo também fácil para que a qualquer pessoa possa utilizá-la.

Os padrões apresentados na Seção 2.2 podem ser impressos em uma impressora comum e devem ser colados em uma superfície plana e rígida, para não carregar um erro indesejado ao sistema. Neste trabalho foi usado uma TV LG 39LB5800 de 39" Full HD para a exibição do padrão. Foi uma maneira mais fácil e rápida de garantir que o padrão estivesse numa superfície plana e rígida, não dependendo de materiais que pudessem levar algum erro.

O padrão utilizado neste trabalho foi do tipo tabuleiro de xadrez com um tamanho 9x6. O quadrado do padrão possui a mesma altura e largura. Em alguns padrões disponibilizados na internet, essa altura e largura geralmente são diferentes. O tamanho do padrão pode ser qualquer um, desde que ele seja conhecido, porém quanto maior o padrão, melhor a calibração.

Figura 15: Padrão de Tabuleiro de Xadrez utilizado

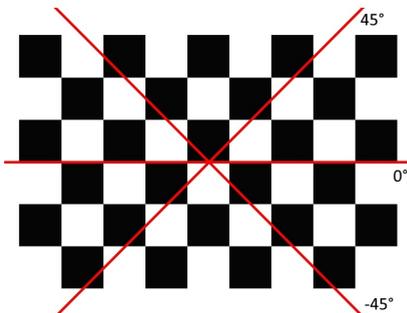


Fonte: Adaptdado de Maghoumi (2017)

Para realizar a calibração, foi feita uma sequência de 45 fotos, na qual as quinze primeiras estavam com 0° de inclinação, as próximas quinze com um ângulo de 45° de inclinação e as últimas quinze com um ângulo de -45° de inclinação, todas em relação ao eixo horizontal. Este eixo está representado na Figura 16 pelo ângulo 0° .

Durante a calibração das câmeras, foram seguidas algumas boas práticas. O uso de um padrão na qual seus quadrados possuíam altura e largura iguais rendeu melhores resultados do que outros padrões que apresentavam altura e largura diferentes. Também notou-se que ao bater as fotos do padrão, a somatória das fotos deve abranger toda a visão da câmera. Ou seja, bater fotos o suficiente para que toda área de visão da câmera seja abrangida.

Figura 16: Orientação do padrão para a obtenção da sequência de imagens



Fonte: Elaborado pelo autor

3.1.1 Calibração de Câmera Estéreo e Retificação

A biblioteca *OpenCV* também fornece uma função chamada *stereoCalibrate()* para realizar a calibração de duas câmeras ao mesmo tempo. Porém a própria documentação da biblioteca recomenda fazer a calibração individual de cada câmera e usar os dados obtidos apenas como parâmetros para a função. Esta função retorna duas matrizes importantes para a próxima etapa, a matriz de rotação entre a 1ª e a 2ª câmera e a matriz de translação entre o sistema de coordenadas da câmera. A retificação segue a mesma linha da Calibração Estéreo.

O processo principal é a calibração individual de cada câmera. A calibração estéreo e a retificação apenas utilizam dos resultados obtidos das funções anteriores. A calibração estéreo utiliza os valores de distorção de lente e os valores intrínsecos. A retificação utiliza as matrizes de translação e rotação obtidas na calibração estéreo, retornando os valores para que se possa corrigir as imagens capturadas.

3.2 MAPA DE DISPARIDADE

Feito todo o processo de calibração das câmeras e retificação, as imagens estão prontas para serem processadas para obter o mapa de disparidade. Porém para a obtenção do mapa, passa-se pré-filtros nas imagens capturadas e um pós-filtro no mapa de disparidade obtido.

3.2.1 Pré-filtros

Os pré-filtros utilizados e aplicados foram equalização de histograma e alisamento Gaussiano, respectivamente.

3.2.1.1 Equalização de Histograma

O histograma de uma imagem é a distribuição da frequência da ocorrência de uma determinada intensidade de um *pixel* em uma imagem em escala de cinza. A equalização de histograma procura distribuir igualmente os níveis de intensidade, normalizando o brilho e aumentando o contraste da imagem (CONCI; AZEVEDO; LETA, 2008)(GONZALES; WOODS, 2010).

As Figuras 17(a) e 17(b) demonstram a aplicação da equalização

de histograma e as Figuras 17(c) e 17(d) são seus respectivos histogramas.

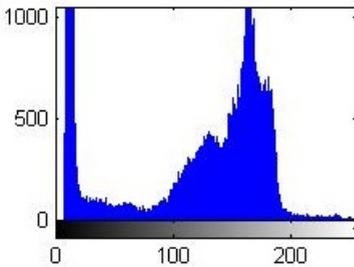
Figura 17: Exemplo de Aplicação da Equalização do Histograma



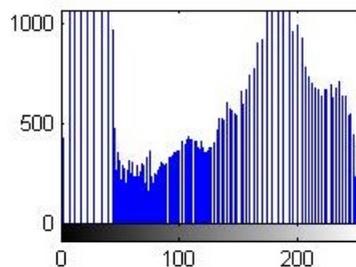
(a) Imagem Original



(b) Imagem Normalizada



(c) Histograma Original



(d) Histograma Normalizado

Fonte: Adaptado de Devinder (2013)

3.2.1.2 Alisamento Gaussiano

O alisamento Gaussiano, conhecido como *Gaussian Smoothing*, ou também borrão Gaussiano (*Gaussian Blur*) foi utilizado para reduzir ruído das imagens e reduzir os detalhes. O tamanho da máscara aplicada nas imagens obtidas foi de 5×5 . A Figura 18 demonstra um exemplo da aplicação do alisamento gaussiano.

Figura 18: Exemplo de Aplicação do Alisamento Gaussiano



Fonte: Adaptado de Kaehler e Bradski (2016)

3.2.2 Obtenção do Mapa de Disparidade

O mapa de disparidade foi obtido utilizando o algoritmo SAD brevemente explicado na Seção 2.4. A biblioteca *OpenCV* trás a implementação deste algoritmo em CUDA através da classe *cv::cuda::StereoBM* que é executada na GPU - *Graphics Processing Unit*. Com esta implementação, a execução do algoritmo é mais rápida do que executada em CPU - *Central Processing Unit*, pois utiliza a computação paralela para maximizar o desempenho do algoritmo.

A biblioteca fornece algumas configurações ao algoritmo, que pelo padrão, possuem seu valor igual a zero. Durante a obtenção do mapa, as configurações modificadas estão descritas na Tabela 3

Quadro 3: Configurações da classe *StereoBM*

Nome Configuração	Valor Utilizado
Texture Threshold	2
NumDisparities	56
BlockSize	51

Fonte: Elaborado pelo Autor.

3.2.3 Pós-Filtro

O pós-filtro utilizado neste trabalho faz parte da biblioteca *OpenCV Contrib*. O filtro do mapa de disparidade tem como base o filtro de mínimos quadrados ponderados (na forma de *Fast Global Smoother* que é muito mais rápido do que as implementações tradicionais de filtro de mínimos quadrados ponderados) e com o uso opcional de confiança baseada na consistência esquerda-direita para refinar os resultados em meias oclusões e áreas uniformes (OPENCV, 2017).

Quadro 4: Configurações do Filtro DisparityWLSFilter

Nome Configuração	Valor Utilizado
LRCThresh	255
Depth Discontinuity Radius	0

Fonte: Elaborado pelo Autor.

3.3 MAPA DE PROFUNDIDADE

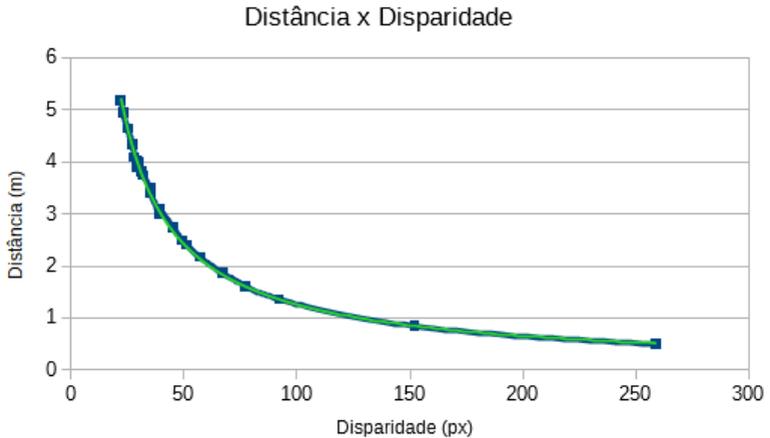
Após a obtenção do mapa de disparidade, podemos utilizar a Equação 4 para calcular a distância de todos os *pixels* que possuem um valor de disparidade válido. Porém, percebeu-se que usando esta formula, a distância calculada era errada. Esse erro será melhor descrito no Capítulo 4.

Para a coleta dos dados foram utilizados uma trena à laser da marca UNI-T modelo UT390B, para a medição da distância entre as câmeras e um objeto determinado, e para a medição das disparidade, foi utilizado o método manual na qual mediasse a disparidade do objeto entre as duas imagens capturadas.

Então foi feito uma tabela na qual a distância era relacionada com a disparidade. Foram feitas 37 medições na qual variaram de uma distância de 0,5 metros até 5,1 metros de distância entre as câmeras e um objeto. Após a coleta dos dados, foi utilizado um programa que fez uma aproximação geométrica retornando a equação 5.

Através dos resultados obtidos, pode-se então por meio de experimentação encontrar uma equação que representasse melhor a relação da distância e a disparidade. A Figura 19 demonstra os resultados obtidos e a Equação 5 representa a formula usada para se obter a distância de todos os pontos. O valor da unidade de medida da Equação 5 é metros.

Figura 19: Gráfico Distância x Disparidade



Fonte: Elaborado pelo autor

$$Z(x, y) = \frac{95,5519389459098}{d(x, y)^{0,940014374744605}} \quad (5)$$

Na Equação 5, a letra d representa a disparidade no $pixel(x, y)$ e a letra Z representa a distância deste mesmo $pixel$.

As operações matemáticas para obter o mapa de profundidade são executadas na GPU para maior performance. Após a obtenção do mapa de profundidade, utiliza-se uma função para colorir o mapa de profundidade, tornando-o mais fácil de perceber pequenas alterações nos valores.

3.4 ALGORITMO PROPOSTO

Através da experiência obtida durante a execução deste trabalho, foi percebido alguns padrões na execução do algoritmo SAD. Uma dos parâmetros que mais influencia na qualidade do mapa de disparidade é o tamanho da janela para realizar o *block matching*. Neste trabalho foi utilizado um valor de 51×51 *pixels*. Porém isso faz com que os objetos obtidos no mapa de disparidade tenham um tamanho maior do que apresentam. Esse foi um dos motivos da utilização de um pós-filtro que utiliza a imagem real obtida com o mapa de disparidade.

Figura 20: Disparidade com Bloco 3×3 Sem Filtro

Fonte: Elaborado pelo autor

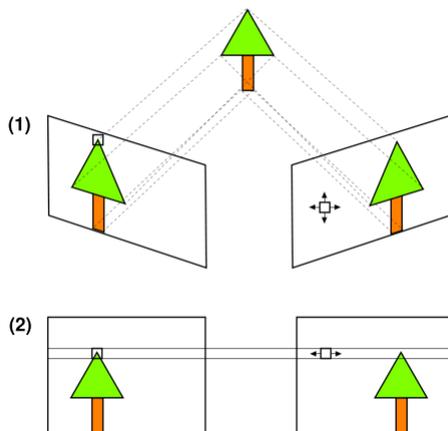
Figura 21: Disparidade com Bloco 51×51 Sem Filtro

Fonte: Elaborado pelo autor

3.4.1 Descrição do Algoritmo Proposto

O conceito do algoritmo vem através do algoritmo SAD, exemplificado na Figura 22. O algoritmo faz uma procura na mesma linha do ponto que se procura na segunda imagem. O problema é que a imagem que obtiver o menor resultado, ou seja, é mais parecida com a imagem da esquerda, será a vencedora e isso pode obter um falso positivo.

Figura 22: Exemplificação do Algoritmo SAD



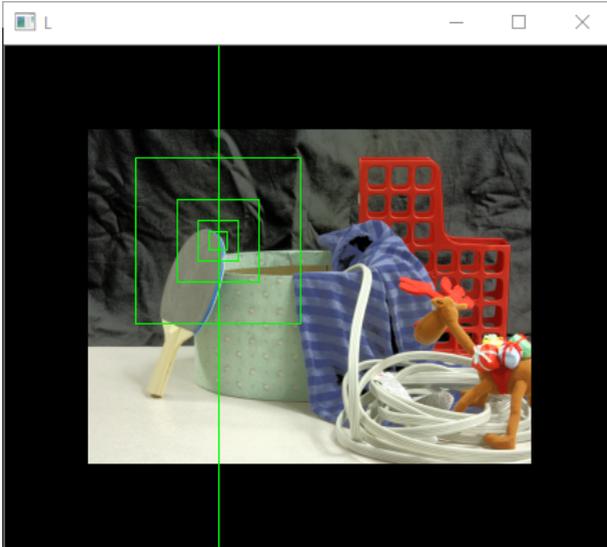
Fonte: Adaptado de Andel (2012)

A ideia por trás do algoritmo proposto é fazer uma busca global, como a janela de procura com um tamanho de 51×51 pixels, por exemplo, e ir dando um “zoom in” nos melhores resultados. Encontrado a melhor janela, então é refeita a busca na mesma linha porém com a janela reduzida em tamanho. Repete-se esse trabalho até que a janela obtenha um tamanho de 3×3 .

O conceito do algoritmo está representado nas Figuras 23 e 24. As fotos utilizadas para demonstrar o conceito do algoritmo foram retiradas do banco de imagens de Middlebury¹.

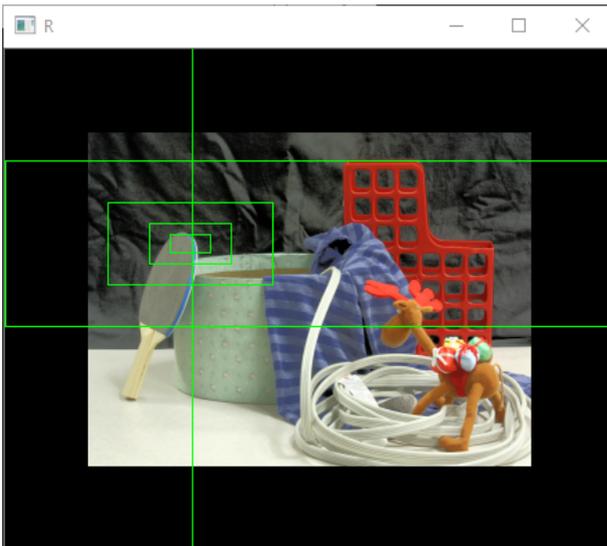
¹<http://vision.middlebury.edu/stereo/data/scenes2014/>

Figura 23: Conceito do novo algoritmo Imagem Esquerda



Fonte: Elaborada pelo Autor

Figura 24: Conceito do novo algoritmo Imagem Direita



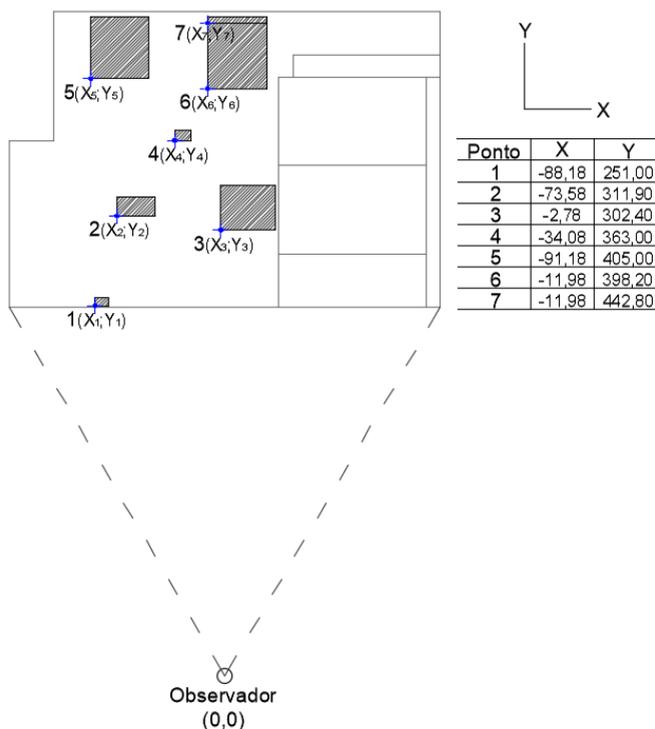
Fonte: Elaborada pelo Autor

4 RESULTADOS

Neste capítulo serão abordados os resultados obtidos no pré-processamento (calibração das câmeras e retificação), processamento (mapa de disparidade) e pós-processamento (mapa de profundidade). Em seguida serão apresentados os resultados das medições realizadas.

Para a realização das medições, foi criada uma cena na qual os objetos contidos nela foram colocados a uma distância conhecida. Sendo a cena utilizada demonstrada na Figura 25. As unidades apresentadas na Figura 25 estão em centímetros.

Figura 25: Cena para captura de dados



Fonte: Elaborada pelo Autor

4.1 PRÉ-PROCESSAMENTO

Os resultados obtidos para a calibração e a retificação são mostrados na Figura 26. Pode-se observar na Figura 26(b) que as linhas demarcadas com 1 e 2 não estão correspondendo a mesma linha nas duas imagens. Já na imagem corrigida, pode-se observar que na Figura 26(a) há o correto alinhamento entre as linhas, porém a imagem capturada pela câmera da direita carrega um erro na região superior esquerda.

As matrizes obtidas com a calibração das câmeras e a retificação estão representadas na Figura 27. Essas matrizes são os parâmetros intrínsecos das câmeras esquerda e direita, e um dos resultados da retificação das câmeras.

Figura 27: Matrizes de Resultados

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1.139e + 03 & 0. & 3.461e + 02 \\ 0. & 1.139e + 03 & 2.611e + 02 \\ 0. & 0. & 1. \end{bmatrix}$$

(a) Matriz de Parâmetros Intrínsecos Câmera Esquerda

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1.139e + 03 & 0. & 3.383e + 02 \\ 0. & 1.139e + 03 & 2.407e + 02 \\ 0. & 0. & 1. \end{bmatrix}$$

(b) Matriz de Parâmetros Intrínsecos Câmera Direita

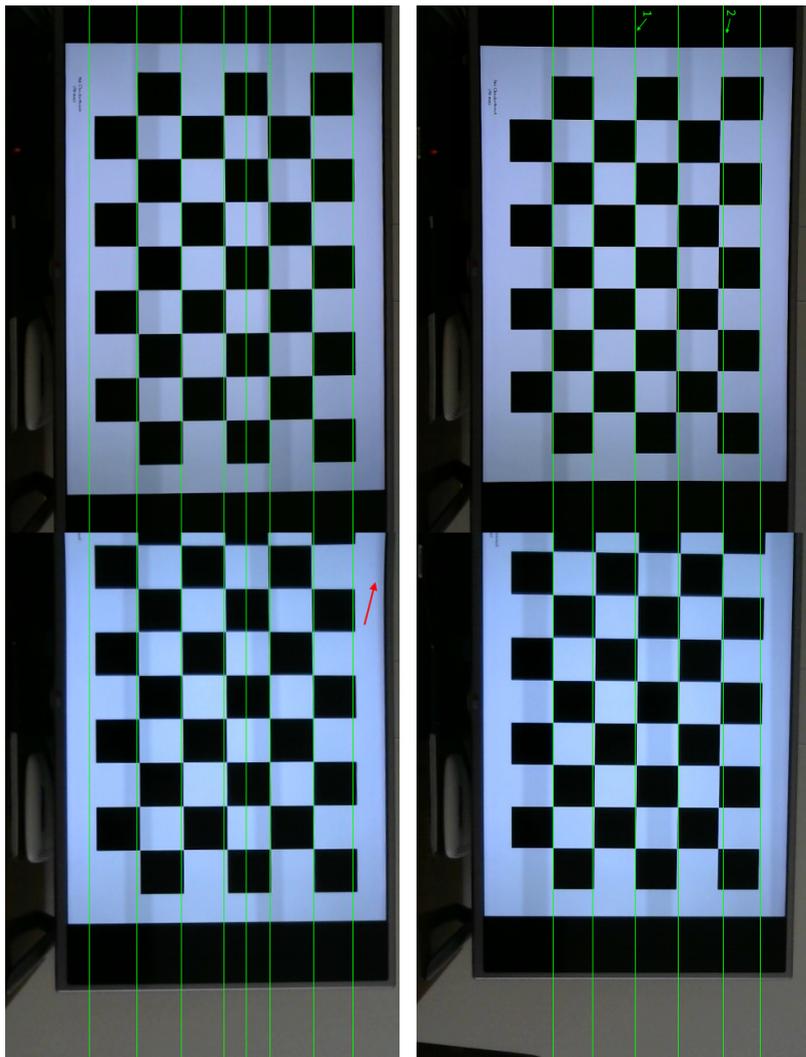
$$\begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & \frac{-1}{T_x} & \frac{(c_x - c_x')}{T_x} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -2.414e + 02 \\ 0 & 1 & 0 & -2.503e + 02 \\ 0 & 0 & 0 & 1.190e + 03 \\ 0 & 0 & 6.666e - 03 & 2.478e - 02 \end{bmatrix}$$

(c) Matriz Q

Fonte: Elaborada pelo Autor.

O erros médios da calibração individual e estéreo foram de, respectivamente, 0,358, 0,561 e 1,036. Sendo o valor de 0 o melhor resultado. A matriz da Figura 27(c) contém alguns dos resultados im-

Figura 26: Resultado da Aplicação da Calibração e Retificação



(a) Calibrada

(b) Normal

Fonte: Elaborado pelo Autor.

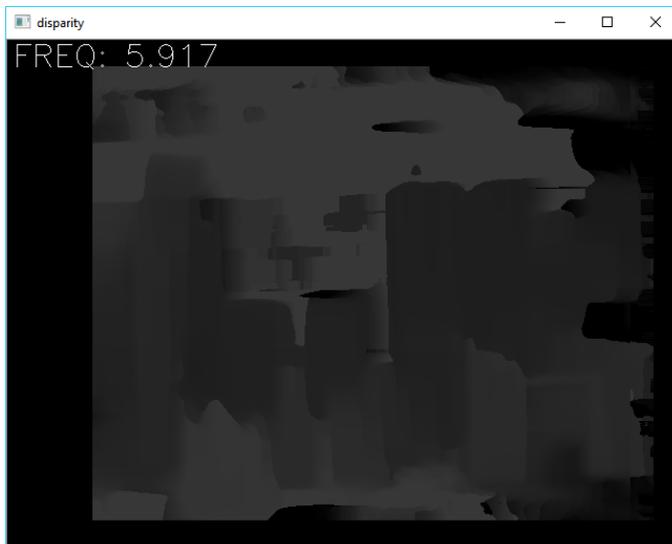
portantes. Na Matriz Q, o valor de $\frac{-1}{T_x}$ é inversamente proporcional a distância entre as duas câmeras. E o valor de f , representada em milímetros, é a distância focal obtida da câmera. O Quadro 1 contém as especificações da câmera e mostra que os valores apresentados não são iguais à especificação. Esse erro é, talvez, o responsável pela instabilidade do sistema anteriormente à execução do cálculo da distância a partir da Equação 5.

$$T_x = \frac{-1}{6.666e - 03} = 150,015mm \quad (6)$$

4.2 PROCESSAMENTO

O mapa de disparidade obtido da cena da Figura 25 está sendo representado pela Figura 28, na qual ela está apenas com os pré-filtros. Esse mapa obtido contém a informação da disparidade, sendo as áreas claras os objetos mais próximos, ou seja, maior disparidade, e as áreas escuras, objetos mais longe e disparidade menor. A aplicação do pós-filtro melhora a visualização deste mapa, representada na Figura 29. O resultado obtido pelo algoritmo SAD com o pós-filtro é satisfatório para o sistema.

Figura 28: Mapa de Disparidade apenas com Pré-Filtro



Fonte: Elaborado pelo autor

Figura 29: Mapa de Disparidade Com Pós-Filtro

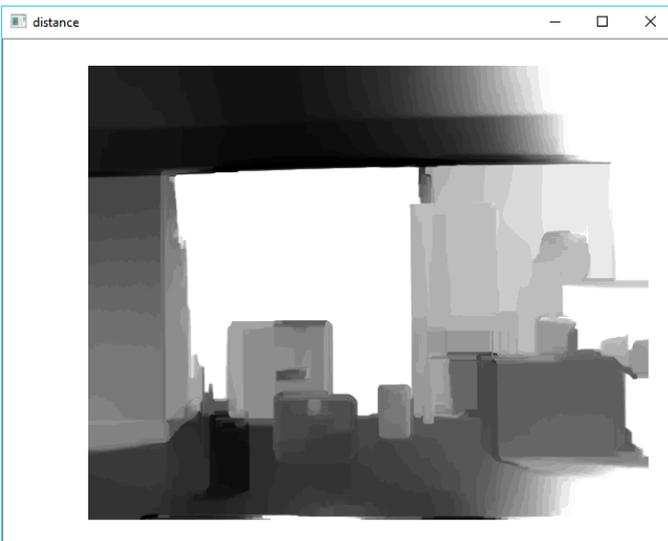


Fonte: Elaborada pelo Autor

4.3 PÓS-PROCESSAMENTO

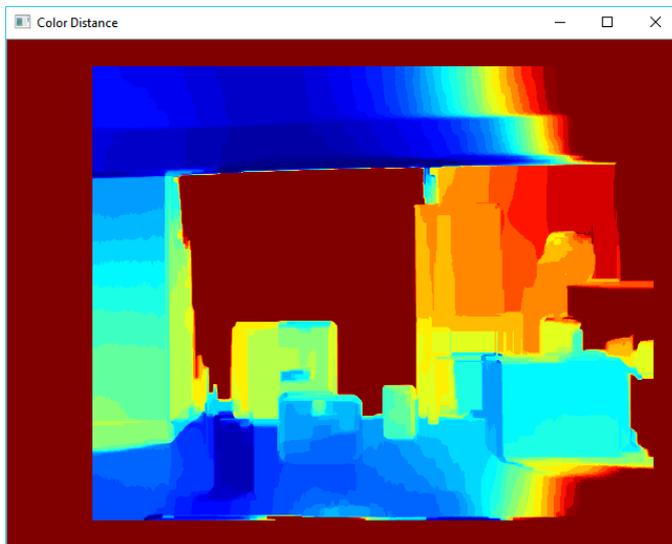
O mapa de profundidade obtido da cena da Figura 25 com o mapa de disparidade da Figura 29 está sendo representado pela Figura 30, sendo a Figura 31 colorida para melhor visualização. O sistema possui uma certa velocidade no processamento de todas as informações. A frequência na Figura 29 mostra o número de conclusões do processo completo em Hertz - Hz.

Figura 30: Mapa de Profundidade



Fonte: Elaborada pelo Autor

Figura 31: Mapa de Profundidade Colorido



Fonte: Elaborada pelo Autor

4.4 OBTENÇÃO DOS DADOS

Os dados foram obtidos de duas formas, em tempo real com captura direta e estática, sendo uma foto da cena. Foram adquiridas 200 amostras nas duas formas, sendo recolhidos os dados de disparidade e distância. Uma outra forma de adquirir a disparidade foi usando o *Template Matching* da biblioteca para validar a disparidade obtida pelo algoritmo SAD. A aquisição do dado foi feita pegando a média da imagem do objeto.

Figura 32: Exemplificação da Aquisição de Dados



Fonte: Elaborada pelo Autor

A Figura 32 ilustra um exemplo da aquisição dos dados. Sendo o dado coletado a média dos valores dentro do retângulo de cada objeto. Essa foi a forma da aquisição dos dados da distância e disparidade. Já no *Template Matching*¹ a disparidade foi calculada seguindo a Equação 1.

4.5 ANÁLISE DOS RESULTADOS

4.5.1 Captura em Tempo Real

Os resultados obtidos foram satisfatórios perante o sistema e tiveram um erro médio máximo de aproximadamente 20 cm, para as capturas realizadas em tempo real. A unidade usada na Tabela 1 é milímetros.

Pode-se observar nos resultados obtidos alguns erros. Nas Figuras 33, 34 e 35 entre as amostras 29 e 43, e amostras 190 a 197 provavelmente houveram falhas na comunicação entre o computador e as câmeras, sendo que esta falha estava ocorrendo com pouca frequên-

¹Traduzido como: Correspondência de modelos

cia no sistema. Entretanto esse erro pode ser descartado tendo em vista que nas demais amostras houve a estabilidade do mesmo e que esses erros foram esporádicos.

Outro erro é na Figura 34 no Objeto 6 na qual seu valor varia muito. Esse erro ocorreu devido ao tamanho do *template*² na qual permitiu que ocorresse falsos positivos. Porém os demais resultados estão corretos, logo pode-se concluir que a aquisição está correta, validando os dados da Figura 33.

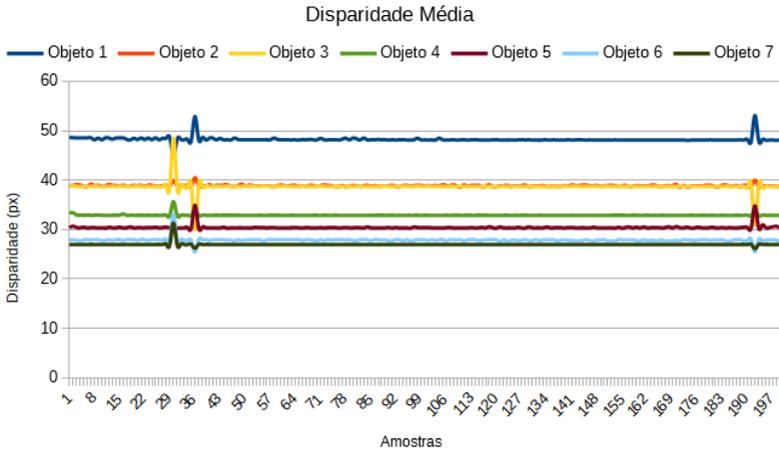
Tabela 1: Resultados Obtidos para Captura Ao Vivo

Objeto	Distância (mm)			
	Laser	Sistema Média	Erro	Desvio Padrão
1	2510	2498,78695	11,21305	25,22089
2	3119	3062,99495	56,00505	13,29633
3	3029	3078,57865	49,57865	87,72833
4	3600	3577,58055	22,41945	21,17891
5	4050	3857,59690	192,40310	47,80050
6	3982	4185,85670	203,85670	53,92004
7	4427	4310,53215	116,46785	40,40968

Fonte: Elaborada pelo Autor.

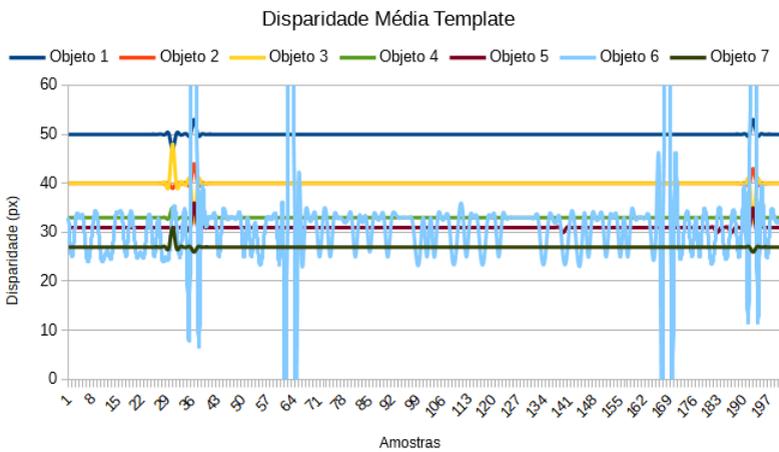
²Traduzido como: modelo

Figura 33: Disparidade Média Captura Ao Vivo



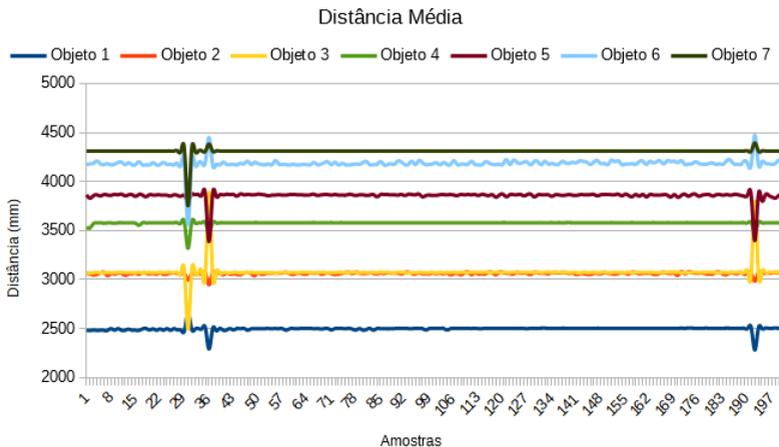
Fonte: Elaborada pelo Autor.

Figura 34: Disparidade Média Template Captura Ao Vivo



Fonte: Elaborada pelo Autor.

Figura 35: Distância Média Captura Ao Vivo



Fonte: Elaborada pelo Autor.

4.5.2 Captura Estática

Os resultados da captura estática também obtiveram resultados satisfatórios, tendo um erro médio máximo de aproximadamente 20 cm também. A unidade na Tabela 2 é milímetros.

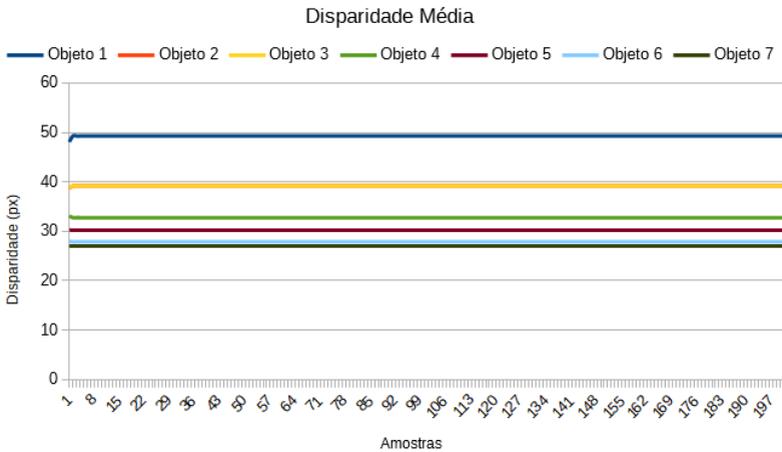
Nos resultados estáticos existem dois erros ocorridos durante a captura dos dados. O primeiro erro é na inicialização da captura estática, percebida na primeira amostra nas Figuras 36, 37 e 38. O segundo erro é no *template* do Objeto 6 na Figura 37, na qual na inicialização da captura, obteve um falso positivo. O motivo disso é que o Objeto 7 é o que possui a maior distância, portando deve possuir a menor disparidade, o que não é verídico na Figura 37, onde o Objeto 6 possui uma menor disparidade que o Objeto 7.

Tabela 2: Resultados obtidos para Captura Estática

Objeto	Distância (mm)				
	Laser	Sistema	Média	Erro	Desvio Padrão
1	2510	2449,87	60,13	0,0	
2	3119	3039,63	79,37	0,0	
3	3029	3038,68	9,68	0,0	
4	3600	3593,84	6,16	0,0	
5	4050	3878,86	171,14	0,0	
6	3982	4185,65	203,65	0,0	
7	4427	4310,67	116,33	0,0	

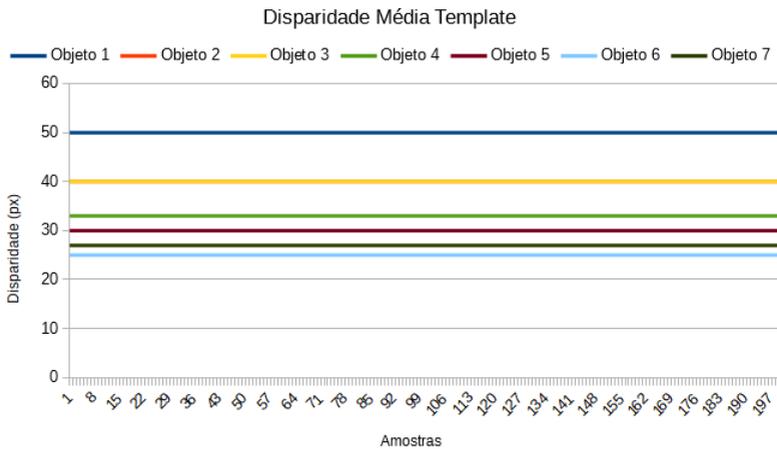
Fonte: Elaborada pelo Autor.

Figura 36: Disparidade Média Captura Estática



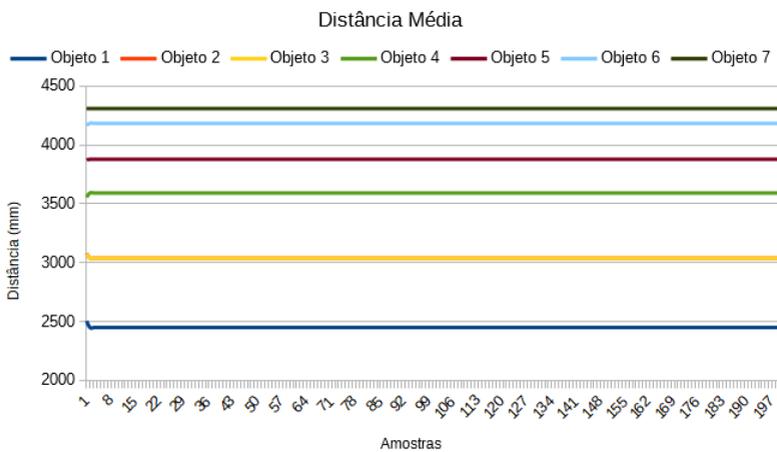
Fonte: Elaborada pelo Autor.

Figura 37: Disparidade Média Template Captura Estática



Fonte: Elaborada pelo Autor.

Figura 38: Disparidade Média Captura Estática



Fonte: Elaborada pelo Autor.

4.6 RESULTADOS DO ALGORITMO PROPOSTO

O conceito do algoritmo proposto foi testado com base no mesmo processo realizado com o algoritmo SAD. O algoritmo em si é apenas uma prévia do que se pode fazer seguindo a mesma lógica proposta. A aprimoração do algoritmo pode ser obtida através do uso de Redes Neurais para a obtenção do mapa de disparidade. Um estudo deve ser feito para saber se ele será promissor para a obtenção do mapa de disparidade.

Tabela 3: Resultados da disparidade do Algoritmo Proposto

Objeto	Disparidade (px)		
	Disparidade Estática	Disparidade Algoritmo	Erro
1	49,2925	45,8696	3,4229
2	39,1977	39,7669	0,5692
3	39,2229	34,6939	4,529
4	32,7875	37,5526	4,7651
5	30,2478	32,8212	2,5734
6	27,8757	27,7564	0,1193
7	27,0132	26,349	0,6642

Fonte: Elaborado pelo Autor.

Tabela 4: Resultados da distância do Algoritmo Proposto

Objeto	Distância (mm)		
	Laser	Sistema Média	Erro
1	2510	2632,48	122,48
2	3119	2998,62	120,38
3	3029	3412,01	383,01
4	3600	3169,34	430,66
5	4050	3624,99	425,01
6	3982	4204,79	222,79
7	4427	4413,95	13,05

Fonte: Elaborado pelo Autor.

Após realizar os experimentos, observa-se que o algoritmo proposto obtém a disparidade entre os *pixels* das imagens e apresenta melhores resultados do que o SAD em determinados cenários (áreas com poucas texturizadas, superfícies lisas que possuem a mesma intensidade

de pixel).

5 CONSIDERAÇÕES FINAIS

O sistema de visão estereoscópica apresentado é apenas um início de futuros trabalhos que podem ser feitos na área de visão computacional no Campus UFSC Araranguá. Com as tecnologias que estão sendo lançadas, como os veículos autônomos, necessitam da obtenção de informações do ambiente para que se possam locomover, tomar decisões perante o ambiente inserido, etc.

Além desse trabalho, um Engenheiro de Computação precisa do conhecimento para realizar a embarcação desse sistema num veículo. A quantidade de sistemas embarcados inseridos no nosso cotidiano aumentou exponencialmente e poder realizar inovações tecnológicas juntamente com sistemas embarcados é um dos desafios de um engenheiro.

O sistema apresentado obteve resultados satisfatórios para aplicações que necessitem de uma precisão de até ± 20 cm. Podendo ser inserido em projetos de grande porte no qual este erro seria insignificante. A calibração das câmeras e a retificação apresentaram resultados satisfatórios, sendo que a distância focal obtida nos resultados não foi igual a dita pela fabricante. O erro médio da calibração das câmeras ficaram próximos de 0, o que é considerado bom. A retificação também teve um bom resultado.

A contribuição realizada pelo autor neste trabalho foi de criação própria. Através das dificuldades e resultados obtidos, foi visto que poderia haver uma melhora no algoritmo SAD. A forma como ele opera na imagem pode excluir falsos positivos, pois quanto maior a janela de procura, maior a acurácia entre o bloco da esquerda e o bloco de procura na direita. Isso exclui lugares nos quais o SAD poderia se perder e dar um falso positivo

A proposta de se implementar o algoritmo em redes neurais é baseada nos trabalhos submetidos ao Benchmark de Middlebury. "As redes neurais são atualmente utilizadas por muitos dos melhores métodos no Benchmark de Middlebury¹."(HILLERSTRÖM, 2017)².

¹<http://vision.middlebury.edu/>

²Texto original: "Neural networks are currently used by many of the top-ranking methods on the Middlebury benchmark."

5.1 DIFICULDADES

Apesar de o conceito de visão estereoscópica ser simples, o caminho para se obter todos os resultados é difícil. Ele abrange áreas complexas de estudos bem divergentes e compreender tudo isso, para reunir em um trabalho, foi uma tarefa complicada.

As áreas foram desde fotografia até computação paralela, sendo conhecimentos esses não obtidos durante a graduação. Porém tudo isso acaba se convergindo na visão estereoscópica. Essas implementações um dia precisarão ser embarcadas em chips e placas capazes de realizar todo esse processamento.

O trabalho passou por um momento de imprecisão na qual não obtinha estabilidade para as medições. Essas fontes de imprecisão era a montagem do sistema, as câmeras não estavam bem alinhadas e paralelas, sendo necessário a reconstrução total do equipamento.

Para resolver esse problema, foi criado um padrão em escala na qual ele possuía várias linhas horizontais, verticais e diagonais e na imagem capturada da câmera era desenhado esse mesmo padrão. Logo, com isso, foi alinhado as câmeras e obtido um bom resultado. O resultado da Equação 6 demonstra isso.

Seguindo nas dificuldades, também foi visto que havia um truncamento nas casas decimais das variáveis obtidas na calibração. A biblioteca recomenda salvar os resultados da calibração como parâmetros intrínsecos e extrínsecos e a partir deles obter o mapa para remapeamento das imagens.

Entretanto com o truncamento da informação, esse mapa, quando recalculado com as variáveis salvas, não obtinha as mesmas informações como quando calculado na primeira vez. Uma das causas disso pode ter sido que as informações já estavam na memória do computador, não ocorrendo o truncamento. Resolvendo esse problema, o trabalho avançou muito e rapidamente, pois boa parte da implementação estava feita.

5.2 PROPOSTAS PARA TRABALHOS FUTUROS

Nesta seção são listadas algumas propostas como trabalhos futuros para a visão estereoscópica:

1. Desenvolver um mecanismo que dê liberdade ao sistema e que não fique preso aos 15 cm proposto neste trabalho. Um meca-

- nismo que possa deslocar uma das câmeras, mantendo a outra fixa, dando a elas uma distância dinâmica e conhecida entre elas;
2. Complementando o Item 1, realizar uma experimentação para adquirir uma equação na qual os parâmetros de entrada nela sejam a distância focal da câmera e a distância entre as duas câmeras;
 3. Desenvolver o algoritmo proposto em Redes Neurais;
 4. Fazer comparações com os algoritmos existentes com o algoritmo proposto;
 5. Desenvolver o algoritmo em computação paralela;
 6. Desenvolver uma Rede Neural que possa ser capaz de realizar a calibração de câmeras;
 7. Otimizar o sistema proposto para maximizar seus resultados;
 8. Realizar o mesmo trabalho com câmeras de maior qualidade e fazer as comparações;
 9. Implementar o sistema em um robô, para que possa resolver os problemas de *Simultaneous localization and mapping* - SLAM;
 10. Implementar o sistema em um sistema embarcado capaz de realizar o processamento completo;
 11. Embarcar o sistema proposto em uma placa NVIDIA Jetson TK1 ou equivalente.

REFERÊNCIAS

- ANDEL, B. van. **Image Rectification**. 2012. Disponível em: <https://en.wikipedia.org/wiki/File:Image_rectification.svg>.
- BAEK, H.-S.; CHOI, J.-M.; LEE, B.-S. Improvement of distance measurement algorithm on stereo vision system(SVS). In: **2010 Proceedings of the 5th International Conference on Ubiquitous Information Technologies and Applications**. IEEE, 2010. Disponível em: <<https://doi.org/10.1109/icut.2010.5678176>>.
- BROWN, D. C. Close-range camera calibration. **PHOTOGRAMMETRIC ENGINEERING**, v. 37, n. 8, p. 855–866, 1971. Disponível em: <<http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.14.6358>>.
- CHANG, J.; JEONG, J.-c.; HWANG, D.-H. Real-time hybrid stereo vision system for hd resolution disparity map. In: **Proceedings of the British Machine Vision Conference**. [S.l.]: BMVA Press, 2014.
- CONCI, A. **Computacao Grafica: Teoria e Pratica - Vol.2**. Elsevier, 2007. ISBN 8535223290. Disponível em: <<https://www.amazon.com/Computacao-Grafica-Teoria-Pratica-Vol-2/dp/8535223290?SubscriptionId=0JYN1NVW651KCA56C102tag=techkie-20linkCode=xm2camp=2025creative=165953creativeASIN=8535223290>>.
- CONCI, A.; AZEVEDO, E.; LETA, F. **Computação Gráfica: Teoria e Prática**. 2. ed. Rio de Janeiro: Elsevier, 2008. ISBN 85-352-2329-0.
- DEVINDER. **Histogram Equalization**. 2013. Disponível em: <<https://www.mathworks.com/matlabcentral/fileexchange/42401-histogram-equalization>>. Acesso em: 2017-11-16.
- GONZALES, R.; WOODS, R. **Processamento Digitais de Imagens**. 3. ed. [S.l.]: Pearson, 2010. 624 p. ISBN 978-85-7605-401-6.
- HIERONYMUS, J. Comparison of Methods for Geometric Camera Calibration. **International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science**, XXXIX-B5,

n. XXII ISPRS Congress, p. 595–599, 2012. ISSN 16821750.
Disponível em: <<http://www2.informatik.hu-berlin.de/cv/>>.

HILLERSTRÖM, P. **GPU-Accelerated Real-Time Stereo Matching**. Dissertao (mathesis) — Chalmers University Of Technology; University Of Gothenburg, Gothenburg, Sweden, 2017.

International Federation of Robotics. **Executive Summary World Robotics 2016 Industrial Robots**. 2016. Disponível em:
<https://ifr.org/downloads/press/02_2016-Presentation_market_overviewWorld_Robotics_29_9_2016.pdf>.

International Federation of Robotics. **Executive Summary World Robotics 2016 Industrial Robots**. 2016. Disponível em:
<https://ifr.org/downloads/press/02_2016-Executive_Summary_Service_Robots_2016.pdf>.

INTERNATIONAL FEDERATION OF ROBOTICS. **Executive Summary World Robotics 2016 Industrial Robots**. 2016.
Disponível em: <https://ifr.org/img/uploads-Executive_Summary_WR_Industrial_Robots_20161.pdf>.

KAHLER, A.; BRADSKI, G. **Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library**. O’Reilly Media, 2016. ISBN 9781491937990. Disponível em: <<https://www.amazon.com/Learning-OpenCV-Computer-Vision-Library-ebook/dp/B01MRXIYAN?SubscriptionId=0JYN1NVW651KCA56C102tag=techie-20linkCode=xm2camp=2025creative=165953creativeASIN=B01MRXIYAN>>.

LOGITECH. **Logitech HD Webcam C270 Technical Specifications**. 2014. Disponível em:
<http://support.logitech.com/en_us/article/17556>.

LOGITECH. **Logitech HD Webcam C270H**. 2016. Disponível em:
<<https://www.logitech.com/en-sg/product/hd-webcam-c270h>>.
Acesso em: 2017-11-15.

MAGHOUMI, M. **Practical Kinect Stereo Calibration for the Highest Accuracy**. 2017. Disponível em:
<<https://www.codefull.org/2017/04/practical-kinect-calibration/>>.
Acesso em: 2017-10-18.

OPENCV. **Detection of ChArUco Corners**. 2015. Disponível em:
<https://docs.opencv.org/3.1.0/df/d4a/tutorial_charuco_detection.html>. Acesso em:
2017-11-13.

OPENCV. **DisparityWLSFilter Class Reference**. 2017. Disponível em: <https://docs.opencv.org/3.3.0/d9/d51-/clascv_1_1ximgproc_1_1DisparityWLSFilter.html>. Acesso em: 2017-11-17.

PEREIRA, J. **Avaliação e correção do modelo cinemático de robôs móveis visando a redução de erros no seguimento de trajetórias**. Dissertao (Mestrado) — Universidade do Estado de Santa Catarina, Joinville, dec 2003.

ROBOCRAFT. 2011. Disponível em: <http://robocraft.ru/files/opencv/acircles_pattern.png>. Acesso em: 2017-11-13.

SILVA, L. R. da. **Análise e Programação de Robôs Móveis Autônomos da Plataforma Eyebot**. Dissertao (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, SC, mar. 2003.

SONG, L. et al. Survey on Camera Calibration Technique. **2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics**, IEEE, p. 389–392, aug 2013. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper-.htm?arnumber=6642768>>.

XU, Z. Z. a. G. **Epipolar Geometry in Stereo, Motion and Object Recognition: A Unified Approach**. 1. ed. Springer Netherlands, 1996. (Computational Imaging and Vision 6). ISBN 978-90-481-4743-4,978-94-015-8668-9. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=9ec6df8824588522383383911962435a>>.

ZHANG, Z. A flexible new technique for camera calibration. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Institute of Electrical and Electronics Engineers (IEEE), v. 22, n. 11, p. 1330–1334, 2000. Disponível em: <<https://doi.org/10.1109/34.888718>>.

APÊNDICE A – Biblioteca *OpenCV*

A biblioteca *OpenCV - Open Source Computer Vision Library* - é uma biblioteca de código aberto para visão computacional e aprendizado de máquina distribuída sob a licença BSD. Ela foi construída para providenciar uma infra-estrutura comum para aplicações de visão computacional e aprendizado de máquina, possuindo mais de 2500 algoritmos, que inclui algoritmos clássicos e do estado da arte.

Esses algoritmos podem ser usados para detectar e reconhecer rostos, identificar objetos, classificar ações humanas em vídeos, rastrear movimentos de câmera, rastrear objetos em movimento, extrair modelos de objetos 3D, produzir nuvens de pontos 3D de câmeras estéreo, juntar imagens para produzir uma imagem de alta resolução de uma cena inteira, encontrar imagens semelhantes de um banco de dados de imagens, remover os olhos vermelhos das imagens tiradas usando o flash, rastrear os movimentos dos olhos, etc. Possui uma comunidade com mais de 47 mil usuários e 14 milhões de downloads. Ela também é amplamente utilizada em empresas, em pesquisa e órgãos governamentais.

A biblioteca foi construída para ser multiplataforma. Seus algoritmos são desenvolvidos em C/C++, sendo os algoritmos mais recentes desenvolvidos em C++. Ela também possui desenvolvimento em Python e Java, com a intenção de atingir um público maior. Isso faz com que a biblioteca seja compatível com plataformas como Windows, Linux, Mac OS, FreeBSD, OpenBSD, Android, Maemo, iOS e também para sistemas como o Raspberry Pi.¹

O *OpenCV* também possui um repositório de contribuições ainda não vinculadas a distribuição principal, conhecida como *OpenCV Contrib*, onde possui novos algoritmos, módulos e exemplos não abrangido pela biblioteca principal.

Neste trabalho é usado uma compilação da biblioteca *OpenCV*, *OpenCV Contrib*, CUDA - *Compute Unified Device Architecture*, Intel® MKL e Intel® TBB. Os hardwares da NVIDIA® possuem uma especificação chamada *compute capability*² e para cada hardware, a compilação da biblioteca deve suportar isso. Neste caso ela foi compilada para uma *compute capability* versão 3.5. Esta versão da biblioteca foi compilada pelo próprio autor.

¹Informações extraídas do site: <https://opencv.org/about.html>

²Traduzido literalmente como: capacidade de cálculo

Quadro 5: Especificações das Bibliotecas Utilizadas

Biblioteca	Versão
OpenCV	3.3.0
OpenCV Contrib	3.3.0
CUDA Toolkit	8.0
Intel® Math Kernel Library (Intel® MKL)	2018.0.0.035
Intel® Threading Building Blocks (Intel® TBB)	2018.0.0.035

Fonte: Elaborada pelo Autor.