

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**UMA METODOLOGIA AUTOMÁTICA PARA
GERAÇÃO DE MAPAS DE DISPARIDADES DE
AMBIENTES REAIS**

CARLOS WILLIAN DE CARVALHO

ORIENTADOR: PROF. DR. EMERSON CARLOS PEDRINO

São Carlos – SP

5 de dezembro de 2017

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**UMA METODOLOGIA AUTOMÁTICA PARA
GERAÇÃO DE MAPAS DE DISPARIDADES DE
AMBIENTES REAIS**

CARLOS WILLIAN DE CARVALHO

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Processamento de Imagens e Sinais

Orientador: Prof. Dr. Emerson Carlos Pedrino

São Carlos – SP

5 de dezembro de 2017

Carvalho, Carlos Willian de

Uma Metodologia Automática para Geração de Mapas de Disparidades de Ambientes Reais / Carlos Willian de Carvalho. -- 2017.

99 f. : 30 cm.

Dissertação (mestrado)-Universidade Federal de São Carlos, campus São Carlos, São Carlos

Orientador: Prof. Dr. Emerson Carlos Pedrino

Banca examinadora: Prof. Dr. Emerson Carlos Pedrino, Prof. Dr. José Hiroki Saito, Prof. Dr. Celso Olivete Júnior

Bibliografia

1. Processamento Digital de Imagens. 2. Visão Estéreo. 3. Mapa de Disparidades. I. Orientador. II. Universidade Federal de São Carlos. III. Título.

Ficha catalográfica elaborada pelo Programa de Geração Automática da Secretaria Geral de Informática (SIn).

DADOS FORNECIDOS PELO(A) AUTOR(A)



UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado do candidato Carlos Willian de Carvalho, realizada em 17/02/2017:

Emerson Carlos Pedrino

Prof. Dr. Emerson Carlos Pedrino
UFSCar

José Hiroki Saito

Prof. Dr. José Hiroki Saito
UFSCar

Celso Olivete Júnior

Prof. Dr. Celso Olivete Júnior
UNESP

À minha mãe Neuza, minha irmã Juliana, e ao meu finado pai Luiz.

AGRADECIMENTOS

À minha mãe que, entre sorrisos e choros, sempre apoiou minhas escolhas; e a meu pai (*in memoriam*), que sempre me incentivou a tentar dar o melhor de mim.

À minha irmã, que quebrou meu galho sempre que precisei pegar um ônibus quatro horas da manhã para chegar a São Carlos (SP). Te devo uma (ou várias).

Ao meu orientador, Prof. Dr. Emerson Carlos Pedrino, pelo apoio e amizade, além de sua competência e atenção (além dos cafés pagos), que foram fatores fundamentais na conclusão deste trabalho. Quando eu crescer, quero ser igual a você!

Aos meus amigos de Santo Anastácio (SP) e região, que sempre estiveram do meu lado quando precisei – mesmo com mais de 400km de distância nos separando.

À galera da república Etanóis (e agregados), que se tornaram grandes amigos, e fizeram a minha vida em São Carlos mais fácil. Vocês são minha segunda família.

A todos os meus amigos e colegas de departamento que, direta ou indiretamente, contribuíram para que eu chegasse aonde estou hoje. Não vou marcar nomes porque poderia soar falso, e correria o risco de esquecer alguém, mas eles sabem que são especiais pra mim. Espero que tenham sucesso nessa caminhada!

À Universidade Federal de São Carlos, em especial ao pessoal do Departamento de Computação, por todo o apoio prestado (e por ter a melhor máquina de café do mundo).

E, por fim, à Coordenação de Aperfeiçoamento de Pessoas de Nível Superior (CAPES), pelo auxílio prestado por meio de bolsa (processo nº 1340099).

*And during the struggle
they will pull us down
but please, please
let use this chance to
turn things around...
And tonight, we can truly say:
together we're invincible.*

Muse, "Invincible"

RESUMO

O objetivo deste trabalho foi a análise, a combinação e os ajustes de um conjunto de algoritmos de estados da arte para desenvolver uma metodologia prática, para representar a profundidade de elementos presentes em uma cena, pelo cálculo do mapa de disparidades desta. Tal mapa é obtido de forma automática, a partir de ambientes reais, com o mínimo possível de intervenção humana. Inicialmente, os cantos dos objetos da cena analisada são obtidos utilizando o algoritmo SURF em um par de imagens estéreo obtidas a partir da mesma. Esses cantos são utilizados para obter um mapeamento de todos os pontos de uma imagem para os respectivos correspondentes na outra imagem, por meio de uma matriz fundamental (obtida por aproximação linear, utilizando *Least Median of Squares* ou MSAC); depois, processos de retificação e de correspondência são realizados, gerando-se o mapa, que passa por um pós-processamento para a correção de distorções. Os mapas inicialmente obtidos foram analisados em relação a tempo de processamento e similaridade com mapas obtidos a partir de um método manual (que usa calibração). Futuramente, pretende-se utilizar os algoritmos estudados como base para um sistema que permita que deficientes visuais possam realizar testes de navegação espacial em ambientes reais.

Palavras-chave: Visão estéreo, matriz fundamental, mapa de disparidades, autocalibração.

ABSTRACT

The objective of this work was the analysis, combination and adjustments of an algorithm set, in state-of-the-art, building a simple methodology, that is able to represent the depth of the elements in a scene, through the calculation of the disparity map. The map is automatically obtained, from real scenes, with minimal human intervention. Initially, the corners of the objects in a scene are obtained by the SURF algorithm in an image stereo pair, obtained from it. These corners are used to obtain a mapping from points in an image to its correspondences into another image, through a fundamental matrix (obtained by linear estimation, using *Least Median of Squares* or MSAC); later, the rectification and matching processes are done, generating the map, which suffers a post-processing to fix the distortions. The maps initially obtained was analysed in order to verify the processing time and the similarity with maps obtained by a manual method (which uses calibration). In the future, it is intended to use the studied algorithms as the basis for a system that will allow blind people to do spatial navigation tests in real environments.

Keywords: Stereo vision, fundamental matrix, disparity map, autocalibration.

LISTA DE FIGURAS

| | | |
|------|--|----|
| 1.1 | Representação do alfabeto tradicional utilizando-se o sistema Braile. | 18 |
| 2.1 | Geração de uma imagem estereoscópica pelo cérebro humano. | 22 |
| 2.2 | Representação espacial de um sistema de visão estéreo. | 23 |
| 2.3 | Projeção do modelo do sistema de visão estéreo para $y = 0$ | 23 |
| 2.4 | Representação do processo de mapeamento de coordenadas tridimensionais para bidimensionais com a calibração. | 25 |
| 2.5 | Geometria epipolar de um sistema estéreo. | 28 |
| 2.6 | Representação da aplicação da retificação planar a um par de imagens. | 29 |
| 2.7 | Esquematisação da retificação cilíndrica para uma imagem. | 30 |
| 2.8 | Cálculo de correspondência entre pontos: um conjunto de pontos que possui a mesma intensidade (ou cor) em ambas as imagens, está a uma mesma distância y do eixo x da origem, e são sequencialmente equivalentes, provavelmente representam o mesmo objeto. Imagens originais extraídas da base de dados Tsukuba (MARTULL; PERIS; FUKUI, 2012). | 32 |
| 2.9 | Tipos de métodos de cálculo de correspondência. As linhas superiores representam um método local, onde a janela de referência (em cinza mais escuro) é reprojeta em ambas as imagens, de modo que a busca por cada correspondência é limitada a esta área; as linhas inferiores representam um método global, onde a janela de referência é a própria linha analisada. | 33 |
| 2.10 | Imagem original e o seu mapa de disparidades correspondente; tons mais claros de cinza indicam objetos mais próximos ao observador. Imagens extraídas da base de dados Tsukuba (MARTULL; PERIS; FUKUI, 2012). | 34 |

| | | |
|------|--|----|
| 2.11 | Esquematisação do processo de geração de um mapa de disparidades passando pelas prováveis três etapas: calibração, retificação e correspondência. | 35 |
| 3.1 | Os pontos de partida para o processo de geração de um mapa de disparidades são os “cantos” dos objetos presentes em cada imagem obtida da cena (representados pelos pontos pretos na imagem). | 37 |
| 3.2 | Correspondências incorretas (representadas em vermelho) entre os cantos de um par de imagens estéreo, que podem ser removidas com o auxílio da matriz fundamental. | 40 |
| 3.3 | Graficamente, o objetivo da matriz fundamental é aproximar a relação entre pontos em um par de imagens estéreo com uma curva. | 40 |
| 3.4 | Representação das janelas utilizadas em um processo de correspondência e/ou disparidade. | 44 |
| 3.5 | Imagem extraída da base de dados Tsukuba (MARTULL; PERIS; FUKUI, 2012), e suas bordas detectadas pelo filtro Sobel. | 46 |
| 4.1 | Modelo da <i>webcam</i> Minoru3D, utilizada neste trabalho. | 50 |
| 4.2 | Um par de imagens capturadas pelo sistema de visão estéreo utilizado neste trabalho. | 52 |
| 4.3 | Representação em blocos do método desenvolvido até o pré-processamento das imagens de entrada. | 52 |
| 4.4 | Conjunto de imagens (da câmera esquerda) capturadas pelo sistema de visão estéreo, utilizadas para a análise dos algoritmos de detecção de cantos. | 53 |
| 4.5 | Gráfico que Apresenta a Taxa Aproximada de Pontos Correspondentes por Cena (%), para cada Detector de Cantos. | 56 |
| 4.6 | Representação em blocos do método desenvolvido até o cálculo dos cantos das imagens de entrada. | 57 |
| 4.7 | Representação em blocos do método desenvolvido até o cálculo das correspondências iniciais. | 57 |
| 4.8 | As correspondências que estão a uma distância muito grande da aproximação linear criada pela matriz fundamental (representadas pelos pontos brancos) não podem ser mapeadas por tal matriz, e são eliminadas durante o processo. | 59 |

| | | |
|------|---|----|
| 4.9 | Gráfico que Representa a Taxa Aproximada de Pontos Correspondentes Aproximáveis pela Matriz Fundamental (%), para cada Métrica de Qualidade (o gráfico da Cena 8 foi omitido por não ter gerado nenhum ponto correspondente aproximável). | 60 |
| 4.10 | Representação em blocos do método desenvolvido até o cálculo da matriz fundamental. | 62 |
| 4.11 | Par de imagens estéreo, com os respectivos pontos correspondentes representados, antes do processo de retificação. | 63 |
| 4.12 | Representação em blocos do método desenvolvido até o processo de retificação. | 64 |
| 4.13 | Par de imagens estéreo, com os respectivos pontos correspondentes representados, após a retificação planar. Percebe-se que os pontos correspondentes estão alinhados em relação ao eixo y das imagens. | 65 |
| 4.14 | Representação em blocos do método desenvolvido até o processo de cálculo do mapa de disparidades. | 66 |
| 4.15 | Mapa de disparidades obtido para o par de imagens da Figura 4.2: tons mais claros de cinza indicam áreas mais próximas do observador. | 67 |
| 4.16 | Mapa de disparidades da Figura 4.15 antes e depois da aplicação da transformada corretora: é possível observar que a área de distorção preta, herdada do processo de retificação, e presente do lado direito do mapa, foi removida. | 67 |
| 4.17 | Uma das imagens originais capturadas pelo sistema de visão estéreo (a), e o mapa de disparidades final, com todas as operações realizadas (b). | 68 |
| 4.18 | Representação em blocos final do método desenvolvido neste trabalho. | 69 |
| 5.1 | Um exemplo de inconsistência presente no cálculo do mapa de disparidades (canto superior direito da Figura (b)): uma região de textura única pode criar valores errados de disparidade no mapa. | 71 |
| 5.2 | Diferentes mapas de disparidades obtidos, por diferentes matrizes fundamentais, para uma mesma cena capturada pelo sistema de visão estéreo. | 73 |
| 5.3 | Mapas de disparidades de uma cena, que podem ser produzidos limitando-se a área de tons de cinza indefinidos em uma porcentagem máxima. | 74 |
| B.1 | Mapa de Disparidades Obtido: Bicycle2 | 91 |

| | | |
|------|---|----|
| B.2 | Mapa de Disparidades Obtido: Computer | 92 |
| B.3 | Mapa de Disparidades Obtido: Crusade | 92 |
| B.4 | Mapa de Disparidades Obtido: Livingroom | 92 |
| B.5 | Mapa de Disparidades Obtido: Plants | 93 |
| B.6 | Mapa de Disparidades Obtido: Djembe | 93 |
| B.7 | Mapa de Disparidades Obtido: Newkuba | 93 |
| B.8 | Mapa de Disparidades Obtido: Australia | 94 |
| B.9 | Mapa de Disparidades Obtido: Classroom2 | 94 |
| B.10 | Mapa de Disparidades Obtido: Hoops | 94 |
| C.1 | Mapas de Disparidades Obtidos: Cena 01 | 95 |
| C.2 | Mapas de Disparidades Obtidos: Cena 02 | 95 |
| C.3 | Mapas de Disparidades Obtidos: Cena 03 | 96 |
| C.4 | Mapas de Disparidades Obtidos: Cena 04 | 96 |
| C.5 | Mapas de Disparidades Obtidos: Cena 05 | 96 |
| C.6 | Mapas de Disparidades Obtidos: Cena 06 | 96 |
| C.7 | Mapas de Disparidades Obtidos: Cena 07 | 97 |
| C.8 | Mapas de Disparidades Obtidos: Cena 08 | 97 |
| C.9 | Mapas de Disparidades Obtidos: Cena 09 | 97 |
| C.10 | Mapas de Disparidades Obtidos: Cena 10 | 97 |

LISTA DE TABELAS

| | | |
|-----|--|----|
| 4.1 | Número de Cantos Obtidos nas Imagens Esquerda (E) e Direita (D) de cada Cena, para cada Detector de Cantos | 54 |
| 4.2 | Número de Pontos Correspondentes por Cena, para cada Detector de Cantos | 55 |
| 4.3 | Taxa Aproximada de Pontos Correspondentes por Cena (%), para cada Detector de Cantos | 56 |
| 4.4 | Número de Pontos Correspondentes Aproximáveis pela Matriz Fundamental, para cada Métrica de Qualidade | 59 |
| 4.5 | Taxa Aproximada de Pontos Correspondentes Aproximáveis pela Matriz Fundamental (%), para cada Métrica de Qualidade | 60 |
| 5.1 | Tempo Estimado para o Processamento de um Mapa de Disparidades (em Segundos) | 75 |
| 5.2 | Taxa de Similaridade Obtida entre os Métodos Desenvolvidos e o de Zhang (2000) | 77 |

SUMÁRIO

| | |
|--|-----------|
| CAPÍTULO 1 – INTRODUÇÃO | 16 |
| 1.1 Tecnologia e Acessibilidade | 16 |
| 1.2 Deficiência Visual | 16 |
| 1.3 Tecnologias Atuais | 17 |
| 1.4 Motivação de Pesquisa | 19 |
| 1.5 Objetivo Principal deste Trabalho | 20 |
| 1.6 Descrição dos Capítulos | 20 |
| | |
| CAPÍTULO 2 – VISÃO ESTÉREO | 21 |
| 2.1 Definição | 21 |
| 2.2 Visão Estéreo Humana | 21 |
| 2.3 Sistemas de Visão Estéreo | 22 |
| 2.4 Etapas da Análise de Visão Estéreo | 24 |
| 2.4.1 Calibração do sistema | 25 |
| 2.4.2 Retificação das imagens | 27 |
| 2.4.3 Correspondência e disparidade entre pontos | 31 |
| 2.5 Esquematização do Desenvolvimento de um Mapa de Disparidades | 34 |
| | |
| CAPÍTULO 3 – ESTUDOS E CONCEITOS RELACIONADOS | 36 |
| 3.1 Considerações Iniciais | 36 |
| 3.2 Técnicas de Detecção de Cantos | 36 |

| | | |
|--|---|-----------|
| 3.3 | Métricas de Qualidade de Uma Matriz Fundamental | 39 |
| 3.4 | Tipos de Métricas de Correspondência | 43 |
| 3.5 | Detecção de Descontinuidades (Bordas): o filtro Sobel | 45 |
| 3.6 | Outros Estudos sobre a Calibração | 46 |
| 3.7 | Outros Estudos sobre o Mapa de Disparidades | 47 |
| CAPÍTULO 4 – DESENVOLVIMENTO DO TRABALHO | | 48 |
| 4.1 | Ideia Utilizada | 48 |
| 4.2 | Modelo de Sistema de Visão Estéreo | 49 |
| 4.3 | Ambiente de Testes | 50 |
| 4.4 | Aquisição das Imagens | 51 |
| 4.5 | Pré-Processamento das Imagens | 51 |
| 4.6 | Detecção dos Pontos Correspondentes Iniciais | 52 |
| 4.6.1 | Obtendo o algoritmo que calcula mais cantos | 53 |
| 4.6.2 | Obtendo o algoritmo que calcula mais correspondências | 54 |
| 4.7 | Estimação da Matriz Fundamental | 58 |
| 4.8 | Etapa de Retificação | 62 |
| 4.9 | Cálculo das Disparidades entre Pontos | 64 |
| 4.10 | Correção do Mapa de Disparidades Obtido | 66 |
| CAPÍTULO 5 – ANÁLISE DOS RESULTADOS OBTIDOS | | 70 |
| 5.1 | Principal Objetivo da Análise | 70 |
| 5.2 | Interferência de Agentes Externos no Mapa | 70 |
| 5.3 | Problemas na Geração de um Mapa de Disparidades | 72 |
| 5.4 | Minimizando Distorções no Mapa de Disparidades | 73 |
| 5.5 | Tempo de Geração de um Mapa de Disparidades | 74 |
| 5.6 | Análise Comparativa com o Método Calibrado de Zhang | 75 |

| | |
|--|-----------|
| CAPÍTULO 6 – CONCLUSÕES | 79 |
| 6.1 Principais Conclusões | 79 |
| 6.2 Possíveis Trabalhos Futuros | 79 |
| REFERÊNCIAS | 82 |
| GLOSSÁRIO | 88 |
| APÊNDICE A – ALGORITMO SIMPLIFICADO DA GERAÇÃO DO MAPA DE DISPARIDADES (SEM CALIBRAÇÃO) | 89 |
| APÊNDICE B – MAPAS DE DISPARIDADES OBTIDOS NO CÁLCULO DE TEMPO DE PROCESSAMENTO | 91 |
| APÊNDICE C – COMPARAÇÃO VISUAL: MÉTODO DESENVOLVIDO E O DE ZHANG | 95 |

Capítulo 1

INTRODUÇÃO

1.1 Tecnologia e Acessibilidade

De uma forma ampla, pode-se dizer que o desenvolvimento da tecnologia se dá com o objetivo de tornar a vida do ser humano mais fácil. Existem incontáveis ferramentas tecnológicas que foram desenvolvidas de modo a favorecer e simplificar atividades do nosso cotidiano como, por exemplo, celulares, *tablets*, computadores e automóveis; e tais ferramentas já foram assimiladas a nossa rotina de uma maneira tão intensa que, atualmente, abrir mão do uso destas é algo impensável.

Uma vez que facilitar o dia-a-dia do ser humano é uma das finalidades da tecnologia, é aceitável pensar que ela pode ser utilizada, também, para facilitar a vida de grupos de pessoas que costumam possuir dificuldades diárias – como os deficientes, por exemplo. Além de serem, eventualmente, vítimas de preconceito e do descaso, poucas atividades do cotidiano são adaptadas para lhes serem úteis.

Dentre os vários tipos de deficiência que existem nos seres humanos, é possível destacar uma: a deficiência visual.

1.2 Deficiência Visual

A deficiência visual é definida como a perda total ou parcial, congênita ou adquirida, da visão. Basicamente, existem vários graus distintos de tal tipo de deficiência; no entanto, baseado em um estudo realizado em 1972, a Organização Mundial da Saúde (OMS) faz uma distinção, utilizada até hoje, entre os dois grandes grupos de deficiência visual: pessoas com baixa visão e pessoas com cegueira. Segundo a OMS:

“Uma pessoa com baixa visão é aquela que tem um comprometimento da função visual mesmo após o tratamento e/ou a correção da refração padrão, e tem uma capacidade visual de menos de 6/18 da percepção de luz, ou um campo visual de menos de 10 graus de um ponto de fixação, mas que utiliza, ou é potencialmente apta a utilizar, a visão para o planejamento e/ou a execução de uma tarefa.” (World Health Organization, 2011)

Já a cegueira é definida pela OMS de forma um pouco mais simples:

“A definição atual [de cegueira] não faz uma distinção entre aqueles que têm cegueira ‘irreversível’ (nenhuma percepção de luz) e aqueles que têm percepção de luz, mas ainda esta sendo menor que 3/60 no melhor olho.” (World Health Organization, 2011)

De acordo com o último censo realizado pelo Instituto Brasileiro de Geografia e Estatística (IBGE), no ano de 2010, cerca de 18,6% da população brasileira apresentava algum grau de deficiência visual, sendo que 3,46% da população brasileira apresentava deficiência visual severa (OLIVEIRA, 2012). Considerando-se que a porcentagem da população brasileira que apresenta qualquer tipo de deficiência (visual, auditiva, motora ou mental/intelectual) é um pouco maior que este valor (23,9%), pode-se concluir que a deficiência visual é o principal tipo de deficiência presente no país. Ainda com base no censo supracitado, a deficiência visual afeta 49,8% das pessoas com mais de 65 anos de idade; ou seja, praticamente a metade da população de terceira idade do país sofre com algum grau de deficiência visual (OLIVEIRA, 2012).

Já mundialmente, a OMS afirma que, também no ano de 2010, 285 milhões de pessoas possuíam algum tipo de deficiência visual. Destes número, 39 milhões de pessoas apresentavam cegueira e, dentro deste último grupo, 82% destas possuíam mais de 50 anos de idade (World Health Organization, 2014).

1.3 Tecnologias Atuais

Há atualmente muitas ferramentas capazes de permitir uma melhor qualidade de vida às pessoas portadoras de algum tipo de deficiência visual. Um dos melhores exemplos é o uso da linguagem Braille, desenvolvida por Louis Braille (1809 – 1852) em 1825. Tal sistema consiste do arranjo de seis pontos em relevo, dispostos em duas colunas com três pontos cada. A forma como estes pontos são distribuídos determina o tipo de caractere que eles representam, como pode ser visto na Figura 1.1.

No âmbito de tecnologias mais avançadas, podem-se citar exemplos tais como as técnicas de *Optical Character Recognition*, que são capazes de detectar os caracteres presentes em uma

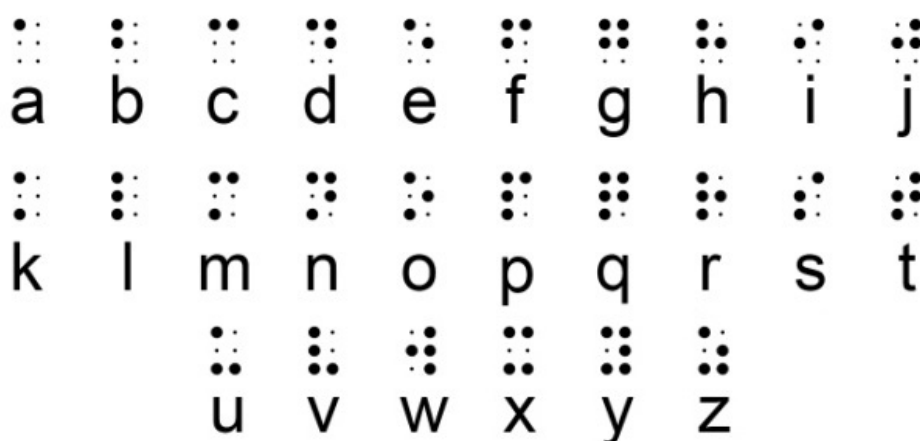


Figura 1.1: Representação do alfabeto tradicional utilizando-se o sistema Braille.

imagem, transmitindo-os a um dispositivo, e posteriormente reproduzindo-os em áudio (CAMPOS; BABU; VARMA, 2009). Além disso, para o uso de ambientes digitais mais complexos (como computadores), existem *softwares* voltados exclusivamente para deficientes visuais como, por exemplo (SONZA; SANTAROSA, 2003):

- O sistema operacional *DOSVOX* (Núcleo do Computação Eletrônica, 2002), desenvolvido no Brasil, que se comunica com o usuário utilizando síntese de voz, além de possuir recursos como impressor/formatador para braille e ampliador de tela para pessoas com visão reduzida, por exemplo;
- O *software Virtual Vision* (MicroPower, 2014), também desenvolvido no Brasil, que consiste em um leitor de telas capaz de informar ao usuário quais os controles (como botões, listas e menus) estão ativos em um determinado momento;
- O *software Jaws* (Freedom Scientific, Inc., 2015), desenvolvido nos Estados Unidos, que além de ser um leitor de telas (incluindo a capacidade de ler páginas da Internet) nos moldes do *Virtual Vision*, permite o uso de teclas de atalho;
- O *software Dirce* (Acessibilidade Brasil, 2014), desenvolvido no Brasil, utilizado para a reprodução via áudio de livros digitais.

Dessa forma, pode-se dizer que existe um amplo campo para desenvolvimento de *softwares* que sejam capazes de promover a inclusão de pessoas com deficiência visual.

1.4 Motivação de Pesquisa

Existem várias pesquisas tecnológicas em que se busca reproduzir funções existentes em seres vivos na natureza, como a representação de emoções na face de robôs (HARA, 2004), a audição (TOURBABIN; RAFAELY, 2013), a fala (NEO; SAKAGUCHI; YOKOI, 2008) e o aprendizado (WERSING et al., 2007). Já em relação ao aspecto da visão, existem na literatura sistemas computacionais capazes de reproduzir características específicas como, por exemplo, a capacidade de reconhecimento de padrões (MESQUITA; MELLO; ALMEIDA, 2014), de sinais que representem informações (CORONADO et al., 2012), de pessoas presentes em uma cena (LI et al., 2013) ou do mapeamento de uma cena em relação ao observador (CHUMERIN et al., 2010; PAUWELS; HULLE, 2012).

Em relação ao mapeamento de uma cena, um dos assuntos mais estudados atualmente é a análise da disparidade dos elementos de uma cena observada. A disparidade é uma medida capaz de quantificar o quanto um elemento presente em uma cena está distante do observador. Os estudos mais antigos sobre tal utilizavam-se de alguns objetos específicos, que eram integrados à cena original, para realizar tal análise; um desses sistemas foi proposto por Zhang (ZHANG et al., 1995; ZHANG, 2000), e é conhecido por utilizar um objeto (no caso, um tabuleiro de xadrez) para se realizar o mapeamento da profundidade dos elementos de uma cena real qualquer. Atualmente, existem métodos capazes de realizar o mesmo procedimento sem a dependência de um elemento externo ou de um conhecimento prévio da cena – como os apresentados pela base de dados de Middlebury (SCHARSTEIN; SZELISKI, 2002; SCHARSTEIN et al., 2014), por exemplo.

Dessa forma, uma vez que há várias pesquisas envolvendo a reprodução de aspectos da visão humana e focando-se na área de mapeamento de uma cena, é natural pensar que uma técnica capaz de mapear a disparidade de elementos em uma cena possa ser utilizada com o objetivo de se desenvolver um sistema para auxiliar um deficiente visual em tarefas do dia-a-dia, mapeando distâncias entre este e os objetos de um ambiente. Porém, dentro desse contexto, deve-se levar em conta:

1. Se o sistema é rápido o suficiente. Se um método utilizado exige um alto tempo de processamento, ele é inviável para uso real;
2. Se o sistema sabe quantificar a distância entre os elementos de uma cena e o observador dela. É interessante dizer não apenas o quê está perto, mas o quanto está perto;
3. Se o sistema precisa conhecer exatamente a forma que um objeto observado em uma cena possui, ou se apenas é necessário dizer se há um objeto próximo ou não;

4. Se o sistema pode ser embarcado em um *hardware* pequeno (e possivelmente leve), de modo que uma pessoa comum possa manuseá-lo facilmente;
5. Se o sistema utiliza *hardware* de baixo custo, de modo a viabilizar uma possível utilização em larga escala.

Ou seja, um sistema útil em aplicações reais para deficientes visuais precisa ser adaptado às limitações dos mesmos: ser rápido, suficientemente simples e de baixo custo.

1.5 Objetivo Principal deste Trabalho

O principal objetivo deste trabalho consistiu na análise, combinação e ajustes entre um conjunto de algoritmos, todos em estado da arte, com a finalidade de obter uma metodologia simples que seja capaz de representar a profundidade de elementos que estão presentes em uma cena. A profundidade dos elementos da cena é representada por meio de um mapa de disparidades, e o cálculo de tal mapa precisa ser realizado de forma automática, em ambientes reais, com o mínimo possível de intervenção humana.

O escopo deste trabalho abrange a geração de mapas de disparidades em *software*, além de alguns tratamentos posteriores. Também são discutidas questões como a análise do tempo de processamento dos mesmos, a comparação dos resultados obtidos pela metodologia apresentada com os resultados obtidos por outra metodologia, e a análise de possíveis problemas do funcionamento de tal metodologia em tempo real.

A ideia deste trabalho é que, devido ao fato do conjunto de algoritmos desenvolvido não apresentar alta complexidade, além de possuir um grau satisfatório de eficiência, em possíveis projetos futuros estes possam ser embarcados em *hardware*, a fim de que pessoas com algum tipo de deficiência visual possam realizar testes de navegação espacial com eles.

1.6 Descrição dos Capítulos

Este trabalho se divide da seguinte maneira: no Capítulo 2 é introduzida a parte teórica sobre a área de visão estéreo, explicando suas etapas. No Capítulo 3, apresentam-se estudos diretamente relacionados com o desenvolvimento do trabalho a ser apresentado. Já no Capítulo 4 mostra-se as etapas de desenvolvimento do trabalho, enquanto no Capítulo 5 é apresentada uma análise sobre os resultados obtidos, e as técnicas para a melhoria dos mesmos. Por fim, as conclusões sobre o trabalho, bem como aspectos futuros, são apresentadas no Capítulo 6.

Capítulo 2

VISÃO ESTÉREO

2.1 Definição

A visão estéreo (também chamada de visão estereoscópica) é uma das principais formas de fornecer à visão humana informações de profundidade. Ela possui tal nome devido ao fato de precisar do uso de ambos os olhos para a formação tridimensional da imagem. A partir das duas imagens levemente diferentes, obtidas uma de cada olho, a noção de profundidade na cena observada pode ser construída pelo cérebro.

Em computação, a visão estéreo é um ramo de visão computacional que procura reproduzir tal processo artificialmente por meio do uso de câmeras, com o auxílio de *hardware* e/ou *software*. Seu objetivo é, a partir de um par de imagens obtidas simultaneamente, buscar a reconstrução da informação tridimensional dos objetos da cena, assim como a visão humana é capaz de realizar.

2.2 Visão Estéreo Humana

A visão tridimensional que o ser humano possui é construída pelo cérebro por meio da interpretação de duas imagens bidimensionais levemente distintas, capturadas pelos olhos a partir de um ponto de vista, além de informações sobre o grau de convergência e divergência destes em relação a cada elemento das imagens. A partir daí, o cérebro consegue “unir” as duas imagens em uma única, como pode ser visto na Figura 2.1. Diferentemente de efeitos de imagem, como perspectiva, iluminação, oclusão, sombra e gradiente, a estereoscopia não está presente diretamente em uma imagem, mas sim sendo interpretada pelo nosso cérebro (SISCOOTTO et al., 2004).

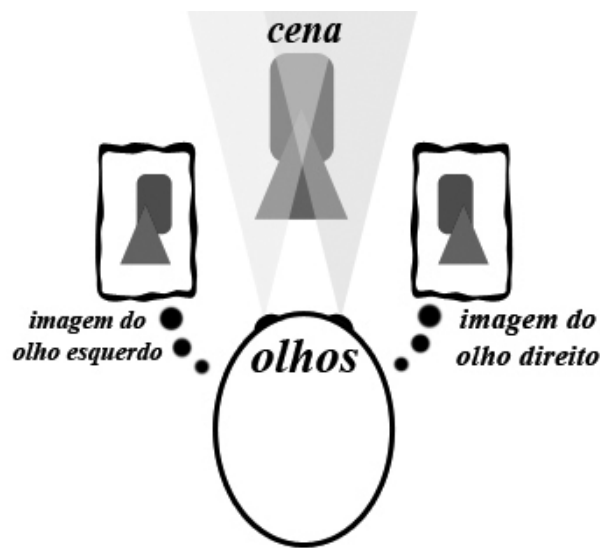


Figura 2.1: Geração de uma imagem estereoscópica pelo cérebro humano.

2.3 Sistemas de Visão Estéreo

Do ponto de vista da computação, o objetivo de qualquer sistema de visão estéreo é combinar duas (ou mais) imagens, a fim de ser possível a identificação de pontos cuja origem seja um mesmo elemento da cena observada. Os pontos que satisfazem tal condição são chamados de pontos correspondentes e, a partir de tais pontos, pode-se calcular o valor da disparidade entre eles – definida como a métrica que representa o deslocamento entre as coordenadas de dois pontos correspondentes (ROMA; VICTOR; TOME, 2002). O processo de encontrar pontos correspondentes em uma cena pode ser altamente complexo, devido a questões como a presença de ruídos nas imagens originais, além de descontinuidades e oclusões de elementos que estão presentes, principalmente em ambientes reais (não controlados).

Na Figura 2.2 é apresentado um modelo padrão de sistema de visão estéreo. Essa exibe um ponto $P(x, y, z)$, que é observado por duas câmeras, definidas como esquerda e direita (E e D , respectivamente), que possuem eixos y paralelos entre si, com P_e e P_d representando o ponto P projetado nos planos definidos pelos eixos ópticos de cada câmera. Tais planos são ortogonais aos segmentos definidos pelas suas respectivas câmeras e o ponto de convergência focal O , comum a estas (GROSSO; TISTARELLI, 1995).

Já a Figura 2.3 exibe os parâmetros de interesse do modelo da Figura 2.2, para o eixo y nulo nos planos de imagem. Sabendo-se os ângulos formados pelas relações entre as câmeras, o ponto de convergência focal O e o ponto P , pode-se definir uma função K , dada pela Equação 2.1 (GROSSO; TISTARELLI, 1995):

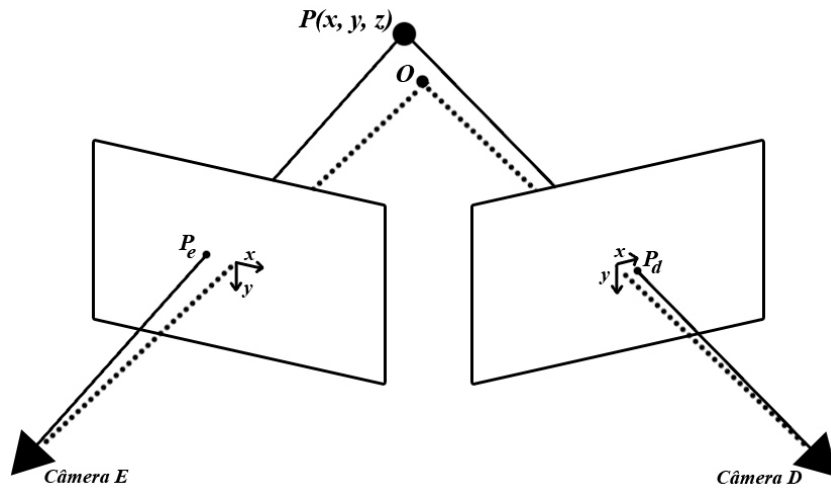


Figura 2.2: Representação espacial de um sistema de visão estéreo.

$$K(\alpha, \beta, \gamma, \delta) = \frac{\tan(\alpha - \gamma) \cdot \tan(\beta + \delta)}{\tan(\alpha - \gamma) + \tan(\beta + \delta)} \quad (2.1)$$

em que α e β correspondem aos ângulos de convergência das câmeras esquerda e direita, respectivamente, e γ e δ definem a disparidade angular de dois pontos correspondentes nos planos de imagem f_e e f_d , onde a posição relativa de tais pontos difere de acordo com a relação $f_d = f_e + d$, com d sendo o parâmetro de disparidade a ser obtido.

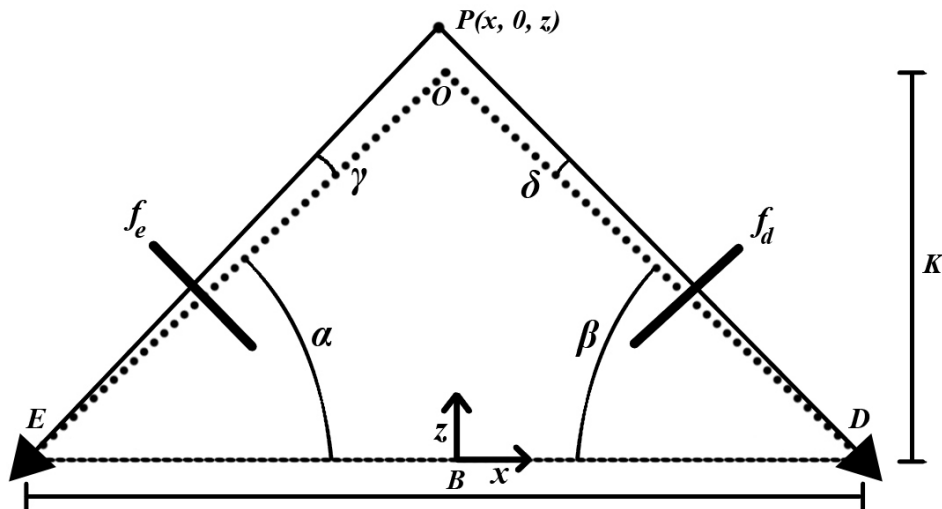


Figura 2.3: Projeção do modelo do sistema de visão estéreo para $y = 0$.

Pode-se equacionar os parâmetros γ e δ como mostram as Equações 2.2 e 2.3, respectivamente (GROSSO; TISTARELLI, 1995):

$$\gamma = \arctan\left(\frac{f_e}{OE}\right) \quad (2.2)$$

$$\delta = \arctan\left(\frac{f_d}{OD}\right) \quad (2.3)$$

em que \overline{OE} e \overline{OD} definem as distâncias focais das câmeras esquerda e direita, respectivamente, em *pixels*. A partir disso, pode-se mostrar que a profundidade do ponto P , em relação ao eixo Z do sistema de coordenadas, pode ser escrita em função dos demais parâmetros citados, além de um parâmetro B fixo (definido como a distância-base entre as câmeras), conforme a Equação 2.4 (GROSSO; TISTARELLI, 1995):

$$Z = B \cdot K(\alpha, \beta, \gamma, \delta) = B \cdot \frac{\tan(\alpha - \gamma) \cdot \tan(\beta + \delta)}{\tan(\alpha - \gamma) + \tan(\beta + \delta)} \quad (2.4)$$

Deve-se observar que a medição da profundidade referenciada no ponto O deve depender apenas dos ângulos de vergência das câmeras, uma vez que, neste ponto, $\gamma = \delta = 0$; tal fato pode ser observado ao se aplicar a Equação 2.1 na Equação 2.4 considerando essa condição, ou seja (GROSSO; TISTARELLI, 1995):

$$Z = B \cdot K(\alpha, \beta, 0, 0) = B \cdot \frac{\tan(\alpha) \cdot \tan(\beta)}{\tan(\alpha) + \tan(\beta)} \quad (2.5)$$

Grosso e Tistarelli (1995) definem que a análise estéreo é uma forma de se conseguir uma correlação entre as imagens obtidas da cena analisada, de modo a se obter o mapa de disparidades desta. Um mapa de disparidades (também chamado de espaço de disparidade da imagem) é, em suma, uma matriz contendo o mapeamento dos valores de disparidade de todos os pontos correspondentes de uma cena. A partir da análise de tal mapa, é possível definir algumas informações sobre a cena analisada, como a distância dos elementos presentes nesta em relação às câmeras, por exemplo.

2.4 Etapas da Análise de Visão Estéreo

Basicamente, existem três etapas essenciais no processo de geração de um mapa de disparidades de uma cena por um sistema: a calibração do sistema, onde a configuração dos parâmetros do sistema de visão estéreo é realizada; a retificação das imagens, onde estas são “alinhas”, de modo que pontos correspondentes das imagens da cena estejam na mesma coordenada na vertical; e o cálculo da correspondência entre os pontos das imagens, de modo que a disparidade entre eles possa ser quantificada. Tais etapas serão discutidas em detalhes a seguir.

2.4.1 Calibração do sistema

A calibração do sistema de visão estéreo tem por objetivo a obtenção dos parâmetros que tornarão possível o mapeamento das coordenadas tridimensionais do mundo em coordenadas bidimensionais de imagens, como mostra a Figura 2.4. Na maioria dos casos, o desempenho geral do sistema de visão estéreo depende, principalmente, da eficiência da calibração das câmeras (HEIKKILA; SILVEN, 1997).

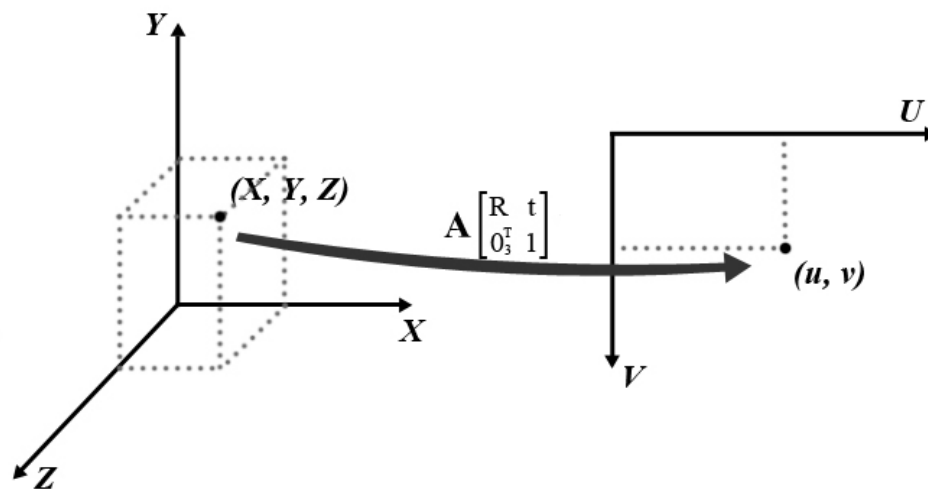


Figura 2.4: Representação do processo de mapeamento de coordenadas tridimensionais para bidimensionais com a calibração.

Existem dois tipos de parâmetros obtidos na calibração das câmeras, e tais parâmetros são utilizados de modo a se conseguir o devido alinhamento das imagens obtidas pelas câmeras (na etapa de retificação). Os **parâmetros intrínsecos** refletem a geometria interna de cada câmera do sistema de visão estéreo, além das características geométricas e ópticas desta (RATEKE et al., 2013). Tais parâmetros são:

- **Distância focal da câmera**, representada por f , que define a distância entre o ponto nodal interior de cada lente das câmeras do sistema e seus respectivos planos de imagem;
- **Fatores de escala**, representados por s_x e s_y , proveniente da diferença entre elementos sensores nas câmeras e *pixels* amostrados pelo *hardware* de digitalização da imagem;
- **Centro projetivo**, representado por (u_0, v_0) , que é basicamente a posição, em *pixels*, da projeção ortogonal do centro óptico (ou seja, a projeção ortogonal do ponto de foco comum das câmeras) no plano de projeção das imagens;
- **Imperfeições geométricas**, representadas pelo coeficiente τ , introduzidas pelo sistema

sensor da câmera. Em geral, as próprias fabricantes das câmeras do sistema fornecem essa informação.

Já os **parâmetros extrínsecos** fornecem a posição da origem do sistema de coordenadas da câmera em relação à origem das coordenadas do mundo real, de modo que seja possível descrever-se o movimento de uma cena estática em torno da câmera e vice-versa (RATEKE et al., 2013). Estes parâmetros são:

- **Matriz de rotação**, representada por R , de dimensões 3×3 , que define a rotação de uma cena em relação às câmeras do sistema;
- **Vetor de translação**, representado por t e pertencente ao espaço \mathfrak{R}^3 , que define o deslocamento de uma cena em relação às câmeras do sistema.

Tais parâmetros são combinados de modo a definir as condições de formação de uma imagem – ou seja, as transformações de pontos em três dimensões (X, Y, Z) , no mundo real, para pontos em duas dimensões (u, v) , na imagem. Matematicamente, o mapeamento entre coordenadas de três e duas dimensões obedece à Equação 2.6 (TARAGLIO; CHIESA, 2011):

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \cong \begin{bmatrix} f \cdot s_x & f \cdot \tau & u_0 & 0 \\ 0 & f \cdot s_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.6)$$

em que os elementos r_{ij} e t_i são os elementos da matriz de rotação R e do vetor de translação t , respectivamente. Em geral, costuma-se representar a matriz com os parâmetros intrínsecos por A , além de $f \cdot s_x$, $f \cdot s_y$ e $f \cdot \tau$ por α , β e γ , respectivamente. Dessa forma, a Equação 2.6 pode também ser representada como a Equação 2.7 (TARAGLIO; CHIESA, 2011):

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \cong A \begin{bmatrix} R & t \\ 0_3^T & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad A = \begin{bmatrix} \alpha & \gamma & u_0 & 0 \\ 0 & \beta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.7)$$

A forma como a calibração é realizada depende da natureza e da capacidade de precisão exigida pela aplicação. Em geral, os parâmetros de calibração costumam ser estimados tendo como base um conjunto de imagens previamente adquiridas de uma mesma cena, pelo sistema

de visão estéreo, sob diferentes “pontos de vista” deste (deslocando-se o sistema no espaço). A partir da análise das imagens adquiridas, e em relação à forma como os parâmetros são estimados, pode-se classificar as técnicas de calibração como sendo lineares ou não-lineares (REMONDINO; FRASER, 2006).

As **técnicas lineares** fazem a estimação dos parâmetros do sistema utilizando aproximações lineares. São mais simples e rápidas mas, em geral, incapazes de contornar problemas de distorção em lentes, além de precisarem de uma sequência de pontos de coordenadas previamente conhecidas, e simplificarem o modelo do sistema de visão estéreo, o que pode levar à baixa confiabilidade dos resultados (REMONDINO; FRASER, 2006). A canônica técnica de transformação linear direta, desenvolvida por Abdel-Aziz e Karara (1971), pode ser utilizada para se fazer uma estimação linear dos parâmetros de calibração, e é uma das principais técnicas lineares conhecida.

Já as **técnicas não-lineares** calculam cada parâmetro de forma independente. Elas possuem a vantagem de não dependerem de conhecimento prévio do sistema – mas exigem uma modelagem mais rigorosa e altamente precisa de entrada e saída da câmera, além do conhecimento do parâmetro de distorção de sua lentes. Uma das técnicas não-lineares mais conhecidas é a de Tsai (1987), onde cada parâmetro é calculado por vez, utilizando métodos como transformações rígidas de coordenadas e a análise da projeção perspectiva das câmeras do sistema.

Além disso, é possível aplicar **técnicas lineares e não-lineares em conjunto**, onde o método linear serve para uma aproximação inicial dos parâmetros de calibração, e em seguida a calibração sejam refinadas através de técnicas não-lineares, como as utilizadas por Weng, Cohen e Herniou (1992) ou Heikkila e Silven (1997), em que a calibração acaba se tornando um processo de duas fases.

Um dos principais problemas da calibração convencional é a exigência de alta precisão nas medições dos objetos nas coordenadas de mundo, nas medições dos pontos correspondentes na imagem ou nos deslocamentos realizados pela câmera. Além disso, os parâmetros intrínsecos encontrados representam a configuração atual da câmera – caso estes sejam alterados, é necessária uma nova calibração, o que pode não ser possível em algumas situações.

2.4.2 Retificação das imagens

A próxima etapa do processo é a retificação, que tem como base de execução a geometria epipolar da cena analisada. Tal geometria é independente da estrutura de tal cena, dependendo somente da posição relativa das câmeras e de seus parâmetros internos (HARLEY; ZISSERMAN,

2004). A geometria epipolar de uma cena capturada por um sistema de visão estéreo envolve o centro da projeção O_i de cada câmera i do sistema, os planos Π_i das imagens capturadas por cada uma destas e um ponto P presente na cena (TARAGLIO; CHIESA, 2011), como pode ser observado na Figura 2.5. A partir daí, pode-se definir as seguintes entidades (HARLEY; ZISSERMAN, 2004):

- **Linha de base**, representada por b , que é a linha que une o centro de projeção das câmeras entre si;
- **Epipolo**, representada por e_i , definida como o ponto de intersecção da linha de base e um plano de imagem;
- **Plano epipolar**, representado por Π , que contém a linha de base;
- **Linha epipolar**, representada por l_i , formada pela intersecção entre um plano epipolar e o plano da imagem.

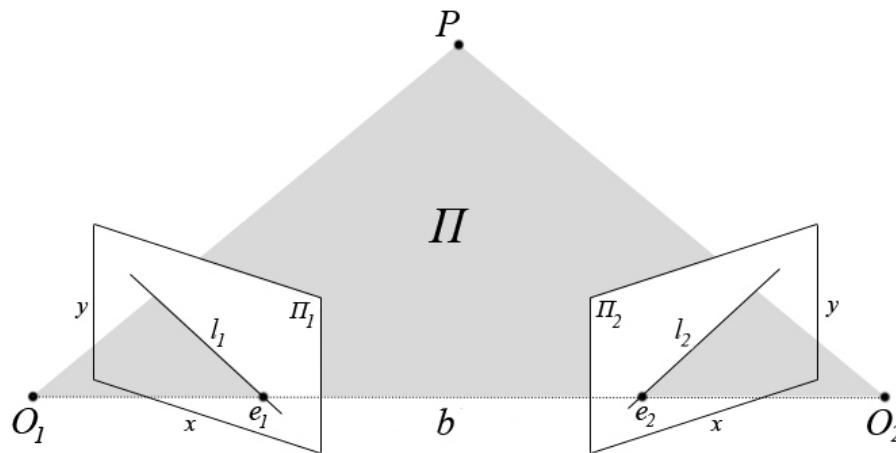


Figura 2.5: Geometria epipolar de um sistema estéreo.

Considerando-se duas imagens de uma mesma cena tridimensional, obtidas pelas câmeras do sistema e seus diferentes pontos de visão, em geral existem divergências entre as posições dos pontos correspondentes (principalmente em relação às alturas deles em ambas as imagens), que precisam ser corrigidas. A retificação tem por objetivo a aplicação de transformações (que podem ser produzidas a partir dos parâmetros obtidos no processo de calibração, se necessário) a um conjunto de imagens estéreo, de modo que as projeções epipolares delas se tornem correspondentes – ou seja, as imagens são manipuladas com o objetivo de que linhas epipolares correspondentes se tornem colineares e paralelas ao eixo x da imagem. Atualmente, existem dois tipos de retificação que podem ser utilizadas: a planar e a cilíndrica.

A **retificação planar** busca reprojeter as imagens em um plano em comum, paralelo à linha de base, de modo que o mapeamento em uma região em comum deste plano garanta linhas epipolares correspondentes com a mesma altura, como pode ser visto na Figura 2.6, onde I_i é uma das imagens originais, enquanto I'_i é a mesma imagem, mas retificada. Existem infinitos planos paralelos à linha de base mas, em geral, utiliza-se um a partir da rotação dos planos de imagem, de forma que eles se tornem coplanares. Para efetuar tal retificação, é necessário o conhecimento dos parâmetros intrínsecos de ambas as câmeras, que permitirão que uma nova matriz de rotação, além de novos parâmetros intrínsecos (de iguais valores para ambas as câmeras), sejam gerados (FUSIELLO; TRUCCO; VERRI, 2000).

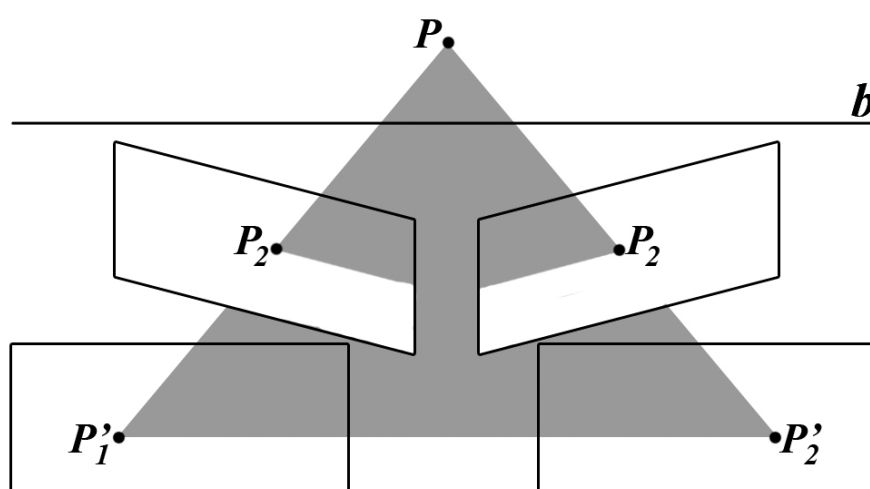


Figura 2.6: Representação da aplicação da retificação planar a um par de imagens.

Em geral, métodos de retificação planar são simples, mas costumam não ser robustos o suficiente para escaparem da influência de movimentos das câmeras. Daí, nesses casos, pode-se utilizar a **retificação cilíndrica**, que parte de um cilindro em comum, de raio unitário e com eixo de revolução na linha de base, para reprojeter cada imagem I_i em uma nova imagem I'_i (ROY; MEUNIER; COX, 1997). Ela mapeia cada *pixel* da imagem em uma coordenada do sistema de coordenadas cilíndrico, de modo que cada ponto da imagem retificada no cilindro possa ser usado como um ponto comum da imagem. Para este mapeamento, as linhas epipolares precisam ser rotacionadas até estarem paralelas à linha de base, para depois serem projetadas no cilindro. Como pode ser visto pela Figura 2.7, este tipo de retificação é mais complexa do que uma retificação planar, uma vez que envolve o uso de operações tridimensionais; no entanto, tais operações podem ser suprimidas se as informações da geometria tanto projetiva quanto epipolar do sistema forem utilizadas. Isso pode ser feito se as transformações necessárias para a retificação ocorrerem não no cilindro centrado na linha de base, mas sim ao redor dos epipolos (que estão no mesmo plano da imagem, evitando as operações no espaço tridimensional) (POLLEFEYS; KOCH; GOOL, 1999).

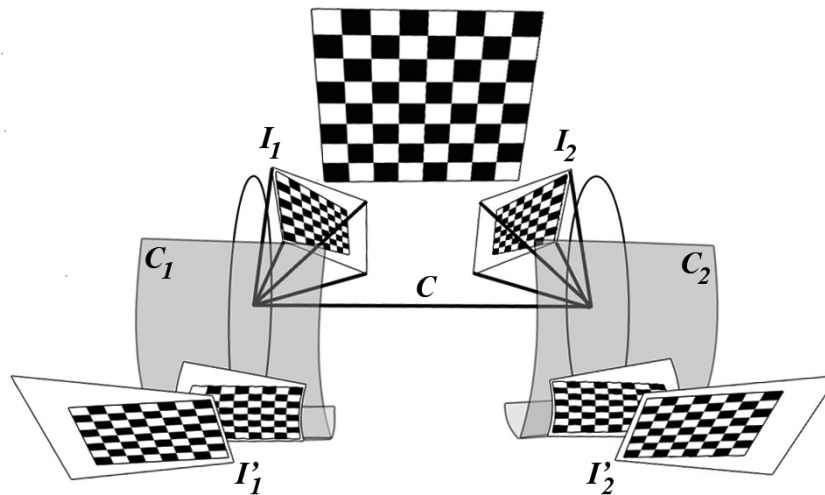


Figura 2.7: Esquemática da retificação cilíndrica para uma imagem.

Para que o processo de retificação possa ser possível, é necessário que se saiba uma forma de relacionar os pontos de ambas as imagens que formam correspondências. A correspondência estéreo, na verdade, pode ser definida como sendo o problema de encontrar, nas imagens esquerda e direita obtidas pelo par de câmeras de um sistema de visão estéreo, os pontos que sejam projeções de uma mesma entidade na cena – chamados de pontos homólogos ou pontos correspondentes (ZUREIKI; DEVY; CHATILA, 2007). O processo de encontrar entre duas ou mais imagens os seus pontos correspondentes é um dos assuntos mais importantes e desafiadores em visão computacional (GU et al., 2008).

Uma forma de se representar a relação entre os pontos correspondentes de um par de imagens é por meio de uma matriz, chamada de **matriz fundamental**, que é capaz de representar algebricamente a geometria epipolar do sistema. A matriz fundamental é uma matriz real capaz de expressar o relacionamento existente entre pontos correspondentes presentes em um par de imagens. Existem várias maneiras com as quais uma matriz fundamental pode ser definida a partir de uma geometria epipolar, partindo-se de pontos presentes em uma das imagens do sistema de visão estéreo, e seu correspondente presente em uma linha epipolar da outra imagem do sistema; neste trabalho, será utilizada uma definição mais simples, que não depende da geometria da cena em si, mostrada a seguir:

Definição (Matriz Fundamental). *Sejam duas imagens capturadas por um sistema de visão estéreo. Seja $p_1 = (x_1, y_1)$ um ponto pertencente ao conjunto de pontos P_1 da primeira imagem e $p_2 = (x_2, y_2)$ um ponto pertencente ao conjunto de pontos P_2 , de modo que p_1 e p_2 sejam correspondentes. A matriz fundamental é uma matriz F , com elementos reais e de dimensões 3×3 , que satisfaz a condição dada pela Equação 2.8 (HARLEY; ZISSERMAN, 2004):*

$$\begin{bmatrix} x_2 & y_2 & 1 \end{bmatrix} \cdot F \cdot \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = 0 \quad (2.8)$$

para todo $p_1 \in P_1$ e $p_2 \in P_2$.

Após o processo de retificação, as imagens capturadas são inseridas em um plano epipolar, onde o eixo x é paralelo a linha de base deste mesmo plano, fazendo com que as linhas epipolares devam coincidir com as linhas de varredura (também chamadas de *scanlines*) horizontais. Kumar et al. (2010) explicam que os pontos correspondentes de um par de imagens retificadas possuirão a mesma coordenada y – logo, eles podem ser encontrados procurando-se apenas ao longo da linha de varredura (ou *scanline*) horizontal, evitando o uso de um possível rastreamento arbitrário de linhas epipolares. Além disso, todas as linhas epipolares se cruzam nos epipolos; assim, um plano epipolar cruza os planos das imagens da esquerda e da direita em linhas epipolares, de modo a definir a correspondência entre tais imagens (HARLEY; ZISSERMAN, 2004).

2.4.3 Correspondência e disparidade entre pontos

Em geral, os algoritmos responsáveis por encontrar a correspondência entre todos os pontos de duas ou mais imagens consideram alguns fatores para a execução, como o fato das imagens obtidas pelas câmeras estarem na mesma altura de observação (característica obtida na fase de retificação) e a distância entre tais câmeras ser pequena demais para permitir uma grande variação de luz entre as imagens capturadas, o que poderia causar diferenças entre as intensidades de cores dos objetos capturados por cada uma. Segundo Rateke et al. (2013) são necessários três pressupostos para a resolução do problema de correspondência na visão estéreo:

- Os pontos correspondentes devem ter a mesma intensidade (ou cor) em ambas as imagens (Pressuposto da Consistência de Cores).
- Os pontos correspondentes em ambas as imagens devem ficar na mesma linha de varredura – ou seja, um mesmo ponto deve ter a mesma altura em ambas as imagens (Pressuposto Epipolar).
- A ordem de ocorrência dos pontos em ambas as imagens deve ser a mesma (Pressuposto da Ordenação).

A Figura 2.8 ilustra a forma como a correspondência entre pontos é obtida em um par de imagens de um sistema de visão estéreo, obedecendo a tais pressupostos. Na realidade, o problema de correspondência estéreo é um problema de minimização, tanto para problemas de energia, representando custos locais de cada correspondência entre pontos (ou seja, onde custos entre correspondências sejam independentes entre si), ou para um função de energia global única, representando o custo global de correspondência entre os pontos das imagens (RATEKE et al., 2013). Existem vários métodos de análise de correspondência estéreo, e esses podem ser classificados de acordo com a forma como a correspondência é realizada.

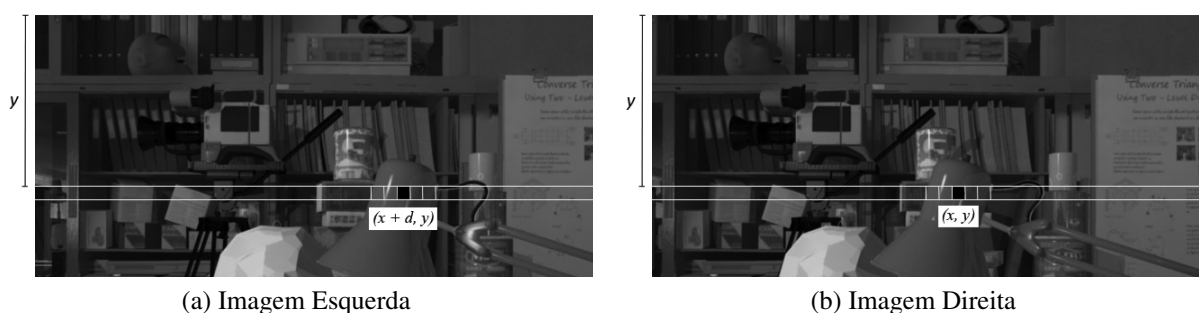


Figura 2.8: Cálculo de correspondência entre pontos: um conjunto de pontos que possui a mesma intensidade (ou cor) em ambas as imagens, está a uma mesma distância y do eixo x da origem, e são sequencialmente equivalentes, provavelmente representam o mesmo objeto. Imagens originais extraídas da base de dados Tsukuba (MARTULL; PERIS; FUKUI, 2012).

Os **métodos locais** calculam a correspondência entre pontos baseando-se nos valores de intensidade de seus pontos vizinhos, por meio do uso de uma janela finita (BARROIS et al., 2010). Esses métodos são mais simples que os métodos globais, apresentam abordagens mais eficientes e estruturas mais simples; no entanto, a homogeneidade de uma região pode dificultar a busca da correspondência entre tais pontos em outras imagens, de modo que a eficiência dos métodos locais depende de dois fatores principais: a medida de correlação entre pontos de uma janela, e a correspondência entre janelas locais (GU et al., 2008).

Já os **métodos globais** consideram a disparidade como sendo um problema de otimização de um custo global entre as correspondências das imagens, como citado anteriormente, e buscam resolver tal problema não apenas para uma determinada região da imagem, mas para uma linha de varredura ou uma imagem inteira, por meio do uso de uma função de custo global, sujeita a uma restrição de suavidade. A análise da correspondência é realizada linha a linha, e cada ponto de uma imagem é comparado com cada ponto na linha epipolar correspondente na outra imagem (BARROIS et al., 2010). A restrição de suavidade é expressa nos métodos globais de uma forma explícita, de modo que o objetivo é tornar o custo o mais baixo possível (SCHARSTEIN; SZELISKI, 2002).

Tanto em um método local quanto em um método global, a métrica de correspondência a ser utilizada no cálculo da similaridade entre regiões das imagens pode influenciar na qualidade do mapa de disparidades a ser obtido. Dentre as várias métricas existentes para o cálculo de tal similaridade, a Soma das Diferenças Quadradas e a Soma das Diferenças Absolutas são as mais conhecidas¹. A Figura 2.9 representa um esboço do funcionamento tanto de um método local quanto de um método global.

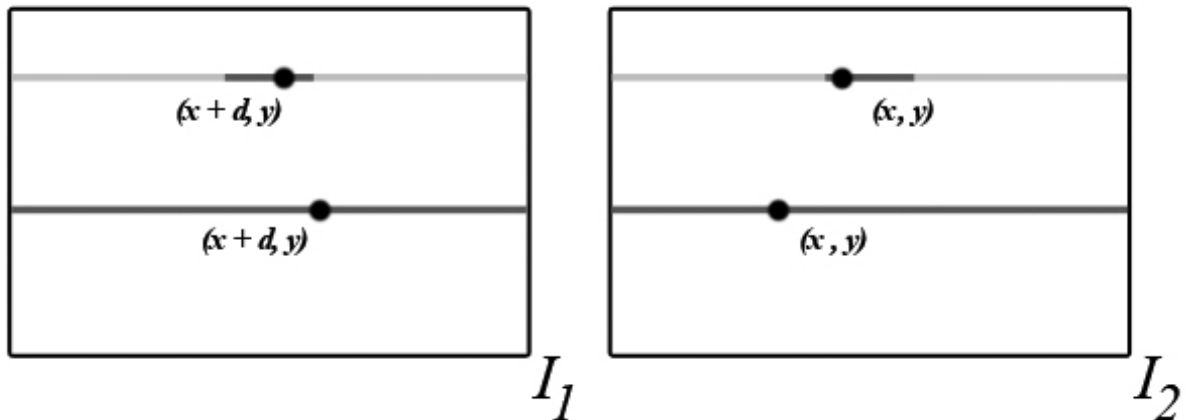


Figura 2.9: Tipos de métodos de cálculo de correspondência. As linhas superiores representam um método local, onde a janela de referência (em cinza mais escuro) é reprojeta em ambas as imagens, de modo que a busca por cada correspondência é limitada a esta área; as linhas inferiores representam um método global, onde a janela de referência é a própria linha analisada.

Uma vez encontrados todos os pares de pontos correspondentes nas imagens do sistema de visão estéreo, a disparidade de cada um desses pares pode ser descrita como a diferença horizontal que os pontos correspondentes de um par têm em ambas as imagens – ou seja, o quanto um ponto em um par está “deslocado” de uma imagem em relação a seu correspondente (RATEKE et al., 2013). Após o cálculo do valor da disparidade de um *pixel*, este pode ser mapeado em uma cor (ou tom de cinza) de uma imagem, de modo a permitir a visualização de regiões com disparidades distintas em uma cena; tal imagem é chamada de **mapa de disparidades**. Dessa forma, é possível ter uma representação visual da “distância” dos objetos da cena em relação ao sistema estéreo, de acordo com a intensidade das cores (ou dos valores de cinza), permitindo saberem quais deles estão mais próximos ou mais afastados em relação às câmeras. A Figura 2.10 exibe o mapa de disparidades para uma cena.

¹Uma descrição detalhada de tais métricas é descrita no Capítulo 3.



Figura 2.10: Imagem original e o seu mapa de disparidades correspondente; tons mais claros de cinza indicam objetos mais próximos ao observador. Imagens extraídas da base de dados Tsukuba (MARTULL; PERIS; FUKUI, 2012).

2.5 Esquematização do Desenvolvimento de um Mapa de Disparidades

Todo o processo de geração de um mapa de disparidades por um sistema de visão estéreo pode ser representado pela esquematização exibida na Figura 2.11. A ordem das etapas pode ser descrita da seguinte maneira:

1. Se necessária a calibração (ou recalibração), esta é realizada, tendo como base um (ou mais) par(es) de imagens de entrada obtidas pelo sistema. Daí, os parâmetros obtidos com a calibração são armazenados pelo sistema;
2. Quando se deseja gerar um mapa de disparidades, um par de imagens é capturado pelas câmeras do sistema, e os parâmetros obtidos na calibração são carregados pelo mesmo sistema. O par de imagens capturado pode sofrer algum pré-processamento a fim de se melhorar a qualidade de suas imagens (como a remoção de ruídos ou a equalização do histograma, por exemplo);
3. Os parâmetros de calibração são utilizados na fase de retificação das imagens capturadas, a fim de se corrigir a divergência entre as alturas dos pontos correspondentes;
4. Quando o processo de retificação termina, as imagens retificadas seguem para o cálculo do valor de disparidade dos pontos correspondentes presentes nestas;
5. Por fim, cada valor obtido no cálculo do valor de correspondência entre as imagens retificadas é mapeado para uma cor (ou tom de cinza), a fim de compor o mapa de disparidades

resultante.

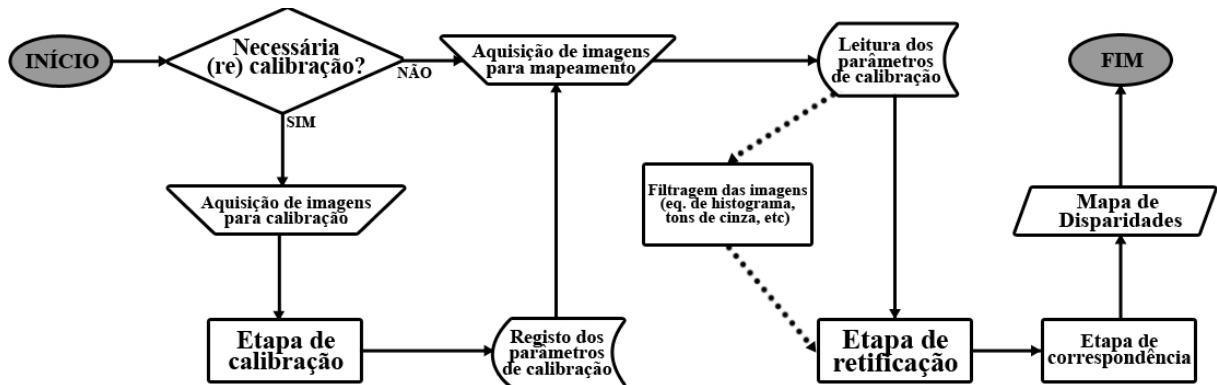


Figura 2.11: Esquematização do processo de geração de um mapa de disparidades passando pelas prováveis três etapas: calibração, retificação e correspondência.

Um detalhe a se observar é que a etapa de calibração deve ser realizada sempre no primeiro uso do sistema ou, então, quando a configuração atual das câmeras for alterada. Caso não seja necessária uma nova calibração, pode-se ir diretamente à etapa de retificação.

Capítulo 3

ESTUDOS E CONCEITOS RELACIONADOS

3.1 Considerações Iniciais

Devido à complexidade de um sistema de visão estéreo, existem várias técnicas e conceitos que este tipo de sistema costuma exigir durante a sua execução – principalmente em relação à calibração do sistema ou ao cálculo da disparidade entre os pontos nas imagens obtidas pelo mesmo. Assim, neste Capítulo é apresentada uma descrição sucinta dos estudos e conceitos que foram considerados relevantes, e que foram utilizados como base para o desenvolvimento deste trabalho.

3.2 Técnicas de Detecção de Cantos

Para que um sistema estéreo possa produzir um mapa de disparidades, se faz necessária a detecção de pontos que sejam entidades de um mesmo objeto em ambas as imagens de uma cena capturada, uma vez que pares de pontos correspondentes são utilizados como base para as etapas do sistema de visão estéreo (calibração, retificação e correspondência).

Inicialmente, pode-se pensar na detecção de pontos correspondentes como a escolha de um ponto arbitrário em uma imagem, e a busca arbitrária por seu correspondente em outra; no entanto, há questões que devem ser consideradas nesse caso. Primeiro, se todos os pontos da imagem forem submetidos a um processo de busca cega pelo seu correspondente, tal processo pode se tornar inviável (mesmo para imagens com dimensões consideradas pequenas) em relação ao tempo de processamento. Segundo, nem todos os pontos podem apresentar correspondências em ambas as imagens, pois a distância entre as duas câmeras do sistema estéreo pode ocasionar efeitos como, por exemplo, oclusão e diferença de luminosidade entre os elementos das

imagens. Por isso, geralmente os pontos utilizados como base no processo inicial de correspondência são aqueles que correspondem aos cantos dos objetos presentes na cena analisada, como pode ser visto na Figura 3.1. Desse modo, é possível contornar questões de variação na luminosidade e oclusões de elementos presentes na cena, além de ser possível reduzir o número de pontos a serem correspondidos, acelerando o processo.

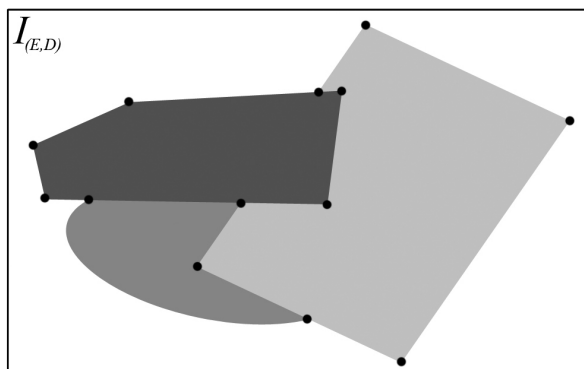


Figura 3.1: Os pontos de partida para o processo de geração de um mapa de disparidades são os “cantos” dos objetos presentes em cada imagem obtida da cena (representados pelos pontos pretos na imagem).

O problema de detecção de cantos em imagens é relativamente antigo em computação, sendo um dos primeiros trabalhos o algoritmo de Harris e Stephens (1988). Atualmente, a maioria dos métodos de detecção de cantos em imagens partem do método conhecido como *Scale-Invariant Feature Transform* (SIFT), desenvolvido por Lowe (2004).

O SIFT é um método de detecção de cantos que fornece um modelo para extração de características distintas e invariantes para o reconhecimento de pontos em um objeto, em uma imagem, de diferentes ângulos. Os descritores extraídos são invariantes a escala e rotação, além de possuírem uma boa tolerância a ruídos, a distorções decorrentes de diferentes perspectivas e a mudanças de iluminação. Lowe (2004) ainda descreve uma abordagem para identificação de objetos utilizando um comparativo com um banco de descritores extraídos de outras imagens usando um algoritmo de vizinhos próximos. A implementação do algoritmo consiste em quatro etapas:

1. Utiliza-se a função de diferença de Gaussianas (LOWE, 2004) aplicadas a diversas escalas da imagem, de modo a encontrar pontos invariantes a escala e a orientação;
2. Para cada candidato encontrado na etapa anterior, determina-se a localização, escala e métricas de estabilidade, de modo a se determinar os pontos mais estáveis da cena. Tais pontos estáveis são os cantos procurados pelo algoritmo;

3. Atribui-se uma ou mais orientações a cada ponto chave escolhido, de acordo com o gradiente local da imagem;
4. Mensura-se os gradientes locais ao redor de cada um dos pontos, e uma representação simplificada é extraída.

Hoje em dia, os métodos mais atuais de detecção de cantos de uma cena são computacionalmente mais eficientes que o SIFT, tanto por utilizarem abordagens diferentes quanto por se basearem em formas otimizadas do método citado. Os principais algoritmos utilizados para a detecção de cantos, de modo sucinto, são:

- **Algoritmo de Harris:** desenvolvido por Harris e Stephens (1988), utiliza segmentação de imagens com uma função pré-definida: uma janela deslizante é deslocada por toda a figura, e dois autovalores para uma função (chamada de função de Harris) são utilizados para calcular a diferença produzida entre a janela deslizante e a região que está sendo sobreposta na imagem original. A partir dessa diferença, é possível conseguir uma pontuação, e se tal pontuação for superior a um limiar definido, aquela janela pode ser considerada um canto da área total da imagem.
- **Minimum Eigen Value (Min8Val):** desenvolvido por Shi e Tomasi (1994), ele é similar ao algoritmo de Harris – no entanto, em vez de partir de dois autovalores para a função de Harris, ele utiliza apenas o menor dentre estes dois. Em geral, tal método é mais rápido que o algoritmo de Harris, e costuma produzir bem mais resultados; no entanto, para conjuntos de imagens similares (como as de um sistema de visão estéreo) o alto número de cantos produzidos pode criar um alto número de correspondências incorretas.
- **Features from Accelerated Segment Test (FAST):** proposto por Rosten e Drummond (2005, 2006) com o objetivo de se identificar pontos chaves em imagens com uma velocidade muito superior ao algoritmo SIFT – porém, tal método não possui informações sobre a orientação dos descritores, e também é altamente sensível a ruídos. O algoritmo FAST propõe uma melhora de desempenho do SIFT original, utilizando aprendizado de máquina, combinado com o uso de uma árvore de decisões.
- **Maximally Stable Extremal Regions (MSER):** desenvolvido por Donoser e Bischof (2006) com o objetivo de ser robusto diante de mudanças de perspectiva. Dos métodos da lista, é o único que originalmente é utilizado com o exclusivo objetivo de detecção de características e alinhamento de imagens estéreo (ou seja, para retificação). O algoritmo localiza pontos extremos em uma imagem buscando identificar regiões conexas, a partir

da intensidade do brilho dos *pixels* presentes nesta. Ele utiliza limiares de diferentes valores para a determinação de cantos de diferentes regiões da imagem, e é capaz de detectar regiões de bordas com grande variação de intensidade.

- **Speeded-Up Robust Features (SURF)**: é um dos mais conhecidos algoritmos, desenvolvido por Bay et al. (2008). É inspirado no algoritmo SIFT, porém apresenta tempo de processamento inferior. Os cantos são detectados utilizando-se uma aproximação inteira do determinante de um detector Hessiano, que pode ser calculado por meio de operações inteiras, utilizando-se uma imagem integrante pré-computada. A orientação deles é obtida por meio da transformada de Haar (*Haar Wavelets*), e os descritores são formados a partir de pulsos dos pontos ao redor dos pontos-chave.
- **Binary Robust Invariant Scalable Keypoints (BRISK)**: proposto por Leutenegger, Chli e Siegwart (2011), baseia-se no algoritmo FAST. Inicialmente, tal método calcula as variações de luminosidade até um limiar dado, identificando possíveis candidatos a cantos; em seguida, é reaplicado às áreas detectadas para extrair-se mais informações sobre os candidatos iniciais. É capaz de evitar a perda de informação sobre a luminosidade da imagem, de modo a garantir que os pontos resultantes desse processo sejam coerentes localmente (na sua vizinhança).

3.3 Métricas de Qualidade de Uma Matriz Fundamental

Uma vez que os pontos que representam cantos em uma cena foram obtidos e correspondidos, é possível a partir deles encontrar uma forma geral para se mapear qualquer ponto de uma imagem ao seu correspondente presente na outra imagem. Porém, é comum que, após o cálculo das correspondências iniciais entre pontos, algumas dessas tenham sido calculadas incorretamente. Isso pode ser representado pela Figura 3.2, onde as correspondências incorretas entre cantos são representadas pelas linhas vermelhas.

A fim de se permitir o mapeamento eficiente entre quaisquer pontos correspondentes em uma cena – e, de quebra, remover correspondências incorretas – costuma-se utilizar a matriz fundamental (ABDEL-AZIZ; KARARA, 1971), que foi descrita inicialmente pela Equação 2.8 apresentada no Capítulo 2. Ela pode ser definida como sendo a matriz capaz de expressar uma relação entre os pontos de ambas as imagens de um sistema estéreo.

A função da matriz fundamental dentro do contexto de correspondência pode ser interpretada de maneira intuitiva por meio de uma representação gráfica. Seja um plano cartesiano $U \times V$, onde o eixo U contenha todos os pontos de uma imagem de um par estéreo, enquanto o

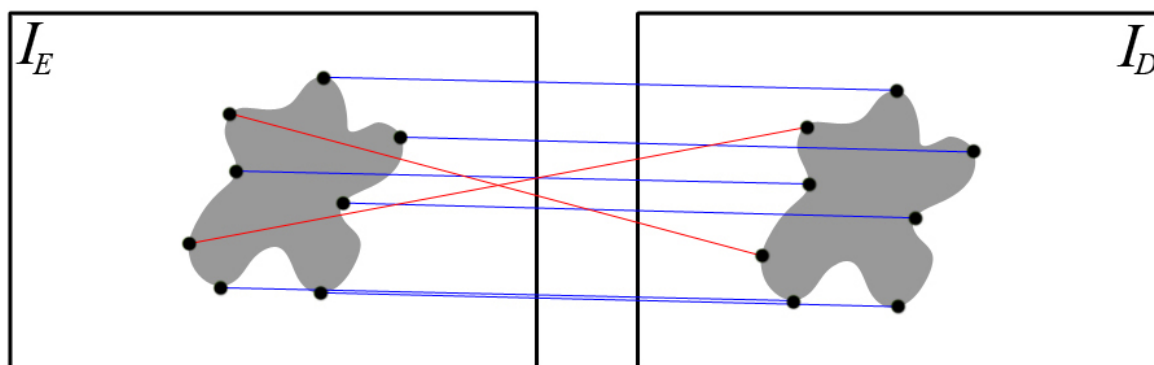


Figura 3.2: Correspondências incorretas (representadas em vermelho) entre os cantos de um par de imagens estéreo, que podem ser removidas com o auxílio da matriz fundamental.

eixo V contém todos os pontos da outra imagem do par. Daí, certo número de correspondências entre os pontos de U e V podem ser escolhidas aleatoriamente, para serem representadas no plano cartesiano. O objetivo da matriz fundamental é fornecer os parâmetros lineares que sejam capazes de aproximar a posição de tais correspondências com uma curva, para qualquer ponto existente no conjunto U ou V , como mostra a Figura 3.3:

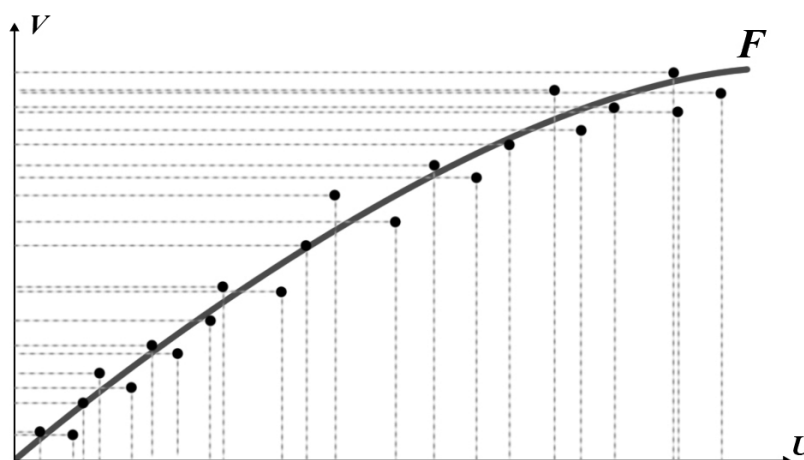


Figura 3.3: Graficamente, o objetivo da matriz fundamental é aproximar a relação entre pontos em um par de imagens estéreo com uma curva.

Todos os modelos existentes na literatura para a obtenção de uma matriz fundamental se baseiam em encontrar uma matriz “ótima”, capaz de fornecer da maneira mais próxima possível a localização de pares de pontos correspondentes presentes em um par de imagens estéreo.

Inicialmente, para o cálculo da matriz fundamental, é necessário definir a forma como a distância entre um par de pontos correspondentes e a curva linear que a matriz fundamental define será quantificada. Na literatura, costuma-se escolher entre dois tipos de fórmulas para tal cálculo: a distância algébrica comum, ou a distância de Sampson. Considerando-se um par

de pontos correspondentes (u_i, v_i) , ambos na forma $[x_i \ y_i \ 1]$, e uma matriz fundamental F , a distância algébrica entre eles pode ser obtida pela Equação 3.1:

$$d(u_i, v_i) = (v_i \cdot F \cdot u_i^T)^2 \quad (3.1)$$

enquanto a distância de Sampson (SAMPSON, 1982) pode ser expressa pela Equação 3.2:

$$d(u_i, v_i) = (v_i \cdot F \cdot u_i^T)^2 \left[\frac{1}{(F \cdot u_i^T)_1^2 + (F \cdot u_i^T)_2^2} + \frac{1}{(v_i \cdot F)_1^2 + (v_i \cdot F)_2^2} \right] \quad (3.2)$$

onde i é o índice dos pontos correspondentes, e $(F \cdot u_i^T)_j^2$ é o quadrado do elemento na entrada j do vetor $F \times u_i^T$. Daí, escolhida a métrica de distância entre pontos correspondentes e a curva linear, a matriz fundamental pode ser computada, de modo geral, da seguinte maneira:

1. Inicializa-se a matriz fundamental F como uma matriz 3×3 , nula;
2. Escolhe-se oito pares de pontos correspondentes aleatórios, e a partir deles se estima uma nova matriz fundamental F' , utilizando um algoritmo de aproximação linear (HARLEY; ZISSERMAN, 2004);
3. Calcula-se a taxa de eficiência para F' : se esta for melhor que a taxa de eficiência de F , a matriz F é substituída por F' ;
4. Repetem-se os passos 2 e 3, quantas vezes forem desejadas.

A taxa de eficiência de uma matriz fundamental, citada no passo 3, serve para indicar o quanto uma matriz fundamental consegue aproximar corretamente a correspondência entre pares de pontos. Existem várias expressões matemáticas que podem calcular a taxa de eficiência de uma matriz, em relação ao conjunto de pontos correspondentes entre um par de imagens, e as principais são:

- **Random Sample Consensus (RANSAC)**: apresentada por Fischler e Bolles (1981), onde a taxa de eficiência é dada por:

$$RANSAC(F) = \sum_{i=1}^N \frac{\text{sgn}(d(u_i, v_i), t)}{N} \quad (3.3)$$

onde:

$$\text{sgn}(a, b) = \begin{cases} 1, & \text{se } a \leq b \\ 0, & \text{caso contrário} \end{cases} \quad (3.4)$$

de modo que o limiar t existente neste método permite definir um limite máximo para a distância entre o par de pontos correspondentes e a aproximação linear definida pela matriz F . Quanto maior o valor obtido pela somatória, melhor a eficiência da matriz fundamental.

- **Least Median of Squares (LMedS)**: desenvolvida por Rousseeuw (1984), em que a taxa de eficiência é dada simplesmente pela mediana das distâncias entre os pontos correspondentes das imagens, ou seja, da forma mostrada na Equação 3.5:

$$\text{LMedS}(F) = \text{mediana}[d(u, v)] \quad (3.5)$$

sendo que quanto menor o valor dessa mediana, melhor a eficiência de uma matriz fundamental. Em geral, devido a este fator, tal método é eficiente quando sabe-se inicialmente que mais da metade das correspondências entre pares de pontos presentes em uma cena são corretas.

- **Least Trimmed of Squares (LTS)**: também desenvolvida por Rousseeuw (1984), onde a taxa de eficiência é dada por:

$$\text{LTS}(F) = \sum_{i \in \Omega} d(u_i, v_i) \quad (3.6)$$

onde Ω é o primeiro valor que for menor que o valor anterior, em um conjunto de $q \cdot N$ pares de pontos, com q sendo a porcentagem de correspondências verdadeiras previamente conhecida. Quanto menor o valor obtido pela somatória, melhor a eficiência da matriz fundamental.

- **M-estimator Sample Consensus (MSAC)**: apresentada por Torr e Murray (1997), é uma variação do método RANSAC, onde a taxa de eficiência é dada por:

$$\text{MSAC}(F) = \sum_{i=1}^N \min(d(u_i, v_i), t) \quad (3.7)$$

onde t também é um limiar que permite definir um limite máximo para a distância entre o par de pontos correspondentes e a aproximação linear definida pela matriz F . No entanto,

nesse método, quanto menor o valor obtido pela somatória, melhor a eficiência da matriz fundamental.

Deve-se ressaltar o fato de que o algoritmo LMedS exige um mínimo de dezesseis pares de pontos correspondentes iniciais, enquanto os algoritmos RANSAC e MSAC exigem como mínimo oito pares de pontos correspondentes. Já o valor mínimo de correspondências do método LTS varia de acordo com a porcentagem de correspondências verdadeiras inicialmente conhecida.

3.4 Tipos de Métricas de Correspondência

Como descrito anteriormente no Capítulo 2, existem dois tipos de métodos de cálculo de correspondências: locais e globais. Tanto em um método local quanto em um método global, há o uso de janelas finitas para o cálculo de correspondências entre pontos; a diferença é que em métodos globais, a janela de referência pode ser a própria linha de varredura que a correspondência analisada está, ou até mesmo o conjunto de linhas em volta de tal correspondência. Também há trabalhos que procuram combinar métodos locais e globais de correspondência – primeiro utilizando janelas deslizantes para a disparidade local, e depois forçando a continuidade da disparidade entre as janelas (HIRSCHMULLER, 2005).

No entanto, além da existência de uma janela de referência, é necessária que exista uma métrica de correspondência, capaz de fornecer uma correlação eficiente entre pontos correspondentes de um par de imagens, que é utilizada tanto para o cálculo de correspondências iniciais (de modo a se gerar uma matriz fundamental futuramente) quanto para o cálculo da própria disparidade entre tais pontos correspondentes, uma vez que a disparidade de um ponto também depende da disparidade de seus pontos vizinhos (HIRSCHMULLER, 2005).

Considerando a Figura 3.4 como base, seja R uma janela de referência de uma das imagens, centrada no ponto (u, v) . A busca pelo correspondente ao ponto (u, v) é realizada reprojando a janela R na outra imagem, como uma nova janela S centrada nas mesmas coordenadas centrais da janela original, e verificando qual o ponto (c, l) da janela S que se relaciona melhor com o ponto (u, v) da janela R original – de modo que (u, v) e (c, l) sejam classificados como uma correspondência.

Sabendo que ambas as janelas são de dimensões $comp \times larg$, pode-se utilizar uma métrica para se calcular a disparidade dos pontos (u, v) e (c, l) . Atualmente, as duas mais utilizadas são:

- **Soma das Diferenças Quadradas**: vista em trabalhos como os de Gaschler, Burschka e

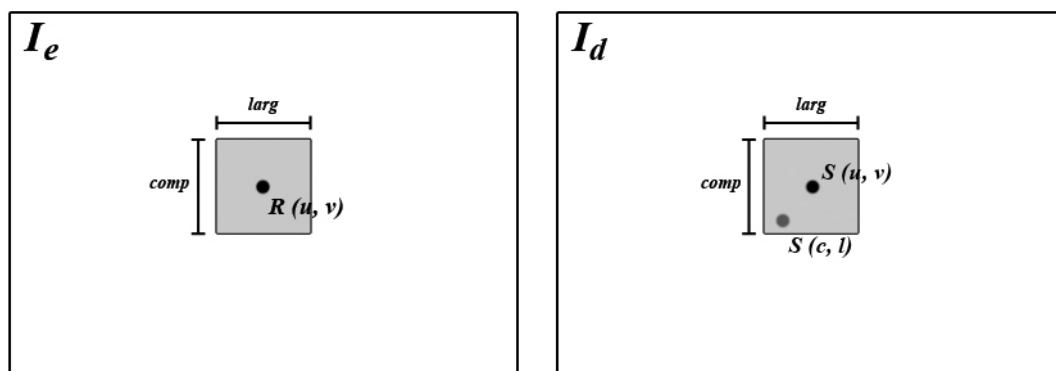


Figura 3.4: Representação das janelas utilizadas em um processo de correspondência e/ou disparidade.

Hager (2010) e Kamencay et al. (2014). A diferença de todos os pontos da janela definida é computada, e depois o quadrado de todas estas diferenças é somado, como mostra a Equação 3.8:

$$SSD(R, S) = \sum_{v=0}^{R_{comp}} \sum_{u=0}^{R_{larg}} [R(u, v) - S(c + u, l + v)]^2 \quad (3.8)$$

As duas janelas que melhor se relacionaram durante o cálculo da correspondência são as que correspondem ao valor mínimo da função. Para o cálculo da disparidade entre pontos, esta é obtida pela diferença horizontal entre as coordenadas dos pontos.

- **Soma das Diferenças Absolutas:** aplicada em trabalhos como os de Jacobi, Cardoso e Borges (2006) e Hamzah, Hamid e Salim (2010). É semelhante ao método de Soma das Diferenças Quadradas; porém, neste caso, as diferenças de todos os pontos da janela definida têm seus módulos somados, como é exibido na Equação 3.9:

$$SAD(R, S) = \sum_{v=0}^{R_{comp}} \sum_{u=0}^{R_{larg}} |R(u, v) - S(c + u, l + v)| \quad (3.9)$$

Assim como no caso da Soma das Diferenças Quadradas, a disparidade pode ser obtida com a diferença horizontal entre coordenadas de pontos.

Há um fato a ser observado nas métricas de correspondência: para o cálculo de disparidades dos pontos da imagem não utiliza-se uma janela de duas dimensões, como representado na Figura 3.4, anteriormente; uma vez que as imagens supostamente já tenham ido para o processo de disparidade retificadas, todos os pontos correspondentes estarão na mesma altura (ou coordenada y). Assim, a janela pode ocupar apenas a linha em que o ponto a ser analisado está.

3.5 Detecção de Descontinuidades (Bordas): o filtro Sobel

Por fim, um dos conceitos que é utilizado neste trabalho envolve o uso de um algoritmo de detecção de bordas. Existem vários tipos de detectores de bordas, e um dos mais utilizados na área de visão estéreo é o filtro Sobel (GONZALES; WOODS, 2007). Tal filtro (também chamado de operador Sobel) é utilizado com o objetivo de obter descontinuidades (bordas) em uma imagem, por meio de uma medida do gradiente espacial desta. Matematicamente, o operador Sobel consiste em um par de matrizes de dimensões 3×3 convoluídas, representadas pela Equação 3.10:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (3.10)$$

e pela Equação 3.11:

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (3.11)$$

Tais matrizes, quando multiplicadas isoladamente a uma matriz que representa uma imagem, criarão duas novas imagens em que cada ponto irá conter uma aproximação das derivadas horizontais e verticais de tal imagem – melhor dizendo, as medidas separadas do gradiente em cada orientação: G_x para o eixo x e G_y para o eixo y . Estas medidas separadas de gradiente podem ser combinadas de modo a se obter tanto a magnitude absoluta do gradiente em qualquer ponto da imagem original, que é dada pela Equação 3.12:

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (3.12)$$

quanto o ângulo de orientação de tal gradiente em relação ao conjunto de *pixels* da imagem, que é dado pela Equação 3.13:

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (3.13)$$

Na prática isso significa que, para cada ponto da imagem original, o filtro Sobel fornece a direção da maior variação de um tom claro para um tom escuro, além de uma medida quan-

titativa de tal variação. Assim, é possível obter-se uma noção de como a luminosidade varia em cada ponto (se esta muda de forma mais suave ou mais abrupta) de modo que, desta forma, é possível estimar a presença de uma transição claro-escuro – o que caracteriza uma borda na imagem original. A Figura 3.5 exibe um exemplo de resultado obtido pela aplicação do filtro Sobel em uma imagem.

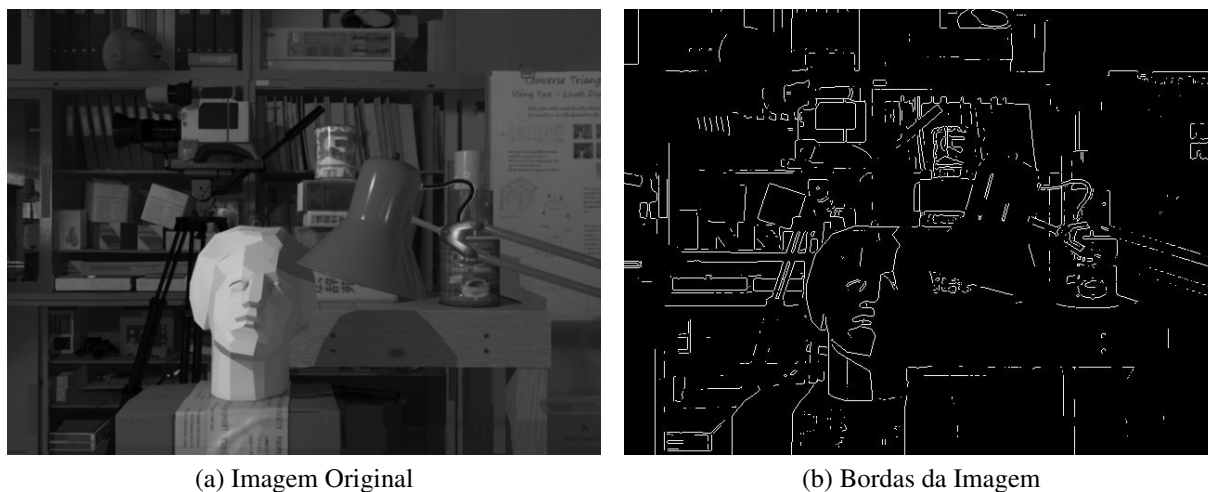


Figura 3.5: Imagem extraída da base de dados Tsukuba (MARTULL; PERIS; FUKUI, 2012), e suas bordas detectadas pelo filtro Sobel.

3.6 Outros Estudos sobre a Calibração

Em relação à calibração, os primeiros métodos utilizados em sistemas de visão estéreo procuravam efetuar a por meio da observação de um objeto presente em uma cena, cujas dimensões são conhecidas (chamado de padrão de calibração), como o método de Zhang (2000). Em geral, tal condição fazia com que, fora de ambientes controlados, o uso de sistemas de visão estéreo se tornasse difícil, principalmente quando havia a necessidade de calibração constante. Dessa forma, surgiram estudos buscando uma forma de automatizar o processo de calibração.

Os estudos sobre autocalibração envolve o desenvolvimento de técnicas que não utilizam objetos conhecidos previamente, mas sim a captura de diferentes imagens com as câmeras, de uma cena estática, sem a alteração dos parâmetros intrínsecos destas. Assim, além de dispensar o uso de um padrão de calibração, possibilita a calibração dinâmica ao sistema – ou seja, durante o seu funcionamento. Além disso, novas calibrações decorrentes da alteração de parâmetros intrínsecos podem ser feitas sem maiores dificuldades. A principal desvantagem desse método, no entanto, consiste no alto tempo de processamento necessário para a realização da correspondência entre os pontos das imagens, a dificuldade de implementação (em geral maior do

que a de métodos que utilizam a calibração convencional), e a geração de resultados menos precisos. Atualmente, os estudos sobre autocalibração buscam realizá-la a partir do conhecimento prévio dos elementos da cena (COLLADO et al., 2006; MUSLEH et al., 2014), ou dos pontos correspondentes desta (KRAMM; BENSRAHAI; MICHE, 2006).

3.7 Outros Estudos sobre o Mapa de Disparidades

Com relação a mapas de disparidades, existem estudos variados. Dentre os vários existentes, pode-se destacar as seguintes vertentes para a criação de tais mapas, que foram surgindo ao longo do tempo:

- Os trabalhos de Belhumeur (1993), Cox et al. (1996) e Birchfield e Tomasi (1998) foram os primeiros a se basearem na minimização de funções de custo, em que são computadas diferenças absolutas ou quadráticas das intensidades de *pixels* em pares de imagens epipolares. Apesar de tais métodos terem apresentado resultados satisfatórios, suas grandes desvantagens estavam no fato de se basearem em programação dinâmica – e, dessa forma, o custo computacional não é viável para sistemas que exigem processamento em tempo real, tornando-os inadequados para implementação em *hardware*.
- Por outro lado, estudos como o de Calin e Roda (2006) e de Werner, Stabernack e Riechert (2014) também se basearam na observação da intensidade dos *pixels*, mas estruturando-se na implementação em dispositivos lógicos programáveis (*hardware*), capazes de conferir a propriedade de processamento a tempo real, embora possuam insensibilidade a oclusões e limitada região de análise para detecção de correspondências.
- Alguns pesquisadores como Grosso e Tistarelli (1995) e Feng e McCarthy (2013) mostraram em seus estudos a possibilidade de se caracterizar cena e navegação, a partir da obtenção, em tempo real, de informações tridimensionais do ambiente, mais especificamente por meio da análise do próprio mapa de disparidades obtido.
- Por fim, mais recentemente, estudos como os de Laureano e Paiva (2007), Zhang, Hou e Yang (2009) e Zbontar e LeCun (2015) buscaram combinar o processo de visão estéreo com o uso de algoritmos genéticos, de modo que um mapa de disparidades ideal para uma cena pudesse ser produzido por meio do processo de cruzamento genético entre uma população inicial de mapas de disparidades.

Capítulo 4

DESENVOLVIMENTO DO TRABALHO

4.1 Ideia Utilizada

Na geração de um mapa de disparidades há três etapas distintas: a calibração do sistema, a retificação das imagens obtidas e o cálculo das correspondências entre tais imagens, conforme visto no Capítulo 2. Muitos sistemas de visão estéreo usam um padrão de calibração para realizar a primeira etapa, baseado no modelo apresentado por Zhang (2000). Como o objetivo é automatizar todo o processo de geração do mapa de disparidades de modo que um deficiente visual possa utilizá-lo futuramente, a necessidade de uso de um padrão de calibração deve ser eliminada.

Uma maneira de se remover tal dependência do padrão de calibração é utilizar a geometria da própria cena a ser analisada, observando os elementos presentes nessa para se chegar a uma matriz fundamental – que por si só já é capaz de permitir a execução de um processo de retificação e, conseqüentemente, do cálculo de disparidade. Para que o cálculo de tal matriz seja possível, no entanto, é necessário o conhecimento de um número mínimo de pares de pontos correspondentes iniciais nas imagens capturadas pelo sistema de visão estéreo (HARLEY; ZISSERMAN, 2004).

Assim, o conjunto de algoritmos desenvolvido procura buscar pares de pontos correspondentes iniciais em uma cena e calcular uma matriz fundamental a partir deles. Esses três passos podem substituir o processo de calibração convencional (uma vez que tal matriz é suficiente para um processo de retificação) e, a partir daí, calcular-se o mapa de disparidades a partir da matriz obtida. Este método de desenvolvimento de um mapa de disparidades é, atualmente, considerado simples se comparado a métodos mais recentes; porém, devido à sua facilidade de implementação e de custo computacional (como mostrado mais adiante), este método é sufi-

ciente para a geração de um mapa de disparidades com o objetivo de permitir o mapeamento da distância de objetos em uma cena. Dessa forma, tal método é, em teoria, simples o suficiente para garantir o mapeamento (ao menos qualitativo) da distância dos elementos presentes em uma cena, de modo que uma pessoa com deficiência visual pudesse utilizá-lo como uma alternativa para se permitir uma orientação espacial em relação a um ambiente.

Este Capítulo apresenta em detalhes a metodologia aplicada no desenvolvimento deste trabalho, de modo a permitir a geração de um mapa de disparidades sem a intervenção humana direta. Além disso, o ambiente de trabalho utilizado, bem como as ferramentas utilizadas e a descrição de cada etapa deste trabalho também estão presentes neste Capítulo.

4.2 Modelo de Sistema de Visão Estéreo

Para o desenvolvimento deste trabalho, foi necessária a utilização de um par de câmeras, de modo a se compor um modelo físico de sistema de visão estereo. Para isso, optou-se por utilizar a Minoru3D, desenvolvida pela Promotion and Display Technology (2015), que é uma *webcam* capaz de capturar imagens em três dimensões.

A Minoru3D pode ser vista na Figura 4.1. Além do baixo custo, a escolha desse modelo de *webcam* se deve a sua construção: ela é composta por duas lentes distintas, separadas por uma distância fixa – o que a torna capaz de ser utilizada como um modelo físico de um sistema de visão estereo. De acordo com a fabricante, suas especificações técnicas são:

- Resolução máxima para cada câmera de 800 por 600 *pixels*;
- Comprimento da linha de base (distância entre as câmeras) de 6 centímetros;
- Taxa máxima de 30 quadros (frames) por segundo;
- Capacidade de foco em objetos a, no mínimo, 10 centímetros de distância;
- Ângulo de visão (tanto vertical quanto horizontal) de 42 graus.

De acordo com a fabricante da Minoru3D, para um bom funcionamento desta câmera, são requeridas as etapas de calibração e retificação. Além disso, a distorção das lentes é baixa o suficiente pra não afetar as aplicações em que a *webcam* for utilizada¹.

¹ Isso significa que o parâmetro τ da matriz de parâmetros intrínsecos do sistema (descrita em detalhes no Capítulo 2) pode ser considerado como zero.



Figura 4.1: Modelo da webcam Minoru3D, utilizada neste trabalho.

4.3 Ambiente de Testes

Existem várias opções de ambiente de testes que poderiam ser utilizadas no desenvolvimento deste trabalho. Dentre elas, é possível citar como exemplo o MatLab (The MathWorks, Inc., 2015) e o Octave (RAWLINGS; EKERDT, 1988). Ambas se destacam por permitirem uma fácil manipulação de imagens e por possuírem ampla aceitação da comunidade de desenvolvedores de *software* na área de processamento digital de imagens.

Partindo do fato anterior, após testes realizados com as duas opções apresentadas, foi determinado que os algoritmos desenvolvidos neste trabalho seriam desenvolvidos no ambiente do MatLab, versão R2014b, de 64 bits. A principal vantagem do uso do MatLab se deve à sua característica de permitir uma fácil manipulação de imagens e por possuir uma biblioteca (*toolbox*) específica para visão estéreo (The MathWorks, Inc., 2015), enquanto o Octave necessita de *frameworks* de terceiros, além do OpenCV (The OpenCV Team, 2017), para poder realizar operações de visão computacional.

Dessa forma, boa parte das funções utilizadas neste trabalho (como as responsáveis pelas etapas de retificação e disparidade), pertencem à biblioteca de visão estéreo do MatLab; no entanto, todas as funções utilizadas tiveram seus parâmetros ajustados empiricamente para se adequarem aos algoritmos desenvolvidos. Já para o uso de *webcams* (como a Minoru3d) dentro do ambiente do MatLab, é necessária a instalação de duas bibliotecas adicionais: a *USB Webcam*, para o reconhecimento de *webcams*, e a *OS Generic Video Interface*, para a extração das imagens capturadas. Ambas podem ser instaladas utilizando o instalador de pacotes dentro do próprio MatLab (cujo acesso pode ser realizado utilizando-se a função *supportPackageInstaller* na linha de comando do console do ambiente).

4.4 Aquisição das Imagens

O primeiro passo no desenvolvimento deste trabalho é a captura das imagens. Cada câmera presente na Minoru3D é inicializada no MatLab por meio de uma função chamada *videoinput*, que retorna um objeto de um tipo com o mesmo nome. Os parâmetros desta função são o tipo de vídeo (no caso, o tipo 'winvideo'), e o número de identificação da câmera a ser inicializada. Tal número de identificação deve ser obtido empiricamente; em geral, quando há apenas a Minoru3D conectada ao computador, os números de identificação são 1 e 2, para as câmeras esquerda e direita (em relação ao ponto de vista da Minoru3D), respectivamente.

Uma vez as câmeras inicializadas, para se capturar uma imagem com uma delas, utiliza-se a função *getsnapshot*, onde o objeto *videoinput* associado à câmera que se deseja utilizar é passado como parâmetro. O retorno da função *getsnapshot* é, basicamente, a imagem capturada pela câmera, em escala de cores RGB (colorida).

4.5 Pré-Processamento das Imagens

A partir das imagens capturadas, um pré-processamento é necessário a fim de ser possível aplicar os algoritmos necessários para se chegar ao mapa de disparidades desejado. Embora questões como luminosidade e qualidade das imagens possam influenciar na eficácia do método, devido ao tipo de câmera utilizada, tais fatores raramente afetam o sistema.

Assim, neste trabalho, apenas a conversão para tons de cinza de tais imagens é realizada, com o uso da função *rgb2gray* do MatLab. Tal função recebe como parâmetro uma imagem em escala de cores RGB, e retorna a mesma imagem, mas em tons de cinza. Na Figura 4.2 é exibido um par de imagens capturadas pelas câmeras do sistema de visão estéreo utilizado neste trabalho, convertidas em tons de cinza.

A partir deste ponto, o método desenvolvido pode ser representado em uma esquematização em blocos, de modo a ilustrar as entradas e saídas de cada função utilizada. A Figura 4.3 exibe a esquematização em blocos do processo até a etapa atual: um par de câmeras obtêm, de uma cena observada, um par de imagens, sendo I_e a imagem da câmera esquerda e I_d a imagem da câmera direita. Daí, tais imagens são convertidas para tons de cinza usando a função *rgb2gray*, de modo que o par de imagens passe a ser C_e para a câmera esquerda e C_d para a câmera direita.

É necessário salientar que a função *rgb2gray* é utilizada duas vezes, uma para cada imagem de entrada; ou seja, é uma etapa que pode ser paralelizada.

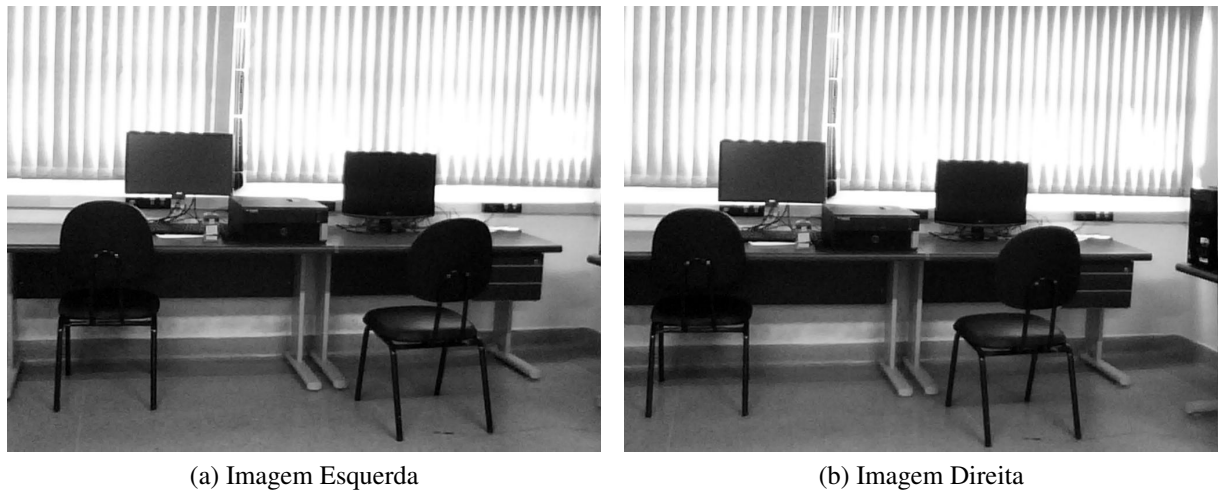


Figura 4.2: Um par de imagens capturadas pelo sistema de visão estéreo utilizado neste trabalho.

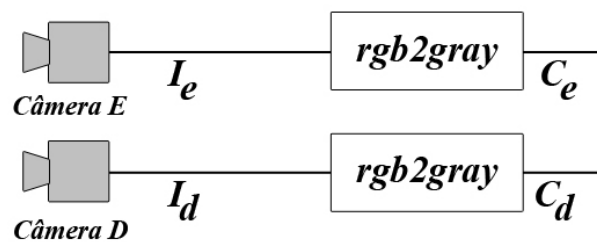


Figura 4.3: Representação em blocos do método desenvolvido até o pré-processamento das imagens de entrada.

4.6 Detecção dos Pontos Correspondentes Iniciais

Com as imagens obtidas pelas câmeras do sistema de visão estéreo pré-processadas, a próxima etapa consiste em obter os pares de pontos correspondentes iniciais, que são usados como base para o cálculo da matriz fundamental mais adiante. É essencial que o número de pontos correspondentes inicialmente obtidos seja o maior possível, de modo a garantir que a matriz fundamental a ser produzida seja a mais correta possível.

Neste trabalho, o processo de detecção dos pontos correspondentes iniciais é dividido em duas partes: primeiro, cantos em ambas as imagens precisam ser encontrados; depois, tais pontos precisam ser analisados de modo a verificar se estes conseguem formar correspondências entre coordenadas das imagens.

4.6.1 Obtendo o algoritmo que calcula mais cantos

Como descrito no Capítulo 3, o cálculo dos pontos correspondentes em uma cena costuma ser realizado a partir dos pontos que correspondem aos cantos dos elementos presentes nesta. Desta forma, inicialmente, um algoritmo de detecção de cantos precisa ser aplicado em ambas as imagens de entrada, de modo a se detectar pontos capazes de formar possíveis correspondências.

A questão da detecção de cantos em uma imagem é altamente conhecida no meio de processamento digital de imagens. Dessa forma, existem vários algoritmos capazes realizar tal procedimento. Levando tal fator em consideração, inicialmente pode-se pensar que a escolha de um algoritmo para uso no método desenvolvido neste trabalho pode ser aleatória; no entanto, uma vez que a matriz fundamental é formada a partir de uma aproximação linear de um conjunto de pontos correspondentes, é possível afirmar que, quanto mais cantos correspondentes o algoritmo de detecção de cantos produzir (e não apenas o maior número de cantos), melhor será a aproximação que a matriz fundamental irá realizar.

Assim, uma comparação entre alguns desses algoritmos de detecção de cantos foi realizada, a fim de se escolher o mais eficiente – ou seja, o que produz o maior número de cantos correspondentes. Para isso, um pequeno conjunto de teste foi definido, com pares de imagens capturadas com o sistema de visão estéreo (dez, no total), representado na Figura 4.4. Tais imagens apresentavam leves diferenças entre si (como número de elementos presentes, profundidade dos objetos e luminosidade, por exemplo).

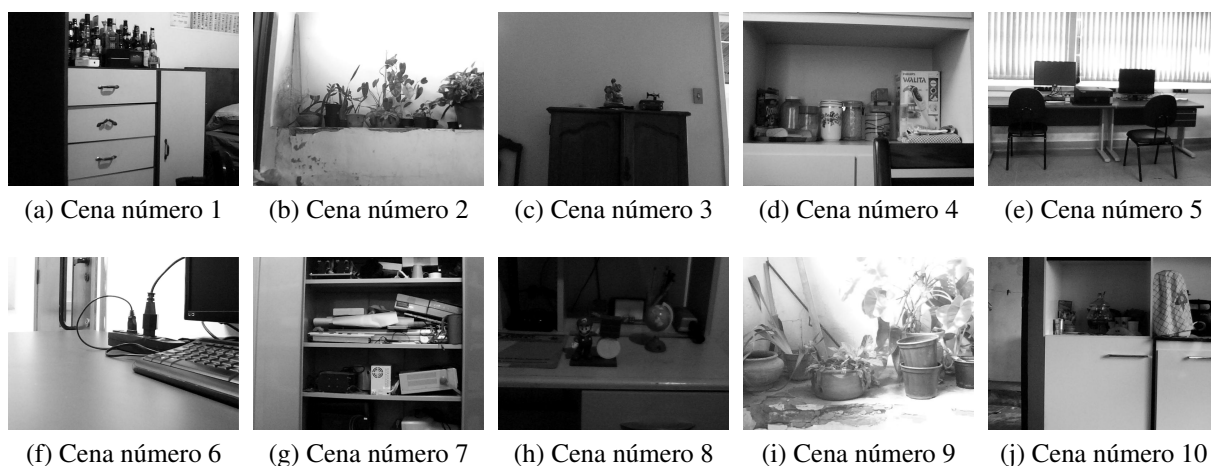


Figura 4.4: Conjunto de imagens (da câmera esquerda) capturadas pelo sistema de visão estéreo, utilizadas para a análise dos algoritmos de detecção de cantos.

Primeiramente, verificou-se qual o número de cantos que cada algoritmo poderia produzir ao ser aplicado a cada imagem do conjunto de teste. Os algoritmos de cantos que foram testados foram o *Binary Robust Invariant Scalable Keypoints* (BRISK) (LEUTENEGGER; CHLI;

SIEGWART, 2011), o *Speeded-Up Robust Features* (SURF) (BAY et al., 2008), o Algoritmo de Harris (HARRIS; STEPHENS, 1988), o *Minimum Eigen Value* ou Min8Val (SHI; TOMASI, 1994), o *Features from Accelerated Segment Test* (FAST) (ROSTEN; DRUMMOND, 2005, 2006) e o *Maximally Stable Extremal Regions* (MSER) (DONOSER; BISCHOF, 2006), explicados anteriormente no Capítulo 3. Eles foram aplicados às imagens do conjunto de teste exibidas anteriormente na Figura 4.4 sem a alteração de nenhum de seus parâmetros originais. O número de cantos que cada algoritmo obteve, tanto na imagem esquerda quanto na imagem direita de cada cena, são exibidos na Tabela 4.1, com os maiores valores obtidos pra cada cena destacados em negrito.

Tabela 4.1: Número de Cantos Obtidos nas Imagens Esquerda (E) e Direita (D) de cada Cena, para cada Detector de Cantos

| Cena Analisada | BRISK | | SURF | | Harris | | Min8Val | | FAST | | MSER | |
|-------------------|-------|-----|------|-----|--------|-----|-------------|-------------|------|-----|------|-----|
| | E | D | E | D | E | D | E | D | E | D | E | D |
| Cena 1 | 300 | 366 | 353 | 401 | 205 | 268 | 667 | 790 | 396 | 497 | 201 | 214 |
| Cena 2 | 242 | 186 | 442 | 397 | 456 | 439 | 1298 | 1295 | 331 | 276 | 227 | 195 |
| Cena 3 | 42 | 8 | 68 | 25 | 99 | 100 | 435 | 406 | 42 | 3 | 38 | 6 |
| Cena 4 | 159 | 230 | 339 | 393 | 239 | 410 | 732 | 966 | 232 | 346 | 214 | 292 |
| Cena 5 | 177 | 192 | 515 | 523 | 177 | 223 | 1073 | 1196 | 107 | 120 | 312 | 326 |
| Cena 6 | 182 | 160 | 627 | 572 | 241 | 135 | 1225 | 1098 | 190 | 115 | 329 | 290 |
| Cena 7 | 236 | 231 | 785 | 755 | 442 | 418 | 1397 | 1613 | 286 | 205 | 359 | 332 |
| Cena 8 | 13 | 67 | 29 | 150 | 27 | 52 | 473 | 388 | 9 | 50 | 32 | 106 |
| Cena 9 | 188 | 225 | 499 | 601 | 864 | 747 | 2849 | 2933 | 220 | 313 | 350 | 385 |
| Cena 10 | 94 | 91 | 242 | 251 | 214 | 270 | 841 | 876 | 114 | 130 | 129 | 139 |

Inicialmente, o algoritmo Min8Val consegue, em todos os casos, detectar o maior número de cantos. Porém, isso não significa necessariamente que os cantos detectados por tal algoritmo podem formar correspondências, e nem se as correspondências formadas estão corretas.

4.6.2 Obtendo o algoritmo que calcula mais correspondências

Embora o procedimento anterior mostre que o algoritmo Min8Val produz um maior número de cantos, é necessário descobrir se os cantos detectados por tal algoritmo realmente podem formar correspondências. A existência de um ponto em uma imagem não garante que este terá uma correspondência na outra, e como pontos correspondentes são essenciais na produção de uma matriz fundamental, os cantos que não formam correspondências devem ser descartados.

Assim, a partir dos cantos obtidos anteriormente, as correspondências que tais cantos poderiam formar foram calculadas, para cada cena, utilizando-se uma métrica de correspondência – neste caso, a Soma das Diferenças Quadradas, descrita anteriormente no Capítulo 3. Tal métrica foi aplicada aos cantos obtidos pelos algoritmos anteriores, de modo a descobrir o número de

correspondências formadas entre um par de imagens do conjunto de teste, produzidas a partir de tais cantos. Os resultados obtidos podem ser vistos na Tabela 4.2, onde os maiores valores de correspondência para cada cena estão destacados em negrito.

Tabela 4.2: Número de Pontos Correspondentes por Cena, para cada Detector de Cantos

| Cena Analisada | BRISK | SURF | Harris | Min8Val | FAST | MSER |
|----------------|-------|------------|--------|-----------|------|------|
| Cena 1 | 15 | 175 | 47 | 89 | 64 | 42 |
| Cena 2 | 15 | 225 | 74 | 115 | 42 | 62 |
| Cena 3 | 1 | 24 | 10 | 14 | 0 | 2 |
| Cena 4 | 8 | 151 | 23 | 82 | 25 | 39 |
| Cena 5 | 12 | 275 | 28 | 68 | 15 | 80 |
| Cena 6 | 1 | 78 | 10 | 25 | 12 | 13 |
| Cena 7 | 13 | 395 | 81 | 145 | 29 | 106 |
| Cena 8 | 0 | 9 | 4 | 12 | 0 | 7 |
| Cena 9 | 0 | 81 | 21 | 43 | 6 | 35 |
| Cena 10 | 3 | 119 | 43 | 64 | 19 | 34 |

Uma vez obtido tanto o número de cantos de uma cena e o número destes cantos que formam correspondências em cada uma, foi possível descrever de forma quantitativa a taxa de eficiência que cada algoritmo detector de cantos tem em produzir correspondências. A eficiência T_c de cada algoritmo foi calculada por meio da Equação 4.1:

$$T_c = \frac{C}{\left[\frac{(I_l + I_r)}{2} \right]} * 100 \quad (4.1)$$

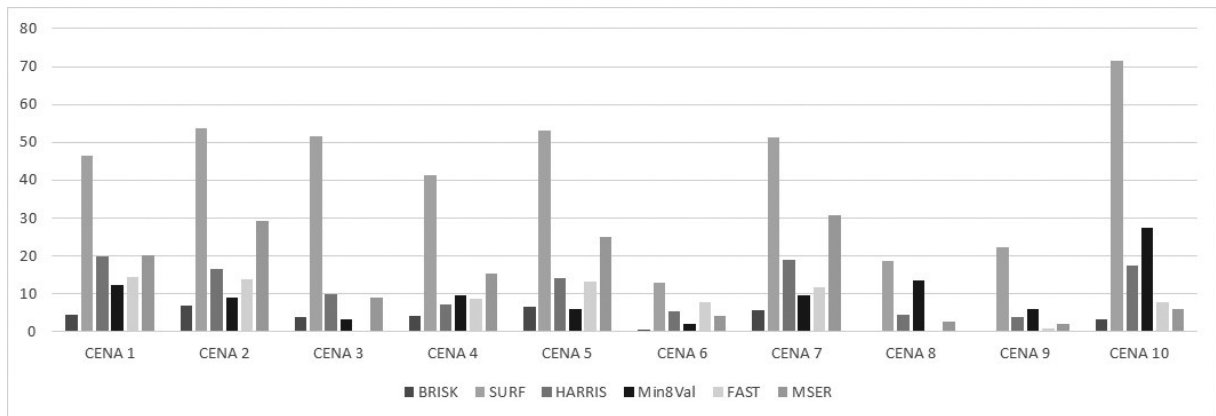
onde C é o número de pontos correspondentes, e I_l e I_r são o número de cantos obtidos nas imagens esquerda e direita, respectivamente. Na prática, a taxa apresentada é a divisão do número de correspondências obtidas pela média do número de cantos obtidos pelo algoritmo, multiplicada por cem, de modo a representar o valor obtido como uma porcentagem.

A taxa de eficiência expressada pela Equação 4.1 foi calculada para cada algoritmo de cantos descrito anteriormente, e os resultados obtidos estão representados na Tabela 4.3, com as maiores taxas para cada cena apresentadas em negrito. Além disso, o gráfico da Figura 4.5 representa visualmente as taxas obtidas.

Anteriormente, o algoritmo Min8Val era o algoritmo que mais encontrava cantos em um par de imagens; no entanto, a porcentagem de correspondências que ele produz a partir de tais cantos é muito baixo, o que o torna ineficiente para o uso neste trabalho. Ainda analisando a Tabela 4.3, é possível perceber que o algoritmo que produz a maior porcentagem de correspondências em todos os casos é o SURF. Em média, metade dos cantos obtidos com ele podem

Tabela 4.3: Taxa Aproximada de Pontos Correspondentes por Cena (%), para cada Detector de Cantos

| Cena Analisada | BRISK | SURF | Harris | Min8Val | FAST | MSER |
|----------------|-------|-------------|--------|---------|------|------|
| Cena 1 | 4.5 | 46.4 | 19.9 | 12.2 | 14.3 | 20.2 |
| Cena 2 | 7 | 53.6 | 16.5 | 8.9 | 13.8 | 29.4 |
| Cena 3 | 4 | 51.6 | 10.1 | 3.3 | 0 | 9.1 |
| Cena 4 | 4.1 | 41.3 | 7.1 | 9.7 | 8.7 | 15.4 |
| Cena 5 | 6.5 | 53 | 14 | 6 | 13.2 | 25.1 |
| Cena 6 | 0.6 | 13 | 5.3 | 2.2 | 7.9 | 4.2 |
| Cena 7 | 5.6 | 51.3 | 18.8 | 9.6 | 11.8 | 30.7 |
| Cena 8 | 0 | 18.7 | 4.5 | 13.5 | 0 | 2.7 |
| Cena 9 | 0 | 22.4 | 3.8 | 5.9 | 0.7 | 1.9 |
| Cena 10 | 3.2 | 71.5 | 17.4 | 27.5 | 7.85 | 6.1 |

**Figura 4.5: Gráfico que Apresenta a Taxa Aproximada de Pontos Correspondentes por Cena (%), para cada Detector de Cantos.**

formar correspondências, o que é um valor muito alto, se comparado com os outros algoritmos analisados.

Dessa forma, para o cálculo das correspondências iniciais, neste trabalho, utiliza-se o algoritmo SURF, por meio da função *detectSURFFeatures* presente em MatLab. Tal função calcula os cantos das imagens C_e e C_d , em tons de cinza, retornando as listas $corners_e$ e $corners_d$, uma para cada imagem de entrada, que contêm as coordenadas dos pontos que representam os cantos de cada imagem. A Figura 4.6 exibe a esquematização em blocos do método desenvolvido neste trabalho, até a etapa de detecção de cantos; da mesma forma que a Figura 4.3, o algoritmo *detectSURFFeatures* é representado como um único bloco para ambas as imagens, de modo a facilitar o entendimento do método como um todo.

Embora o algoritmo SURF seja utilizado na detecção de cantos, ele não é capaz de fazer as correspondências entre estes. No entanto, uma vez que a Soma das Diferenças Quadradas é a métrica de correspondência utilizada para validar a qualidade do algoritmo SURF, ela já

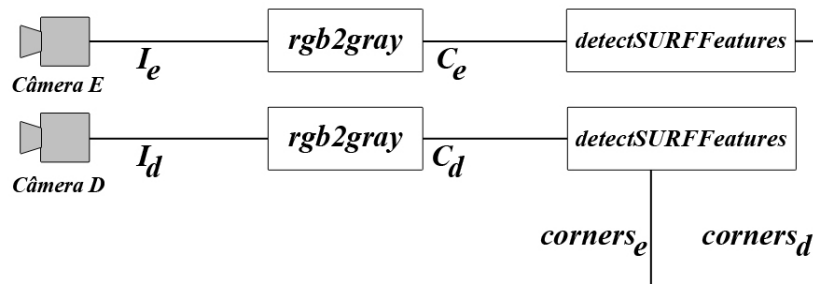


Figura 4.6: Representação em blocos do método desenvolvido até o cálculo dos cantos das imagens de entrada.

pode ser acrescentada neste trabalho como a métrica utilizada no cálculo das correspondências iniciais. Assim, é utilizada neste trabalho a Soma das Diferenças Quadradas como métrica de correspondência inicial, por meio da função *matchFeatures*, disponível em MatLab. Tal função recebe como parâmetros os cantos obtidos pelo algoritmo SURF, para cada imagem, bem como as próprias imagens de entrada (em tons de cinza) e o tipo de métrica de correspondência utilizada (no caso, a Soma das Diferenças Quadradas). A função tem como retorno dois conjuntos de pontos $matches_e$ e $matches_d$, um para cada imagem, de modo que pontos que possuam mesmo índice são correspondentes. Dessa forma, a Figura 4.6 pode ser atualizada, tornando-se a Figura 4.7, que representa a metodologia acrescida do cálculo das correspondências iniciais.

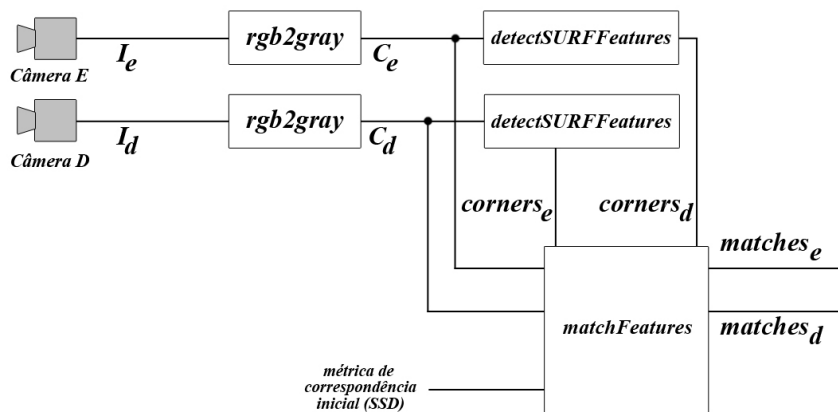


Figura 4.7: Representação em blocos do método desenvolvido até o cálculo das correspondências iniciais.

Partindo-se dos cantos correspondentes iniciais, o próximo passo é o cálculo da matriz fundamental, de modo a se obter o mapeamento entre todos os pontos correspondentes presentes nas imagens.

4.7 Estimação da Matriz Fundamental

Como descrito anteriormente no Capítulo 3, a matriz fundamental é utilizada como uma aproximação linear entre as coordenadas de pontos correspondentes em um par de imagens estéreo, de modo que um ponto possa ser mapeado em seu correspondente por meio do uso de tal matriz. Em geral, a matriz fundamental é estimada utilizando um algoritmo de aproximação linear em um subconjunto aleatório do conjunto de pontos correspondentes, de modo a se obter uma matriz, que deve ser submetida a uma métrica capaz de representar a sua taxa de eficiência. Caso a eficiência de tal matriz seja comprovada, ela pode ser utilizada como uma possível matriz fundamental.

Enquanto o cálculo de uma matriz fundamental segue um padrão definido (que é a escolha de correspondências aleatórias e a estimação linear da matriz), a taxa utilizada para se dizer a eficiência de uma matriz fundamental pode mudar de acordo com o escopo do problema em que a matriz fundamental será aplicada. Atualmente, na literatura, encontram-se várias métricas capazes de expressar o quanto uma matriz fundamental é eficiente; no entanto, a escolha aleatória de uma dessas pode impactar profundamente a eficiência do método desenvolvido neste trabalho. Isso ocorre principalmente devido ao fator de aproximação linear da matriz fundamental: algumas correspondências podem não serem mapeadas corretamente por uma matriz fundamental (principalmente se tais correspondências forem incorretas), de modo que estas acabam sendo descartadas nos processos posteriores de retificação e correspondência.

Como exemplo, na Figura 4.8 (baseada na Figura 3.3 do Capítulo 3) é representado um plano cartesiano onde os eixos contêm os pontos de um par de imagens U e V , de modo que as correspondências entre os pontos estão marcadas como pontos em tal plano. A curva cinza representa a aproximação linear dos pontos no plano cartesiano, definida pela matriz fundamental F . É possível observar que alguns pontos (marcados em branco) estão a uma distância da curva grande o suficiente para tornar impossível o seu mapeamento correto pela matriz fundamental, de modo que estas correspondências são descartadas durante o processo.

Dessa forma, outra análise comparativa foi efetuada, de modo a testar a qualidade das métricas existentes que definem a eficiência de uma matriz fundamental. Cada métrica foi aplicada a um conjunto de imagens que, na prática, é o mesmo conjunto de imagens utilizado anteriormente para testar os algoritmos de detecção de cantos. Os cantos de cada imagem foram calculados com o algoritmo SURF, e depois a Soma das Diferenças Quadradas foi aplicada de modo a se obter o número de correspondências iniciais de cada cena; na prática, os valores obtidos são os mesmos que apareceram anteriormente na Tabela 4.2, na coluna do algoritmo

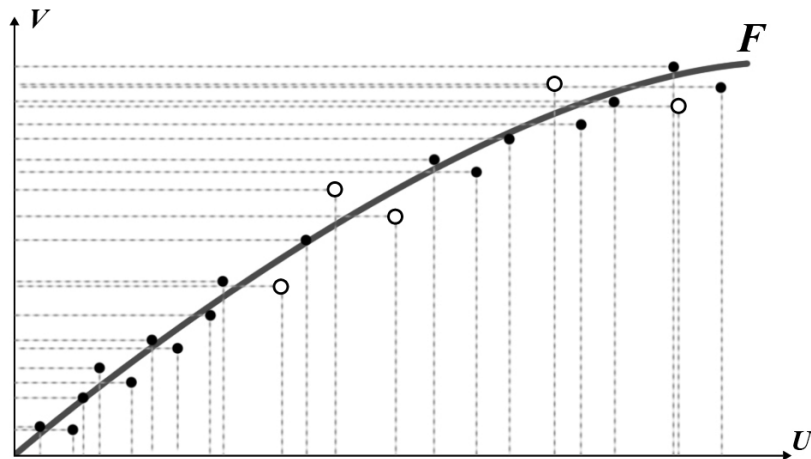


Figura 4.8: As correspondências que estão a uma distância muito grande da aproximação linear criada pela matriz fundamental (representadas pelos pontos brancos) não podem ser mapeadas por tal matriz, e são eliminadas durante o processo.

SURF.

As métricas de eficiência testadas foram a *Least Median of Squares* (LMedS) (ROUSSEEUW, 1984), a *Random Sample Consensus* (RANSAC) (FISCHLER; BOLLES, 1981), a *M-estimator Sample Consensus* (MSAC) (TORR; MURRAY, 1997) e a *Least Trimmed of Squares* (LTS) (ROUSSEEUW, 1984), todas apresentadas anteriormente no Capítulo 3. Para o cálculo da distância entre o par de pontos correspondentes e a aproximação linear da matriz fundamental, utilizou-se a distância de Sampson (SAMPSON, 1982), também descrita no Capítulo 3. Dessa forma, a quantidade de pontos correspondentes iniciais que podem ser aproximados por cada métrica está representada na Tabela 4.4.

Tabela 4.4: Número de Pontos Correspondentes Aproximáveis pela Matriz Fundamental, para cada Métrica de Qualidade

| Cena Analisada | Originais (SURF) | LMedS | RANSAC | MSAC | LTS |
|----------------|------------------|-------|--------|------|-----|
| Cena 1 | 175 | 88 | 44 | 48 | 87 |
| Cena 2 | 225 | 113 | 64 | 55 | 112 |
| Cena 3 | 24 | 12 | 13 | 13 | 12 |
| Cena 4 | 151 | 76 | 44 | 43 | 75 |
| Cena 5 | 275 | 138 | 57 | 56 | 137 |
| Cena 6 | 78 | 39 | 12 | 15 | 39 |
| Cena 7 | 395 | 198 | 95 | 105 | 197 |
| Cena 8 | 9 | 0 | 0 | 0 | 0 |
| Cena 9 | 81 | 41 | 24 | 19 | 40 |
| Cena 10 | 119 | 60 | 33 | 28 | 59 |

A partir daí, uma taxa de pontos que uma matriz fundamental consegue aproximar usando

determinada métrica foi obtida, para cada métrica analisada, dividindo o número de pontos correspondentes C_F obtidos após o cálculo da matriz fundamental pelo número de pontos correspondentes C_{SURF} originais do algoritmo SURF, e multiplicando o resultado obtido por cem, como representado na Equação 4.2:

$$T_s = \frac{C_F}{C_{SURF}} * 100 \quad (4.2)$$

Assim, os resultados obtidos para cada métrica de qualidade de uma matriz fundamental estão representados na Tabela 4.5 (com as maiores taxas para cada cena representadas em negrito), assim como no gráfico da Figura 4.9.

Tabela 4.5: Taxa Aproximada de Pontos Correspondentes Aproximáveis pela Matriz Fundamental (%), para cada Métrica de Qualidade

| Cena Analisada | LMedS | RANSAC | MSAC | LTS |
|----------------|-------------|-------------|-------------|-----------|
| Cena 1 | 50.3 | 25.1 | 27.4 | 49.7 |
| Cena 2 | 50.2 | 28.4 | 24.4 | 49.8 |
| Cena 3 | 50 | 54.2 | 54.2 | 50 |
| Cena 4 | 50.3 | 29.1 | 28.5 | 49.6 |
| Cena 5 | 50.9 | 20.7 | 20.4 | 49.8 |
| Cena 6 | 50 | 15.3 | 19.2 | 50 |
| Cena 7 | 50.1 | 24 | 26.6 | 49.8 |
| Cena 8 | 0 | 0 | 0 | 0 |
| Cena 9 | 50.6 | 29.6 | 23.5 | 49.4 |
| Cena 10 | 50.4 | 27.7 | 23.5 | 49.6 |

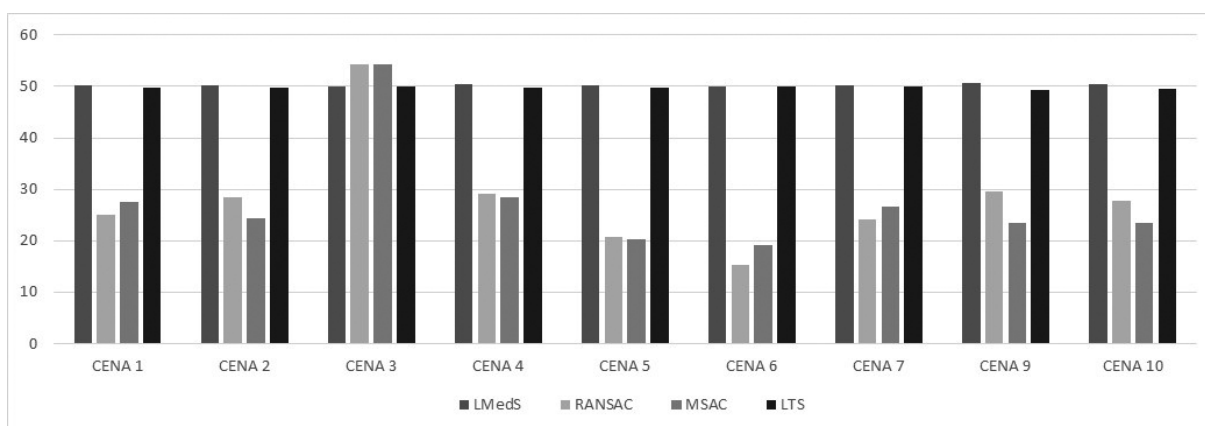


Figura 4.9: Gráfico que Representa a Taxa Aproximada de Pontos Correspondentes Aproximáveis pela Matriz Fundamental (%), para cada Métrica de Qualidade (o gráfico da Cena 8 foi omitido por não ter gerado nenhum ponto correspondente aproximável).

A análise da Tabela 4.5 mostra que, em praticamente todos os casos, a métrica de qualidade LMedS é a que consegue garantir que a matriz fundamental aproxime a maior porcentagem de

pontos. Dessa forma, essa métrica é a escolhida como a principal neste trabalho; no entanto, há uma outra questão a ser considerada.

Das quatro métricas de qualidade de uma matriz fundamental analisadas, a métrica MSAC apresenta uma vantagem em relação às outras. Enquanto as outras métricas só podem ser utilizadas caso exista ao menos dezesseis correspondências, a métrica MSAC pode ser utilizada com, no mínimo, oito. Como é possível que, em ambientes reais, algumas cenas a serem analisadas pelo método desenvolvido neste trabalho possam ser de má qualidade, de modo que a quantidade de correspondências formadas seja baixa, é possível que a métrica MSAC possa ser vista como uma alternativa à métrica LMedS caso não seja possível utilizar esta última, mesmo que em teoria a quantidade de correspondências que a métrica MSAC consiga aproximar em relação a uma matriz fundamental seja menor que a métrica LMedS.

Assim, a métrica LMedS é a padrão deste trabalho, mas quando uma cena real é analisada, se esta produzir poucas correspondências (mais precisamente, entre oito e dezesseis) a métrica MSAC é utilizada. No entanto, caso haja menos de oito correspondências iniciais, mesmo que todas sejam possíveis de serem aproximadas por uma matriz fundamental, o cálculo desta última não pode ser realizado – tal condição, por exemplo, é a responsável por alguns dos valores nulos para os pares correspondentes da cena 8 da Tabela 4.5. Dessa forma, não é possível prosseguir com o processo de geração de um mapa de disparidades para a cena a partir deste ponto ².

No MatLab, a função *estimateFundamentalMatrix* é a função utilizada para a estimação da matriz fundamental. Seus parâmetros principais são os pares de pontos correspondentes e o tipo de métrica de qualidade a ser utilizada. Tal função retorna, além da matriz fundamental F , dois conjuntos contendo novas correspondências $matches'_e$ e $matches'_d$ para as imagens esquerda e direita, respectivamente. Diferente dos conjuntos de correspondências iniciais $matches_e$ e $matches_d$, estes novos conjuntos contêm apenas as correspondências que a matriz fundamental consegue mapear corretamente.

A Figura 4.10 exibe a representação em blocos atualizada com o bloco responsável pela estimação da matriz fundamental. Além disso, há um outro bloco, que recebe como parâmetro o número de correspondências inicialmente obtidas, e que retorna o tipo de métrica de qualidade a ser utilizada pelo bloco da matriz fundamental (LMedS para mais de dezesseis correspondências, ou MSAC para mais de oito).

Uma vez conhecida a matriz fundamental, a partir dela é possível mapear qualquer ponto de uma das imagens de entrada em um ponto na outra imagem de entrada. A partir de tal informação, é possível realizar um processo de retificação para o par de imagens obtido, já que

²Mais detalhes sobre tal problema são discutidos no Capítulo 5.

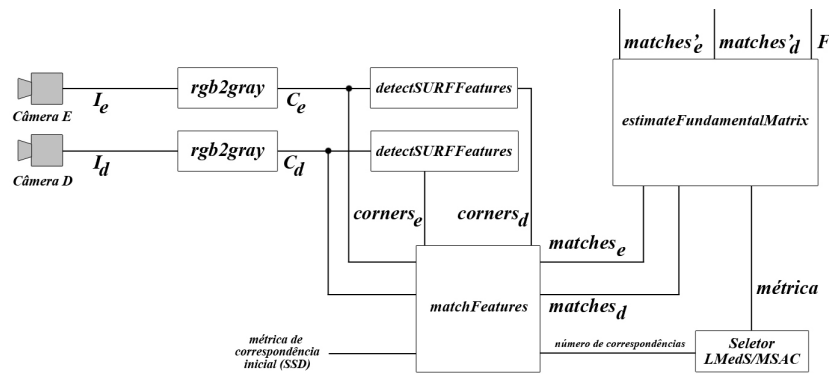


Figura 4.10: Representação em blocos do método desenvolvido até o cálculo da matriz fundamental.

agora é possível determinar as coordenadas dos pontos que precisam ficar na mesma altura em ambas as imagens. Assim, além da simplicidade exigida, o processo realizado até agora elimina a exigência de um padrão de calibração para o mapeamento dos elementos presentes na cena.

Obtida a matriz fundamental a partir da geometria da própria cena, a próxima etapa deste trabalho envolve a retificação do par de imagens obtidas, de modo a ser possível o posterior cálculo da disparidade dos elementos da cena.

4.8 Etapa de Retificação

O objetivo do processo de retificação é, basicamente, garantir que os pontos correspondentes em um par de imagens estejam na mesma altura (ou seja, com a mesma coordenada y) em ambas.

Como exemplo, na Figura 4.11 é exibido um par de imagens estéreo, com alguns pontos representados. As linhas amarelas estão representadas no par de imagem para indicar os pares de pontos correspondentes. É possível observar que as linhas não estão exatamente paralelas ao eixo x das imagens, o que significa que os pontos correspondentes não estão na mesma altura.

Em visão estéreo, a retificação cilíndrica costuma ser um processo mais complexo que a retificação planar, de modo que esta última é o tipo de retificação utilizada neste trabalho. Assim, uma vez que a matriz fundamental permite mapear qualquer par de pontos correspondentes presentes nas imagens capturadas pelo sistema de visão estéreo, ela é utilizada no método desenvolvido para estimar qual o grau de deslocamento que cada plano da imagem capturada deve sofrer, em três direções: altura, largura e perspectiva. Daí, no momento em que as imagens forem reprojadas no plano comum que a retificação planar cria, garante-se que pontos correspondentes possuam a mesma coordenada y (HARLEY; ZISSERMAN, 2004).

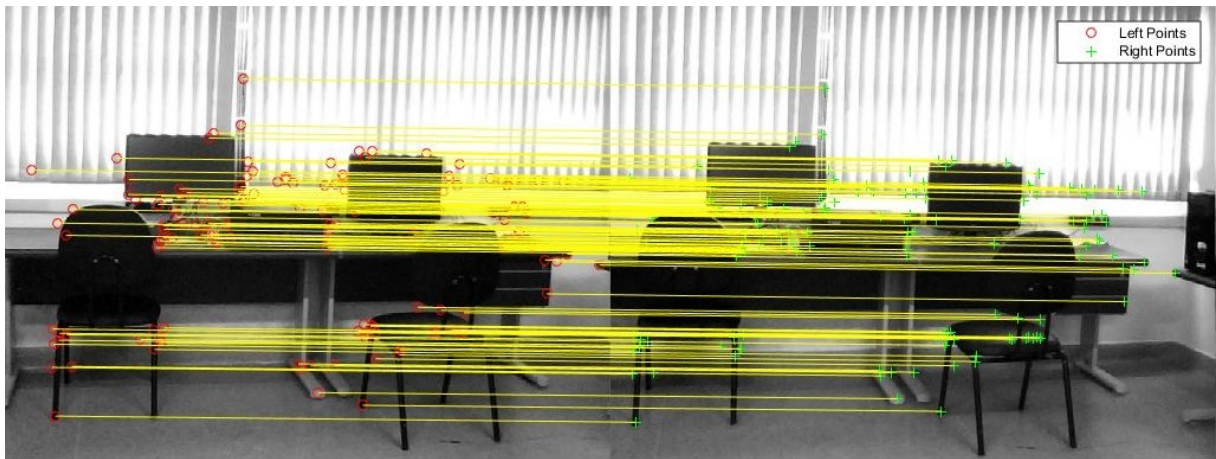


Figura 4.11: Par de imagens estéreo, com os respectivos pontos correspondentes representados, antes do processo de retificação.

Em MatLab, utiliza-se a função *estimateUncalibratedRectification* para auxiliar o processo de retificação de um par de imagens estéreo. Seus parâmetros incluem o par de pontos correspondentes $matches'_e$ e $matches'_d$ e a matriz fundamental F . Nenhuma imagem é passada a tal função como parâmetro; na verdade, o retorno desta função é um par de matrizes T_e e T_d , que representam as transformações projetivas que os planos das imagens originais devem sofrer para terem seus pontos correspondentes retificados (FUSIELLO; TRUCCO; VERRI, 2000). De modo sucinto, as transformações projetivas a serem aplicadas às imagens originais do sistema de visão estéreo se baseiam na matriz fundamental, de acordo com as etapas a seguir (HARLEY; ZISSERMAN, 2004; PEREK; MAKOWSKI; NAPIERALSKI, 2016):

1. A câmera esquerda do sistema é rotacionada (utilizando a matriz fundamental), de modo que as epipoles da imagem de tal câmera se encontrem com o eixo x apenas no infinito;
2. A mesma rotação anterior é aplicada à câmera direita do sistema, de modo a restaurar a geometria da cena completa;
3. A câmera direita sofre uma nova rotação, desta vez baseada na matriz de rotação do sistema de visão estéreo;
4. A escala de ambas as câmeras é ajustada; cada matrizes de homografia entre as posições originais e as novas posições de cada câmera é uma transformação projetiva.

Uma vez obtidas as transformações projetivas, a função *imwarp* é utilizada para aplicar tais transformações nas imagens correspondentes. Ela recebe como parâmetro uma imagem C_e (ou C_d), além de uma transformação projetiva T_e (ou T_d), para retornar uma nova imagem R_e (ou R_d) que representa a imagem de entrada com a transformação aplicada. Na verdade, observa-se que

a função anterior não realiza a retificação de fato; ela apenas fornece os parâmetros para isso. Na Figura 4.12 é exibida a representação em blocos do método desenvolvido neste trabalho, até o processo de retificação; para facilitar o entendimento, ambas as funções foram representadas como um único bloco, para cada uma.

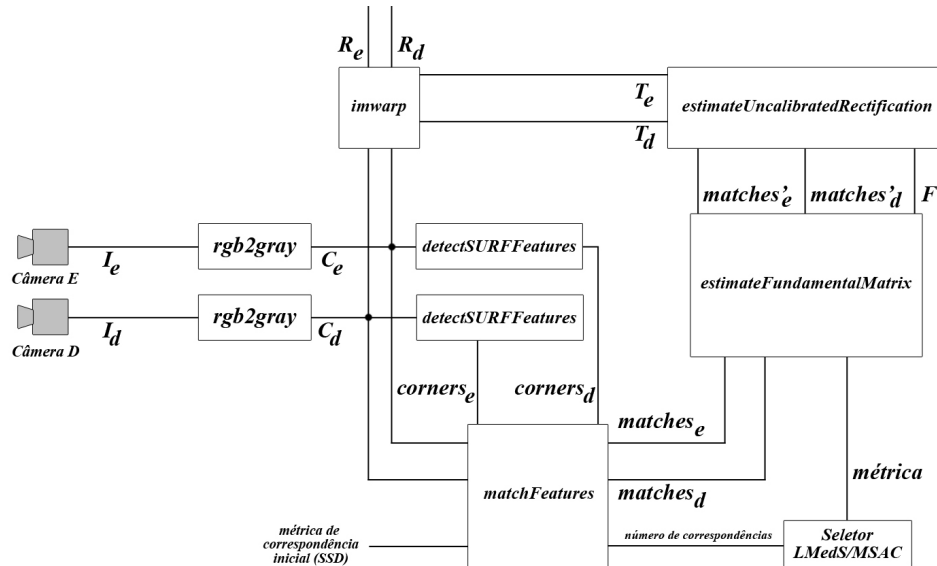


Figura 4.12: Representação em blocos do método desenvolvido até o processo de retificação.

Finalizando a etapa de retificação, a Figura 4.13 exibe o resultado da retificação aplicada ao par de imagens da Figura 4.11. Há uma questão a se observar no processo de retificação planar: dependendo do caso, as transformações projetivas calculadas pela retificação podem ser tão distintas que são capazes de distorcer, em muito, o plano das imagens do par de imagens a ser retificado. Isso ocorre, em geral, quando há linhas epipolares se cruzando em pelo menos um dos planos das imagens – mais especificamente, quando o plano da imagem intersecciona a linha de base. Em geral, quando tal condição ocorre, interrompe-se o processo de geração do mapa de disparidades, devido à má qualidade que este irá possuir³.

4.9 Cálculo das Disparidades entre Pontos

Uma vez que o par de imagens obtidas pelo sistema de visão estéreo está retificado, é possível calcular a disparidade dos pontos presentes na cena originalmente analisada. Geralmente, uma função de disparidade calcula o mapa de disparidades aplicando uma métrica de correspondência a todos os pares de pontos correspondentes de uma imagem, quantificando os valores de disparidade obtidos, e depois representando tais valores para cada ponto como uma imagem em tons de cinza.

³Mais detalhes sobre isso são descritos no Capítulo 5.



Figura 4.13: Par de imagens estéreo, com os respectivos pontos correspondentes representados, após a retificação planar. Percebe-se que os pontos correspondentes estão alinhados em relação ao eixo y das imagens.

No trabalho desenvolvido, a disparidade é calculada utilizando um método semi-global, que funciona como uma combinação entre os tipos de métodos locais e globais. Primeiro, uma medida de contraste de ambas as imagens, obtida com o uso do filtro Sobel, é calculada; depois, utilizando a métrica de Soma das Diferenças Absolutas (exibida anteriormente no Capítulo 3), blocos de pontos presentes em ambas as imagens têm seus valores de disparidade computados. Por fim, a similaridade entre os valores de disparidade de blocos vizinhos é “forçada”, de modo a garantir a consistência das disparidades no mapa todo (HIRSCHMULLER, 2005).

Em MatLab, a função responsável pelo cálculo de um mapa de disparidades de uma cena é a função *disparity*. Ela recebe como entrada um par de imagens estéreo R_e e R_d , previamente retificados, e tem como retorno uma matriz D_{map} , contendo todos os valores de disparidades dos pontos de uma cena, calculados pelo método semi-global. Tal função também pode receber parâmetros adicionais, como a faixa de valores que a disparidade pode assumir, o tamanho da janela utilizada pela métrica de disparidade, e um indicador para o método local ser utilizado no lugar do método semi-global. Na Figura 4.14 está representada a esquematização em blocos do método desenvolvido neste trabalho, até o processo de criação de um mapa de disparidades.

Uma vez obtido o mapa de disparidades, este pode ser convertido em uma imagem, de modo a se obter uma representação visual dos diferentes valores de disparidade presentes em uma cena. Como exemplo, na Figura 4.15 é exibido um mapa de disparidades obtido para o par de imagens da Figura 4.2. O método utilizado no cálculo das disparidades de tal mapa foi o semi-global, utilizando o tamanho padrão de janela da função *disparity* (15×15 pixels) e definindo dezesseis possíveis valores para uma disparidade. Os tons mais claros de cinza indicam áreas mais próximas do observador (no caso, as câmeras do sistema de visão estéreo).

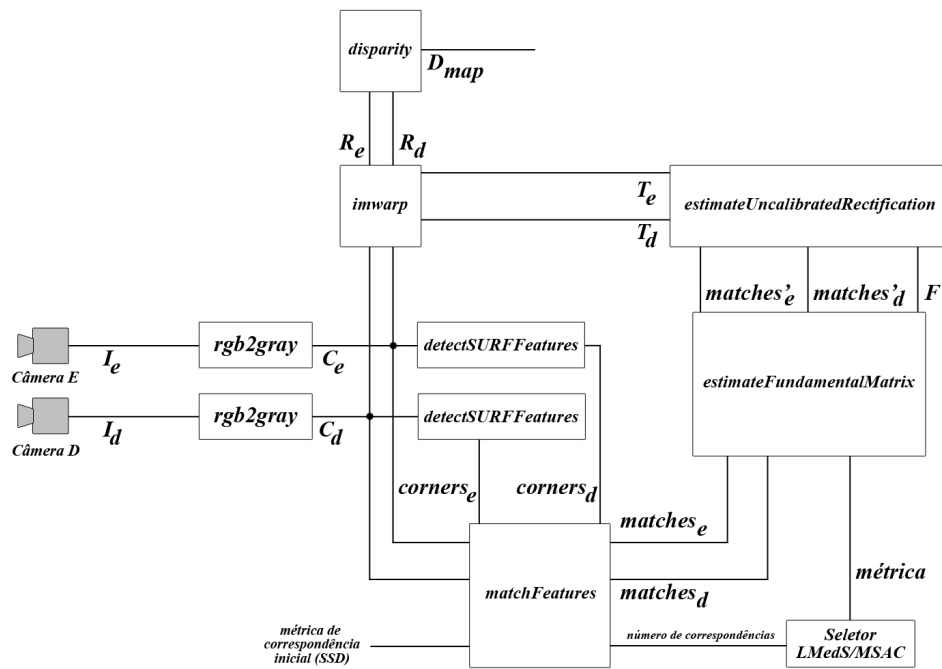


Figura 4.14: Representação em blocos do método desenvolvido até o processo de cálculo do mapa de disparidades.

4.10 Correção do Mapa de Disparidades Obtido

Após a geração de um mapa de disparidades, este é submetido a um pós-processamento, com o objetivo de corrigir algumas distorções que surgem durante o processo. Observando o mapa de disparidades representado na Figura 4.15, é possível perceber que o processo de retificação aplicado antes da etapa de cálculo das disparidades produz algumas áreas no mapa que não fazem parte da cena original. Na realidade, tais áreas são as regiões representadas pela cor preta nas imagens retificadas da Figura 4.13.

Essa característica ocorre devido ao fato da retificação “distorcer” os planos das imagens originais para o alinhamento horizontal dos pontos correspondentes. Uma vez que, no método desenvolvido neste trabalho, a retificação utiliza duas transformações projetivas para ser executada, é possível utilizar essas mesmas transformações projetivas para se corrigir a posição do mapa, de modo que o plano de imagem de tal mapa fique o mais próximo possível dos planos das imagens originais.

Pensando nesse fator, uma nova transformação projetiva, baseada nas duas transformações utilizadas na retificação, é calculada com o objetivo de realizar tal correção planar no mapa de disparidades. Essa nova transformação projetiva é calculada como sendo a inversa das matrizes obtidas pela média aritmética das duas outras matrizes que representam transformações projetivas, e que foram utilizadas no processo de retificação. Matematicamente, sendo M_E e M_D as

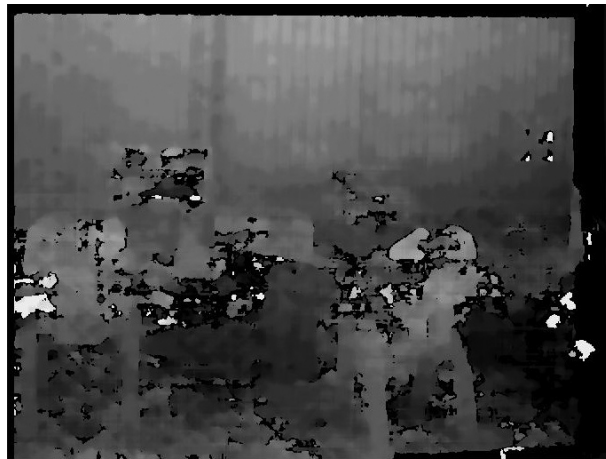
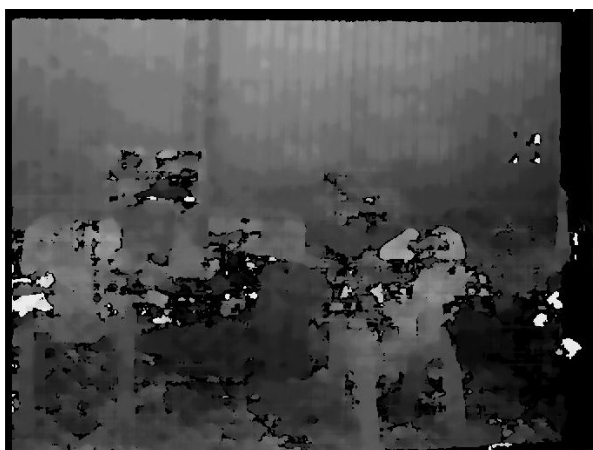


Figura 4.15: Mapa de disparidades obtido para o par de imagens da Figura 4.2: tons mais claros de cinza indicam áreas mais próximas do observador.

matrizes correspondentes às transformações projetivas das imagens esquerda e direita, respectivamente, ambas utilizadas no processo de retificação, a transformação corretora M_C é dada pela Equação 4.3:

$$M_C = \left(\frac{M_E + M_D}{2} \right)^{-1} \quad (4.3)$$

Assim, essa transformada corretora é aplicada ao mapa de disparidades obtido, de modo que este volte a apresentar uma perspectiva mais fiel às imagens inicialmente obtidas pelo sistema de visão estéreo. Na Figura 4.16 está representado o mapa de disparidades presente na Figura 4.15, antes e depois da aplicação da transformação corretora.



(a) Antes da Correção



(b) Depois da Correção

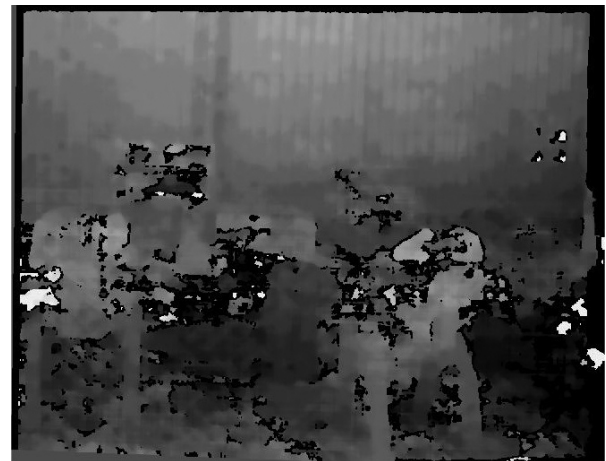
Figura 4.16: Mapa de disparidades da Figura 4.15 antes e depois da aplicação da transformada corretora: é possível observar que a área de distorção preta, herdada do processo de retificação, e presente do lado direito do mapa, foi removida.

Com a aplicação da transformada corretora, o mapa de disparidades assume uma perspectiva mais próxima das imagens originais do sistema de visão estéreo, facilitando possíveis visualizações entre áreas do mapa de disparidades e áreas da cena original que estão relacionadas. No entanto, ao utilizar tal transformada, blocos com um único tom de cinza (correspondendo a um valor indefinido de disparidade) costumam aparecer em um dos lados das imagens. Como tais blocos não são resultantes do cálculo da disparidade, e sim da operação anterior, devem ser reduzidos o máximo possível – uma vez que não possuem utilidade no mapa de disparidades.

Devido a tal problema, aplica-se um algoritmo que remove os blocos verticais (ou colunas) de pontos que contenham somente um valor indefinido, e que partam dos eixos verticais do mapa de disparidades obtido pelo processo anterior. Essa remoção de blocos verticais é feita até que o primeiro bloco vertical contendo ao menos um ponto com valor definido para a disparidade seja encontrado; dessa forma, a área contendo pontos indefinidos é reduzida ao máximo possível. Uma vez tal algoritmo aplicado, o mapa de disparidades resultante é o mapa final, um dos objetivos buscados neste trabalho. Assim, o mapa resultante do processo desenvolvido neste trabalho pode ser visto na Figura 4.17.



(a) Imagem Original



(b) Mapa de Disparidades Final

Figura 4.17: Uma das imagens originais capturadas pelo sistema de visão estéreo (a), e o mapa de disparidades final, com todas as operações realizadas (b).

Por fim, a representação em blocos, na Figura 4.18 é apresentado o método desenvolvido completo, com os dois últimos blocos adicionados: o que é responsável por aplicar a transformação corretora, e o que é responsável pela eliminação das bordas incorretas. O primeiro exige como parâmetros o mapa de disparidades D_{map} inicialmente obtido, e as duas transformações projetivas utilizadas no processo anterior de retificação. Ele executa o processo de correção e retorna um novo mapa D_{map} , que é passado como parâmetro ao segundo bloco;

daí, este último bloco executa o processo de remoção das laterais irrelevantes do mapa e, por fim, retorna o mapa D_{map} totalmente corrigido.

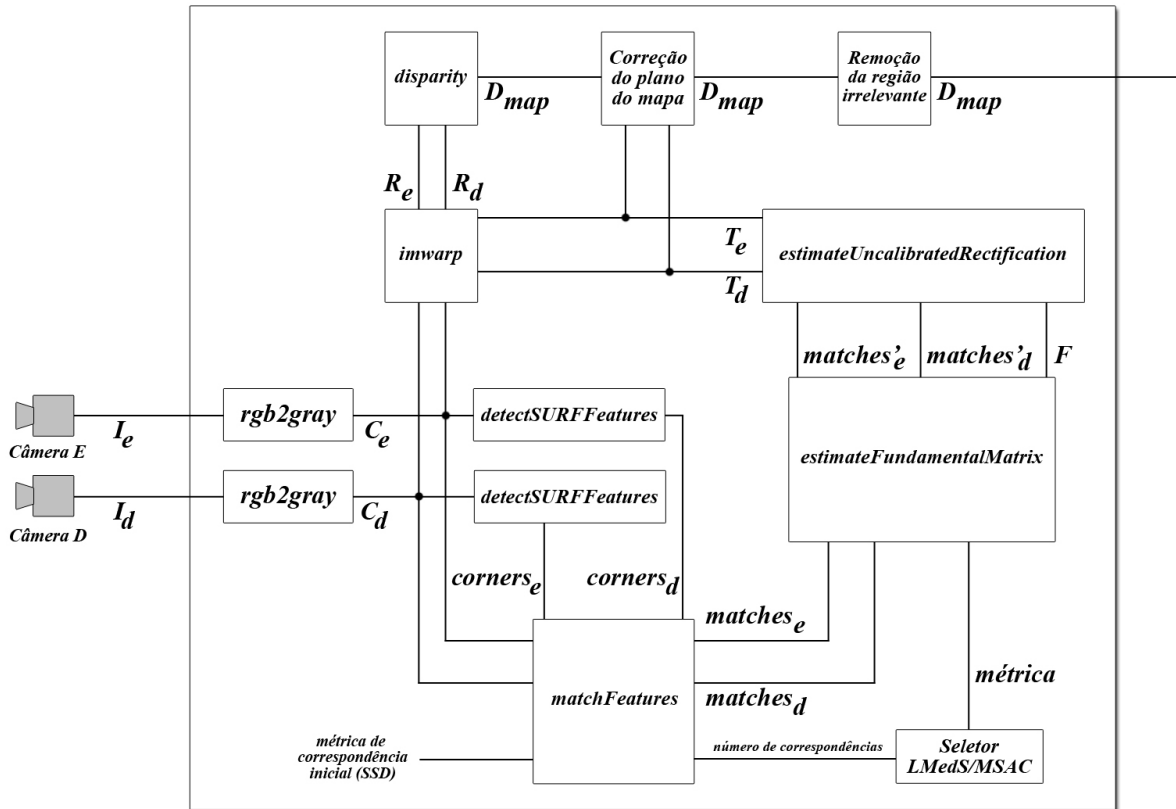


Figura 4.18: Representação em blocos final do método desenvolvido neste trabalho.

O Apêndice A apresenta uma representação simplificada do método desenvolvido completo, em pseudocódigo (incluindo alguns tratamentos de entrada, que são explicados mais adiante, Capítulo 5).

Capítulo 5

ANÁLISE DOS RESULTADOS OBTIDOS

5.1 Principal Objetivo da Análise

Após o desenvolvimento do processo de geração de um mapa de disparidades estar completo, resultados iniciais puderam ser obtidos. Assim, a eficiência do processo pôde ser questionada, tanto com o objetivo de garantir a qualidade dos resultados inicialmente obtidos, quanto comparando os mesmos com resultados obtidos por algum outro método existente na literatura, também capaz de produzir mapas de disparidades. Os resultados de tais análises são discutidos neste Capítulo.

5.2 Interferência de Agentes Externos no Mapa

Pelo fato do método desenvolvido ser voltado ao uso em ambientes reais, não é raro que, dependendo da estrutura de uma cena, agentes externos do ambiente contendo-a possam provocar inconsistências nos mapas de disparidades produzidos. Um exemplo de ocorrência desta inconsistência pode ser visto na Figura 5.1, onde uma cena e seu possível mapa de disparidades é apresentado. A parede de fundo na imagem original possui um padrão que se repete, e que apresenta textura única; assim, tal área “confunde” o cálculo das disparidades do mapa, de modo que tal região aparenta estar mais próxima do que qualquer outro elemento da cena.

Uma forma de entender o surgimento de tais inconsistências é entender como cada agente externo pode afetar o mapa de disparidades, de modo que seja possível descrever uma maneira de contorná-lo. Assim, é possível destacar os seguintes pontos:

- Um mapa de disparidades é altamente afetado pelas condições de luminosidade em uma cena: quando a cena analisada é afetada por uma forte intensidade de luz (ou a ausência

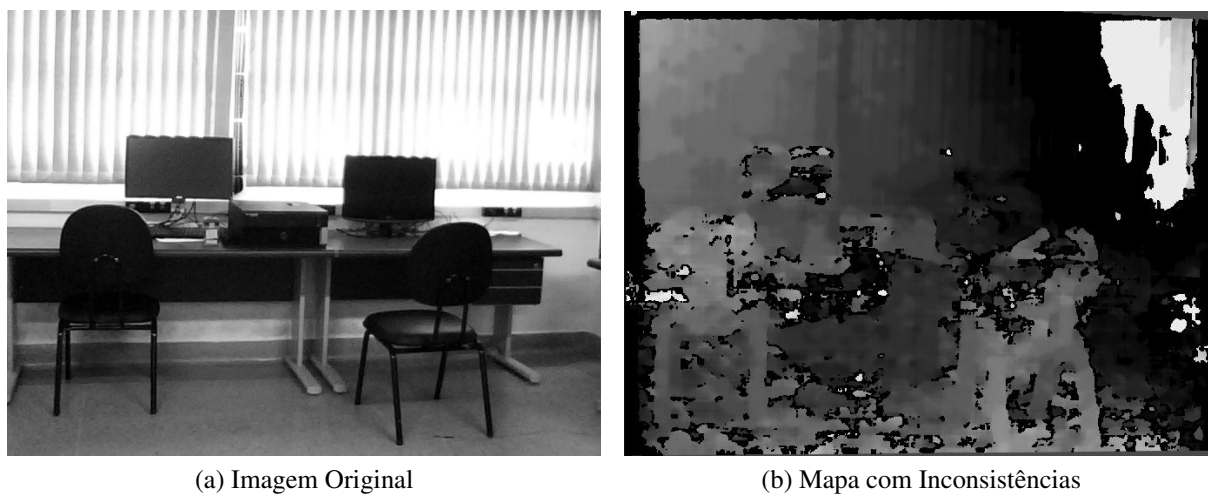


Figura 5.1: Um exemplo de inconsistência presente no cálculo do mapa de disparidades (canto superior direito da Figura (b)): uma região de textura única pode criar valores errados de disparidade no mapa.

desta), é mais difícil mapeá-la, uma vez que fica mais difícil encontrar pontos correspondentes em um par de imagens extraído a partir dela.

- Ruídos nas imagens podem afetar a precisão do cálculo das correspondências de uma cena. Por exemplo, um ponto de ruído pode ser mapeado de maneira incorreta como um canto, pelo algoritmo SURF;
- A repetição de padrões em uma cena (como no caso exibido pela Figura 5.1) pode fazer com que áreas que deveriam apresentar um alto valor de disparidade apresentem um baixo valor para esta, ou vice-versa;
- Elementos que forem sobrepostos por outros elementos durante o processo de captura das imagens podem não ter a disparidade mapeada, uma vez que eles podem aparecer apenas em uma das imagens do par;
- Reflexões, ou transparências, podem fazer com que o mapa de disparidade exiba objetos inexistentes na cena original.

Dessa forma, é possível afirmar que um sistema de visão estéreo é altamente sensível a agentes externos de uma cena. No entanto, há como esboçar possíveis formas de se contornar os problemas que possíveis agentes possam causar.

No caso da luminosidade, da repetição de padrões e de reflexões ou transparências, é possível que o mapa de disparidades tenha de ser gerado várias vezes, ou até mesmo que as

cenar tenham de ser recapturadas, até que os cantos de uma cena possam ser capturados corretamente. Já no caso de elementos sobrepostos, uma vez que eles estão em tal condição, o mapeamento dos objetos que os sobrepõem já é suficiente. Por fim, no caso de ruídos, a aplicação de alguns filtros antes da detecção dos cantos correspondentes pode resolver o problema.

5.3 Problemas na Geração de um Mapa de Disparidades

No Capítulo 4 são descritos dois casos em que o processo de geração de um mapa de disparidades não pode ser executado. Tais casos são descritos como:

1. Quando o número de pares de pontos correspondentes é menor do que oito. Nesse caso, o baixo número de correspondências impede o cálculo da matriz fundamental, uma vez que não há correspondências suficientes para se obter uma aproximação linear do modo descrito no Capítulo 3.
2. Quando linhas epipolares se cruzam nos planos das imagens retificadas. Isso implica que as transformações projetivas, obtidas a partir da matriz fundamental, ao serem aplicadas às imagens originais, irão causar uma distorção muito alta nos planos de tais imagens. Desse modo, é impossível manter os pares de pontos correspondentes verticalmente alinhados de maneira correta.

Em geral, quando um desses problemas ocorre, a continuidade do processo de geração do mapa de disparidades é, pelo menos inicialmente, inviável. No entanto, alguns pontos podem ser observados sobre tais problemas.

No primeiro caso, em geral, o baixo número de pontos correspondentes se deve à má qualidade das imagens capturadas para uma cena. Conseqüentemente, tal condição pode levar a uma baixa produção de cantos pelo método SURF nas imagens capturadas. Dessa forma, em aplicações reais, o problema pode ser contornado refazendo a captura das imagens pelo sistema estéreo, e reiniciando o processo de geração do mapa de disparidades. No trabalho desenvolvido, quando o processo de geração de um mapa é influenciado por tal problema, o processo é interrompido; porém, é possível adaptar o método desenvolvido para se contornar tal problema: basta fazer com que, caso o número de pontos correspondentes detectados inicialmente seja baixo, o método refaça a captura das imagens da cena, com o objetivo de reiniciar o processo com um par de imagens que supostamente tenham uma melhor qualidade.

Já no segundo caso, as linhas epipolares aparecem devido à forma como a matriz fundamental foi calculada, provavelmente utilizando alguma correspondência incorreta. Uma vez que a

natureza do cálculo da matriz fundamental é aleatória (por esta se basear na escolha arbitrária de pares de pontos correspondentes), o processo não precisa ser reiniciado do zero quando tal problema ocorre; basta recalcular-se a matriz fundamental, quantas vezes forem necessárias, até que se chegue a uma que não permita o aparecimento de linhas epipolares nos planos das imagens retificadas. No trabalho desenvolvido, o processo de geração da matriz fundamental segue tal princípio.

5.4 Minimizando Distorções no Mapa de Disparidades

Devido à natureza aleatória da forma de cálculo da matriz fundamental utilizada no método desenvolvido, não é raro que o mapa de disparidades representado ainda não seja eficiente o bastante para uso em uma análise estéreo, mesmo após a aplicação das correções. A razão para tal problema está no processo de produção da matriz fundamental: durante o processo de retificação, ela pode produzir um par de transformadas projetivas capazes de causar uma grande distorção nas imagens originais capturadas pelo sistema de visão estéreo, de modo que a área de tons de cinza indefinidos seja muito grande. Como exemplo, pode-se observar a Figura 5.2, que exhibe diferentes mapas de disparidades obtidos para uma mesma cena, de acordo com a matriz fundamental inicialmente calculada para cada mapa.

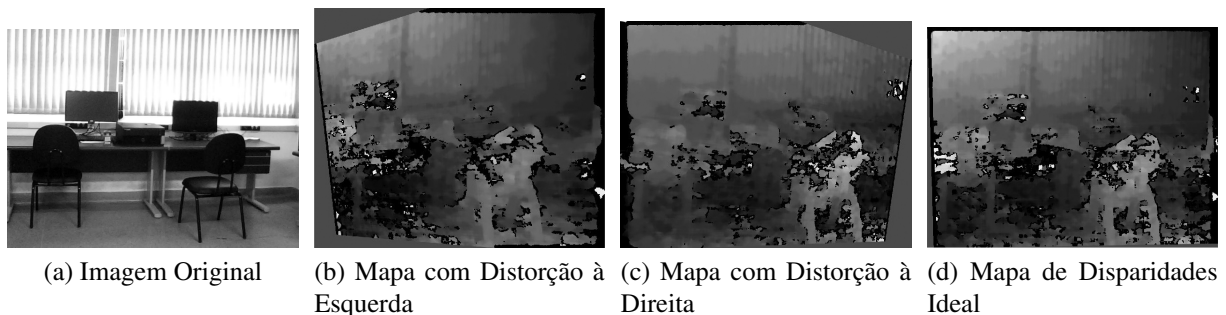


Figura 5.2: Diferentes mapas de disparidades obtidos, por diferentes matrizes fundamentais, para uma mesma cena capturada pelo sistema de visão estéreo.

A imagem (a) da Figura 5.2 representa uma das imagens capturadas de uma cena, enquanto (b), (c) e (d) representam diferentes mapas de disparidades obtidos a partir da mesma, usando matrizes fundamentais distintas. Dentre os três mapas obtidos, o mapa (d) é o que melhor se aproxima de um mapa de disparidades ideal, já que os mapas (b) e (c) possuem uma grande distorção em seus planos de projeção, devido à alta distorção que as imagens originais sofreram na etapa de retificação. Devido a esse fator, após os resultados iniciais terem sido obtidos, decidiu-se desenvolver uma maneira com que os mapas de disparidades produzidos possuam uma distorção mínima em seu plano de projeção.

Para um mapa de disparidades ser considerado ideal, a distorção planar que a aplicação da matriz fundamental irá causar em ambas as imagens do sistema de visão estéreo durante o processo de retificação deve ser a mínima possível, de modo que a distorção causada no mapa de disparidades resultante também seja mínima. Considerando tal fator, desenvolveu-se um algoritmo simples, que calcula a porcentagem de tons de cinza indefinidos que estão presentes em um mapa de disparidades produzido para uma determinada cena. Caso tal mapa apresente essa porcentagem maior que um limiar definido, este mapa é descartado, uma nova matriz fundamental é recalculada, e um novo mapa de disparidades é produzido para a cena analisada. Esse processo é repetido até que a porcentagem de tons de cinza nulos seja menor que o limiar definido, de modo a se produzir um mapa o mais próximo possível do ideal.

Nos testes realizados com algumas cenas obtidas, a análise empírica dos mapas obtidos mostra que, em geral, os melhores mapas de disparidade são gerados ao limitar-se a área de tons de cinza indefinidos para, no máximo, 5% do mapa de disparidades, como exibido na Figura 5.3.

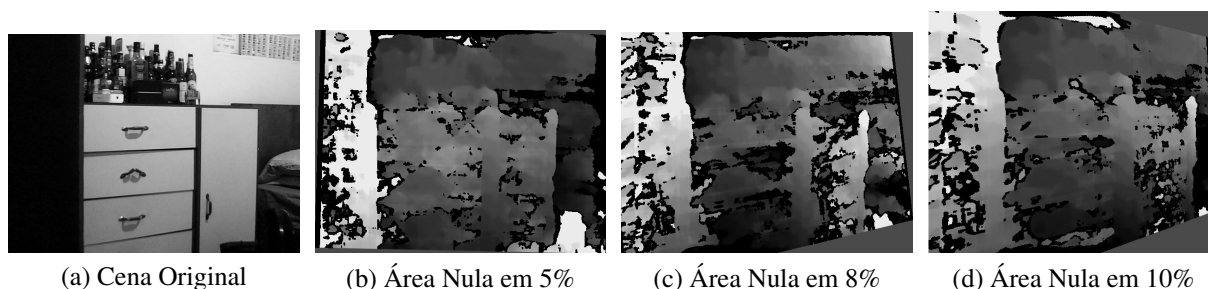


Figura 5.3: Mapas de disparidades de uma cena, que podem ser produzidos limitando-se a área de tons de cinza indefinidos em uma porcentagem máxima.

A vantagem de se limitar o tamanho da área de tons de cinza indefinidos presentes no mapa de disparidades envolve a questão da qualidade do mapa; quanto menor tal área, mais dados relevantes sobre a disparidade da cena estarão presentes neste. No entanto, tal método para se conseguir um mapa considerado ideal pode consumir um bom tempo de processamento – pois vários mapas gerados, considerados de qualidade insuficiente, podem ser descartados antes disso – o que pode prejudicar a usabilidade de tal algoritmo em situações onde tempo de execução é importante.

5.5 Tempo de Geração de um Mapa de Disparidades

Visando a análise do tempo de processamento estimado para a geração de um mapa de disparidades, o método desenvolvido foi aplicado a um conjunto de imagens pré-definidas,

verificando-se o tempo necessário para a geração de cada mapa de disparidade.

A Tabela 5.1 mostra o tempo de processamento necessário, em segundos, para a geração de cada tipo de mapa de disparidade, a partir de um par de imagens previamente capturadas. Tais imagens pertencem ao conjunto de testes do banco de imagens de Middlebury (SCHARSTEIN; SZELISKI, 2002; SCHARSTEIN et al., 2014), e o mapa de disparidades de cada imagem teve seu tempo de processamento calculado cinco vezes. As imagens utilizadas, bem como os respectivos mapas obtidos, estão apresentadas no Apêndice B. O *hardware* utilizado para os testes de tempo consistiu de um *notebook* com um processador Intel® Core™I5 de 1.70GHz, além do uso de 4Gb de RAM; não houve utilização de GPU.

Tabela 5.1: Tempo Estimado para o Processamento de um Mapa de Disparidades (em Segundos)

| Cena | Tempo de Processamento em Segundos | | | | | | |
|--------------|------------------------------------|--------|--------|--------|--------|---------------|---------------|
| | 1 | 2 | 3 | 4 | 5 | Média | Desvio Padrão |
| Bicycle2 | 0.7744 | 0.8124 | 0.8086 | 0.7855 | 0.7846 | 0.7985 | 0.0165 |
| Computer | 0.8629 | 0.4347 | 0.4188 | 0.4801 | 0.4765 | 0.5721 | 0.1854 |
| Crusade | 0.7132 | 0.7711 | 0.7337 | 0.7603 | 0.7470 | 0.7393 | 0.0227 |
| Livingroom | 0.7536 | 0.7760 | 0.7555 | 0.7794 | 0.7941 | 0.7617 | 0.0171 |
| Plants | 0.8228 | 0.8019 | 0.8312 | 0.8411 | 0.8737 | 0.8186 | 0.0264 |
| Djembe | 0.7727 | 0.7734 | 0.7696 | 0.8014 | 0.8225 | 0.7719 | 0.0232 |
| Newkuba | 0.7827 | 0.7443 | 0.7875 | 0.8063 | 0.8086 | 0.7715 | 0.0259 |
| Australia | 0.7613 | 0.7943 | 0.7658 | 0.7951 | 0.7714 | 0.7738 | 0.0160 |
| Classroom2 | 0.7803 | 0.7530 | 0.7937 | 0.7792 | 0.7395 | 0.7757 | 0.0222 |
| Hoops | 0.7537 | 0.7680 | 0.7684 | 0.7946 | 0.7701 | 0.7634 | 0.0147 |
| <i>Média</i> | 0.7778 | 0.7429 | 0.7623 | 0.7588 | 0.7433 | 0.7547 | 0.0146 |

A Tabela 5.1 mostra que a média geral do tempo de processamento de um mapa de disparidades, combinada com o desvio padrão, significa que tal mapa pode ser processado com menos de um segundo de custo, o que é um tempo viável. No entanto, uma ressalva deve ser feita: os tempos da tabela foram obtidos de modo que em nenhuma vez o mapa precisou ser refeito devido a algum problema com a matriz fundamental (falta de qualidade ou a alta distorção na retificação). Caso essa condição aconteça, é possível que este tempo aumente, dependendo de quantas vezes a matriz fundamental precisar ser refeita.

5.6 Análise Comparativa com o Método Calibrado de Zhang

Com o objetivo de verificar a qualidade do método desenvolvido neste trabalho, os mapas de disparidades produzidos por tal método foram comparados a mapas produzidos por algum outro método conhecido.

Uma maneira de obter o grau de qualidade de um mapa de disparidades é por meio do uso de alguma métrica de similaridade. A mais utilizada em bases de dados de visão estéreo envolve o cálculo do erro médio que um mapa tem em relação a outro mapa (ou a um mapa de disparidades perfeito, também chamado de *groundtruth*). Tal erro médio é dado pela média aritmética dos valores obtidos em um mapa de erros E , que é uma matriz bidimensional com as mesmas dimensões dos mapas de disparidades analisados. O erro médio em cada ponto (x, y) de E é calculado da seguinte forma, representada na Equação 5.1:

$$E(x, y) = \begin{cases} 0, & \text{se } d(x, y) < t \\ 1, & \text{caso contrário} \end{cases} \quad (5.1)$$

onde $d(x, y)$ é a métrica comparativa utilizada, e t é um limiar que reflete a maior diferença aceitável entre pontos de coordenadas iguais, presentes tanto no mapa de disparidades M , quanto no mapa de disparidades G , de modo a comparar o primeiro mapa em relação ao último. Já a métrica comparativa utilizada pode ser tanto as diferenças absolutas entre as intensidades dos pontos na posição (x, y) de cada mapa, como representada pela Equação 5.2:

$$d(x, y) = |M(x, y) - G(x, y)| \quad (5.2)$$

ou então as diferenças quadradas entre as intensidades dos mesmos pontos, como representada na Equação 5.3:

$$d(x, y) = [M(x, y) - G(x, y)]^2 \quad (5.3)$$

Independente da métrica utilizada pelo erro médio, quanto mais próximo de 0 a média do mapa de erros, mais eficiente é o mapa. Bases de dados como as de Middlebury (SCHARSTEIN; SZELISKI, 2002; SCHARSTEIN et al., 2014) costumam utilizar a diferença absoluta como métrica para a qualidade de um mapa de disparidades, embora a métrica de diferenças quadradas seja capaz de realçar melhor grandes diferenças entre mapas de disparidades.

O método desenvolvido neste trabalho utiliza para o cálculo de disparidade o algoritmo *SemiGlobal* (HIRSCHMULLER, 2005), como descrito anteriormente no Capítulo 4. Tal algoritmo já teve sua taxa de eficácia calculada em relação às imagens da base de dados de Middlebury, de modo que este apresentou uma taxa de erro de 25% – ou seja, sua eficácia é de 75% em média (Middlebury College, 2016). Por esse motivo, a comparação apresentada neste trabalho foi realizada apenas com a metodologia de Zhang (2000), uma vez que esta é semelhante

à metodologia deste trabalho. O Apêndice C apresenta a comparação visual entre os mapas de disparidades obtidos tanto pelo método de Zhang quanto pelo método desenvolvido neste trabalho. As imagens utilizadas para comparação foram todas capturadas a partir de cenas reais do dia-a-dia.

Assim, na Tabela 5.2 é exibida a porcentagem de similaridade obtida entre os mapas do método apresentado neste trabalho e os mapas obtidos pelo método de Zhang (2000), tanto para a métrica de Diferenças Absolutas quanto para a métrica de Diferenças Quadradas, onde o limiar definido entre as intensidades dos pontos foi de dez tons de cinza entre os pontos analisados.

Tabela 5.2: Taxa de Similaridade Obtida entre os Métodos Desenvolvidos e o de Zhang (2000)

| Cena Analisada | Taxa de Similaridade (%) | |
|----------------------|--------------------------|--------------------------|
| | com Diferenças Absolutas | com Diferenças Quadradas |
| Cena 1 | 72.83 | 70.86 |
| Cena 2 | 80.45 | 75.28 |
| Cena 3 | 75.44 | 71.76 |
| Cena 4 | 83.80 | 82.51 |
| Cena 5 | 76.07 | 74.23 |
| Cena 6 | 72.82 | 69.03 |
| Cena 7 | 71.45 | 67.91 |
| Cena 8 | 65.10 | 60.10 |
| Cena 9 | 71.54 | 67.73 |
| Cena 10 | 77.60 | 74.23 |
| <i>Média</i> | 74.71 | 71.36 |
| <i>Desvio Padrão</i> | 5.23 | 5.91 |

A média de quase 75% de similaridade obtida pela análise da Tabela 5.2 é um resultado satisfatório, uma vez que esta média é idêntica ao resultado de 75% de similaridade que o algoritmo de disparidade *SemiGlobal* (HIRSCHMULLER, 2005), utilizado neste trabalho, apresenta na tabela de resultados disponíveis na base de dados de Middlebury (Middlebury College, 2016).

Essa característica do método desenvolvido de manter a qualidade de um algoritmo de disparidade é algo importante, uma vez que isso vem do fato de que o cálculo do mapa de disparidades, no método desenvolvido neste trabalho, é algo que pode ser tratado de forma independente. Assim, caso exista a necessidade de melhora de qualidade dos mapas a serem obtidos, basta que o algoritmo *SemiGlobal* seja trocado por outro algoritmo cuja qualidade seja considerada como melhor, sem afetar a estrutura do método desenvolvido.

Visualmente, o método desenvolvido também apresenta um grau de semelhança próximo ao método de Zhang. Já no que tange à questões de usabilidade, uma das vantagens do método desenvolvido é a de que não há o uso de um padrão de calibração para a geração do mapa de

disparidades de uma cena, uma vez que os parâmetros utilizados são calculados utilizando-se a geometria da própria cena a ser analisada – diferente do método utilizado por Zhang, que em geral exige uma etapa de calibração, onde há intervenção humana. Além disso, a dispensa do padrão de calibração no método desenvolvido também abre a possibilidade de uso em tempo real de um sistema de visão estéreo, o que dificilmente o método de Zhang permitiria – uma vez que, neste último, qualquer ajuste na configuração dos parâmetros intrínsecos e extrínsecos do sistema exigiria uma nova calibração, consumindo tempo.

Capítulo 6

CONCLUSÕES

6.1 Principais Conclusões

O objetivo deste trabalho foi a análise, a combinação e o uso de ajustes entre um conjunto de algoritmos, conhecidos em visão estéreo e em estado da arte, para se obter uma metodologia capaz de gerar um mapa de disparidades, para uma determinada cena real, representando a profundidade dos elementos desta mesma cena. O mapa de disparidades foi obtido partindo-se da geometria da cena, analisada por um par de câmeras, para se chegar a um mapeamento entre os pontos existentes nas imagens capturadas por cada câmera. Utilizando conceitos geométricos e métricas de correspondência, foi possível a criação do mapa de disparidades, que passou por um processamento posterior (para a correção projetiva) para ser considerado ideal para uso.

A análise dos resultados obtidos com a metodologia apresentada mostrou que esta produz mapas com cerca de 75% de similaridade em relação a mapas produzidos por outros métodos, com média de menos de um segundo de tempo de processamento, o que valida a eficiência de tal metodologia.

6.2 Possíveis Trabalhos Futuros

Em trabalhos futuros, pretende-se utilizar os algoritmos apresentados a fim de torná-los algo embarcado, e que questões como a caracterização de cena e navegação possam ser analisadas, de modo a permitir o seu teste por pessoas com algum tipo de deficiência visual. Mas, para tais testes ocorrerem, os aspectos apresentados a seguir, dentre outros, devem ser analisados:

1. Como determinar a localização dos objetos mais próximos em uma cena (por exemplo: esquerda, direita, centro) em relação ao sistema?

2. Como determinar qual a distância aproximada dos objetos próximos em relação ao sistema?
3. Como fornecer tal informação sobre a cena a um deficiente visual? Áudio, toque, ou algo do tipo?
4. O tempo de processamento de um mapa (e da análise deste) é algo viável para uso em tempo real?

Os itens 1 e 2 são, de certa forma, rapidamente solucionáveis: basta o uso de alguma técnica de segmentação, combinada com alguma função capaz de mapear a intensidade de um *pixel* do mapa em uma distância real (metros, por exemplo).

O item 3 é uma característica a ser estudada, pois o áudio apresenta vantagem em relação ao toque: teoricamente é fácil informar a um cego, via áudio, tanto a localização de um objeto em uma cena quando a distância em relação a este (do ponto de vista do sistema), por exemplo, se comparado a utilizar alguma sensação de vibração.

Já o item 4 é, provavelmente, o mais complexo, pois o tempo de processamento de um mapa de disparidades deve ser o menor possível, o que exigiria o máximo de otimização possível de código. Uma das alternativas seria substituir o uso de ambientes de desenvolvimento pelo uso de dispositivos lógicos programáveis, embarcando os algoritmos utilizados e, como consequência, reduzindo o tempo de processamento destes.

Embora outras alternativas devam ser estudadas a respeito da questão de dispositivos lógicos programáveis, uma das alternativas na produção dos algoritmos propostos em tais dispositivos consistiria no uso da biblioteca HDL Coder, disponível na própria ferramenta MatLab, que tem por objetivo justamente a conversão de código escrito na linguagem padrão da ferramenta para código em Verilog ou VHDL. O código gerado pelo HDL Coder pode ser utilizado tanto em um *Field Programmable Gate Array* (FPGA) ou em um *Application Specific Integrated Circuits* (ASIC). Dessa forma, por meio dessa biblioteca, os algoritmos apresentados aqui poderiam ser embarcados mais facilmente.

Por fim, uma última observação que pode ser realizada é a de que novos algoritmos podem ser estudados em trabalhos futuros, de modo a adaptar a metodologia desenvolvida de acordo com os objetivos futuros. Alguns algoritmos utilizados atualmente neste trabalho já são considerados canônicos – porém, isso pode ser explicado pelo fato de um sistema capaz de ser utilizado por pessoas com algum tipo de deficiência visual exigir a necessidade de ser funcional, e tais algoritmos são estáveis. De qualquer forma, como o campo de visão estéreo é uma

área amplamente estudada na literatura atual - principalmente sobre questões como retificação (PEREK; MAKOWSKI; NAPIERALSKI, 2016) e disparidade (SCHARSTEIN et al., 2014; MA et al., 2016; DROUYER et al., 2017) - novas técnicas devem ser analisadas, de modo que elas talvez possam substituir as atualmente utilizadas, com o objetivo de atingir um alto grau de eficiência do sistema.

REFERÊNCIAS

- ABDEL-AZIZ, Y. I.; KARARA, H. M. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. In: *Symposium on Close-Range Photogrammetry*. Urbana, Illinois, USA: American Society of Photogrammetry, 1971. v. 1, p. 1–18.
- Acessibilidade Brasil. *Dirce*. Rio de Janeiro, RJ: [s.n.], 2014. Disponível em: <<http://www.acessibilidadebrasil.org.br/joomla/softwares?id=686>>.
- BARROIS, B. et al. Resolving stereo matching errors due to repetitive structures using model information. *Pattern Recognition Letters*, Elsevier Science Inc., New York, NY, USA, v. 31, n. 12, p. 1683–1692, set. 2010.
- BAY, H. et al. SURF: Speeded up robust features. *Computer Vision and Image Understanding*, Barcelona, Spain, v. 110, n. 3, p. 346–359, jun. 2008.
- BELHUMEUER, P. N. A binocular stereo algorithm for reconstructing sloping, creased, and broken surfaces in the presence of half-occlusion. In: *International Conference on Computer Vision*. Berlin, Germany: IEEE Computer Society, 1993. p. 431–438.
- BIRCHFIELD, S.; TOMASI, C. Depth discontinuities by pixel-to-pixel stereo. In: *International Conference of Computer Vision*. Bombay, India: IEEE Computer Society, 1998. p. 1073–1080.
- CALIN, G.; RODA, V. O. Real-time disparity map extraction in a dual head stereo vision system. *Latin American Applied Research*, v. 37, p. 21–24, 2006.
- CAMPOS, T. E.; BABU, B. R.; VARMA, M. Character recognition in natural images. In: *International Conference on Computer Vision Theory and Applications*. Lisboa, Portugal: Springer International Publishing, 2009. v. 68.
- CHUMERIN, N. et al. Learning eye vergence control from a distributed disparity representation. *International Journal of Neural Systems*, World Scientific Publishing Company, v. 20, n. 4, p. 267–278, 2010.
- COLLADO, J. et al. Self-calibration of an on-board stereo-vision system for driver assistance systems. In: *Intelligent Vehicles Symposium*. Tokyo, Japan: IEEE Computer Society, 2006. p. 156–162.
- CORONADO, G. A. P. et al. Detection and classification of road signs for automatic inventory systems using computer vision. *Integrated Computer-Aided Engineering*, IOS Press, Amsterdam, The Netherlands, v. 19, n. 3, p. 285–298, jul. 2012.

- COX, I. J. et al. A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding*, v. 63, p. 542–567, maio 1996.
- DONOSER, M.; BISCHOF, H. Efficient maximally stable extremal region (MSER) tracking. In: *International Conference on Computer Vision and Pattern Recognition*. Beijing, China: IEEE Computer Society, 2006. v. 1, p. 553–560.
- DROUYER, S. et al. Sparse stereo disparity map densification using hierarchical image segmentation. In: *International Symposium of Mathematical Morphology and Its Applications to Signal and Image Processing*. Cham, Switzerland: Springer International Publishing, 2017. p. 172–184.
- FENG, D.; MCCARTHY, C. Enhancing scene structure in prosthetic vision using iso-disparity contour perturbation maps. In: *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Osaka, Japan: IEEE Computer Society, 2013. p. 5283–5286.
- FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, ACM, New York, NY, USA, v. 24, n. 6, p. 381–395, jun. 1981.
- Freedom Scientific, Inc. *JAWS Screen Reader*. St. Petersburg, FL, USA: [s.n.], 2015. Disponível em: <<http://www.freedomscientific.com/Products/Blindness/JAWS>>.
- FUSIELLO, A.; TRUCCO, E.; VERRI, A. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, Springer International Publishing, Secaucus, NJ, USA, v. 12, n. 1, p. 16–22, jul. 2000.
- GASCHLER, A.; BURSCHEKA, D.; HAGER, G. Epipolar-based stereo tracking without explicit 3d reconstruction. In: *International Conference on Pattern Recognition*. Istanbul, Turkey: IEEE Computer Society, 2010. p. 1755–1758.
- GONZALES, R. C.; WOODS, R. E. *Digital Image Processing*. 3. ed. London, England: Pearson, 2007.
- GROSSO, E.; TISTARELLI, M. Active/dynamic stereo vision. *Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, v. 17, n. 9, p. 868–879, set. 1995.
- GU, Z. et al. Local stereo matching with adaptative support-weight, rank transform and disparity calibration. *Pattern Recognition Letters*, Chengdu, China, p. 1230–1235, 2008.
- HAMZAH, R. A.; HAMID, A. M. A.; SALIM, S. I. M. The solution of stereo correspondence problem using block matching algorithm in stereo vision mobile robot. In: *International Conference on Computer Research and Development*. Kuala Lumpur, Malaysia: IEEE Computer Society, 2010. p. 733–737.
- HARA, F. Artificial emotion of face robot through learning in communicative interactions with human. In: *International Workshop on Robot and Human Interactive Communication*. Kurashiki, Okayama, Japan: IEEE Computer Society, 2004. p. 7–15.
- HARLEY, R.; ZISSERMAN, A. *Multiple View Geometry in Computer Vision*. 2. ed. Cambridge, MA, USA: Cambridge University Press, 2004.

- HARRIS, C.; STEPHENS, M. A combined corner and edge detector. In: *Alvey Vision Conference*. Manchester, England, UK: University of Sheffield Printing Unit, 1988. p. 147–151.
- HEIKKILA, H.; SILVEN, O. A four-step camera calibration procedure with implicit image correction. In: *International Conference on Computer Vision and Pattern Recognition*. San Juan, Puerto Rico: IEEE Computer Society, 1997. p. 1106–1112.
- HIRSCHMULLER, H. Accurate and efficient stereo processing by semi-global matching and mutual information. In: *International Conference on Computer Vision and Pattern Recognition*. San Diego, CA, USA: IEEE Computer Society, 2005. v. 2, p. 807–814.
- JACOBI, R. P.; CARDOSO, R. B.; BORGES, G. A. VoC: a reconfigurable matrix for stereo vision processing. In: *Parallel and Distributed Processing Symposium*. Rhodes Island: IEEE Computer Society, 2006.
- KAMENCAY, P. et al. 3D image reconstruction from 2D CT slices. In: *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*. Budapest, Hungary: IEEE Computer Society, 2014. p. 1–4.
- KRAMM, S.; BENSRAHAI, A.; MICHE, P. Self calibration of a road stereo color vision system using a similarity criterion. In: *Annual Conference on Industrial Electronics*. Paris, France: IEEE Computer Society, 2006. p. 3544–3549.
- KUMAR, S. et al. Stereo rectification of uncalibrated and heterogeneous images. *Pattern Recognition Letters*, Via Della Science, Udine, Italy, v. 31, p. 1445–1452, mar. 2010.
- LAUREANO, G. T.; PAIVA, M. S. V. Algoritmo genético para criação de mapas de disparidade estéreo. *Workshop de Visão Computacional*, São José do Rio Preto, São Paulo, Brazil, 2007.
- LEUTENEGGER, S.; CHLI, M.; SIEGWART, R. BRISK: Binary robust invariant scalable keypoints. In: *International Conference on Computer Vision*. Barcelona, Spain: IEEE Computer Society, 2011. p. 2548–2555.
- LI, D. et al. Integrating a statistical background-foreground extraction algorithm and svm classifier for pedestrian detection and tracking. *Integrated Computer-Aided Engineering*, IOS Press, Amsterdam, The Netherlands, v. 20, n. 3, p. 201–216, jul. 2013.
- LOWE, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, Kluwer Academic Publishers, Dordrecht, Netherlands, v. 60, n. 2, p. 91–110, 2004.
- MA, N. et al. Accurate dense stereo matching based on image segmentation using an adaptive multi-cost approach. *Symmetry*, MDPI AG, Basel, Switzerland, v. 8, n. 12, p. 159–180, dez. 2016.
- MARTULL, S.; PERIS, M.; FUKUI, K. Realistic CG stereo image dataset with ground truth disparity maps. In: *International Conference of Pattern Recognition*. Tsukuba, Japan: IEEE Computer Society, 2012. p. 40–42.

MESQUITA, R. G.; MELLO, C. A. B.; ALMEIDA, L. H. E. V. A new thresholding algorithm for document images based on the perception of objects by distance. *Integrated Computer-Aided Engineering*, IOS Press, Amsterdam, The Netherlands, v. 21, n. 2, p. 133–146, 2014.

MicroPower. *Virtual Vision*. São Caetano do Sul, SP: [s.n.], 2014. Disponível em: <<http://www.virtualvision.com.br/>>.

Middlebury College. *Middlebury Stereo Evaluation - Version 3*. 2016. Disponível em: <<http://vision.middlebury.edu/stereo/eval3/>>.

MUSLEH, B. et al. Extrinsic parameter self-calibration and nonlinear filtering for in-vehicle stereo vision systems at urban environments. In: *International Conference on Computer Vision Theory and Applications*. Lisboa, Portugal: IEEE Computer Society, 2014. v. 3, p. 427–434.

NEO, E. S.; SAKAGUCHI, T.; YOKOI, K. A humanoid robot that listens, speaks, sees and manipulates in human environments. In: *International Conference on Multisensor Fusion and Integration for Intelligent Systems*. Seoul, South Korea: IEEE Computer Society, 2008. p. 419–425.

Núcleo do Computação Eletrônica. *Projeto DOSVOX*. Rio de Janeiro, RJ: [s.n.], 2002. Disponível em: <<http://intervox.nce.ufrj.br/dosvox/>>.

OLIVEIRA, L. M. B. *Cartilha do Censo 2010 - Pessoas com Deficiência*. Brasília, DF, Brasil: Secretaria dos Direitos Humanos da Presidência da República (SDH/PR), 2012.

PAUWELS, K.; HULLE, M. M. V. Head-centric disparity and epipolar geometry estimation from a population of binocular energy neurons. *International Journal of Neural Systems*, World Scientific Publishing Company, v. 22, n. 3, 2012.

PEREK, P.; MAKOWSKI, D.; NAPIERALSKI, A. Efficient uncalibrated rectification method for stereo vision systems. In: *International Conference Mixed Design of Integrated Circuits and Systems*. Lodz, Poland: IEEE Computer Society, 2016. v. 1, n. 1, p. 89–92.

POLLEFEYS, M.; KOCH, R.; GOOL, L. V. A simple and efficient rectification method for general motion. In: *International Conference on Computer Vision*. Kerkyra, Greece: IEEE Computer Society, 1999. v. 1, p. 496–501.

Promotion and Display Technology. *Minoru 3D Webcam*. 2015. Disponível em: <<http://www.minoru3d.com/>>.

RATEKE, T. et al. Segmentação de regiões de interesse em mapas de disparidades. *Computer on the Beach*, Florianópolis, SC, Brazil, v. 1, n. 1, p. 288–297, 2013.

RAWLINGS, J. B.; EKERDT, J. G. *GNU Octave*. 1988. Disponível em: <<https://www.gnu.org/software/octave/>>.

REMONDINO, F.; FRASER, C. Digital camera calibration methods: Considerations and comparisons. *Image Engineering and Vision Metrology*, Dresden, Germany, v. 36, n. 5, p. 266–272, 2006.

- ROMA, N.; VICTOR, J. S.; TOME, J. A comparative analysis of cross-correlation matching algorithms using a pyramidal resolution approach. In: CHRISTENSEN, H. I.; PHILLIPS, P. J. (Ed.). *Empirical Evaluation Methods in Computer Vision*. Stockholm, Sweden: World Scientific Press, 2002. v. 50, p. 117–142.
- ROSTEN, E.; DRUMMOND, T. Fusing points and lines for high performance tracking. In: *International Conference on Computer Vision*. Beijing, China: IEEE Computer Society, 2005. v. 2, p. 1508–1515.
- ROSTEN, E.; DRUMMOND, T. Machine learning for high-speed corner detection. In: LEONARDIS, A.; BISCHOF, H.; PINZ, A. (Ed.). *Computer Vision – ECCV 2006*. Heidelberg, Germany: Springer International Publishing, 2006, (Lecture Notes in Computer Science, v. 3951). p. 430–443.
- ROUSSEEUW, P. J. Least median of squares regression. *Journal of the American Statistical Association*, Delft, The Netherlands, v. 79, n. 388, p. 871–880, dez. 1984.
- ROY, S.; MEUNIER, J.; COX, I. J. Cylindrical rectification to minimize epipolar distortion. In: *International Conference on Computer Vision and Pattern Recognition*. San Juan, Puerto Rico: IEEE Computer Society, 1997. p. 393–399.
- SAMPSON, P. D. Fitting conic sections to “very scattered” data: An iterative refinement of the bookstein algorithm. *International Journal of Computer Graphics and Image Processing*, Academic Press, Chicago, Illinois, USA, p. 97–108, jan. 1982.
- SCHARSTEIN, D. et al. High-resolution stereo datasets with subpixel-accurate ground truth. In: *German Conference on Pattern Recognition*. Munster, Germany: IEEE Computer Society, 2014. v. 1, p. 31–42.
- SCHARSTEIN, D.; SZELISKI, R. A taxonomy and evaluation of dense two-framed stereo correspondence algorithms. *International Journal of Computer Vision*, Springer International Publishing, Amsterdam, Netherlands, v. 47, p. 7–42, abr. 2002.
- SHI, J.; TOMASI, C. Good features to track. In: *International Conference on Computer Vision and Pattern Recognition*. Seattle, WA, USA: IEEE Computer Society, 1994. v. 1, p. 593–600.
- SISCOOTTO, R. A. et al. Estereoscopia. In: KIRNER, C.; TORI, R. (Ed.). *Realidade Virtual: Conceitos e Tendências – Livro do Pré-Simpósio SVR 2004*. São Paulo, SP, Brasil: Mania de Livro, 2004. cap. 11, p. 179–201.
- SONZA, A. P.; SANTAROSA, L. M. C. Ambientes digitais virtuais: Acessibilidade aos deficientes visuais. *Centro Interdisciplinar de Novas Tecnologias na Educação*, Porto Alegre, RS, Brasil, v. 1, n. 1, fev. 2003.
- TARAGLIO, S.; CHIESA, S. Evolutionary approach to epipolar geometry. In: BHATTI, A. (Ed.). *Advances in Theory and Applications of Stereo Vision*. Rijeka, Croatia: InTech, 2011. cap. 1, p. 1–16.
- The MathWorks, Inc. *Stereo Vision - MATLAB & Simulink*. 2015. Disponível em: <<http://www.mathworks.com/help/vision/stereo-vision.html>>.
- The OpenCV Team. *OpenCV Library*. 2017. Disponível em: <<https://opencv.org/>>.

- TORR, P. H. S.; MURRAY, D. W. The development and comparison of robust methods for estimating the fundamental matrix. *International Conference on Advanced Robotics*, Oxford, England, UK, v. 24, n. 3, p. 271–300, 1997.
- TOURBABIN, V.; RAFAELY, B. Theoretical framework for the design of microphone arrays for robot audition. In: *International Conference on Acoustics, Speech and Signal Processing*. Vancouver, BC, Canada: IEEE Computer Society, 2013. p. 4290–4294.
- TSAI, R. Y. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. *Journal of Robotics and Automation*, IEEE Computer Society, v. 3, n. 4, p. 323–344, ago. 1987.
- WENG, J.; COHEN, P.; HERNIOU, M. Camera calibration with distortion models and accuracy evaluation. *Transactions Pattern Analysis and Machine Intelligence*, IEEE Computer Society, Los Alamitos, CA, USA, v. 14, n. 10, p. 965–980, out. 1992.
- WERNER, M.; STABERNACK, B.; RIECHERT, C. Hardware implementation of a full hd real-time disparity estimation algorithm. *Transactions on Consumer Electronics*, IEEE Computer Society, Reading, Berkshire, U.K., v. 60, n. 1, p. 66–73, fev. 2014.
- WERSING, H. et al. Online learning of objects in a biologically motivated visual architecture. *International Journal of Neural Systems*, World Scientific Publishing Company, v. 17, n. 5, p. 219–230, ago. 2007.
- World Health Organization. *Change the Definition of Blindness*. 2011. Disponível em: <<http://www.who.int/blindness/Change%20the%20Definition%20of%20Blindness.pdf>>.
- World Health Organization. *10 Facts about Blindness and Visual Impairment*. 2014. Disponível em: <<http://www.who.int/features/factfiles/blindness/en/>>.
- ZBONTAR, J.; LECUN, Y. Computing the stereo matching cost with a convolutional neural network. In: *International Conference on Computer Vision and Pattern Recognition*. Boston, MA, USA: IEEE Computer Society, 2015.
- ZHANG, Z. A flexible new technique for camera calibration. *Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, Los Alamitos, CA, USA, v. 22, n. 11, p. 1330–1334, nov. 2000.
- ZHANG, Z. et al. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, Elsevier Science Publishers Ltd., Essex, England, UK, v. 78, p. 87–119, out. 1995.
- ZHANG, Z.; HOU, C.; YANG, J. A stereo matching algorithm based on genetic algorithm with propagation stratagem. In: *International Workshop on Intelligent Systems and Applications*. Wuhan, China: IEEE Computer Society, 2009. p. 1–4.
- ZUREIKI, A.; DEVY, M.; CHATILA, R. Stereo matching using reduced-graph cuts. In: *International Conference on Image Processing*. San Antonio, TX, USA: IEEE Computer Society, 2007. v. 1, p. 237–240.

GLOSSÁRIO

BRISK – *Binary Robust Invariant Scalable Keypoints*

FAST – *Features from Accelerated Segment Test*

LMedS – *Least Median of Squares*

LTS – *Least Trimmed of Squares*

MSAC – *M-estimator Sample Consensus*

MSER – *Maximally Stable Extremal Regions*

Min8Val – *Minimum Eigen Value*

RANSAC – *Random Sample Consensus*

SIFT – *Scale-Invariant Feature Transform*

SURF – *Speeded-Up Robust Features*

Apêndice A

ALGORITMO SIMPLIFICADO DA GERAÇÃO DO MAPA DE DISPARIDADES (SEM CALIBRAÇÃO)

Nesta Apêndice é exibido o algoritmo simplificado do método completo, descrito anteriormente no Capítulo 4, desenvolvido com o objetivo de se obter um mapa de disparidades para uma cena, obtida por um sistema de visão estéreo, de uma cena real.

Deve-se observar o fato de que a produção de um mapa de disparidades depende de alguns fatores (também incluídos neste algoritmo) como o número de correspondências inicialmente obtidas a partir dos resultados obtidos com o algoritmo de detecção de cantos, e a ausência de linhas epipolares produzidas pela matriz fundamental. Mais detalhes de tais condições do método foram discutidos no Capítulo 5.

ALGORITMO: METODOLOGIA DA GERAÇÃO DE UM MAPA DE DISPARIDADES

Entrada: Imagens I_e (esquerda) e I_d (direita), obtidas pelo Sistema de Visão Estéreo

Saída: Mapa de Disparidades D_{map} (se possível)

1 **início**

```

2   // Pré-processamento (conversão para cinza)
3    $C_e \leftarrow \text{rgb2gray}(I_e)$ ;
4    $C_d \leftarrow \text{rgb2gray}(I_d)$ ;
5
6   // Detecção de cantos com SURF (apresenta melhores resultados)
7    $\text{edges}_e \leftarrow \text{SURF}(C_e)$ ;
8    $\text{edges}_d \leftarrow \text{SURF}(C_d)$ ;
9
10  // Cálculo das correspondências iniciais (usando SAD)
11   $\text{matches}_{e,d} \leftarrow \text{SAD}(\text{edges}_e, \text{edges}_d)$ ;
12
13  // Estimação da matriz fundamental F (LMedS ou MSAC)
14  se  $|\text{matches}_{e,d}| \geq 8$  então
15      se  $|\text{matches}_{e,d}| \geq 16$  então
16           $F \leftarrow \text{LMedS}(\text{matches}_{e,d})$ ;           // Mais de 16 pares, usa LMedS
17      senão
18           $F \leftarrow \text{MSAC}(\text{matches}_{e,d})$ ;           // Entre 8 e 16 pares, usa MSAC
19      fim
20
21  // Separa as correspondências verdadeiras
22   $\text{matches}_{e,d} \leftarrow \text{inliers}(\text{matches}_{e,d}, F)$ ;
23
24  // Se não houverem linhas epipolares em F
25  se  $\text{IsEpipolarLinesIn}(F) \neq \text{true}$  então
26      // Etapa de retificação
27       $(T_e, T_d) \leftarrow \text{rectify}(F, \text{matches}_{e,d})$ ;
28       $C_e \leftarrow \text{warp}(C_e, T_e)$ ;
29       $C_d \leftarrow \text{warp}(C_d, T_d)$ ;
30
31      // Cálculo do mapa de disparidades
32       $D_{map} \leftarrow \text{disparity}(C_e, C_d)$ ;
33
34      // Correção das distorções
35       $T_{average} \leftarrow (T_e + T_d)/2$ ;
36       $T_{average} \leftarrow T_{average}^{-1}$ ;
37       $D_{map} \leftarrow \text{warp}(D_{map}, T_{average})$ ;
38       $D_{map} \leftarrow \text{crop}(D_{map})$ ;
39
40      retorna  $D_{map}$  ;
41
42  senão
43      retorna erro(F);           // A matriz fundamental produziu epipolos
44  fim
45
46  senão
47      retorna erro( $\text{matches}_{e,d}$ );           // Poucas correspondências detectados
48  fim
49
50 fim
```

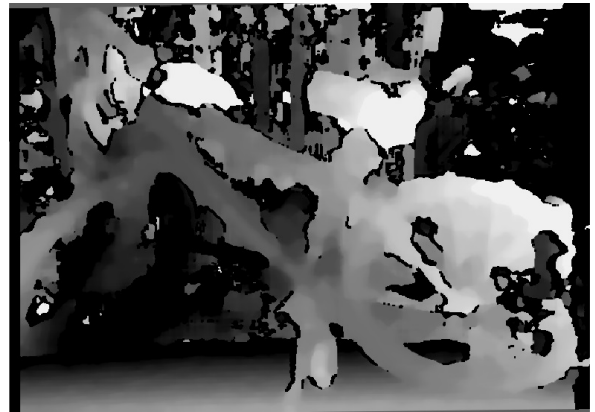
Apêndice B

MAPAS DE DISPARIDADES OBTIDOS NO CÁLCULO DE TEMPO DE PROCESSAMENTO

Os mapas de disparidade obtidos a partir do método desenvolvido neste trabalho, e utilizados na análise de tempo de processamento explicada anteriormente no Capítulo 5, podem ser visualizados nas Figuras B.1 a B.10. Todas as imagens originais pertencem ao banco de imagens estéreo de Middlebury (SCHARSTEIN; SZELISKI, 2002; SCHARSTEIN et al., 2014).



(a) Imagem Original (Esquerda)

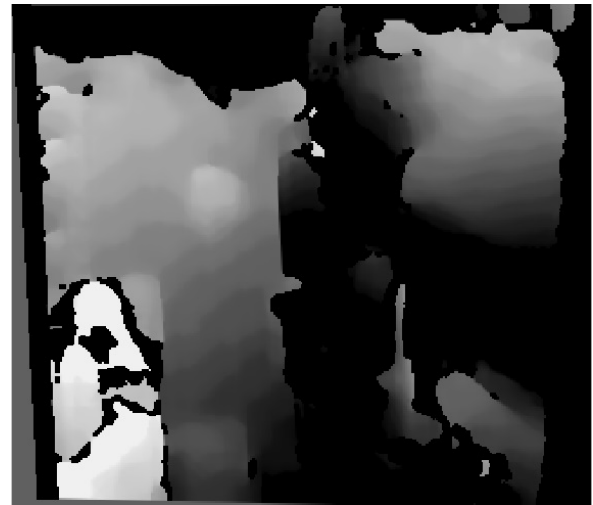


(b) Mapa de Disparidades

Figura B.1: Mapa de Disparidades Obtido: Bicycle2



(a) Imagem Original (Esquerda)

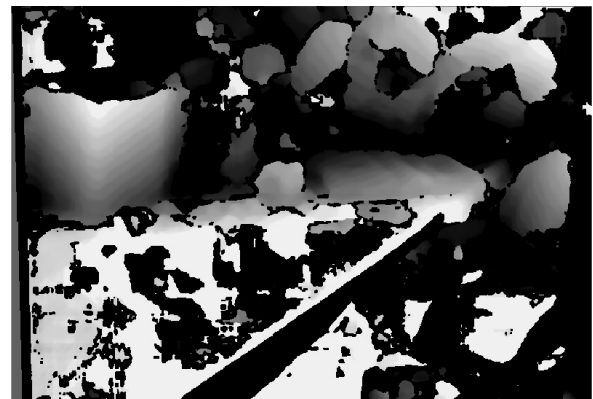


(b) Mapa de Disparidades

Figura B.2: Mapa de Disparidades Obtido: Computer



(a) Imagem Original (Esquerda)

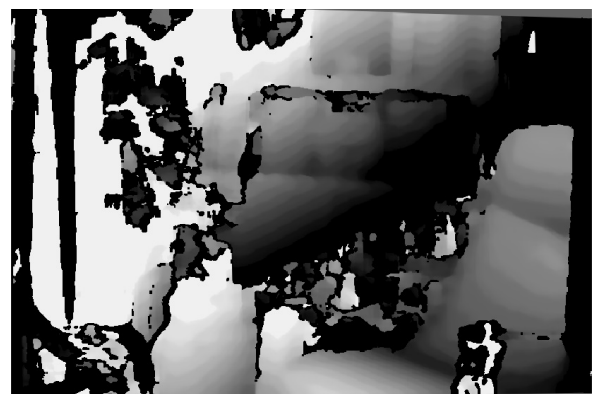


(b) Mapa de Disparidades

Figura B.3: Mapa de Disparidades Obtido: Crusade



(a) Imagem Original (Esquerda)



(b) Mapa de Disparidades

Figura B.4: Mapa de Disparidades Obtido: Livingroom



(a) Imagem Original (Esquerda)

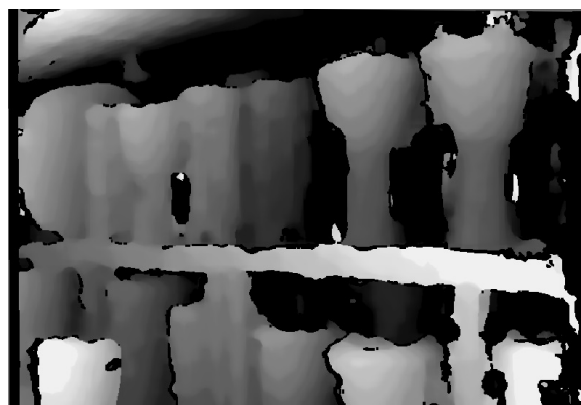


(b) Mapa de Disparidades

Figura B.5: Mapa de Disparidades Obtido: Plants



(a) Imagem Original (Esquerda)

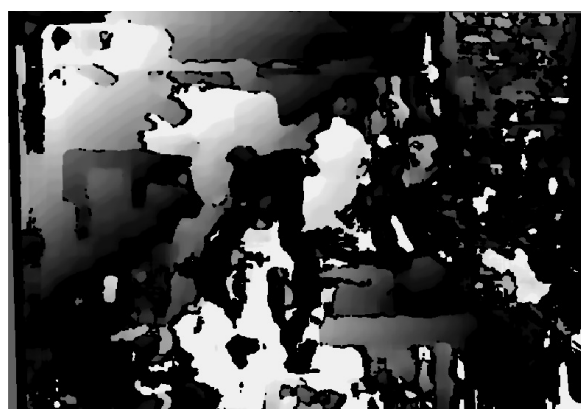


(b) Mapa de Disparidades

Figura B.6: Mapa de Disparidades Obtido: Djembe



(a) Imagem Original (Esquerda)

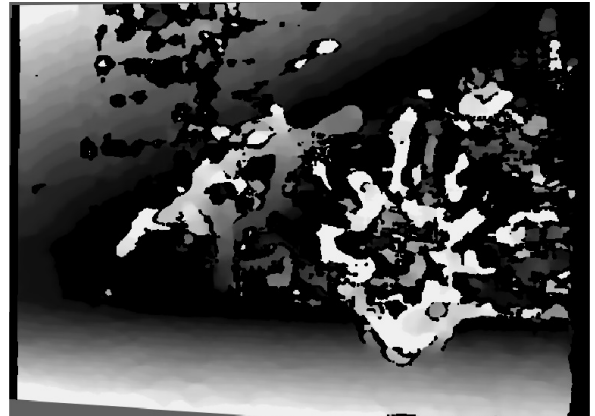


(b) Mapa de Disparidades

Figura B.7: Mapa de Disparidades Obtido: Newkuba

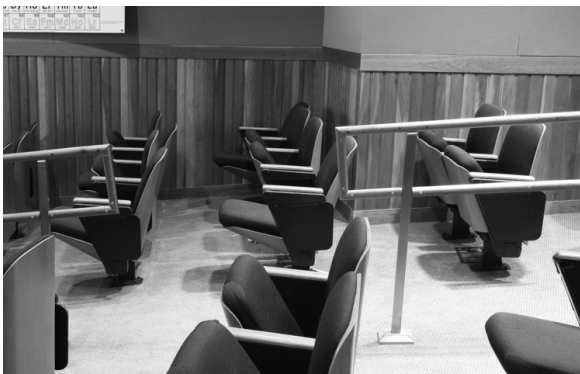


(a) Imagem Original (Esquerda)

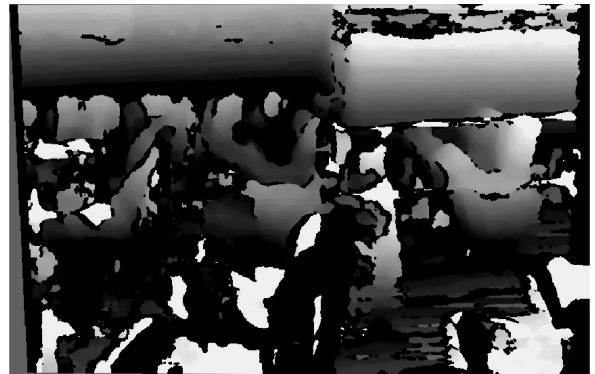


(b) Mapa de Disparidades

Figura B.8: Mapa de Disparidades Obtido: Australia



(a) Imagem Original (Esquerda)



(b) Mapa de Disparidades

Figura B.9: Mapa de Disparidades Obtido: Classroom2



(a) Imagem Original (Esquerda)



(b) Mapa de Disparidades

Figura B.10: Mapa de Disparidades Obtido: Hoops

Apêndice C

COMPARAÇÃO VISUAL: MÉTODO DESENVOLVIDO E O DE ZHANG

Os mapas de disparidades obtidos tanto a partir do método de Zhang (2000) quanto a partir do método desenvolvido neste trabalho, como descrito anteriormente no Capítulo 5, podem ser visualizados nas Figuras C.1 a C.10. Todas as imagens foram obtidas a partir de ambientes reais.

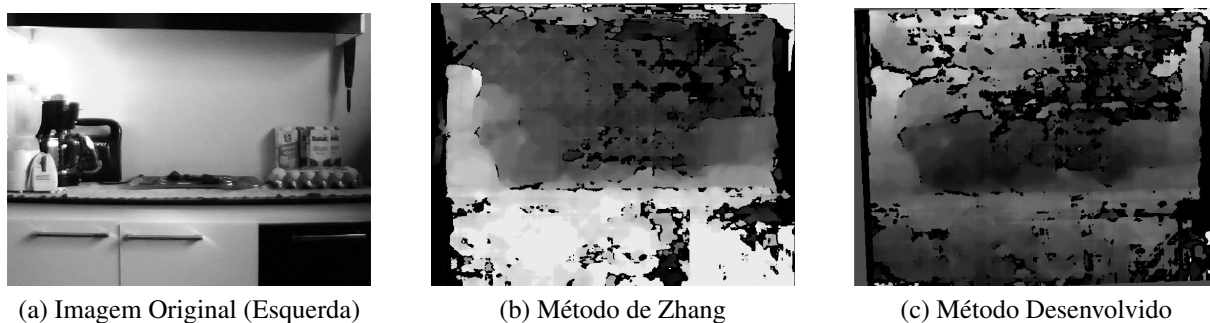


Figura C.1: Mapas de Disparidades Obtidos: Cena 01

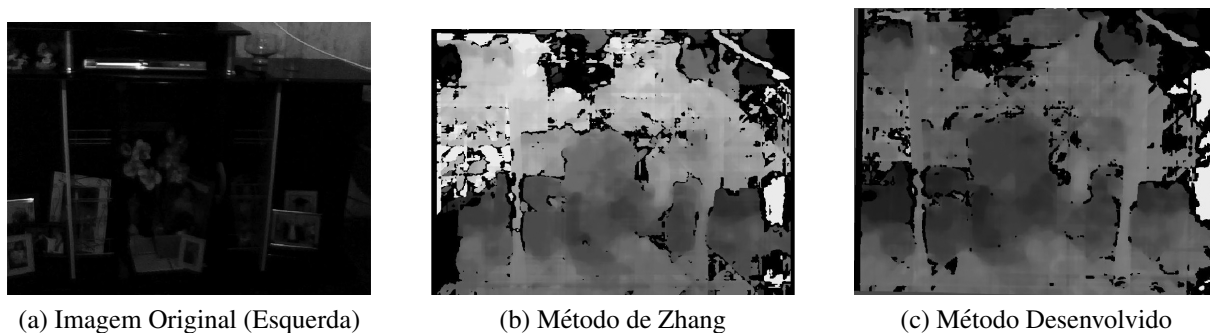


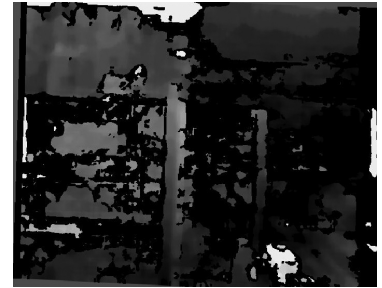
Figura C.2: Mapas de Disparidades Obtidos: Cena 02



(a) Imagem Original (Esquerda)



(b) Método de Zhang

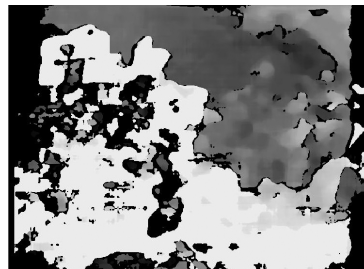


(c) Método Desenvolvido

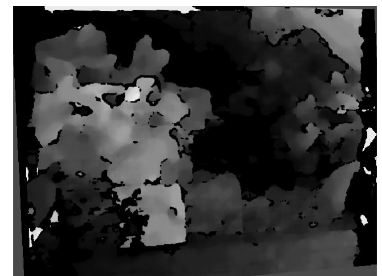
Figura C.3: Mapas de Disparidades Obtidos: Cena 03



(a) Imagem Original (Esquerda)



(b) Método de Zhang



(c) Método Desenvolvido

Figura C.4: Mapas de Disparidades Obtidos: Cena 04



(a) Imagem Original (Esquerda)



(b) Método de Zhang

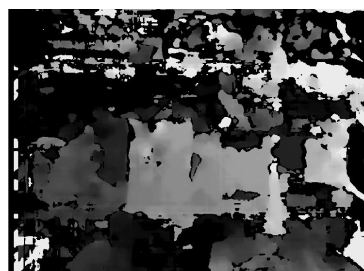


(c) Método Desenvolvido

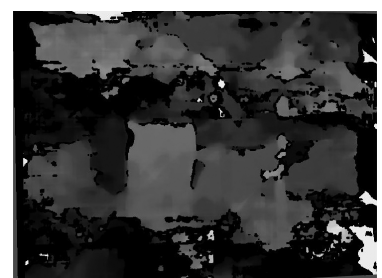
Figura C.5: Mapas de Disparidades Obtidos: Cena 05



(a) Imagem Original (Esquerda)



(b) Método de Zhang



(c) Método Desenvolvido

Figura C.6: Mapas de Disparidades Obtidos: Cena 06



(a) Imagem Original (Esquerda)



(b) Método de Zhang



(c) Método Desenvolvido

Figura C.7: Mapas de Disparidades Obtidos: Cena 07



(a) Imagem Original (Esquerda)



(b) Método de Zhang



(c) Método Desenvolvido

Figura C.8: Mapas de Disparidades Obtidos: Cena 08



(a) Imagem Original (Esquerda)



(b) Método de Zhang



(c) Método Desenvolvido

Figura C.9: Mapas de Disparidades Obtidos: Cena 09



(a) Imagem Original (Esquerda)



(b) Método de Zhang



(c) Método Desenvolvido

Figura C.10: Mapas de Disparidades Obtidos: Cena 10