

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

Cristiano Flores dos Santos

**UM FRAMEWORK PARA AVALIAÇÃO DE
MAPEAMENTO TRIDIMENSIONAL
UTILIZANDO TÉCNICAS DE
ESTEREOSCOPIA E ODOMETRIA VISUAL**

Santa Maria, RS

2016

Cristiano Flores dos Santos

**UM FRAMEWORK PARA AVALIAÇÃO DE MAPEAMENTO TRIDIMENSIONAL
UTILIZANDO TÉCNICAS DE ESTEREOSCOPIA E ODOMETRIA VISUAL**

Dissertação apresentada ao Curso de Mestrado em Computação do Programa de Pós-Graduação em Informática (PPGI), da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestre em Ciência da Computação**

Orientadora: Prof^ª. Dra. Simone Regina Ceolin

Santa Maria, RS
2016

Ficha catalográfica elaborada através do Programa de Geração Automática da Biblioteca Central da UFSM, com os dados fornecidos pelo(a) autor(a).

Santos, Cristiano Flores dos
UM FRAMEWORK PARA AVALIAÇÃO DE MAPEAMENTO
TRIDIMENSIONAL UTILIZANDO TÉCNICAS DE ESTEREOSCOPIA E
ODOMETRIA VISUAL / Cristiano Flores dos Santos.-2016.
81 p. ; 30cm

Orientador: Simone Regina Ceolin
Dissertação (mestrado) - Universidade Federal de Santa
Maria, Centro de Tecnologia, Programa de Pós-Graduação em
Informática, RS, 2016

1. Mapeamento 3D 2. Estereoscopia 3. Odometria
Visual I. Ceolin, Simone Regina II. Título.

Cristiano Flores dos Santos

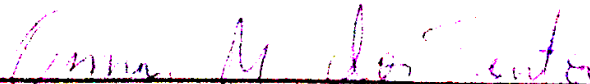
**UM FRAMEWORK PARA AVALIAÇÃO DE MAPEAMENTO
TRIDIMENSIONAL UTILIZANDO TÉCNICAS DE ESTEREOCOPIA E
ODOMETRIA VISUAL**

Dissertação apresentada ao Curso de Pós-Graduação em Ciência da Computação, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do título de **Mestre em Ciência da Computação**.

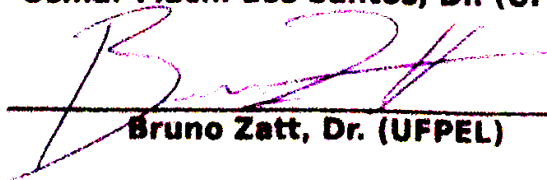
Aprovado em 30 de março de 2016:



Simone Regina Ceolin, Dra. (UFSM)
(Presidente/Orientadora)



Osmar Machi dos Santos, Dr. (UFSM)



Bruno Zatt, Dr. (UFPEL)

Santa Maria, RS
2016

AGRADECIMENTOS

Ao final de cada etapa é importante refletirmos sobre tudo o que passou e lembramos de todas as pessoas que tornaram possível alcançar os objetivos traçados.

Quero agradecer primeiramente a minha orientadora, por ter acreditado em mim desde a seleção para o programa de pós-graduação até os momentos mais difíceis. Sempre me incentivado, exigente mas também muito solidária. Obrigado professora Simone Regina Ceolin.

Agradecimento especial a Fapergs, órgão que financiou meus estudos durante o período do mestrado, sem este incentivo a dedicação exclusiva não seria possível. Além disto, agradeço aos professores e demais pessoas que integram o programa de pós-graduação em Ciência da Computação da Universidade Federal de Santa Maria.

Mesmo com palavras simples, sem muito estudo e noção do que eu tinha que enfrentar, eles que me deram a vida, muito amor e dedicação, onde sempre encontrei palavras de incentivo ou até mesmo de consolo. Obrigado minha mãe Vera Lúcia Flores dos Santos e ao simples mas um grande homem, meu pai Aldori Silva Dutra dos Santos.

Agradeço ao meu irmão Luciano Flores dos Santos que mesmo na impossibilidade de se expressar com palavras, os seus gestos e sorriso bastaram para me motivar.

Não poderia deixar de fora as pessoas que considero muito importantes, os meus sogros Rogério Unfer e Verônica Faller. Mesmo nos momentos os quais me ausentei sempre me mandavam muita força, além de bolo, cuca e por aí vai.

Até mesmo minha cunhada que embora com alguns pequenos conflitos, sempre estava presente pra receitar alguma forma de enfrentar o cansaço, ou mesmo para me fazer sorrir.

Aos meu colegas e amigos de GMicro, Thiago Trugillo, Rafaelo Pinheiro, Ermilo Lima, Tiago Mantay, Vinícius Barbosa e Juliano Andrade. Obrigado pela amizade, discussões técnicas, momentos de cultura inútil e gargalhadas.

Agradeço ainda a vó Nercilda, aos padrinhos João e Rosane Rossini, Amauri e Rosângela. A todos os primos e demais amigos por serem compreensivos durante todo o período que me fiz ausente de festas familiares e demais momentos. Meu muito obrigado a todos.

E por último, mas minha base para tudo, a minha noiva Natale Aline Unfer. Sempre dedicada, compreensiva, carinhosa e companheira. A pessoa que me incentivou a ingressar no mestrado e sempre esteve em todos os momentos servindo como base para meu amadurecimento técnico, pessoal e me mostrando uma melhor forma de ver as coisas. Obrigado meu amor.

“Seja fiel no cumprimento de todos os seus deveres. Execute com capricho e amor todas as tarefas que recebe, embora pareçam insignificantes. Qualquer coisa que esteja fazendo, por menor que seja, é um passo à frente em seu progresso. Realize suas tarefas todas, como se delas dependesse - como de fato depende - todo o seu futuro.”

— C. TORRES

RESUMO

Dissertação de Mestrado
Programa de Pós-Graduação em Informática
Universidade Federal de Santa Maria

UM FRAMEWORK PARA AVALIAÇÃO DE MAPEAMENTO TRIDIMENSIONAL UTILIZANDO TÉCNICAS DE ESTEREOSCOPIA E ODOMETRIA VISUAL

AUTOR: CRISTIANO FLORES DOS SANTOS
ORIENTADORA: SIMONE REGINA CEOLIN

O mapeamento tridimensional de ambientes tem sido intensivamente estudado na última década. Entre os benefícios deste tema de pesquisa é possível destacar a adição de autonomia a automóveis ou mesmo drones. A representação tridimensional também permite a visualização de um dado cenário de modo iterativo e com maior riqueza de detalhes. No entanto, até o momento da elaboração deste trabalho não foi encontrado um *framework* que apresente em detalhes a implementação de algoritmos para realização do mapeamento 3D de ambientes externos que se aproximasse de um processamento em tempo real. Diante disto, neste trabalho foi desenvolvido um *framework* com as principais etapas de reconstrução tridimensional. Para tanto, a estereoscopia foi escolhida como técnica para a aquisição da informação de profundidade do cenário. Além disto, neste trabalho foram avaliados 4 algoritmos de geração do mapa de profundidade, onde foi possível atingir a taxa de 9 quadros por segundo.

Palavras-chave: Mapeamento 3D. Estereoscopia. Odometria Visual.

ABSTRACT

Master Thesis
Post-Graduate Program in Informatics
Federal University of Santa Maria

THREE-DIMENSIONAL MAPPING OF EXTERNAL ENVIRONMENTS THROUGH STEREOSCOPY AND VISUAL ODOMETRY TECHNIQUES

AUTHOR: CRISTIANO FLORES DOS SANTOS

ADVISOR: SIMONE REGINA CEOLIN

The three-dimensional mapping environments has been intensively studied in the last decade. Among the benefits of this research topic is possible to highlight the addition of autonomy for car or even drones. The three-dimensional representation also allows viewing of a given scenario iteratively and with greater detail. However, until the time of this work was not found one framework to present in detail the implementation of algorithms to perform 3D mapping outdoor approaching a real-time processing. In view of this, in this work we developed a framework with the main stages of three-dimensional reconstruction. Therefore, stereoscopy was chosen as a technique for acquiring the depth information of the scene. In addition, this study evaluated four algorithms depth map generation, where it was possible to achieve the rate of 9 frames per second.

Keywords: 3D mapping. Stereoscopy. Visual Odometry.

LISTA DE FIGURAS

Figura 2.1 – Modelo de câmara <i>pinhole</i> e a geometria envolvida.....	20
Figura 2.2 – Modelo de câmara pinhole modificado, com o ponto Q projetado sobre plano projetivo.	21
Figura 2.3 – Geometria de um sistema estéreo (AHMED, 2006).....	23
Figura 2.4 – Principais parâmetros Geometria Epipolar.	25
Figura 2.5 – Representação da calibração estéreo.	27
Figura 2.6 – Mapa de disparidade em escala de cinza a partir do par de imagens.....	29
Figura 2.7 – Representação de correspondência na mesma linha e a procura pelo mesmo ponto entre o par iniciando na disparidade mínima e movendo para a esquerda até um intervalo de disparidade máxima definido.....	30
Figura 2.8 – Os planos horizontais representam a influência entre os parâmetros de disparidade mínima e máxima para a distância de objetos no cenário e na vertical que o algoritmo irá considerar de profundidade.....	31
Figura 2.9 – Poses relativas $T_{k,k-1}$, de posições adjacentes das câmeras que são calculadas através de características visuais e concatenadas para obter as poses absolutas c_k tendo como coordenada do frame inicial o $k = 0$	36
Figura 2.10 – Diagrama em blocos dos principais componente da Odometria visual.	37
Figura 2.11 – Visão geral do sistema Libviso.	40
Figura 2.12 – Definição do sistema de coordenadas.	41
Figura 3.1 – A nuvem de pontos do mapeamento 3D no topo. Na parte inferior o mapa de disparidades esparso.	43
Figura 3.2 – Mapeamento 3D de ambientes internos dividido em multi-sessões.	44
Figura 3.3 – Visão geral do sistema RGB-D SLAM.....	45
Figura 3.4 – Visualização do Mapeamento 3D através de OctoMaps.	46
Figura 4.1 – Visão geral do sistema desenvolvido organizado em blocos.	48
Figura 4.2 – Diagrama de classes do sistema.	50
Figura 4.3 – Diagrama de sequência.	52
Figura 4.4 – Exemplo da biblioteca PCL.	54
Figura 4.5 – Dezesesseis caminhos para o refinamento da correspondência utilizados pelo SGM.	56
Figura 4.6 – Possíveis regiões de picos e problemas de fundo sem textura.....	56
Figura 4.7 – Exemplo de aplicação de SAD em um problema de correspondência estéreo.	57
Figura 4.8 – Algoritmo BM proposto por Konolige (1998).	58
Figura 4.9 – Comparativo entre BM (CPU) e BM (GPU).....	59
Figura 4.10 – Comparativo entre métodos locais e ELAS. No topo e a direita da janela, um exemplo de processado de método local com uma janela de 5×5 , na parte inferior a esquerda, o método local com janela de 25×25 . Na parte inferior a direita o método ELAS.	60
Figura 4.11 – Telas do sistema desenvolvido em execução.....	61
Figura 5.1 – Plataforma de aquisição de dados é equipada com quatro câmeras (sistema estéreo em escala de cinza e sistema para captura em RGB), um scanner laser 3D e um sistema que combina GPS/IMU.	63
Figura 5.2 – Avaliação do mapa gerado por ELAS. A imagem do topo apresenta o cenário real, seguida da imagem do mapa de disparidades e a imagem que apresenta o erro.	65

Figura 5.3 – Avaliação do mapa gerado por SGM. A imagem do topo apresenta o cenário real, seguida da imagem do mapa de disparidades e a imagem que apresenta o erro	66
Figura 5.4 – Avaliação do mapa de disparidade gerado pelo BM.....	67
Figura 5.5 – Avaliação do mapa gerado por BM GPU. A imagem do topo apresenta o cenário real, seguida da imagem do mapa de disparidades e a imagem que apresenta o erro	68
Figura 5.6 – Evaluation of Visual Odometry	71
Figura 5.7 – Imagem tridimensionais da sequência 07.....	74
Figura 5.8 – Imagem tridimensionais da sequência 01.....	75

LISTA DE ABREVIATURAS E SIGLAS

FPS	<i>Frames per Second</i>
EKF	<i>Extended Kalman Filter</i>
KF	<i>Kalman Filter</i>
RANSAC	<i>Random Sample Consensus</i>
RMSE	<i>Root Mean Squared Error</i>
SLAM	<i>Simultaneous Localization and Mapping</i>
SURF	<i>Speeded-Up Roboust Features</i>
VO	<i>Visual Odometry</i>
PCL	<i>Point Cloud Library</i>
OPENCV	<i>Open Source Computer Vision</i>
UML	<i>Unified Modeling Language</i>
GPS	<i>Global Positioning System</i>
GPU	<i>Graphics Processing Unit</i>
IMU	<i>Inertial Measurement Unit</i>
ROS	<i>Robot Operation System</i>

LISTA DE SÍMBOLOS

π	Plano da imagem da câmera.
f	Distância focal.
C	Centro de projeção da câmera.
c_x	Ponto principal da imagem no eixo x
c_y	Ponto principal da imagem no eixo y.
Q	Matriz de reprojeção.
q	Vetor de ponto no plano 2D.
F	Matriz Fundamental.
t	Vetor de translação que descreve a posição relativa entre as origens de dois sistemas de coordenadas.
R	Matriz de Rotação 3x3 que descreve a orientação relativa entre as origens de dois sistemas de coordenadas.
e	Epipolo.
X	Coordenada do eixo x do sistema 3D.
x	Coordenada do eixo x do sistema 2D.
Y	Coordenada do eixo y do sistema 3D.
y	Coordenada eixo y do sistema 2D.
Z	Coordenada eixo z do sistema 3D.

SUMÁRIO

1 INTRODUÇÃO	16
1.1 Motivação	17
1.2 Objetivos	18
1.2.1 Objetivos Específicos	18
1.3 Estrutura	18
2 FUNDAMENTOS TEÓRICOS	19
2.1 Estereoscopia	19
2.1.1 Modelo de Câmera	19
2.1.2 Distorções das Lentes	21
2.1.3 Triangulação	22
2.1.4 Geometria Epipolar	24
2.1.5 Etapas do Sistema de Estereoscopia	25
2.1.5.1 Calibração	26
2.1.5.2 Retificação	27
2.1.5.3 Correspondência	28
2.1.5.4 Reprojeção	34
2.2 Odometria Visual	35
2.2.1 Estimativa de Movimento	37
2.2.2 Movimentação de Imagens por características 2D-para-2D	38
2.2.3 Movimentação de Imagens por características 3D-para-3D	38
2.2.4 Movimentação de Imagens por características 3D-para-2D	38
2.3 LibViso	39
3 TRABALHOS RELACIONADOS	42
3.0.1 Estereoscopia	42
3.0.2 Monocular	43
3.0.3 Sensores RGB-D	43
3.0.4 Considerações	46
4 SISTEMA PROPOSTO	48
4.1 Visão Geral	48
4.2 Algoritmo	49
4.3 Ferramentas Utilizadas	53
4.3.1 <i>OpenCV</i>	53
4.3.2 PCL	54
4.4 Semi-Global	55
4.5 Block Match	56
4.6 Block Match (GPU)	58
4.7 ELAS	59
4.8 Visualização do Sistema	60
5 RESULTADOS	62
5.1 Plataforma de Simulação	62
5.2 Base de dados de imagens	62
5.3 Análises e Resultados	63

6 CONSIDERAÇÕES FINAIS	76
6.1 Trabalho Futuros	76
6.1.1 Aperfeiçoamentos	76
6.1.2 Aplicação	77
REFERÊNCIAS	78

1 INTRODUÇÃO

Em visão computacional um dos grandes desafios é a reconstrução tridimensional de ambientes. Esse tema de pesquisa consiste na aplicabilidade de uma série de conceitos e abordagens as quais é possível citar o reconhecimento de padrões, extração de características, tratamento de imagens via filtros e a utilização de vários sensores de captura de imagens (LUCAS; KANADE et al., 1981) e (SZELISKI, 2010).

Entre as possibilidades computacionais para aquisição de informação de tridimensionalidade da cena, a estereoscopia apresenta resultados de destaque na literatura (ENGEL; STUCKLER; CREMERS, 2015). Essa técnica consiste na utilização de um par de câmeras homólogas a forma de visão humana. Diante disso, o efeito da aquisição de duas imagens com deslocamento horizontal permite a noção de profundidade. A estereoscopia se destaca frente a outras formas de aquisição de informação de profundidade como laser e sensores ultra-sônicos por viabilizar a aquisição da informação de profundidade da cena juntamente com a imagem do ambiente (BELLAVIA et al., 2013). Além disso, é aplicável em ambientes de grandes distância bem como pequenas, bastando o ajuste correto da disposição dos sensores (LEMAIRE et al., 2007).

Para realizar o mapeamento 3D de ambientes, além da aquisição da informação de profundidade da cena é necessário calcular o deslocamento do sistema diante do cenário. A odometria visual foi o termo concebido por (NISTER; NARODITSKY; BERGEN, 2004) para conceituar a técnica de estimar movimento através de imagens. Esse termo nasceu da semelhança da odometria de rodas (*Wheel odometry*) que incrementa estimativas de movimento pelo número de voltas das rodas de um veículo em um determinado tempo (NISTER; NARODITSKY; BERGEN, 2004). Assim a odometria visual permite mapear ambientes de forma incremental por meio de estimativas de movimento de sucessivos quadros.

A aplicabilidade do mapeamento de ambientes pode se estender à navegação autônoma de robôs e veículos, sistemas de auxílio a mobilidade de deficientes visuais e reconstrução tridimensional (VARGAS et al., 2012), (PRADEEP; MEDIONI; WEILAND, 2010) e (GEIGER; ZIEGLER; STILLER, 2011).

Embora nos últimos dez anos uma ampla quantidade de trabalhos relacionados a mapeamento de ambientes tenha sido publicada, apenas uma pequena parte dos trabalhos citam o desempenho próximo do tempo real para a tarefa de reconstrução tridimensional (GEIGER; ZI-

EGLER; STILLER, 2011), (STEINBRUCKER; STURM; CREMERS, 2011), (ENDRES et al., 2014) e (ENGEL; STUCKLER; CREMERS, 2015).

Neste contexto, este trabalho apresenta conceitos em torno do mapeamento 3D tendo como único sensor de captura de informação do ambiente, um sistema de estereoscopia. Além disso, é desenvolvido um *framework* de mapeamento tridimensional que se aproxima do tempo real com a utilização do algoritmo de odometria visual denominado Libviso (Geiger 2011). Como alternativa para aumento da taxa de quadros, é feita uma análise de quatro algoritmos em destaque na literatura no que se refere a aquisição de informação de profundidade da cena.

Finalmente é descrita a arquitetura do *framework* desenvolvido neste trabalho, onde são obtidos resultados de até 9 quadros por segundo (fps) em comparação a abordagem proposta por (GEIGER; ZIEGLER; STILLER, 2011) usada neste trabalho como referência e que atinge 4 fps.

1.1 Motivação

As técnicas para mapear ou identificar deslocamentos através de imagens foram inicialmente propostas pela Nasa (LACROIX et al., 1999) e (OLSON et al., 2000) quando propôs a inserção de sensores de captura de imagens em seus robôs para mapear e se locomover em ambientes desconhecidos e de difícil acesso, onde a odometria por rodas era inviável. As tecnologias desenvolvidas inicialmente eram baseados no "para e anda".

Nos últimos anos tem havido esforços de pesquisadores para viabilizar a autonomia de automóveis assim como dispositivos robóticos (GEIGER; LENZ; URTASUN, 2012) e (PI et al., 2014). Porém, no atual contexto é necessário diminuir o custo computacional para que a tarefa de mapeamento seja realizada em tempo real. Além da autonomia, essa abordagem pode oferecer a aquisição de informação tridimensional de cenários inacessíveis ou até mesmo que ofereçam algum perigo ao ser humano. Esse tipo abordagem tem como potencial a aplicação em missões militares ou até mesmo para mapeamento de áreas com fins arquitetônicos.

Nesse sentido, tendo em vista que até o momento da conclusão deste trabalho não foi encontrado um *framework* para o mapeamento tridimensional em tempo real através de dispositivos de estereoscopia. Buscou-se através desse, desenvolver um *software* que pudesse cobrir esta necessidade.

A motivação deste trabalho se concentra no desenvolvimento e disponibilização de um *framework* que encapsule as funcionalidades básicas para realização do mapeamento 3D de am-

bientes externos que se aproxime do tempo real. Dessa forma, viabiliza estudos futuros para o problema, proporcionando avanços em pesquisas de forma mais ágil, dado que a construção de um software de mapeamento despreza tempo. Estudos encontrados na literatura sobre mapeamento 3D, sugerem aplicação em dispositivos como automóveis, drones e submarinos, possibilitando a autonomia e aquisição de informações tridimensionais do cenário as quais podem estar inseridos.

1.2 Objetivos

Este trabalho tem como objetivo descrever um *framework* de mapeamento 3D de ambientes que mais se aproxime de um desempenho em tempo real, dado que até o momento da elaboração deste trabalho não foi encontrada uma abordagem que apresentasse uma descrição detalhada com foco nesse problema. Ainda buscou-se como objetivo avaliar a taxa de quadros diante da análise de algoritmos de correspondência mais relevantes disponíveis na literatura.

1.2.1 Objetivos Específicos

- Estudar técnicas de odometria visual;
- Analisar algoritmos de correspondência estéreo;
- Descrever um *framework* para mapeamento tridimensional de ambientes externos;
- Avaliar a influência de diferentes resoluções para a reconstrução tridimensional.

1.3 Estrutura

Este trabalho foi organizado em seis Capítulos. O Capítulo 2, intitulado Revisão Bibliográfica, descreve os principais fundamentos teóricos envolvidos no trabalho. Os principais trabalhos relacionados são descritos no Capítulo 3, com o objetivo de realizar um paralelo entre a literatura disponível e o presente trabalho. O Capítulo 4 denominado Solução Proposta visa descrever os detalhes do sistema desenvolvido, assim como ferramentas utilizadas e estrutura diante de alguns diagramas UML. Por fim, são apresentados os resultados obtidos no Capítulo 5 e as considerações finais no Capítulo 6.

2 FUNDAMENTOS TEÓRICOS

Neste capítulo serão apresentados os principais fundamentos teóricos envolvidos no sistema de mapeamento 3D. O objetivo é servir como base conceitual para os próximos capítulos.

Neste contexto os conceitos sobre a estereoscopia e aspectos importantes para aquisição da informação de profundidade da cena será explanada na Seção 2.1, seguida da geometria e teoria dos problemas que cercam a Odometria Visual na Seção 2.2.

2.1 Estereoscopia

A estereoscopia ou visão estéreo consiste em um sistema que torna possível adquirir a informação de profundidade através de duas imagens de uma mesma cena em diferentes ângulos. Este procedimento é análogo ao funcionamento da visão humana (GONZALEZ; WOODS, 2000).

Considerando a utilização de duas câmeras dispostas em paralelo e deslocadas horizontalmente, é possível obter um par de imagens no qual os objetos de uma imagem terão um deslocamento em relação a outra. O deslocamento é chamado disparidade e se apresenta maior para elementos mais próximos do sistema de captura e menor para elementos mais distantes. Nesse sentido, através da diferença das posições de um mesmo ponto entre as imagens é possível aferir a profundidade relativa e conseqüentemente gerar um mapa de disparidades da cena (BRADSKI, 2008).

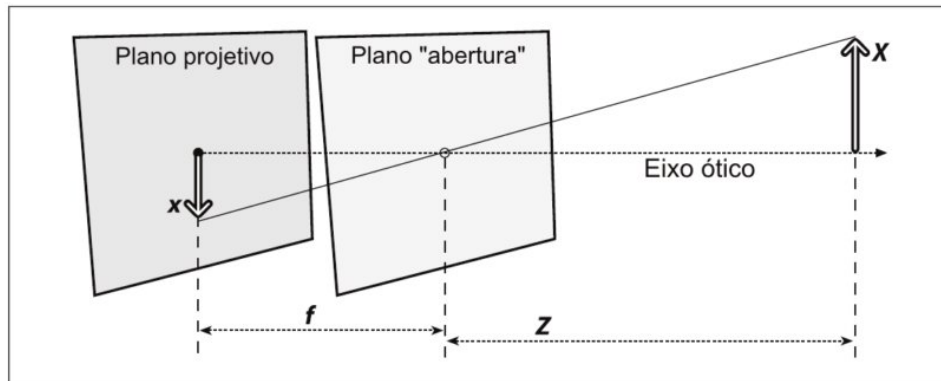
Para compreensão de um sistema de estereoscopia segue a apresentação de um modelo de uma câmera comum na Seção 2.1.1, posteriormente é apresentada a geometria básica de um sistema de visão estéreo na Seção 2.1.3 e Seção 2.1.4 e na última parte as etapas que constituem o sistema na Seção 2.1.5.

2.1.1 Modelo de Câmera

A apresentação do modelo de câmera se faz necessária para o entendimento de conceitos e parâmetros utilizados no sistema de visão estéreo. Neste contexto, o modelo de câmera real *pinhole* ou “buraco de agulha” é ilustrado na Figura 2.1.

A Figura 2.1 ilustra o modelo que possui uma abertura por onde passa a luz e o plano projetivo, onde a luz é projetada formando a imagem. A distância entre a abertura e o plano

Figura 2.1 – Modelo de câmera *pinhole* e a geometria envolvida.



Fonte: (BRADSKI, 2008).

projetivo é denominada distância focal, sendo representada por f . O parâmetro Z refere-se a distância entre a abertura e um dado objeto, X é o comprimento deste objeto e x o comprimento do mesmo no plano projetivo. Com o conhecimento da distância focal f é possível adquirir a informação do comprimento de qualquer ponto no plano projetivo. Através da similaridade entre triângulos $-x/f = X/Z$, chega-se a Fórmula 2.1 (BRADSKI, 2008):

$$-x = f \frac{X}{Z} \quad (2.1)$$

Rearranjando o modelo *pinhole* com o objetivo de torná-lo mais atraente, conforme a Figura 2.2, o objeto é projetado na parte superior do eixo x . O ponto *pinhole* é reinterpretado como centro da projeção. O ponto de intersecção do plano projetivo e o eixo óptico refere-se ao ponto principal.

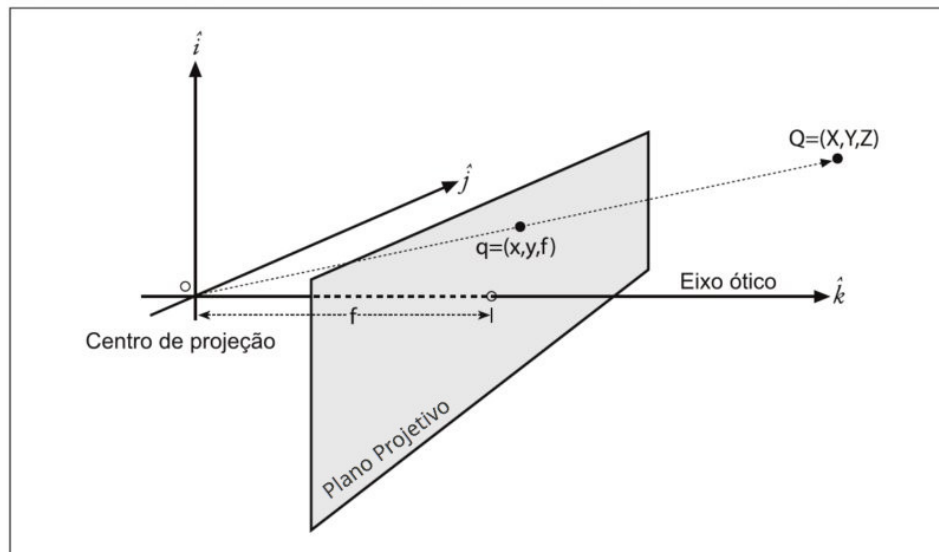
Em um sistema real tem-se ainda parâmetros como c_x e c_y . Esses parâmetros são referentes ao deslocamento entre o ponto principal (centro do plano projetivo) e a abertura da câmera. Os parâmetros c_x e c_y são necessários pois câmeras reais possuem distorções que permitem diferenças entre o alinhamento dos sensores de luz e a abertura.

Como resultado desse modelo, o ponto Q no plano projetivo é adquirido através das Equações 2.2 e 2.3:

$$x = f_x \frac{X}{Z} + c_x \quad (2.2)$$

$$y = f_y \frac{Y}{Z} + c_y \quad (2.3)$$

Figura 2.2 – Modelo de câmera pinhole modificado, com o ponto Q projetado sobre plano projetivo.



Fonte: (BRADSKI, 2008).

Por meio das coordenadas homogêneas, a projeção de pontos podem ser representadas por multiplicação de matrizes como as que seguem:

$$q = MQ, \quad \text{Onde :} \quad q = \begin{bmatrix} x \\ y \\ w \end{bmatrix}, \quad M = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.4)$$

A Equação $q = MQ$ propicia uma maneira simplificada de representar a transformação projetiva sofrida pelo ponto Q , apresentado na Equação 2.16. A matriz M possui os valores intrínsecos da câmera e parâmetros referentes a distorção projetiva provocada pelas lentes, sendo que esses parâmetros são adquiridos do procedimento de calibração descrito com maiores detalhes na Seção 2.1.5.1.

2.1.2 Distorções das Lentes

As lentes das câmeras em geral possuem distorções, principalmente as câmeras de baixo custo como *webcams*. Isto ocorre por questões de fabricação, onde é mais fácil fabricar uma lente esférica do que uma matematicamente parabólica ideal. Também existe um problema mecânico voltado a dificuldade de alinhar as lentes com os sensores de captura.

Os dois principais tipos de distorções de lentes são a distorção radial e distorção tangencial. A distorção radial é provocada pelo formato imperfeito da lente, problema que afeta

os pontos mais próximos das bordas do sensor e tem um comportamento incremental partindo do centro em direção as extremidades. Essa distorção é caracterizada por uma série de *Taylor*, como exposto nas Equações 2.5 e 2.6 (BRADSKI, 2008):

$$x_{\text{corrigido}} = x(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (2.5)$$

$$y_{\text{corrigido}} = y(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (2.6)$$

Onde x e y representam as posições originais dos pontos, $x_{\text{corrigido}}$ e $y_{\text{corrigido}}$ a nova posição dos pontos após a correção e os coeficientes k_n são os termos da série considerados para a correção.

A segunda distorção é a tangencial provocada pelo alinhamento defeituoso da lente em relação ao sensor, representado pela Equações 2.7 e 2.8 (BRADSKI, 2008):

$$x_{\text{corrigido}} = x + [2p_1y + p_2(r^2 + 2x^2)] \quad (2.7)$$

$$y_{\text{corrigido}} = y + [p_1(r^2 + 2y^2) + 2p_2x] \quad (2.8)$$

São no total cinco coeficientes de distorção, k_1 , k_2 , k_3 , representados nas Equações 2.5 e 2.6, p_1 e p_2 presentes nas Equações 2.7 e 2.8. Esses constituem uma matriz de uma coluna e cinco linhas chamado de matriz de distorção ou simplesmente matriz D. Os parâmetros da matriz D com a matriz M citada na Seção 2.1.1 constituem os parâmetros chamados de intrínsecos, que representam informações internas das câmeras.

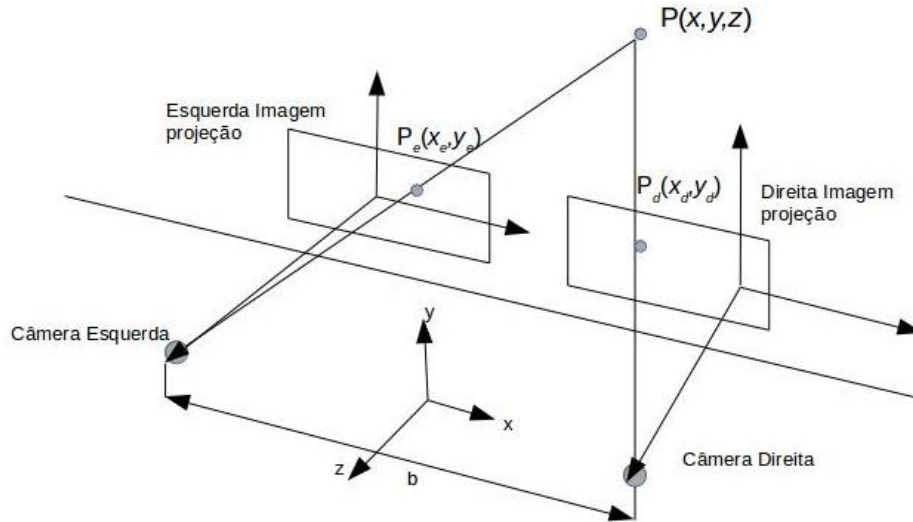
2.1.3 Triangulação

Conforme descrito na Seção 2.1.1 a informação de profundidade é perdida a partir do momento em que é adquirida a imagem através de uma câmera do modelo *pinhole*. Porém, por meio da visão estéreo é possível recuperar a profundidade de um dado ponto no espaço, a partir da distância relativa entre os pontos que o representam no par de imagens. Esta distância é denominada disparidade (STIVANELLO; STEMMER, 2012).

A Figura 2.3 ilustra a geometria envolvida em um sistema de visão estéreo para aquisição da informação de profundidade de um ponto P no mundo. Considerando um sistema perfeitamente calibrado, os eixos ópticos das câmeras são alinhados em paralelo e separado por

uma linha de distância b . Um sistema de coordenadas é vinculado de forma que o eixo x está em paralelo com a linha base e o eixo z está em paralelo com os eixos ópticos, em outras palavras, neste exemplo é desconsiderada os possíveis desalinhamentos que possam existir entre as câmeras.

Figura 2.3 – Geometria de um sistema estéreo (AHMED, 2006).



Fonte: (AHMED, 2006).

Na Figura 2.3 o ponto marcado na “Câmera esquerda” e “Câmera direita” é referente ao ponto P no plano projetivo de ambas as câmeras. Através de uma geometria simples conforme as igualdades apresentadas pelas Equações 2.9, 2.10 e 2.11:

$$\frac{x_e}{f} = \frac{x + b/2}{z} \quad (2.9)$$

$$\frac{x_d}{f} = \frac{x - b/2}{z} \quad (2.10)$$

$$\frac{y_e}{f} = \frac{y_d}{f} = \frac{y}{z} \quad (2.11)$$

Note que:

$$\frac{x_e - x_d}{f} = \frac{b}{z} \quad (2.12)$$

Seguem as Equações 2.9, 2.10 e 2.11, rearranjadas em 2.13, 2.14 e 2.15, para resolver as coordenadas (x, y, z) do ponto P :

$$x = b \frac{(x_e + x_d)/2}{x_e - x_d} \quad (2.13)$$

$$y = b \frac{(y_e + y_d)/2}{x_e - x_d} \quad (2.14)$$

$$z = b \frac{f}{x_e - x_d} \quad (2.15)$$

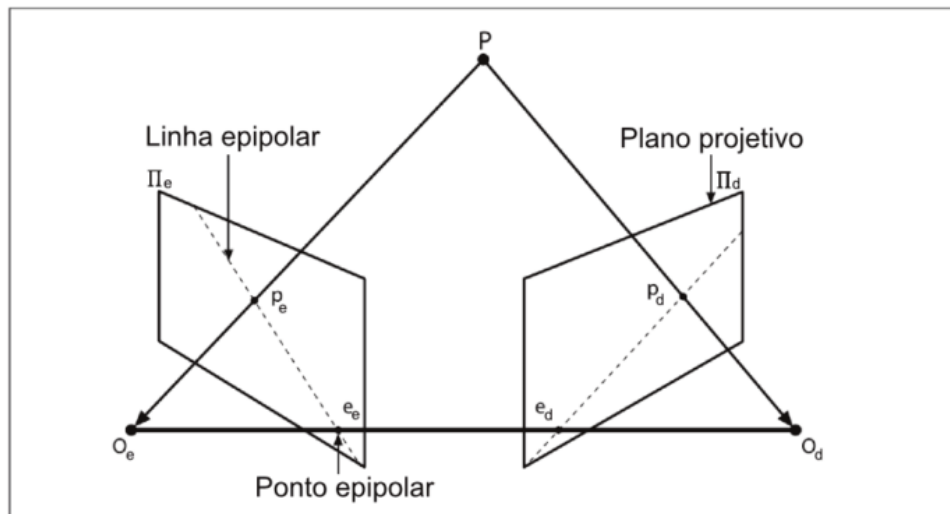
As Equações 2.13, 2.14 e 2.15, mostram que a distância é inversamente proporcional a disparidade e que é diretamente proporcional a linha base. É possível observar que objetos a uma grande distância vão parecer não ter qualquer disparidade. Sendo a disparidade e a linha de base proporcionais, o aumento da linha de base fará possível detectar uma disparidade em objetos que estão mais longe. No entanto, nem sempre é vantajoso aumentar a linha de base porque os objetos que estão mais próximos irão desaparecer da visão de uma ou ambas as câmaras (AHMED, 2006).

Nessa seção foi considerado um sistema ideal, sabe-se que um sistema de estereoscopia pode ser construído com câmeras individualmente colocadas em paralelo, o que propicia desalinhamento de angulo entre as câmeras. Para este e outros tipos de problemas deve ser considerada a etapa de calibração e retificação, descritas respectivamente nas Seções 2.1.5.1 e 2.1.5.2. Além disso, o encontro dos pontos correspondentes entre o par de imagens é um problema desafiador em sistemas estéreos, sendo assim, são apresentadas algumas propostas de solução na Seção 2.1.5.3.

2.1.4 Geometria Epipolar

A geometria básica do sistema de estereoscopia é denominada geometria epipolar. A importância desta geometria esta relacionada a busca por pontos similares entre o par de imagens. A Figura 2.4 ilustra o modelo de geometria epipolar e seus parâmetros.

Figura 2.4 – Principais parâmetros Geometria Epipolar.



Fonte: (BRADSKI, 2008).

Na Figura 2.4, os centros das projeções são representados por O_e e O_d e o par de planos projetivos correspondentes são π_e e π_d , o ponto P é o ponto do mundo físico e tem como projeção os pontos rotulados como p_e e p_d em cada plano projetivo. As intersecções entre a linha que liga os centros de projeção e os planos de projeção são denominados de pontos epipolares e_e e e_d . Levando em consideração o ponto P , tem-se ainda o plano epipolar formado por p_e , p_d , e_e e e_d . A intersecção do plano epipolar com cada plano projetivo resulta em linha epipolares.

Este sistema estabelece a chamada restrição epipolar. Nesse contexto, ao procurar na imagem da direita um ponto correspondente ao ponto p_e tendo este como projeção do ponto P na imagem da direita, este ponto pode ser encontrado na linha epipolar da direita. Essa geometria proporciona através do conhecimento dos valores de e_e , p_e e e_d calcular a linha epipolar da imagem direita e restringir a busca pelo ponto p_d a esta linha. Dessa forma é reduzido o escopo da busca que com a ausência da restrição poderia ser bidimensional (BRADSKI, 2008).

2.1.5 Etapas do Sistema de Estereoscopia

Em um sistema de estereoscopia é necessário tratar questões como distorções presentes nas lentes das câmeras, ajustes para viabilizar o encontro de pontos em comum entre o par de imagens, algoritmo de busca e cálculo do mapa de disparidades e a reprojeção dos pontos para 3D. Assim são descritas quatro etapas que contemplam estas questões (BRADSKI, 2008):

- Calibração: estima parâmetros intrínsecos e extrínsecos que possibilitam a remoção de distorções presentes nas imagens (Seção 2.1.5.1);
- Retificação: consiste em ajustar alinhamento horizontal entre o par de imagens e remover possíveis imperfeições no par de imagens (Seção 2.1.5.2);
- Correspondência: tem como objetivo encontrar os pontos similares entre par de imagens. Esta etapa tem como resultado a geração do mapa de disparidades, sendo a disparidade a diferença das coordenadas x que representam o mesmo ponto no par de imagens: $x^e - x^d$ (Seção 2.1.5.3);
- Reprojeção: reprojeta pontos através do conhecimento dos parâmetros geométricos das câmeras e por triangulação obtendo-se assim o mapa de profundidade (Seção 2.1.5.4).

2.1.5.1 Calibração

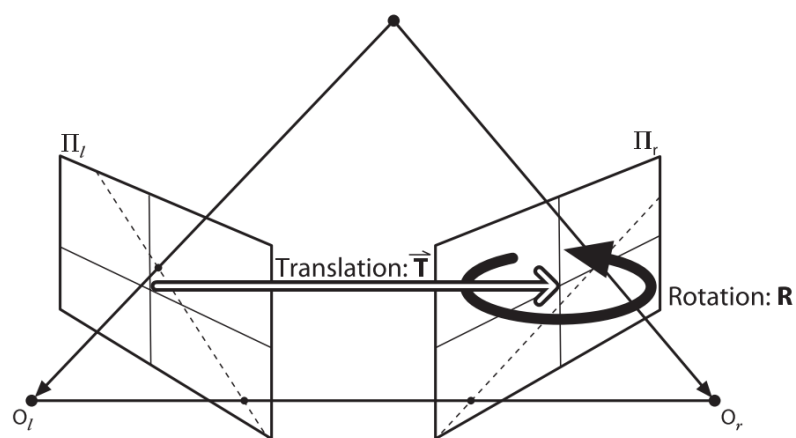
Existem diferentes métodos para a realização da calibração, alguns são dinâmicos com o objetivo de calibrar o sistema em diferentes momentos durante a captura (CROWLEY; BOBET; SCHMID, 1993) e (DANG; HOFFMANN; STILLER, 2009), e outro métodos são estáticos, onde o processo de calibração é realizado antes do início da captura de imagens pelo sistema estéreo (BRADSKI, 2008). A calibração proposta por (ZHANG, 2000), considera que em um sistema que não existam alterações das posições entre o par de câmeras, o processo de calibração deve ser realizado somente uma vez.

O processo de calibração pode ser aplicado para uma única câmera para determinar os parâmetros intrínsecos como mencionado na Seção 2.1.2, assim como para determinar parâmetros em um sistema de estereoscopia. Para realização da calibração para uma única câmera é utilizada a proposta de (ZHANG, 2000), que basicamente utiliza de um padrão de objeto já conhecido geometricamente e de fácil reconhecimento pelo sistema de visão computacional. Sendo assim, é utilizado um tabuleiro de xadrez que viabiliza a extração das coordenadas dos cantos dos quadros sob diferentes orientações. O processo realiza a identificação e reprojeção de sucessivos pontos que possibilitam a aquisição das informações dos parâmetros de distorção da câmera e parâmetros internos. Maiores detalhes da geometria que envolve a aquisição dos parâmetros de calibração podem ser encontrados em (ZHANG, 2000) e (DUANE, 1971).

A calibração estéreo é o processo de calcular o relacionamento geométrico entre o par de câmeras no espaço. Diferente do processo de retificação que tem como objetivo a correção

de imagens individuais para alinhar o par horizontalmente. O foco dessa etapa é orientar o sistema de forma que o eixo óptico de ambas as imagens se intersectem no infinito. Esse procedimento pode ser realizado sequencialmente ao sistema de calibração das câmeras. A partir do momento em que os pontos dos cantos dos quadros do tabuleiro são identificados, estes podem ser utilizados como informação para realizar as transformações de translação e rotação em ambas as imagens conforme ilustrado na Figura 2.5.

Figura 2.5 – Representação da calibração estéreo.



Fonte: (BRADSKI, 2008).

Como resultado da etapa de calibração de um sistema estéreo pode ser gerada uma matriz denominada fundamental ou ainda uma outra denominada essencial. A matriz fundamental contém as informações de rotação e translação que relacionam as duas câmeras fazendo valer a restrição epipolar apresentada na Seção 2.1.4, além de informações de parâmetros intrínsecos de ambas as câmeras. A matriz essencial diferentemente da matriz fundamental não possui os parâmetros intrínsecos das câmeras.

2.1.5.2 Retificação

O procedimento de retificação tem como propósito realizar o alinhamento entre o par de imagens de forma a proporcionar a redução da complexidade do problema de correspondência. Após a realização da retificação a área de busca entre pontos correspondentes no sistema estéreo é reduzido. O sistema de estereoscopia passa a considerar deslocamentos entre os pontos similares restritos a linhas horizontais, dispensando a necessidade de verificação bidimensional.

Existem diferentes algoritmos para tratar da retificação, porém (BOUGUET, 2015) con-

sidera um sistema previamente calibrado beneficiando uma abordagem mais viável e menos propensa a erros. O algoritmo de Bouguet simplesmente minimiza as distorções restantes do sistema maximizando a visão em uma área comum (BRADSKI, 2008).

O algoritmo usa os parâmetros de rotação e translação das câmeras gerados pela etapa de calibração. Para minimizar a distorção da imagem de projeção cada imagem do par de câmera é rotacionado pela metade diante do matriz de rotação, fazendo com que cada raio principal finalize em posição paralela entre o par de imagens. Para calcular a retificação das imagens, o epipolo da câmera da esquerda deve apontar para o infinito e alinhar as linhas epipolares horizontalmente, criando uma matriz de rotação inicialmente apontando para a direção do próprio epipolo e_1 . O ponto principal da imagem da esquerda é utilizado como centro de projeção para ambas as câmeras. Posteriormente é computada uma nova matriz de projeção para ambas as imagens e uma matriz de reprojeção denominada Q .

$$Q = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & -1/T_x & (c_x - c'_x)/T_x \end{bmatrix} \quad (2.16)$$

Todos os parâmetros são da imagem da esquerda usada como referência, com a exceção do parâmetro c'_x que representa o ponto principal da imagem da direita. Ao considerar que os raios principal das imagens se intersectam no infinito, os pontos principais do par de imagens são iguais. Dessa forma $(c_x - c'_x)/T_x = 0$ (BRADSKI, 2008).

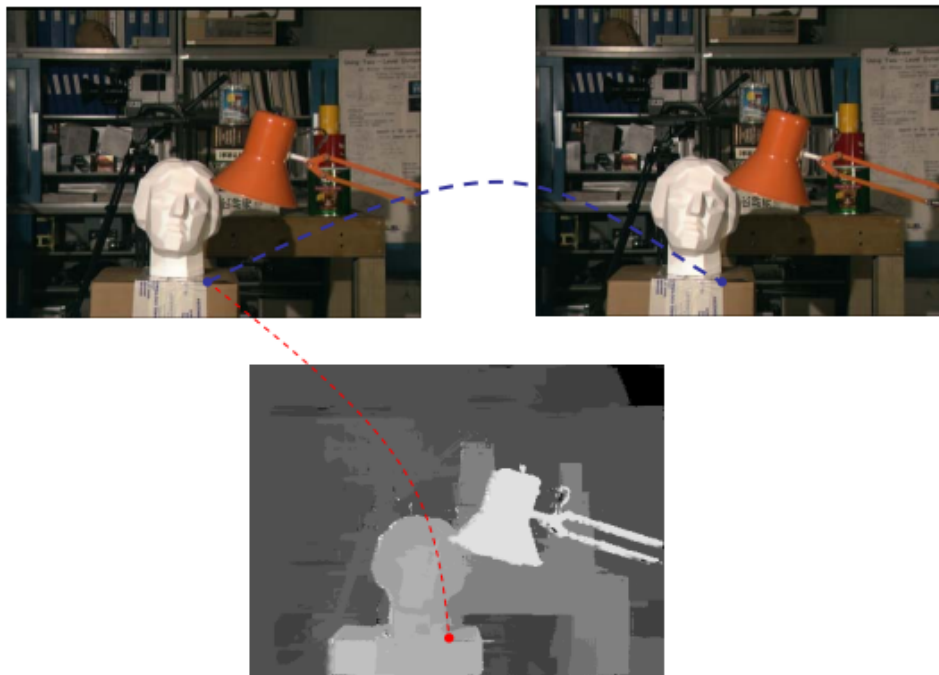
2.1.5.3 Correspondência

A correspondência em visão estéreo (*stereo matching*) refere-se ao procedimento de encontrar a projeção do mesmo ponto em ambas as imagens do par (BRADSKI, 2008). Este é um tópico que representa um grande desafio para estereoscopia, pois tem como problemas a serem tratados: regiões oclusas, problemas de iluminação, distorções e texturas fracas ou homogêneas (SCHARSTEIN; SZELISKI, 2002).

O resultado desse procedimento visa a aquisição do mapa de disparidades, que consiste em uma imagem que possui como valores de intensidade dos *pixels* o resultado obtido pelo cálculo da subtração entre um ponto x da imagem da esquerda e um ponto x da imagem da direita que representem o mesmo ponto do mundo, esse cálculo refere-se aos deslocamentos dos pontos homólogos no eixo x . Usualmente são utilizadas imagens em tons de cinza (8 bits), que proporciona valores de até 256 níveis de profundidade. Dessa forma, deslocamentos maio-

res representam pontos mais próximos e intensidades em tons mais claros, já os deslocamentos menores são pontos mais distantes e representados por tons mais escuros (BRADSKI, 2008). A posição dos pontos no mapa de disparidades tem como referência a imagem da esquerda, também chamada de imagem base do sistema estéreo. A Figura 2.6 ilustra o mapa de disparidade adquirido do par de imagens de uma cena clássica utilizada na literatura denominada Tkusuba.

Figura 2.6 – Mapa de disparidade em escala de cinza a partir do par de imagens.

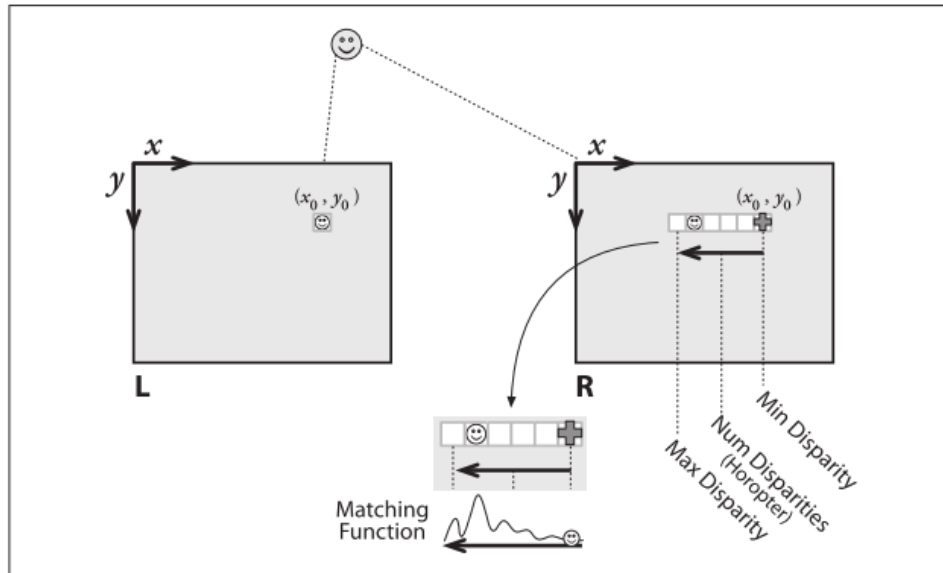


Fonte: (BRADSKI, 2008).

A disparidade também é tratada como um parâmetro em algoritmos e correspondência. Inicialmente deve ser definido o intervalo de disparidade mínima e máxima que vai ser considerado na geração do mapa de disparidades. A disparidade máxima define o deslocamento máximo que algoritmo pode considerar e a disparidade mínima define a profundidade que pode ser considerada no mapeamento da cena. Neste contexto, é importante ressaltar que disparidades menores são pontos que convergem no sistema de estereoscopia, logo são ponto mais distantes. A Figura 2.7 ilustra o impacto dos parâmetros de mínima e máxima disparidade na busca de ponto similares.

A Figura 2.8 ilustra a interferência desses parâmetros na etapa de correspondência ao considerar a distância dos objetos frente ao par de câmeras. Nesse cenário a distância máxima pode incluir ou não objetos mais próximos da cena. Para o cálculo do mapa de disparidade,

Figura 2.7 – Representação de correspondência na mesma linha e a procura pelo mesmo ponto entre o par iniciando na disparidade mínima e movendo para a esquerda até um intervalo de disparidade máxima definido.



Fonte: (BRADSKI, 2008).

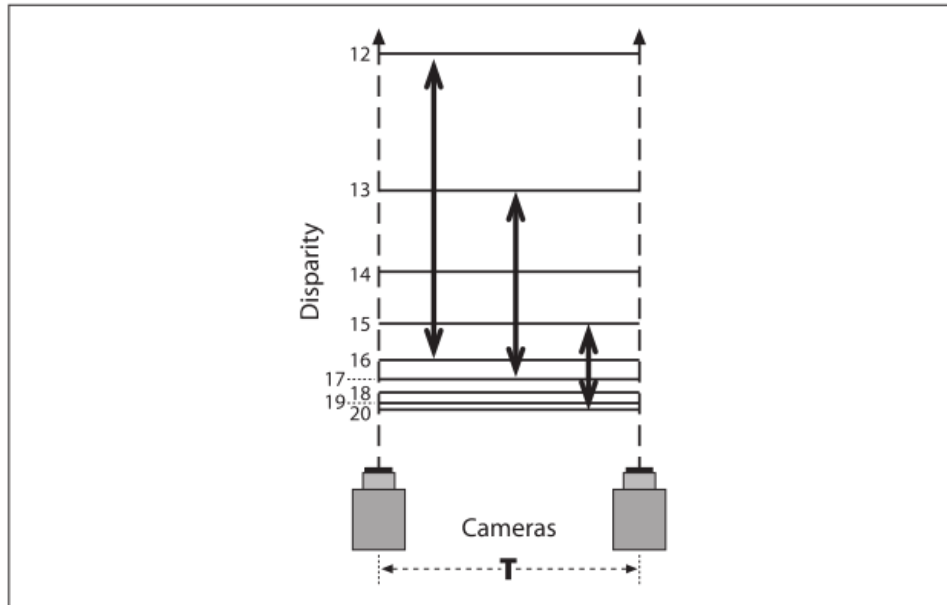
o que estiver fora no intervalo de disparidades tende a ser considerado desconhecido e consequentemente pode gerar ruído.

A taxonomia proposta por (SCHARSTEIN; SZELISKI, 2002) sugere a divisão do algoritmo de correspondência em "blocos", essa estrutura é utilizada por grande parte das soluções encontradas na literatura. A taxonomia foi proposta com o objetivo de avaliar diferentes componentes que constituem a correspondência. Neste sentido são quatro blocos que contemplam a etapa de correspondência:

- Computação do Custo de Correspondência;
- Agregação de Custo;
- Computação da Disparidade e Otimização;
- Refinamento da Disparidade.

Os algoritmos presentes na literatura são ainda classificados como locais e globais, diferenciando-se basicamente na forma de implementação aplicada para corresponder pontos do par de imagens (SCHARSTEIN; SZELISKI, 2002).

Figura 2.8 – Os planos horizontais representam a influência entre os parâmetros de disparidade mínima e máxima para a distância de objetos no cenário e na vertical que o algoritmo irá considerar de profundidade.



Fonte: (BRADSKI, 2008).

Diante dos blocos que compõem a estrutura proposta por (SCHARSTEIN; SZELISKI, 2002), a Computação do Custo de Correspondência tem como objetivo avaliar a similaridade entre pontos no par de imagens, para isso emprega uma métrica de similaridade. Comumente são aplicadas como métricas a Diferença Absoluta de intensidade (AD) ou a Diferença Quadrática (SD). O bloco denominado Agregação de Custos é utilizado somente em métodos de busca local. Esse aplica como critério para corresponder pontos, estruturas do tipo janela que visam resolver problemas como o baixo poder discriminativo e a sensibilidade a ruídos que podem ocorrer em uma avaliação *pixel á pixel*. Dessa forma, através de um conjunto de pontos de uma janela na imagem da esquerda é realizada uma busca por uma janela que apresenta maior similaridade na imagem da direita. Em geral os métodos aplicados para este bloco utilizam somatórios ou médias sobre janelas que podem ser sobre um conjunto de duas dimensões (x, y) ou três dimensões $(x, y, disparidade)$. A implementação do bloco de Computação da Disparidade em métodos locais é um processo trivial, pois nesse método é dado o enfoque para Computação do Custo de Correspondência e a Agregação de Custo, assim basta selecionar para cada *pixel* a disparidade associada que apresenta menor valor de custo. No entanto, para métodos globais todo o trabalho de encontrar pontos similares e calcular a similaridade se concentra no bloco de

Computação da Disparidade, isso porque métodos globais pulam a etapa de agregação. Por último, o Refinamento da Disparidade é feito por remoção de picos por filtro de média, checando a consistência através do cálculo da disparidade usando a imagem da esquerda e posteriormente a imagem da direita como referencia ou interpolando lacunas entre outras abordagens que melhoram a qualidade do mapa de disparidades.

Abordagem Local

A abordagem local conforme o nome sugere, é feito o uso de uma janela finita ou região onde assume-se que os pontos vizinhos tendem a ter uma disparidade similar. A correspondência para os pontos de cada imagem é realizada entre a região da imagem de referência (esquerda) com uma janela movendo sobre todas as posições em um espaço de busca na imagem da direita (SCHARSTEIN; SZELISKI, 2002). Em outras palavras, os métodos locais levam em consideração apenas intensidades próximas aos *pixels* candidatos.

Na etapa de Computação do Custo de Correspondência é definida uma métrica de similaridade a ser utilizada para comparação entre os *pixels* do par de imagens. Segue a Fórmula 2.17, que define a métrica de Diferença Absoluta (AD), onde C é o custo, p é o *pixel* da imagem esquerda ou base, q da imagem da direita ou alvo, d a disparidade e I representa a intensidade.

$$C(p, q, d) = |I_p(x, y) - I_q(x + d, y)| \quad (2.17)$$

A etapa de Agregação de Custo utiliza uma região (janela) para realizar o cálculo de similaridade. Uma limitação do uso de regiões de suporte de *pixels* em geral é o baixo poder discriminativo e sensibilidade a ruídos, pois a relação de agregação supõem uma disparidade constante, o que ocasiona em um mapa de disparidade com pequenos objetos borrados (SZELISKI, 2010).

A Equação 2.18 representa esta etapa, onde U é o conjunto contendo os valores de intensidade de x e y determinado pelo tamanho do bloco.

$$C(p, q, d) = \sum_{i, j \in U} |I_p(x + i, y + j) - I_q(x + i + d, y + j)| \quad (2.18)$$

Para computar a disparidade final é escolhido cada *pixel* que disparidade apresenta o valor mínimo de custo. Assim, na busca local é aplicada a estratégia do vencedor fica com tudo "*winner-take-all*".

A solução de aplicar janelas pode ainda gerar erros nas discontinuidades de profundidade ocasionadas por regiões oclusas. Para tratar esse problema diferentes abordagens foram sugeridas, como aplicação do processo de correspondência considerando o cálculo da disparidade em dois sentidos, primeiramente utilizando a imagem da esquerda como referência e posteriormente o cálculo com a imagem da direita como referência.

No processo de correspondência, a similaridade dos *pixels* entre o par de imagens é determinante para o cálculo de disparidade. Nesse aspecto, o problema de iluminação, além dos diferentes ganhos de luminosidade dos sensores podem influenciar no resultado final. Para solucionar este problema uma solução muito empregada é a utilização da normalização das imagens por meio do filtro *Laplacian of the Gaussian* LOG. O filtro de LOG possui a filtragem gaussiana e laplaciana. Neste sentido, o filtro laplaciano reduz a influência da iluminação gerando respostas zero nas áreas com intensidades constantes e negativa ou positiva nas proximidades de bordas. E o filtro gaussiano reduz a influência de ruído, já que o laplaciano se apresenta muito sensível a este tipo de ruído (SZELISKI, 2010).

Abordagem Global

Abordagens do tipo global não implementa a etapa de Agregação de Custo, assumindo assim como solução uma otimização global (SZELISKI, 2010). A tarefa desempenha a rotulagem de cada *pixel* com a solução disparidade d que minimiza uma função global de energia. A Equação 2.19 esboça como isto é realizado através de uma representação matemática.

$$E(d) = E_{dados(d)} + \lambda E_{restricao(d)} \quad (2.19)$$

Na Equação 2.19 $E_{dados(d)}$ é responsável pela Computação do Custo das combinações entre o par de imagens e $E_{restricao}$ são restrições a fim de tornar a busca computacionalmente tratável e λ regula as restrições. Uma restrição pode ser a intensidade dos *pixels* com os seus vizinhos, considerando que essa variação deva ser suave.

A Computação do Custo de Correspondência é ilustrada pela Formula 2.20. O termo $C(x, y, d(x, y))$ representa o intervalo de correspondência.

$$E_d(d) = \sum_{x,i} C(x, y, d(x, y)), \quad (2.20)$$

O termo de restrição $E_{restricao}(d)$ é similar a etapa de Agregação de Custos dos métodos

locais e tem como objetivo minimizar falsas correspondências e conseqüentemente gerar mapas de disparidades mais suaves possível. Uma forma de representar matematicamente a codificação para suavidade é a minimização das diferenças na disparidades entre os *pixels* vizinhos como é mostrada na Equação 2.21.

$$E_s(d) = \sum_{x,i} \rho(d(x, y) - d(x + 1, y)) + \rho(d(x, y) - d(x, y + 1)), \quad (2.21)$$

Onde ρ é uma função monótona de incremento das diferenças das disparidades. Porém essa abordagem é muito simples e tende a gerar mapa de disparidade com suavização ao ponto de excluir as bordas. Assim o termo de restrição pode ser modificado ficando como na Equação 2.22.

$$E_s(d) = \sum_{x,i} \rho_d(d(x, y) - d(x + 1, y)) + \rho_I(||I(x, y) - I(x, y + 1)||), \quad (2.22)$$

Como apresentado na Equação 2.22 o termo de restrição pode ser modificado para fazer depender da diferença das intensidades. Uma minimização energética é realizada com o objetivo de encontrar a função d (disparidade), que minimize a energia global dada pela Equação 2.19.

Métodos globais em sua grande maioria apresentam resultados superiores em relação aos métodos locais, porém possuem maior custo computacional (BRADSKI, 2008).

2.1.5.4 Reprojeção

A reprojeção ou também chamada de reconstrução está relacionada á recuperação da informação de profundidade a partir dos mapas de disparidades. Embora em alguns casos o mapa de disparidade é suficiente para detectar objetos próximos ou distantes do sistema estéreo, a etapa de reconstrução é a mais indicada para recuperar a informação de tridimensionalidade da cena de forma mais fiel. Para este procedimento é necessária a execução de todas as etapas, calibração, retificação e correspondência (BRADSKI, 2008).

A etapa de retificação através do algoritmo proposto por (BOUGUET, 2015) gera uma matriz denominada matriz de reprojeção ou matriz Q . Essa matriz possui as informações de ponto principal de c_x e c_y , distância focal f e linha base T , sendo que através dela os pontos referente as coordenadas 3D podem ser recuperados re-arranjando as Equações presentes na Seção 2.1.1 e 2.1.3.

$$X = \frac{(x - c_x) \cdot T}{f} \quad (2.23)$$

$$Y = \frac{(y - c_y) \cdot T}{f} \quad (2.24)$$

$$Z = \frac{f \cdot T}{d} \quad (2.25)$$

As Equações 2.23, 2.24 e 2.25 calculam as coordenadas do mundo real a partir da matriz Q do sistema de estereoscopia. Porém, para representar os pontos em 3D é necessário os valores das intensidades, para isto são atribuídos os valores da imagem da esquerda do par estéreo.

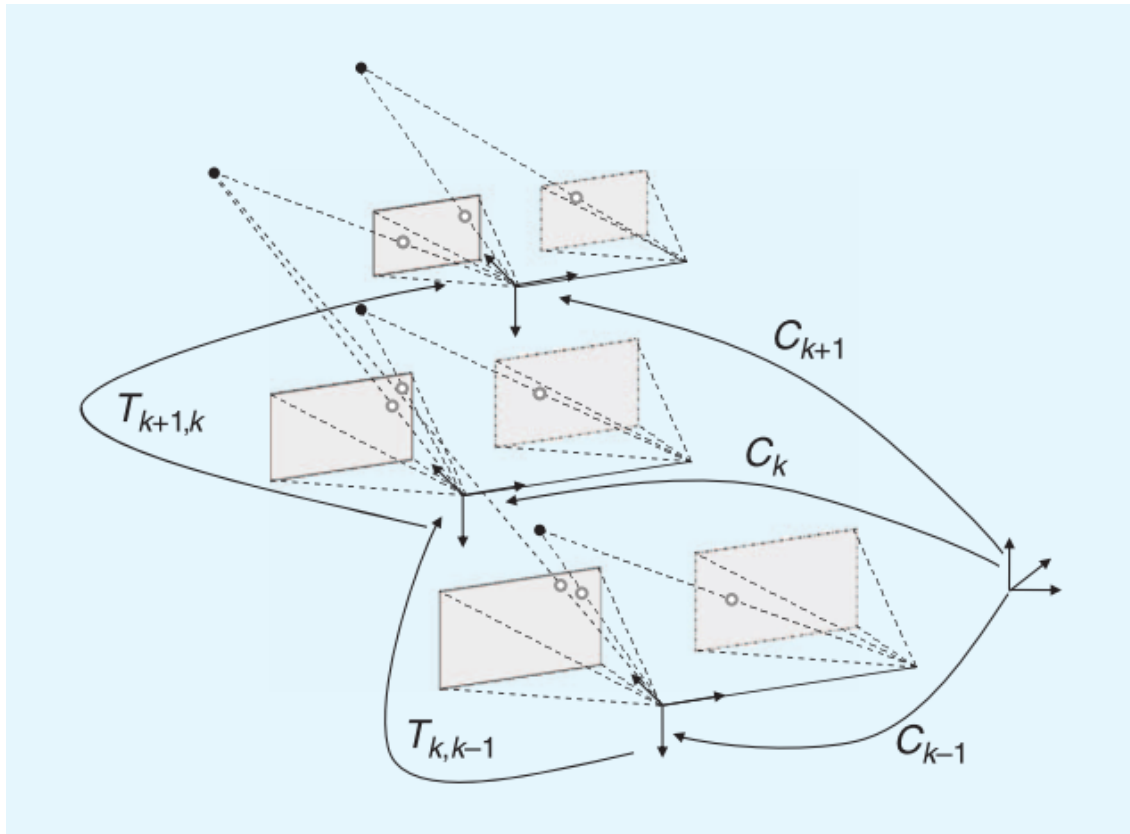
2.2 Odometria Visual

A odometria visual é um processo que estima movimento seja este realizado por um robô, um humano ou veículo utilizando uma ou varias câmeras (NISTER; NARODITSKY; BERGEN, 2004). Este nome foi designado pela semelhança com a odometria que utiliza rodas (*whell odometry*), e estima incrementos dos movimentos pelo números de voltas por um determinado tempo. Da mesma forma, a odometria visual estima movimento através de incrementos de quadros capturados. Para que seja possível a utilização desta abordagem é necessária a presença de iluminação na cena e também é desejável um ambiente com pouco dinamismo e com textura suficiente para que permita a extração de movimento aparente (SCARAMUZZA; FRAUNDORFER, 2011a).

A vantagem desta abordagem é o fato de não ser afetada por deslizamentos em oposição a abordagem que utilizam rodas. Além disso, é mais precisa ao fornecer estimativas de trajetórias e se mostra mais atraente em comparação a propostas que utilizam sensores como sistema de posicionamento global (GPS) e odometria a laser, diante da aplicação em ambientes aéreos, aquáticos ou ainda de ambientes de difícil acesso.

A Figura 2.9 ilustra a movimentação de um sistema de câmeras em instantes discretos k . Os instantes são denotados por $I_{l,0:n} = I_{l,0}, \dots, I_{l,n}$ e $I_{r,0:n} = I_{r,0}, \dots, I_{r,n}$. Nesta representação, as coordenadas das câmeras também são as coordenadas do agente, seja ele, um robô, carro ou pessoa com o sistema de captura. A posição das câmeras é dada por transformações de corpo rígido $T_{k,k-1} \in R^{4 \times 4}$ como na Fórmula 2.26.

Figura 2.9 – Poses relativas $T_{k,k-1}$, de posições adjacentes das câmeras que são calculadas através de características visuais e concatenadas para obter as poses absolutas C_k tendo como coordenada do frame inicial o $k = 0$.



Fonte: (SCARAMUZZA; FRAUNDORFER, 2011a).

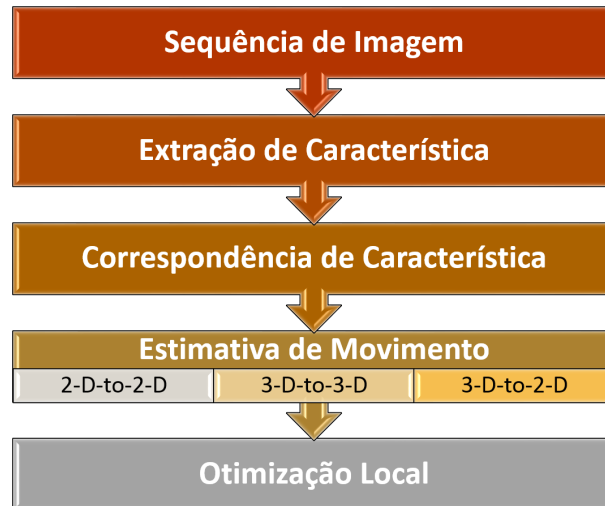
$$T_{k,k-1} = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix}, \quad (2.26)$$

Na Equação 2.26, $R_{k,k-1}$ é a matriz de rotação, e $t_{k,k-1}$ o vetor de translação. O conjunto $T_{1:n} = T_{1,0}, \dots, T_{n,n-1}$ contém todos os movimentos sucessores. A pose das câmeras é representado pelo conjunto $C_{0:n} = C_0, \dots, C_n$ com as transformações das câmeras com coordenada inicial $k = 0$. A posição atual pode ser calculada concatenando todas as transformações $T_k (K = 1, \dots, n)$, e em seguida por, $C_n = C_{n-1}T_n$ com C_0 sendo a pose da câmera no instante $k = 0$, que pode ser dado arbitrariamente.

A odometria visual calcula transformações T_k das imagens I_k e I_{k-1} , concatenando as transformações para recuperar todo o trajeto das câmeras $C_0 : n$. Um refinamento é realizado sobre as últimas m poses de forma iterativa minimizando a soma dos quadrados dos erros de reprojeção dos pontos 3D reconstruídos. O refinamento em Odometria visual é denominado

windowed-bundle adjustment, por conta da realização em janelas dos m frames.

Figura 2.10 – Diagrama em blocos dos principais componente da Odometria visual.



Fonte: Autoria própria.

A Figura 2.10 ilustra os principais componentes para a odometria visual. As duas primeiras etapas consistem em detectar e corresponder as características similares entre quadros adjacentes, a terceira etapa consiste em calcular a movimentação T_k , entre instantes $K - 1$ e K . E por fim, o refinamento é realizado sobre os m últimos quadros.

2.2.1 Estimativa de Movimento

A estimativa de movimento é o núcleo do sistema de odometria visual. Mais precisamente a etapa de estimar movimento entre o quadro corrente e o quadro anterior. As transformações T_k entre duas imagens I_k e I_{k-1} , podem ser estimadas de dois conjuntos de características que correspondam, f_{k-1} e f_k no instante de tempo $k - 1$ e k , respectivamente. Existem diferentes métodos para corresponder as características que podem ser em 2D ou 3D:

- 2D-para-2D: Nesse caso, tanto f_{k-1} e f_k são coordenadas em 2D;
- 3D-para-3D: As características tanto de f_{k-1} e f_k são utilizadas em 3D. Assim é necessária a triangulação a cada instante de tempo e para isto é utilizada visão estéreo;
- 3D-para-2D: Nessa abordagem, f_{k-1} é especificada em 3D e f_k são as re-projeções correspondentes em 2D na imagem I_k .

2.2.2 Movimentação de Imagens por características 2D-para-2D

Para a identificar a movimentação por características considerando somente duas dimensões dos passos a serem executados são:

1. Capturar um novo quadro i_k ;
2. Extrair e corresponder características entre I_{k-1} e I_k ;
3. Calcular a matriz essencial do par de imagens I_{k-1}, I_k ;
4. Decompor a matriz essencial em R e t_k e formar T_k ;
5. Calcular a escala relativa e reescalar t_k ;
6. Concatenar as transformações pelo cálculo $C_k = C_{k-1}t_k$;
7. Repetir os passo novamente.

2.2.3 Movimentação de Imagens por características 3D-para-3D

Esta metodologia é propensa a erros, diante do fato que depende do cálculo correto das triangulações, que não pode ser garantida (NISTER; NARODITSKY; BERGEN, 2004).

1. Capturar dois pares de imagens estéreo $I_{l,k-1}, I_{r,k-1}$ e $I_{l,k}, I_{r,k}$;
2. Extrair e corresponder características entre I_{k-1} e I_k ;
3. Triangular as características correspondidas para cada par estéreo;
4. Calcular T_k para características 3D X_{k-1} e X_k ;
5. Concatenar as transformações pelo cálculo $C_k = C_{k-1}t_k$;
6. Repetir os passos novamente.

2.2.4 Movimentação de Imagens por características 3D-para-2D

Esta abordagem propõem maior robustez e mais amplamente adotada pela literatura e sugerida por (NISTER; NARODITSKY; BERGEN, 2004) e (SCARAMUZZA; FRAUNDORFER, 2011b).

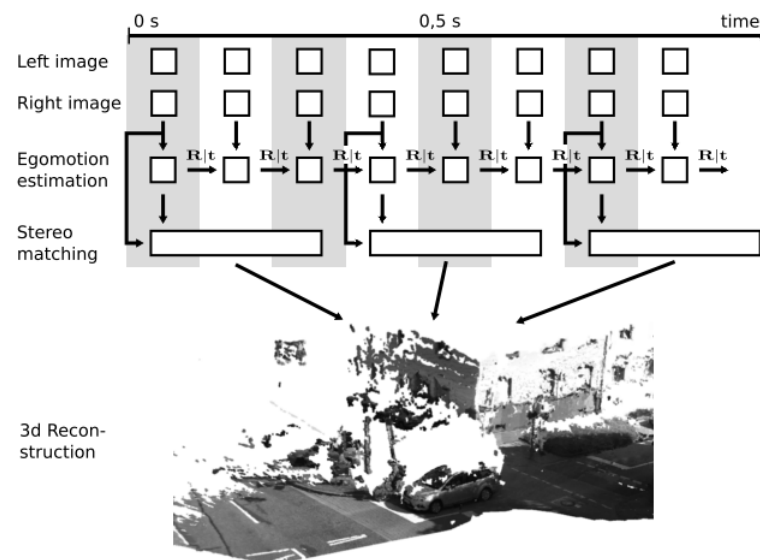
1. Realiza somente para a primeira iteração;
2. Capturar dois quadro i_{k-1} e i_{k-2} ;
3. Extrair e corresponder características entre os quadros;
4. Realiza a triangulação de I_{k-2} , I_{k-1} ;
5. Realiza as demais iterações;
6. Capturar um novo quadro i_k ;
7. Extrai características e corresponde com o quadro anterior I_{k-1} ;
8. Calcular a pose da câmera através de correspondência 3D para 2D;
9. Triangular todas as características correspondidas entre i_{k-1} e i_{k-2} ;
10. Repetir os passos novamente.

2.3 LibViso

A proposta de (GEIGER; ZIEGLER; STILLER, 2011), realiza o cálculo da odometria visual com a taxa de *frames* atingindo até 20 *frames* por segundo (fps). É feito uso de uma abordagem de correspondência de características esparsas em conjunto com um algoritmo de odometria visual robusto.

Conforme a Figura 2.11, são utilizados características de quatro *frames*, correspondendo a imagem da direita e esquerda do par de imagens de *frames* consecutivos. Com o propósito de encontrar as características locais estáveis, a imagem de entrada é filtrada com *blob* e *corner mask* $5x5$. Posteriormente é aplicado no filtro de imagens *non-maximum-* e *non-minimum-supression* resultando em quatro classes de características candidatas, isto é, *blob max*, *blob min*, *corner max* e *corner min*. Com o objetivo de reduzir o custo computacional, são correspondidas características que atendam somente estas classes. Dados os pontos característicos é realizada a comparação em blocos de “11 X 11” com filtro de Sobel horizontal e vertical, através da Soma das diferenças absolutas (SAD) como métrica de similaridade. Para acelerar a correspondência, é quantificada a resposta para 8 bits e a soma das diferenças sobre um conjunto espaço de 16 locais, ao invés de considerar uma janela sobre um bloco inteiro. Uma vez que o SAD

Figura 2.11 – Visão geral do sistema Libviso.



Fonte: (GEIGER; ZIEGLER; STILLER, 2011).

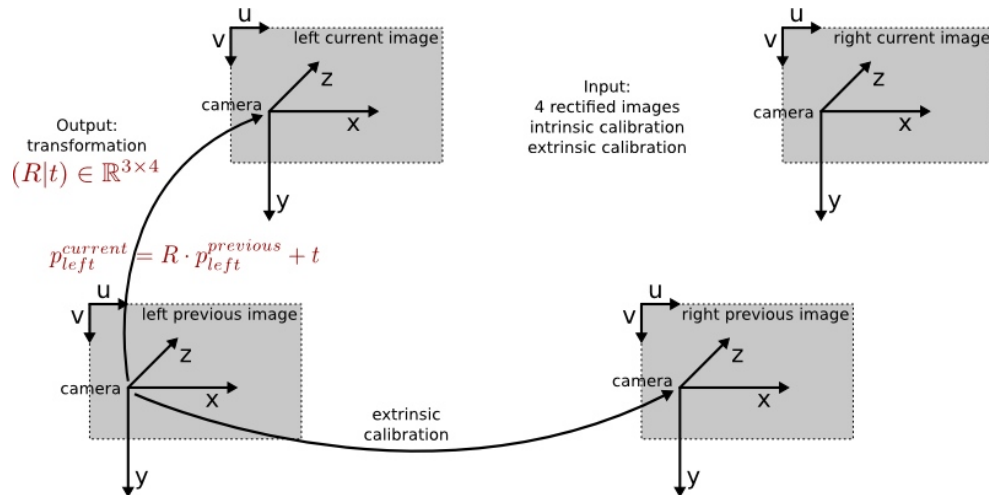
de 16 bytes pode ser calculado de forma eficiente é necessária somente duas chamadas (Sobel horizontal + vertical).

A forma de realizar a correspondência considerando quatro *frames*, é similar a idéia de círculo. Para todas as características candidatas do *frame* corrente é procurado a correspondência com o *frame* anterior, com uma janela de busca de $M \times M$, em seguida o *frame* anterior da imagem da direita, é feita a busca com corrente também da esquerda e assim o corrente do par passa ser o anterior da próxima iteração e assim consecutivamente.

O cálculo da estimativa de movimento, é realizada através da minimização da soma de erros reprojados e o uso da estimativa de velocidade através do filtro de Kalman.

Conforme a Figura 2.12, os pontos característicos do *frame* anterior projetado por triangulação, usando os parâmetros de calibração. Para o *frame* seguinte, são projetados pontos característicos para o 2D, em ambas as imagens.

Figura 2.12 – Definição do sistema de coordenadas.



Fonte: (GEIGER; ZIEGLER; STILLER, 2011).

Por meio da otimização *Gauss-Newton*, iterativamente são minimizados os respectivos parâmetros de (r, t) , como mostrado na Equação 2.27:

$$\sum_{i=1}^N \|x_i^{(l)} - \pi^{(l)}(X_i; r, t)\|^2 + \|x_i^{(r)} - \pi^{(r)}(X_i; r, t)\|^2 \quad (2.27)$$

Para tornar mais robusta a estimativa é utilizada a abordagem denominada *Random sample consensus* (RANSAC), eliminando os possíveis *outliers*.

3 TRABALHOS RELACIONADOS

Este capítulo tem como objetivo apresentar trabalhos encontrados na literatura que utilizam diferentes sensores de captura de imagens para realizar o mapeamento 3D. Embora o número de trabalhos relacionados ao mapeamento 3D por meio de técnicas de visão computacional tenha crescido na última década, poucos atingem o desempenho em tempo real, em outras palavras que apresentem processamento com *delay* pouco significativo com taxas de *frames* na ordem de milissegundos. Os trabalhos relacionados apresentados nesse capítulo são técnicas que dispensam sensores de posicionamento. Sendo assim, as abordagens são organizadas considerando os diferentes tipos de sensores de captura de imagens.

3.0.1 Estereoscopia

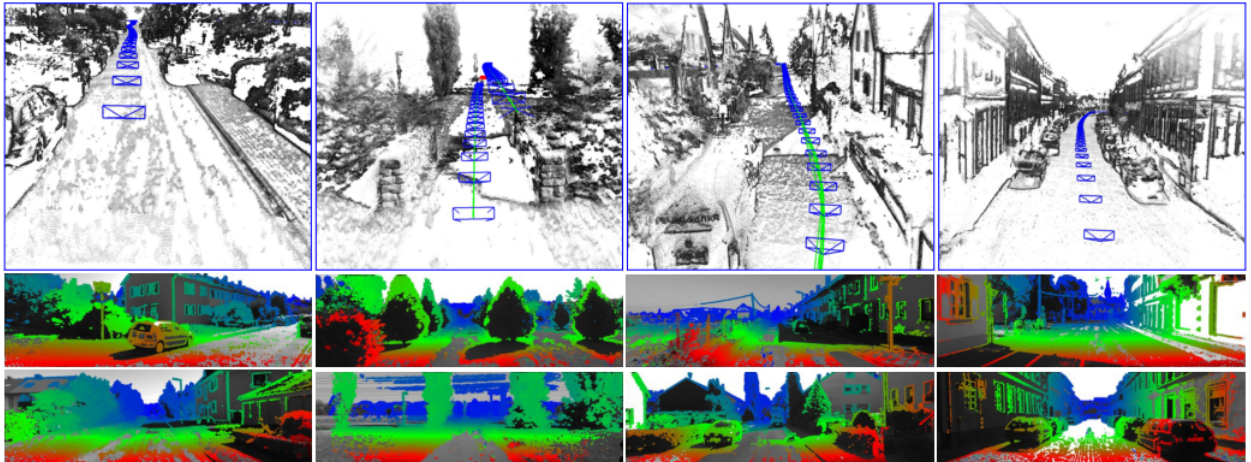
A estereoscopia é um tema desafiador conforme apresentado na Seção 2.1.5.3. Muitas abordagens se aproximam da solução de aquisição de informação de profundidade da cena diante dos desafios de texturas homogêneas e objetos oclusos, mas deixam a desejar no quesito desempenho.

O trabalho “*Large-Scale Direct SLAM with Stereo Cameras*” proposto por (ENGEL; STUCKLER; CREMERS, 2015) é o estado da arte tendo em vista o desempenho em tempo real para o mapeamento de ambiente 3D. Esse trabalho utiliza uma técnica de reconstrução 3D semi-densa, onde são consideradas partes da imagem mais relevantes como bordas, cantos e texturas mais evidentes. A correspondência é realizada de forma direta entre as regiões sobressalentes, ocorrendo simultaneamente com a estimativa de movimento. Assim, os quadros chaves são gerados por meio da estimativa de movimento em adição a disparidade. A Figura 3.1 exibe a nuvem de pontos esparsa assim como o mapa de disparidades adquirido por essa abordagem.

Outra abordagem de destaque é proposta por (GEIGER; ZIEGLER; STILLER, 2011), denominada “*StereoScan: Dense 3d Reconstruction in Real-time*” como mencionada na Seção 2.11 que utiliza um algoritmo próprio para correspondência gerando um mapa de disparidades denso. Os resultados obtidos mostram um taxa de 3 fps.

A proposta intitulada “*Stereo Visual SLAM System in Underwater Environment*” (PI et al., 2014), realiza mapeamento de ambientes aquáticos. A extração de características do ambiente que são usadas para estimar a trajetória de um robô que possui o sistema estéreo acoplado. Para isto o algoritmo *Speeded Up Robust Features* (SURF) é aplicado. O método

Figura 3.1 – A nuvem de pontos do mapeamento 3D no topo. Na parte inferior o mapa de disparidades esparso.



Fonte: (ENGEL; STUCKLER; CREMERS, 2015).

Extend Kalman Filter (EKF) resolve o problema de estimativa em tempo real.

3.0.2 Monocular

Trabalhos que utilizam sensores monoculares para realização de mapeamento 3D possuem como vantagem a aplicação tanto em ambientes internos como externos. Esse dinamismo também representa um desafio, tendo em vista que o método é baseado na movimentação do sistema de captura para estimar a profundidade e conseqüentemente adquirir as coordenadas tridimensionais da cena. A propensão de acúmulo de erros representa uma desvantagem desse método (ENGEL; SCH; CREMERS, 2014).

A abordagem proposta por Engel, intitulada “*LSD-SLAM: Large-Scale Direct Monocular SLAM*” considera pequenos deslocamentos para reconstruir ambientes 3D em tempo real através de mapas de profundidade semi-densos. A estimativa de movimento é realizada de forma direta, denominada Odometria Visual Direta, habilitando o uso da informação de toda a imagem.

3.0.3 Sensores RGB-D

O mapeamento 3D de ambientes internos em tempo real foi viabilizado nos últimos anos pelo surgimento de sensores RGB-D como Kinect da empresa *Microsoft*. Este sensor possibilita a captura de imagens RGB simultaneamente com a imagem de profundidade da cena com uma

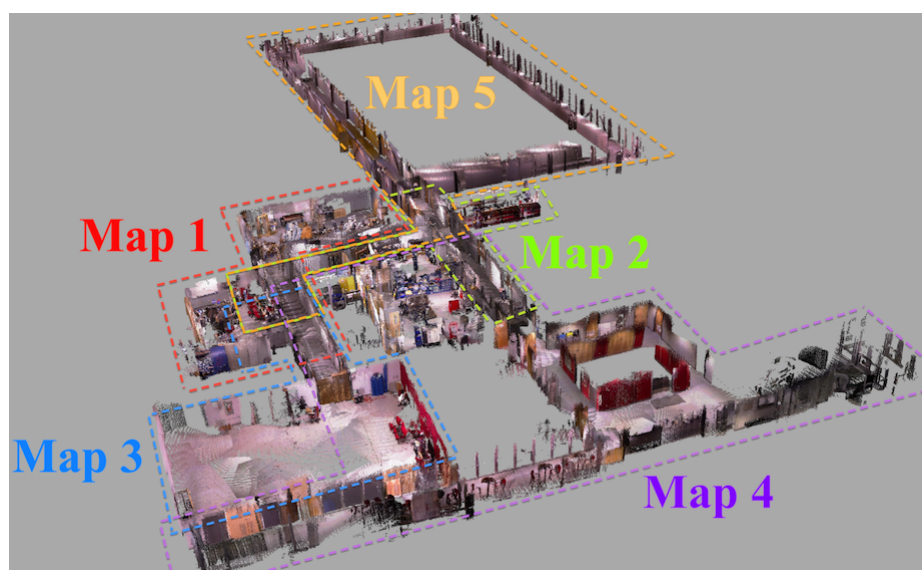
taxa de 30 fps.

As duas abordagens de referência disponíveis na literatura que utilizam sensores RGB-D em conjunto com técnicas de SLAM visual é o software denominado RTAB-Map (*Real-Time Appearance-Based Mapping*), proposto por (LABBE; MICHAUD, 2014) e vencedor do desafio de navegação autônoma na Conferência Internacional de Robôs e Sistemas Inteligentes (IROS), ocorrido na cidade de Chicago no ano de 2014. Além dessa abordagem, se destaca a proposta intitulada "3D Mapping with an RGB-D camera" desenvolvida por (ENDRES et al., 2014).

O software RTAB-Map proposto por Labbé tem como base o desenvolvimento de três artigos, sendo: "*Online Global Loop Closure Detection for Large-Scale Multi-Session Graph-Based SLAM*" publicado em "*Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*" no ano de 2014, "*Appearance-Based Loop Closure Detection for Online Large-Scale and Long-Term Operation*" publicado no ano de 2013 na "*IEEE Transactions on Robotics*", e o mais antigo, "*Memory management for real-time appearance-based loop closure detection*" publicado no "*Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*" no ano 2011.

O trabalho de Labbé descrito nos três artigos e presente no RTAB-Map tem como objetivo principal gerenciar a memória para o problema de fechamento de repetição (*Loop Closure*) em ambientes de grande escala para propiciar o processamento em tempo real.

Figura 3.2 – Mapeamento 3D de ambientes internos dividido em multi-sessões.



Fonte: (LABBE; MICHAUD, 2014).

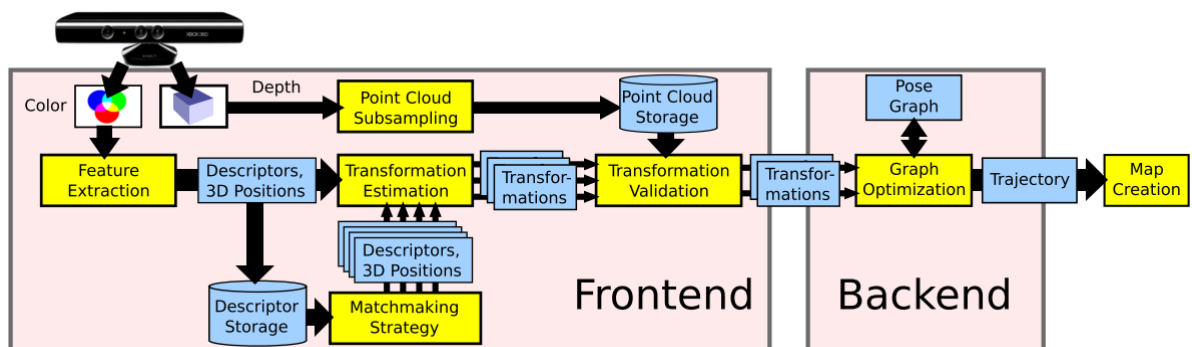
Para realização de teste em larga escala conforme ilustrado na Figura 3.2 foi utilizado

sensor de laser para auxiliar o mapeamento devido ao tamanho do ambiente. Dessa forma, através dos teste realizados verificou-se que houve maior precisão (LABBE; MICHAUD, 2014).

O trabalho descrito em “*3D Mapping with an RGB-D camera*” utiliza o sensor kinect sem o auxilio de outros sensores. Essa proposta controla a quantidade de características armazenadas para detecção de “*Loop closure*” no decorrer do mapeamento 3D através de uma estratégia composta por três etapas. Primeiro é realizada a transformação das posições anteriores, posteriormente para reduzir o erro, é feita a procura por “*Loop closure*” no vizinhos do grafo de posições dos *frames* anteriores. É calculada através da árvore geradora mínima e são eliminados os predecessores por um limiar “*n*” para evitar duplicações. Para encontrar *Loop closure* em larga escala é utilizada a seleção aleatória dos quadros chave. Por considerar ambientes internos de pequena escala esta abordagem apresenta bons resultados.

A visão geral do sistema é ilustrada na Figura 3.3. Cabe ressaltar, que os melhores resultados obtiveram uma taxa de 15 fps e diante da alteração do cenário houve uma diminuição desta taxa ficando em torno de 5 fps.

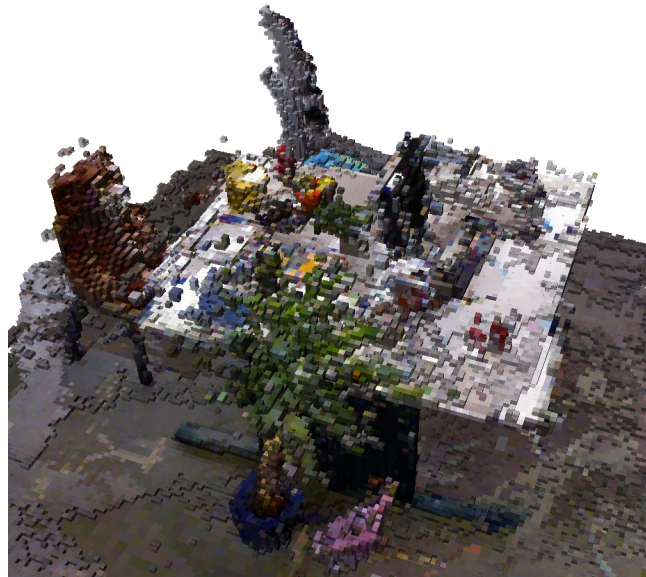
Figura 3.3 – Visão geral do sistema RGB-D SLAM.



Fonte: (ENDRES et al., 2014).

Como apresentado por (ENDRES et al., 2014), a visualização do mapeamento 3D é através de OctoMap como ilustrado na Figura 3.4. O uso de OctoMap é justificado pela redução de armazenamento em memória. Porém como citado no mesmo artigo, OctoMap exige mais recursos computacionais para a geração do mapeamento tridimensional.

Figura 3.4 – Visualização do Mapeamento 3D através de OctoMaps.



Fonte: (ENDRES et al., 2014).

3.0.4 Considerações

Os trabalhos apresentados neste capítulo, em sua maioria apresentam a implementação baseada na biblioteca *Robot Operating System* (ROS) que necessita de troca de mensagens entre os diferentes módulos de implementação. Embora tenha suporte para embarcar os algoritmos em diferentes modelos de *hardwares* robóticos, possui problemas de incompatibilidade entre diferentes versões da biblioteca.

O algoritmo estado da arte proposto por (ENGEL; STUCKLER; CREMERS, 2015), possui uma abordagem em que o mapa de disparidades é parte do algoritmo de cálculo da odometria visual, impossibilitando a geração de um mapa denso. Além disso, como essa proposta executa a extração de características mais evidentes, é possível que esse não seja adequado para automatização de veículos aéreos ou mesmo terrestre, pois diante de texturas homogêneas a abordagem tende a falhar. A dependência entre as etapas que constitui o mapeamento pode inviabilizar a alteração do algoritmo do mapa de disparidades.

No momento da elaboração deste trabalho as abordagens que realizavam o mapeamento 3D de ambientes não disponibilizavam seus códigos ou mesmo apresentavam com clareza de detalhes as implementações para estudos ou melhorias das técnicas. A reconstrução tridimensional é uma importante etapa para visualização das fragilidades dos métodos de geração dos mapas de disparidades e de cálculo de odometria visual. Além disso, a união de diferentes eta-

pas para construção do mapeamento 3D possibilita do surgimento de novas aplicações ou problemas de pesquisas. Neste contexto, é necessária a existência de um *framework* que apresente com simplicidade e clareza as etapas que constitui o mapeamento tridimensional de ambientes, que ainda seja modular a ponto de viabilizar a inserção de novos métodos e depender o minimamente de bibliotecas ou *softwares* auxiliares. O *framework* também deve necessariamente atingir resultados compatíveis as técnicas mais atuais ou ainda supera-las.

Este trabalho propõem um *framework* para o mapeamento 3D utilizando somente as bibliotecas *OpenCV* e *PCL*. A proposta contempla a possibilidade de seleção de diferentes algoritmos para a etapa de correspondência, viabilizando a geração de um mapa denso. O software também é modular e segue descrito em detalhes no Capítulo 4.

4 SISTEMA PROPOSTO

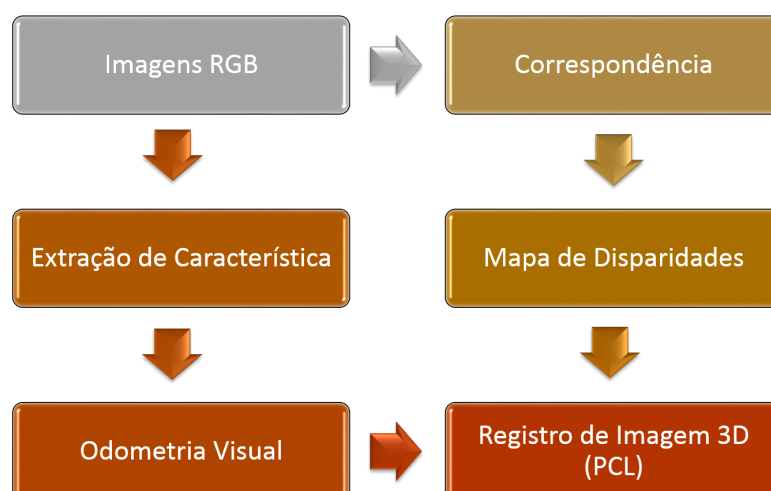
Neste capítulo será apresentada inicialmente uma visão geral de alto nível das principais etapas do sistema de mapeamento 3D desenvolvido. Posteriormente será descrita a arquitetura e os algoritmos. Também são apresentadas as bibliotecas que viabilizaram o desenvolvimento e a descrição das quatro abordagens utilizadas para a etapa de correspondência. Por último serão mostradas as telas de saída do sistema.

4.1 Visão Geral

A proposta deste trabalho consiste no desenvolvimento de um sistema que possa ser utilizado como *framework* para o mapeamento 3D de ambientes externos baseado em estereoscopia. São utilizados 4 métodos disponíveis na literatura para a etapa de correspondência, onde é realizada uma avaliação de desempenho durante a reconstrução tridimensional. Para o cálculo da odometria visual é aplicada a abordagem proposta por (GEIGER; ZIEGLER; STILLER, 2011) apresentada na Seção 2.11.

Para uma melhor compreensão da construção do sistema optou-se por organizar um diagrama de blocos conforme ilustra a Figura 4.1. O diagrama é constituído pelas etapas que consolidam as principais funcionalidades do sistema.

Figura 4.1 – Visão geral do sistema desenvolvido organizado em blocos.



Conforme a Figura 4.1 a entrada do sistema prevê um par de imagens que podem ser adquiridas através de duas câmeras ou de uma base de dados. Posteriormente, considerando um sistema calibrado, é realizado o procedimento de correspondência para geração do mapa de disparidades. A extração de características é realizada para possibilitar o cálculo da Odometria Visual. Como etapa final é gerada uma nuvem de pontos 3D que é computada com a utilização da informação do mapa de disparidades e posicionada pelo cálculo de odometria visual.

Para a correspondência foram utilizadas as abordagens denominadas, *Block Match* (BM), *Semiglobal Matching* (SGM) e BM com o processamento realizado por meio de *Graphics Processing Unit* (GPU) que será chama de BM GPU, essas técnicas estão implementadas na biblioteca *OpenCV* utilizada neste trabalho. O algoritmo *Efficient Large-scale Stereo* (ELAS) (GEIGER; ROSER; URTASUN, 2010) também foi incorporado ao sistema, compondo dessa forma os métodos avaliados neste *framework*.

4.2 Algoritmo

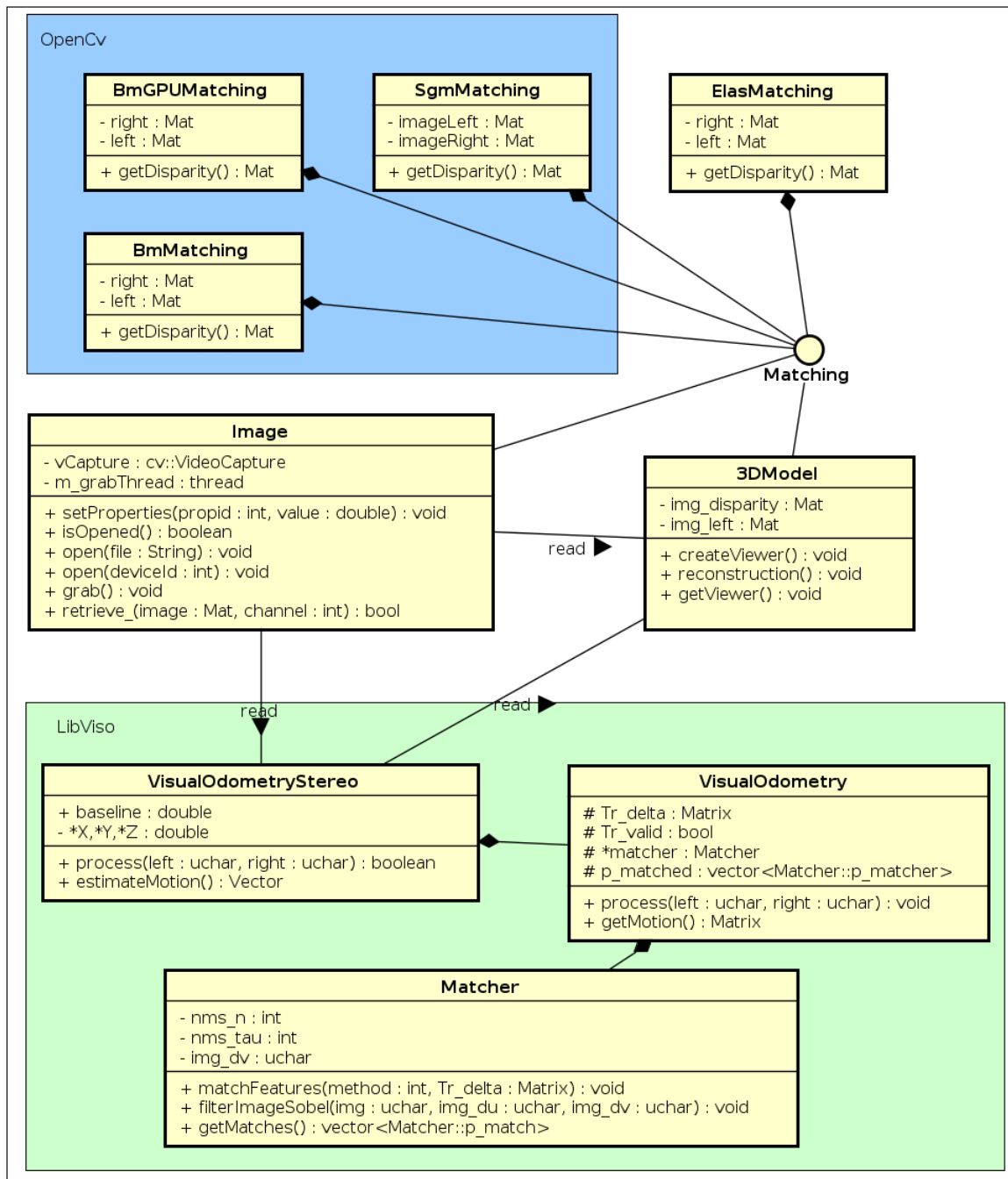
Esta seção apresenta a estrutura do sistema desenvolvido através de um diagrama de classes contendo métodos e atributos que compõem o sistema. Segue ainda a exposição de trechos de pseudocódigos de algumas partes relevantes da implementação.

A Figura 4.2 esboça o diagrama das classes implementadas neste trabalho. As classes foram concebidas considerando a idéia de *framework*, onde a estrutura geral prioriza a facilidade de implementação de novos módulos para o sistema ou ainda a alteração ou inserção de métodos tanto de correspondência como para o cálculo da odometria visual.

Conforme ilustrado na Figura 4.2, as classes *BmGPUMatching*, *BmMatching* e *Sgm-Matching* foram escritas com o objetivo de organizar internamente a chamada das respectivas classes que tratam dos métodos de correspondência implementados pela biblioteca *OpenCv*. Dessa maneira, essas classes herdaram a interface *Macthing* para padronizar a chama ao método *getDisparity*, ou seja, ao criar uma instância de qualquer uma das classes responsáveis por implementar o cálculo da correspondência, a chamada para o método que retorna o mapa de disparidades é o mesmo. Além disso, representa uma facilidade para testar as diferentes abordagens de correspondência, onde as peculiaridades de cada um dos métodos é tratado internamente pela classe, ficando dessa forma mais transparente a sua utilização e também tornando acessível a inserção de novas técnicas de correspondência.

A classe *ElasMatching* foi construída para tratar a configuração e utilização do método

Figura 4.2 – Diagrama de classes do sistema.



Fonte: Autoria própria.

de correspondência proposto por (GEIGER; ROSER; URTASUN, 2010). Como as demais classes que tratam da geração do mapa de disparidades, essa herda a interface *Matching*.

As imagens de *datasets* ou adquiridas de câmeras são configuradas através da classe *Image*. Para isso, a classe tem como responsabilidade inserir valores para os atributos de configuração de resolução com o método *setProperties*, abrir as câmeras ou *datasets* (método *open*)

e sincronizar o par de imagens com *grab* e com método reescrito *retrieve*. A classe foi desenvolvida para concentrar tudo que se refere a configuração e leitura das imagens.

As classes *VisualOdometry*, *Matcher* e *VisualOdometryStereo* são classes da biblioteca LibViso e não sofreram modificações. A classe *VisualOdometry* implementa a classe *Matcher* que realiza a extração e correspondência de características entre diferentes *Frames*. A classe *VisualOdometryStereo* estende a classe *VisualOdometry* e trata de implementar métodos para estimar movimento através de par de *frames*.

Por fim, a classe *3DModel* contempla métodos para geração e visualização da imagem tridimensional do mapeamento. Nesse sentido a biblioteca PCL é utilizada nessa classe, que é estendida para realizar a escrita de métodos adequados para a inserção das nuvens de pontos como descrito no pseudocódigo 1.

Algorithm 1 Reconstrução 3D

```

1: procedure RECONSTRUCAOPCL(RGB_ESQ, DISP_MAPS, POSE)
2:   ponto : pontoXYZRGB
3:   while rgb_esq.linhas do
4:     while rgb_esq.cols do
5:       disparidade = disps_maps[linhas][cols]
6:       ponto.X ← (rgb_esq.X - cx) * baseline / disparidade
7:       ponto.Y ← (rgb_esq.Y - cy) * baseline / disparidade
8:       ponto.Z ← (f * baseline) / disparidade
9:     end while
10:  end while
11:  reposicionaPontos(ponto, pose)
12:  AdPontos(ponto)
13: end procedure

```

O pseudocódigo 1 apresenta o algoritmo de construção do modelo tridimensional e a operação de transformação aplicada a cada *frame* computado pelos algoritmos de odometria visual.

Filtros são aplicados sobre a nuvem de pontos para cada *frame* reconstruído, corrigindo assim as falhas do cálculo do mapa de disparidades, como regiões ruidosas. O filtro varre os pontos em um determinado raio (configurável) e elimina os pontos que são superior a um limiar de vizinhança, também parametrizável como apresentado no pseudocódigo 2.

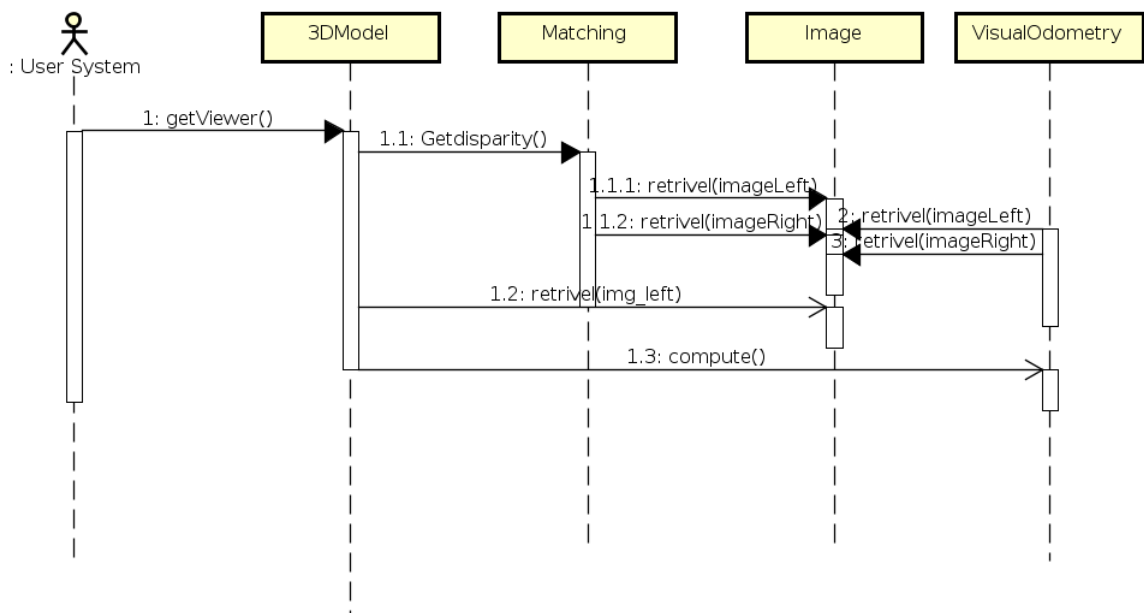
Algorithm 2 Filtragem para nuvem 3D

```

1: procedure FILTER3DPOINTS(REF PTR POINTCLOUD, MINRADIUS, MAXRADIUS)
2:   RadiusOutliersRemoval = newrmOutliers
3:   rmOutliers.SetInputCloud(pointCloud)
4:   rmOutliers.SetRadius(maxRadius)
5:   rmOutliers.SetMinNeighborsInRadius(minRadius)
6:   rmOutliers.filer(*pointCloud)
7: end procedure
  
```

Um diagrama de sequência de alto nível é ilustrado na Figura 4.3 e apresenta os passos de execução do sistema de mapeamento. Por meio desse é identificável a troca de mensagens entre as classes propostas no diagrama de classes da Figura 4.2. Para apresentar maiores detalhes da implementação, segue ainda um pseudocódigo do ciclo de execução principal do sistema. Destaca-se, a utilização de *threads* para que seja viabilizada a interação com a visualização da nuvem de pontos simultaneamente a inserção de novos pontos. O uso de *threads* proporciona a aplicação do conceito de processo leve, viabilizando a execução desse por diferentes núcleos do processador.

Figura 4.3 – Diagrama de sequência.



Fonte: Autoria própria.

A implementação de uma única *thread* para o cálculo de odometria visual e o mapa de disparidade se justifica pelo fato de buscar simplificar o procedimentos de implementação,

pois para tratar estes métodos em diferentes *threads* é necessária a etapa de sincronização. Caso contrário pode ser gerada uma matriz de pose sem a existência de um mapa de disparidade referente a profundidade do *frame* que deveria sofrer a transformação de rotação ou translação.

Algorithm 3 Fluxo de iteração principal

```

1: while verdadeiro do
2:   threadCalcDisp_Odometria(objCorrespondencia, objOdometriaVisual)
3:   threadReconstrói3D(objModelo3D)
4:   threadVisualiza3D(objModelo3D)
5: end while

```

O pseudocódigo 3 representa a execução do sistema onde os principais processos são executados em *threads*. Assim, a geração e manipulação da nuvem de pontos são implementadas em *threads* separadas e o cálculo do mapa de disparidades e da odometria são processados em uma mesma *thread*.

4.3 Ferramentas Utilizadas

Como ferramentas de *software* utilizadas nesta abordagem é importante destacar as bibliotecas *Open Source Computer Vision (OpenCV)* e *Point Cloud Library (PCL)*, com objetivo de viabilizar a aplicação dos principais algoritmos base para tratamento e processamento de imagens e a visualização do ambiente 3D através de uma nuvem de pontos respectivamente.

4.3.1 *OpenCV*

A biblioteca *OpenCV* possui a implementação de uma vasta quantidade de algoritmos para aplicação de técnicas de visão computacional. Por meio dessa biblioteca é possível realizar a captura das imagens através de câmeras ou ainda por meio da leitura de imagens providas de *datasets*. Além disso, a biblioteca foi utilizada principalmente para a etapa de calibração e retificação, por possuir a implementação dos métodos conforme mencionado nas Seções 2.1.5.1, 2.1.5.2 e 2.6.

Para o contexto deste trabalho, a biblioteca *OpenCV* proporcionou a realização de testes de diferentes algoritmos para a geração do mapa de disparidades, visto que a biblioteca possui a implementação de métodos BM, BM GPU e SGM.

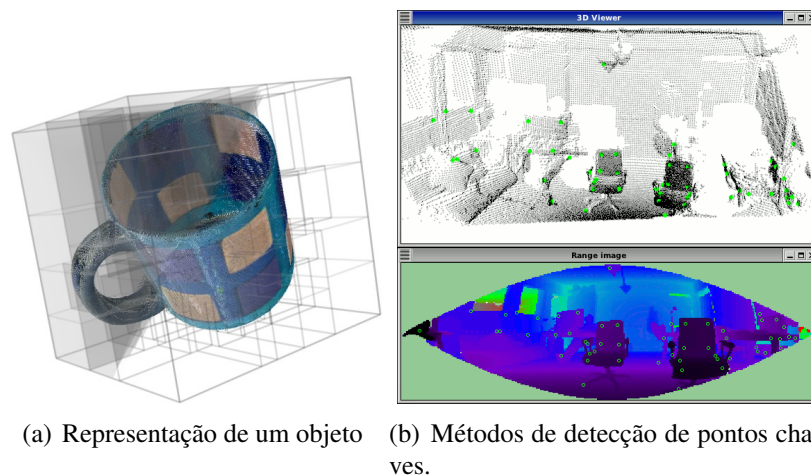
4.3.2 PCL

A biblioteca *Point Cloud Library* (PCL) proporciona a geração de uma nuvem de pontos para representação de imagens tridimensionais. Esta é largamente utilizada em abordagens que visam processamento em tempo real (ENDRES et al., 2012), (GEIGER; ZIEGLER; STILLER, 2011), (ENGEL; STURM; CREMERS, 2013) e (ENGEL; SCH; CREMERS, 2014). Entre as vantagens de sua utilização está o fato de ser de código aberto, baixo custo computacional e fácil criação e manipulação de representações tridimensionais.

PCL é baseada na linguagem C++, também utilizada para o desenvolvimento desta proposta. Possui um desenvolvimento contínuo baseado em colaboração de várias universidades entre elas Stanford, Freiburg e Washington (RUSU; COUSINS, 2011).

Embora a biblioteca possua muitas funcionalidades para tratamentos de dados tridimensionais, a principal aplicação para este trabalho é a visualização do mapeamento de ambientes tridimensionais e filtro para remoção de pontos rejeitados (*outliers*).

Figura 4.4 – Exemplo da biblioteca PCL.



Fonte: (HIRSCHMULLER, 2008).

A Figura 4.4 ilustra algumas das aplicações da biblioteca PCL que engloba desde de métodos simples de criação e visualização de nuvens de pontos até métodos de filtragem de informação, extração de características e reconhecimento de padrões.

4.4 Semi-Global

O método semi-global proposto por (HIRSCHMULLER, 2008) é utilizado neste trabalho. Métodos Semi-globais são uma alternativa a métodos locais e globais conforme descrito na Seção 2.1.5.3, esses apresentam um balanço entre desempenho e custo computacional.

O método calcula o custo de disparidade hierarquicamente por informação mútua. A agregação de custo é realizada por aproximação de uma função de energia global através de otimizações por partes para todas as direções da imagem. A Computação da Disparidade é feita assim como em métodos locais com o “*Winner takes all*” e otimização por refinamentos como checagem e interpolação de *subpixel*. Além disso, otimizações incluem filtro sobre picos, seleção de consistência de intensidade da disparidade.

Assim, mais especificamente o algoritmo prevê como métodos para as etapas:

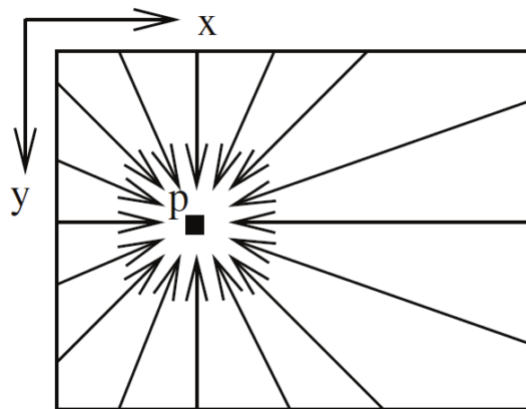
- Custo de correspondência -Informação Mútua Hierarquica; -Operação subsequente consiste na convolução.
- Agregação de custos

O cálculo ponto a ponto é geralmente ambíguo e correspondências falhas podem facilmente ter custo menor que corretas, devido ao ruído. Portanto, uma restrição adicional é adicionada para suportar suavidade penalizando de disparidades vizinhas. O custo ponto a ponto e as regras de suavidade são expressadas definindo a energia $E(D)$ que depende da imagem de disparidade D .

A novidade dessa abordagem é tratar a disparidade não somente em uma direção ou linha mais sim igualmente para todas as direções. O custo agregado (suavizado) $S(p, d)$ para um pixel p e a disparidade d é calculada pela soma dos custos de todos os caminhos de custo mínimo 1D que terminam em pixels p a disparidade d , conforme mostrado na Figura 4.5.

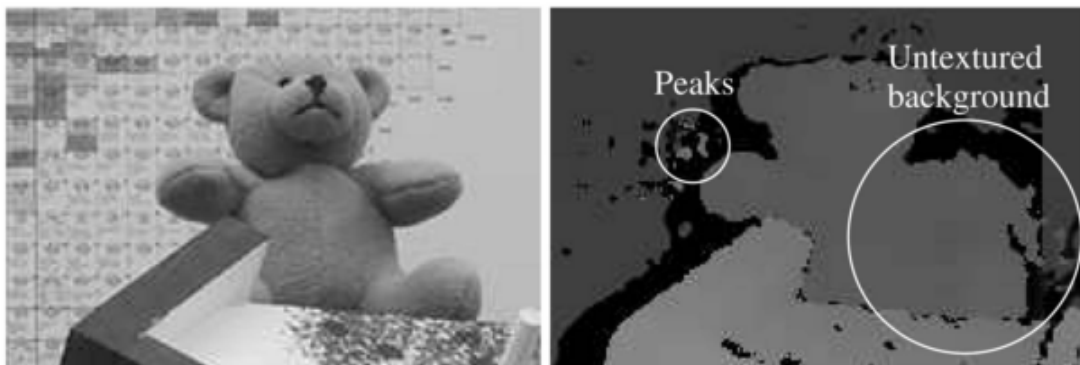
- O cálculo de disparidades é semelhante aos métodos locais, selecionando o *pixel* correspondente que represente o custo mínimo, ou seja, o mais próximo. *Outliers* são removidos por filtro de média considerando um janela de tamanho 3×3 .

Figura 4.5 – Dezesseis caminhos para o refinamento da correspondência utilizados pelo SGM.



Fonte: (HIRSCHMULLER, 2008).

Figura 4.6 – Possíveis regiões de picos e problemas de fundo sem textura.



Fonte: (HIRSCHMULLER, 2008).

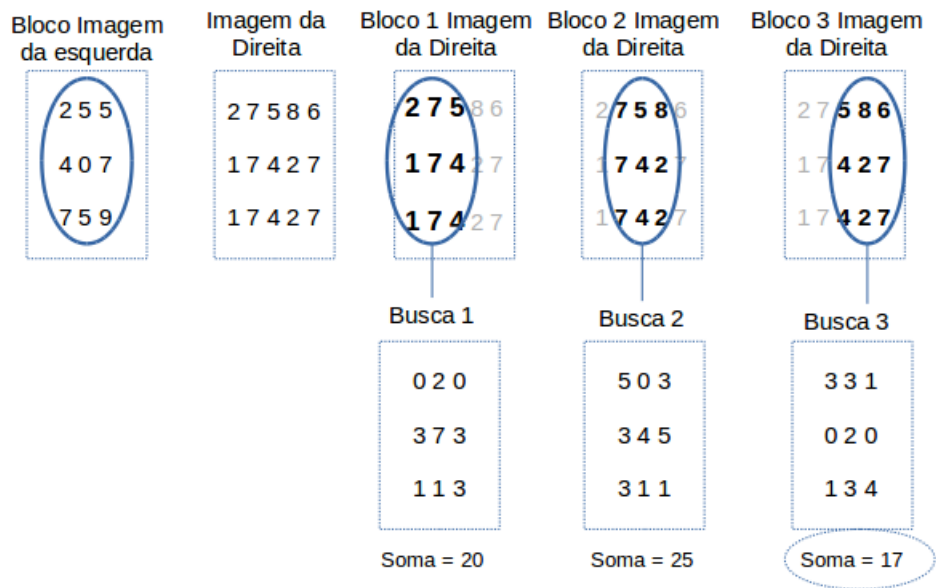
A Figura 4.6 ilustra possíveis problemas de *outliers* que são tanto regiões de picos ou problemas com regiões com fundo sem textura que podem ser resolvidos pelo filtro de média.

4.5 Block Match

O algoritmo *Block Match* (BM) implementado na biblioteca *OpenCV* é baseado no trabalho de (KONOLIGE, 1998) intitulado “*Small Vision Systems: Hardware and Implementation*”. Este é um método considerado clássico na literatura diante do problema de correspondência. Basicamente uma das etapas que compreendem a proposta é a correlação entre pequenas regiões do par de imagens. O valor que representa a correlação de cada *pixel* é determinado, assim a melhor correspondência entre a disparidade esquerda e direita é endereçada. Nesse

sentido, implementa a soma das diferenças absolutas (SAD), como medida de similaridade de blocos entre as imagens. Basicamente este método considera uma matriz de tamanho a ser definido de forma parametrizada. No exemplo da Figura 4.7, o bloco é configurado como 3X3, e imagem da esquerda é usada como referência. Dessa forma, é buscada melhor correspondência na imagem da direita diante do bloco da imagem da esquerda.

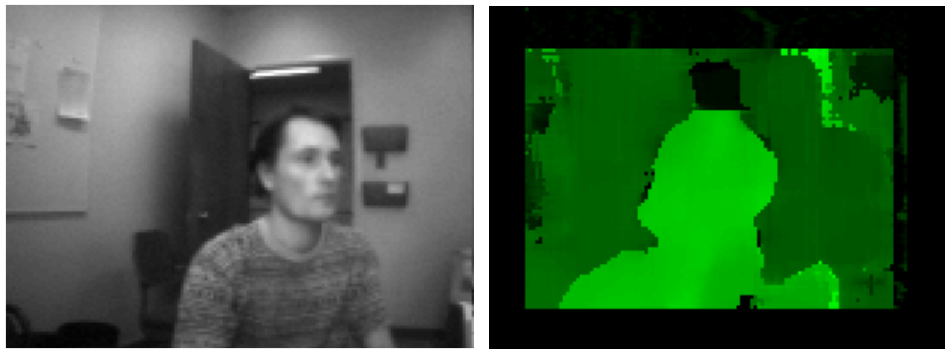
Figura 4.7 – Exemplo de aplicação de SAD em um problema de correspondência estéreo.



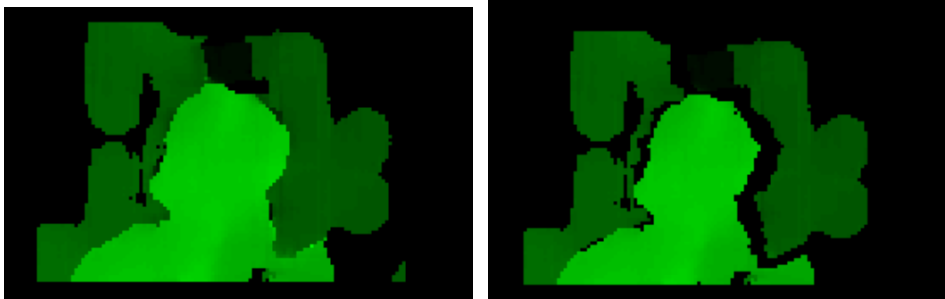
Fonte: Autoria própria.

Como ilustrado na Figura 4.7 é realizado basicamente a subtração ponto a ponto em cada bloco, resultando em blocos ilustrados com legenda de "Busca 1", "Busca 2" e "Busca 3". Posteriormente é realizada a soma de todos os valores pertencentes aos blocos resultantes, o menor valor final representa o bloco mais similar. Nesse exemplo, o bloco da busca 3.

Figura 4.8 – Algoritmo BM proposto por Konolige (1998).



(a) Imagem da esquerda do sistema estereo (Escala de cinza). (b) Imagem da disparidade por correlação.



(c) Aplicação do filtro de textura.

(d) Filtro de textura aplicado por entre esquerda e direita e ao direita esquerda.

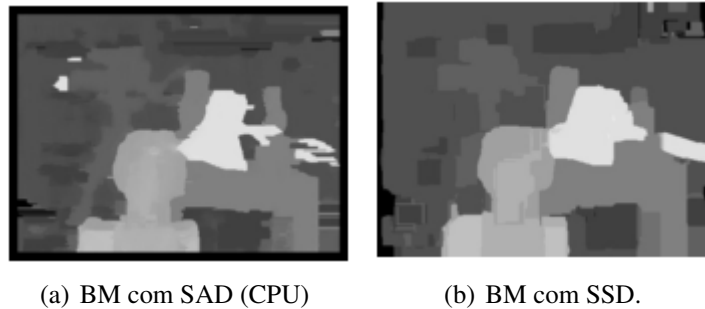
Fonte: (GEIGER; ROSER; URTASUN, 2010).

A Figura 4.8 ilustra o procedimento do cálculo da disparidade e as saídas baseadas nos filtros pós-processamento baseado em *Laplacian of Gaussian*.

4.6 Block Match (GPU)

A implementação do método *Block Match* disponível na *OpenCv* para processamento em GPU, implementa a Soma dos Quadrados das Diferenças (SSD) diferentemente do método proposto por (KONOLIGE, 1998) que usa SAD. A diferença ainda pode ser maior ao considerar o fato que os parâmetros de configuração como tamanho da janela para os filtros de pré e pós-processamento são inexistente, ficando todos estáticos na implementação em *OpenCv* para GPU. Tais diferenças implicam na construção de mapas de disparidade diferentes para a abordagem em GPU em comparação com a CPU disponíveis em *OpenCv* conforme ilustra a Figura 4.9.

Figura 4.9 – Comparativo entre BM (CPU) e BM (GPU).



Fonte: (SCHARSTEIN; SZELISKI, 2002).

Para as demais etapas como algoritmos de filtragem as implementações possuem as mesmas abordagens. A proposta SSD tende a gerar mapas de disparidades mais densos e menos propenso a ruídos porém apresentam maior custo computacional (SCHARSTEIN; SZELISKI, 2002).

4.7 ELAS

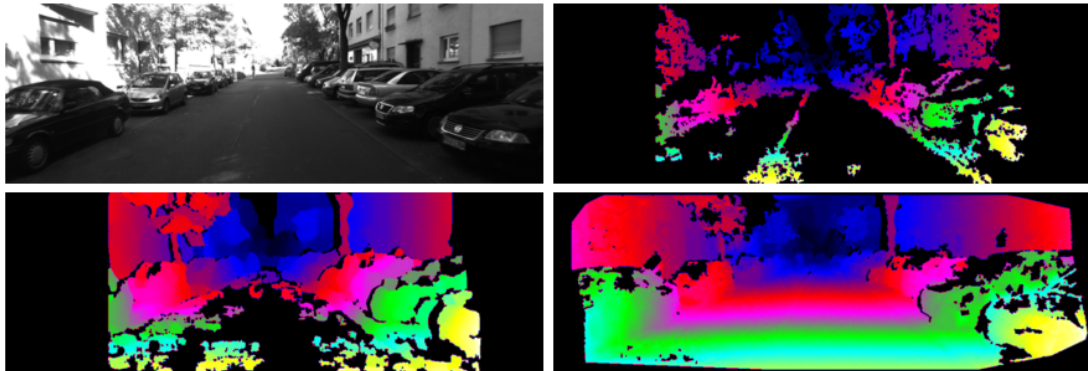
A proposta de (GEIGER; ROSER; URTASUN, 2010) de título “*Efficient Large-Scale Stereo Matching*” utiliza um modelo probabilístico generativo. Assim é construída uma prévia sobre o espaço de disparidade através de uma triangulação em conjunto com correspondências combinadas, chamada de ponto de apoio. Tal proposta não sofre com regiões de texturas homogêneas ou pobres, sendo baseada em abordagem *Bayesiana*. Define-se como etapas desse método primeiramente o cálculo da disparidade de um conjunto esparsos de pontos de suporte. As coordenadas de regiões de suporte da imagem são usadas para criar um *mesh 2D* através da triangulação de *Delaunay*. Uma prévia de incerteza é computada fazendo o processo ser eficiente por restrições de procura de regiões plausíveis.

É definido como pontos de suporte os *pixels* que podem ser combinados devido a sua singularidade e textura. Para a correspondência de pontos de suporte, uma região regular é formada por linhas horizontais e verticais do filtro de *Sobel* com uma configuração de 9x9.

Para atingir consistência, as correspondências são consideradas somente se atendem o cálculo entre direita e esquerda e também o inverso. Ainda para evitar ambigüidades, todos os pontos em que a relação entre a melhor e a segunda melhor disparidade exceder um dado limiar são removidos, eliminando dessa forma os pontos que apresentam valores de disparidades

diferentes de todos os pontos de suporte circundantes. Ainda para cobrir toda a imagem, pontos de apoio são adicionados nas bordas atribuindo disparidades dos pontos mais próximos.

Figura 4.10 – Comparativo entre métodos locais e ELAS. No topo e a direita da janela, um exemplo de processado de método local com uma janela de 5×5 , na parte inferior a esquerda, o método local com janela de 25×25 . Na parte inferior a direita o método ELAS.



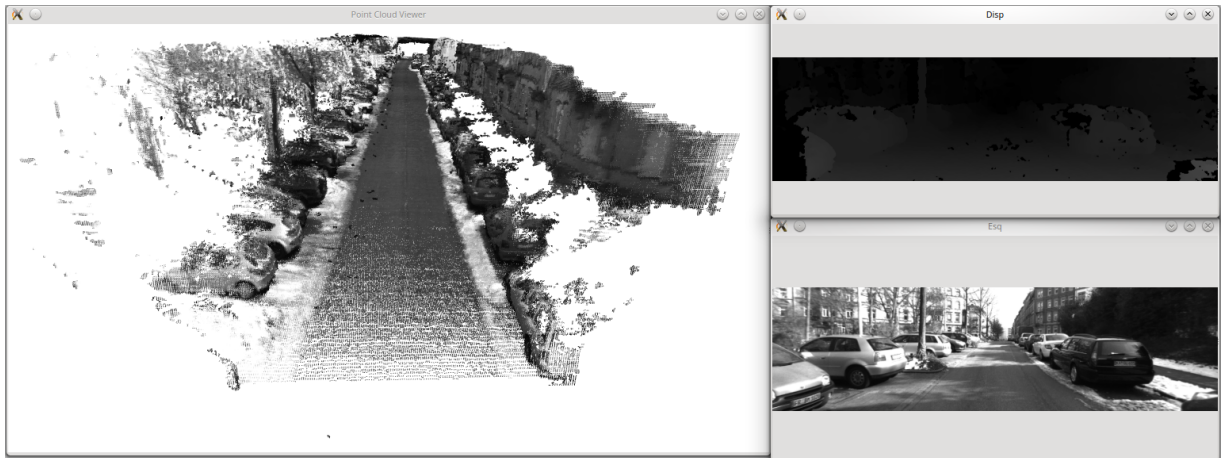
Fonte: (GEIGER; ROSER; URTASUN, 2010).

A Figura 4.10 exibe um comparativo entre o método ELAS e um método local clássico com diferente tamanhos de áreas de correspondência.

4.8 Visualização do Sistema

Para o sistema proposto, a visualização das informações devem seguir o exemplo da Figura 4.11, onde na janela da direita superior deve ser exibido o mapa de disparidades gerado por algum dos algoritmos. Na parte inferior deve ser mostrada a imagem câmera da esquerda do sistema estéreo, isso porquê na literatura essa é usada como imagem de referência. A parte Esquerda da imagem deve exibir o ambiente em coordenadas tridimensionais.

Figura 4.11 – Telas do sistema desenvolvido em execução.



Fonte: Autoria própria.

A telas devem ser atualizadas simultaneamente para que se possa ter maior clareza em tempo real se o mapeamento 3D é condizente com as imagens.

5 RESULTADOS

A apresentação dos resultados obtidos por este trabalho segue primeiramente com a descrição do *hardware* utilizado, logo é apresentado o banco de dados de imagens e *Ground Truth* do mapeamento. Finalmente são apresentados os testes e respectivamente os resultados alcançados.

5.1 Plataforma de Simulação

Para execução dos teste foi utilizado um *laptop* com processador I5 de 5ª Geração, com 8 GBs de memória RAM, SSD e placa de vídeo NVIDIA do modelo 740M de 2GBs de memória e suporte a programação em GPU. O sistema operacional usado foi o Linux Ubuntu versão 14.04.

5.2 Base da dados de imagens

A base de dados de imagens utilizada para os testes foi baseada na proposta de (GEIGER et al., 2013) adquirida através um conjunto de sensores que foram acoplados em um veículo automotivo, como ilustra a Figura 5.1. Assim, com essa configuração de sistema de aquisição de informação tanto de localização quanto de imagens, o veículo percorreu diferentes cenários urbanos da cidade de Karlsruhe na Alemanha. As imagens foram disponibilizadas na versão calibrada e retificada no formato png.

A configuração dessa plataforma permitiu a aquisição de imagens em RGB e escala de cinza além de um mapa de posições *Ground Truth* para trabalhos futuros que pretendam a utilização de um sistema baseado em imagem para realização de mapeamentos.

Para viabilizar a avaliação de sistemas de odometria visual, foram disponibilizadas sequências de imagens de diferentes cenários. Algumas delas possuem a trajetória de treinamento, ou seja informações de Sistema de Posicionamento Global (GPS) e Unidade de Medida Inercial (IMU), as demais sequência são usadas para submissão de resultados que devem ser avaliados automaticamente através de um endereço *online*¹.

Também foi utilizado neste trabalho um *dataset* proposto por (MENZE; GEIGER, 2015) para avaliação da qualidade dos mapas de disparidades. Esse *dataset* foi concebido por meio

¹ <http://www.cvlibs.net/datasets/kitti/evalodometry.php>

Figura 5.1 – Plataforma de aquisição de dados é equipada com quatro câmeras (sistema estéreo em escala de cinza e sistema para captura em RGB), um scanner laser 3D e um sistema que combina GPS/IMU.



Fonte: (GEIGER et al., 2013).

da edição de informações adquiridas de laser e de imagens através de *computer aided design* (CAD).

A utilização da base de imagens para exposição dos resultados deste trabalho justifica-se pelo fato de apresentar ambientes suficientemente grandes e desafiadores para a geração do mapeamento 3D. Essa base de imagens tem sido amplamente utilizada por trabalhos atuais no que se refere a mapeamento de ambientes reais. Dessa forma, a utilização desse *dataset* em específico é motivada diante do fato que outros *datasets* existentes na literatura não apresentam ambientes reais e tratam os problemas relacionados a qualidade dos mapas de disparidades em imagens sintéticas.

5.3 Análises e Resultados

Conforme descrito na Seção 4.1 correspondente a visão geral do sistema de mapeamento 3D proposto neste trabalho, quatro algoritmos de correspondência são avaliados com o objetivo de expor as características e verificar a possibilidade de atingir taxas de quadros melhores que a proposta inicial de (GEIGER; ZIEGLER; STILLER, 2011) que faz uso do algoritmo denominado ELAS. Os resultados apresentados diante do problema de mapeamento 3D por meio do

algoritmo ELAS foram de 3 a 4 fps, conforme (GEIGER; ZIEGLER; STILLER, 2011).

O mapa de disparidades gerado pelas quatro abordagens também são avaliados, para que dessa forma seja possível verificar a qualidade da informação de profundidade adquirida que pode influenciar na também na qualidade mapeamento tridimensional.

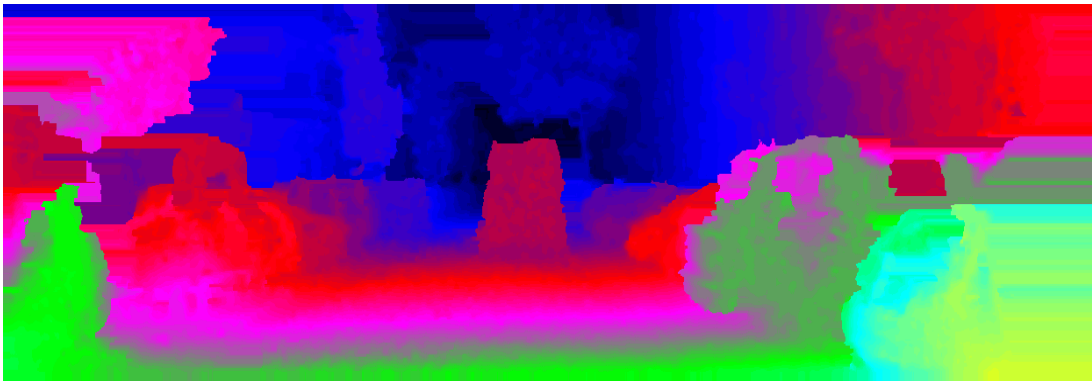
O *dataset* para avaliação dos mapas de disparidades possui um conjunto de imagens denominada *non-occluded*, isto é, regiões em que a correspondência esta dentro do domínio da imagem e ainda *occluded*, refere-se a todas as regiões da imagem que *ground truth* pode ter de informação, isso inclui regiões que foram mapeadas de pontos externos da imagem de domínio, ou seja, por outro ângulo ou ponto de visão.

As avaliações dos mapas de disparidades são realizadas por meio do processamento das imagens de *ground truth* em busca de *bad pixels* como segue a imagem 5.2. O conjunto de imagens sugeridos para realização dos testes contemplam 20 pares de imagens. Neste trabalho, foi realizada a avaliação sobre o conjunto completo, mas é apresentada somente uma das primeiras imagens da série de treinamento e a tabela que quantifica informações como: o percentual de erro geral (*all*), a percentual de erro de todas as imagens que apresentam um cenário estático (*all/est*), o percentual sobre o *ground truth non-occluded* e o *ground truth non-occluded* com conjunto estático. No presente trabalho, se refere a cenário estático considerando os que não apresentam de pessoas ou veículos em deslocamento.

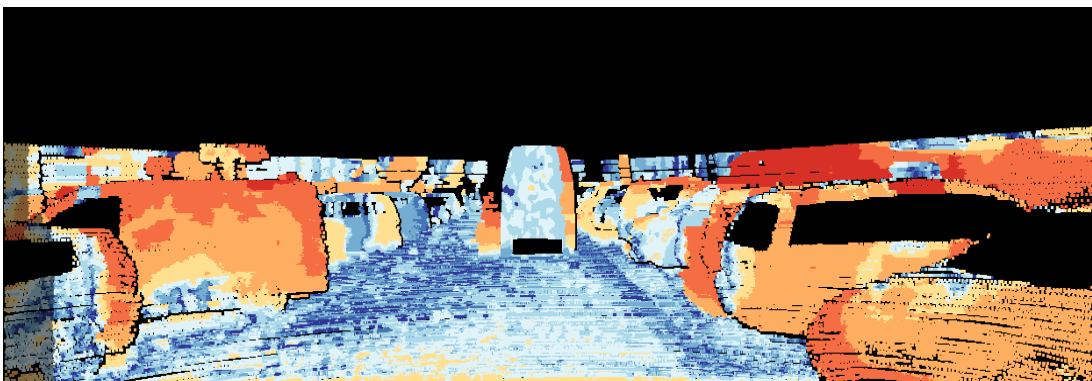
Figura 5.2 – Avaliação do mapa gerado por ELAS. A imagem do topo apresenta o cenário real, seguida da imagem do mapa de disparidades e a imagem que apresenta o erro.



(a) Imagem original.



(b) Imagem disparidade ELAS.



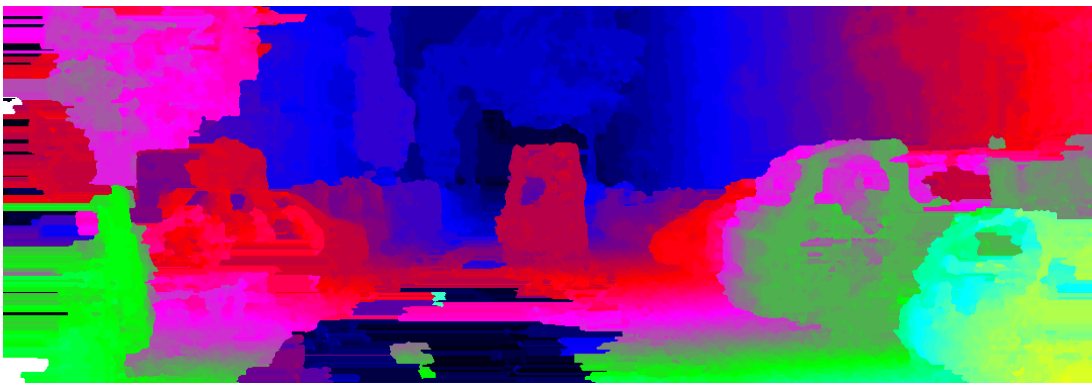
(c) Imagem do cálculo de erro.

Fonte: Autoria própria.

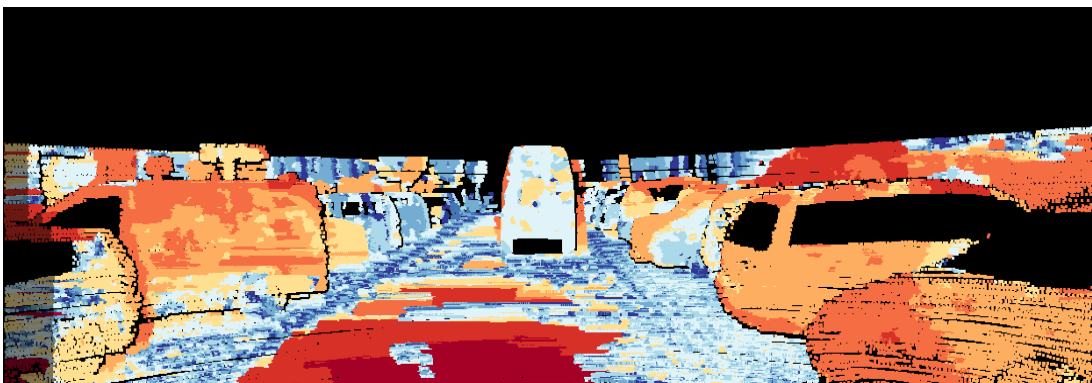
Figura 5.3 – Avaliação do mapa gerado por SGM. A imagem do topo apresenta o cenário real, seguida da imagem do mapa de disparidades e a imagem que apresenta o erro



(a) Imagem original.



(b) Imagem disparidade SGM.



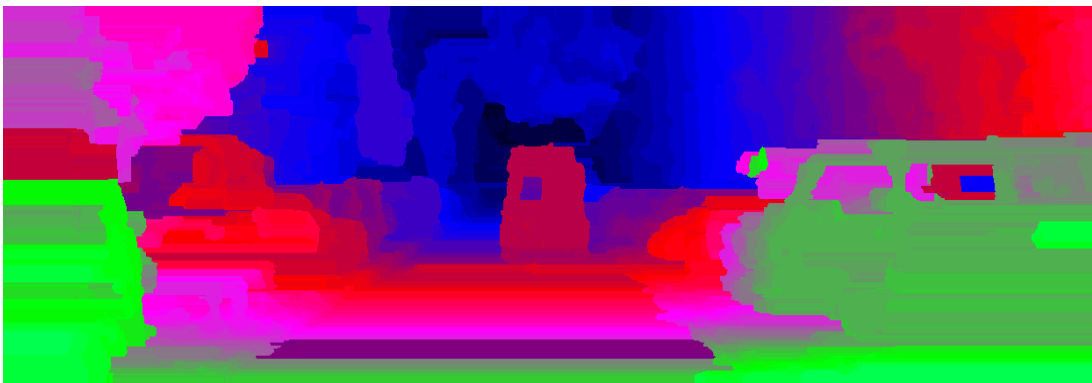
(c) Imagem do cálculo de erro.

Fonte: Autoria própria.

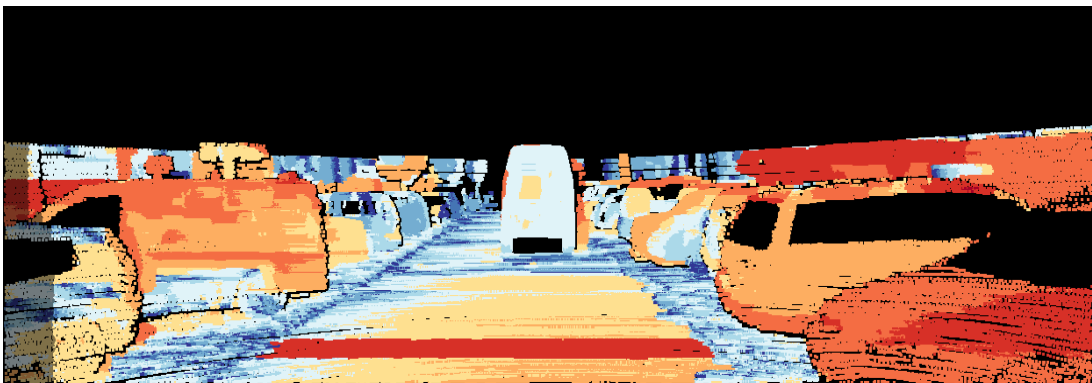
Figura 5.4 – Avaliação do mapa de disparidade gerado pelo BM.



(a) Imagem original.



(b) Imagem disparidade BM.



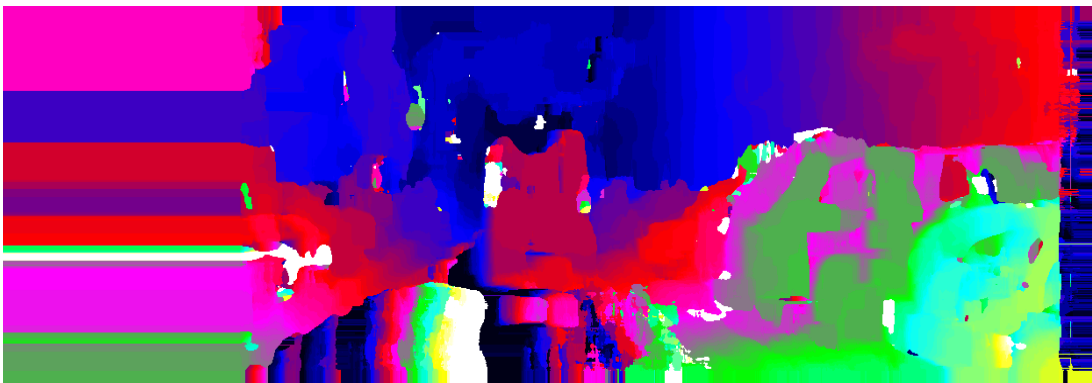
(c) Imagem do cálculo de erro.

Fonte: Autoria própria.

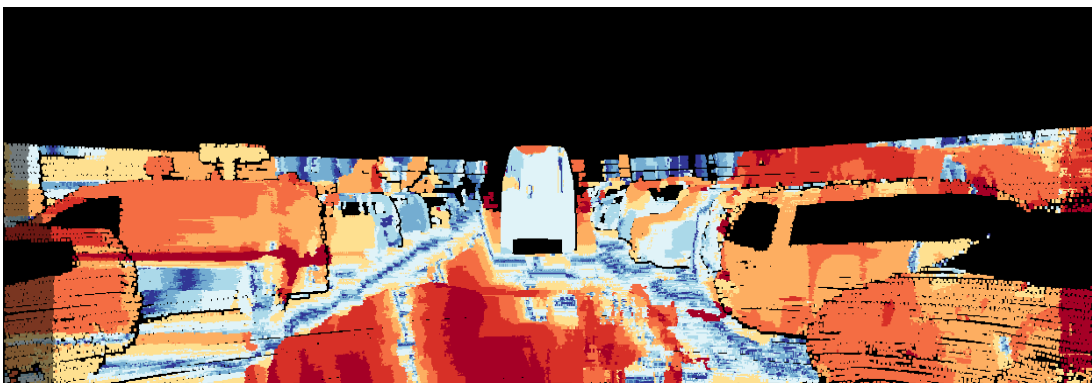
Figura 5.5 – Avaliação do mapa gerado por BM GPU. A imagem do topo apresenta o cenário real, seguida da imagem do mapa de disparidades e a imagem que apresenta o erro



(a) Imagem original.



(b) Imagem disparidade BM GPU.



(c) Imagem do cálculo de erro.

Fonte: Autoria própria.

Para todos os mapas de disparidade submetidos a avaliação são geradas imagens que apresentam em tonalidade azul as regiões validadas (corretas) diante da imagem de *ground truth* e as regiões que se aproximam da cor vermelha são pontos falhos. As Tabelas 5.1, 5.2 e 5.3 esboçam os dados quantitativos de D1-bg (disparidade para *background*) D1-fg (disparidade para *foreground*) e D1-all para ambos (MENZE; GEIGER, 2015).

Tabela 5.1 – Avaliação do mapa de disparidade (ELAS) (MENZE; GEIGER, 2015).

Error	D1-bg	D1-fg	D1-all
All / All	7.86 %	19.04 %	9.72 %
All / Est	5.25 %	12.30 %	6.36 %
Noc / All	6.88 %	17.73 %	8.67 %
Noc / Est	5.11 %	11.99 %	6.19 %

Os métodos SGM e ELAS apresentam dados bastante similares. Os resultados variam de acordo com o cenário analisado, mas em um contexto como o conjunto de imagens analisados, o método ELAS apresenta melhores resultados.

Tabela 5.2 – Avaliação do mapa de disparidade (SGM) (MENZE; GEIGER, 2015).

Error	D1-bg	D1-fg	D1-all
All / All	8.92 %	20.59 %	10.86 %
All / Est	4.45 %	13.24 %	5.86 %
Noc / All	7.62 %	18.81 %	9.47 %
Noc / Est	3.98 %	12.56 %	5.35 %

Os métodos BM e sua implementação em GPU apresentam os piores resultados em relação a qualidade tanto visualmente quanto quantitativamente. O método BM GPU ficou de fora dessa avaliação por apresentar erros extremos e discrepantes.

Tabela 5.3 – Avaliação do mapa de disparidade (BM) (MENZE; GEIGER, 2015).

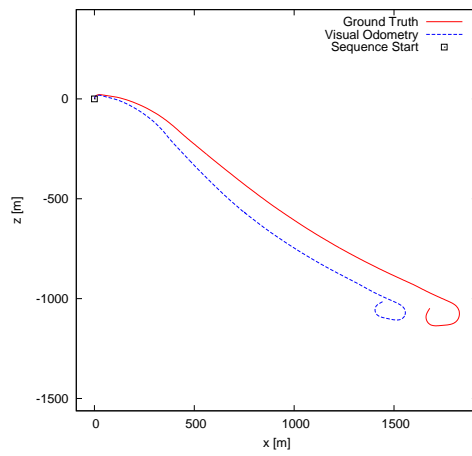
Error	D1-bg	D1-fg	D1-all
All / All	24.29 %	30.13 %	25.27 %
All / Est	12.11 %	16.52 %	12.94 %
Noc / All	23.76 %	29.11 %	24.64 %
Noc / Est	12.11 %	16.52 %	12.94 %

Para realização da avaliação da reconstrução tridimensional, foram selecionadas 5 sequências de imagens disponíveis no *dataset* de (GEIGER et al., 2013). A sequência 01 do *dataset* possui 2202 imagens contabilizando o par estéreo e apresenta o deslocamento do veículo sobre uma rodovia sem obstáculo, com poucos carros e uma trajetória sem curvas. A sequência 03 compõem um trajeto urbano com algumas curvas, poucos carros e totaliza 1602 imagens. A menor sequência é a 04 que totaliza 542 imagens e apresenta um cenário urbano com vários veículos trafegando. Os dois ultimo conjuntos de imagens apresentam 2202 imagens, são as sequências 06 e 07 que constituem um cenário com várias curvas e pouco dinamismo de carros e obstáculos. Essas sequências foram selecionadas por apresentar diferentes contextos e serem amplamente utilizadas por abordagens atuais na literatura no que se refere a algoritmos de odometria visual e reconstrução tridimensional. A Figura 5.6 ilustra os diferente mapas topológico desses cenários. Assim são apresentadas na cor vermelha as coordenadas do mapeamento *Ground Truth* adquirido do sensor de GPS e em azul o mapeamento realizado neste trabalho com a utilização da LibViso. Fica evidente que o método utilizado para o cálculo de odometria visual não é livre de erros, porém tem resultados aceitáveis sendo até mesmo aplicado em outros trabalhos, como em (NEGRE; BONIN-FONT; OLIVER, 2014) e (PI et al., 2014).

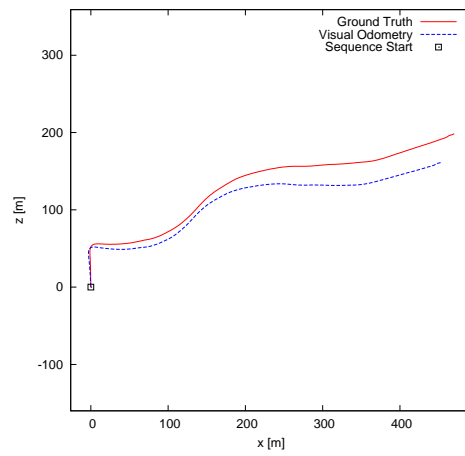
As 5 sequências de imagens selecionadas serviram para a realização de testes de reconstrução tridimensional com o objetivo de mensurar a taxa de *frames*, a quantificar de pontos gerados pela nuvem tridimensional após a conclusão do processo, contabilizar o tempo desprezado para o cálculo da odometria visual e também para o processo de construção da representação 3D por completa. Neste contexto as Tabelas 5.4 e 5.5 apresentam os resultados quantitativos para o processo de mapeamento 3D realizado pelo *framework* desenvolvido neste trabalho. A Tabela 5.4 apresenta os resultados sobre a resolução de 672X186 *pixels* e a Tabela 5.5 apresenta os dados do mapeamento em resolução de 336x93 *pixels*.

As Tabelas identificam informações como o método de correspondência, a sequência de imagens referida (*Dataset*), a taxa de *frames* (FPS), o tempo gasto para calcular a odometria visual de todo o mapa ("Tempo VO") com valores em milésimos de segundos, o tempo total para a conclusão do processo de geração do modelo 3D, a quantidade de pontos por *frame* e o total de pontos do modelo tridimensional.

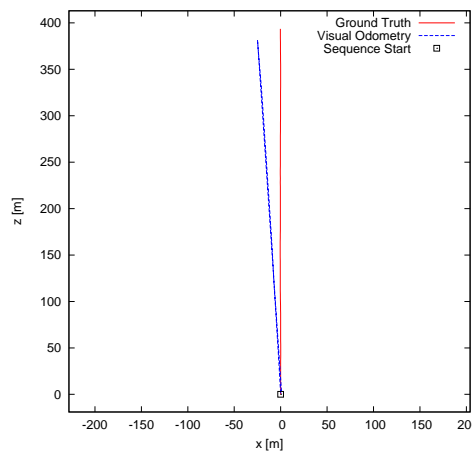
Figura 5.6 – Evaluation of Visual Odometry .



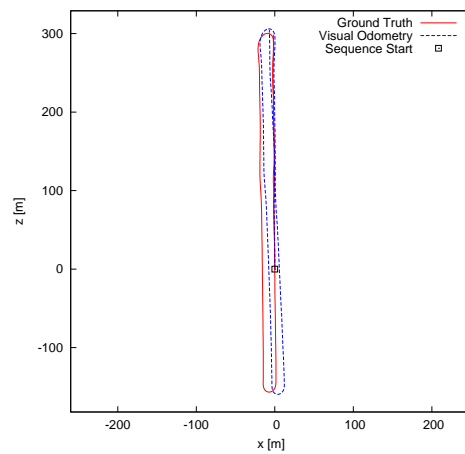
(a) Sequence 01.



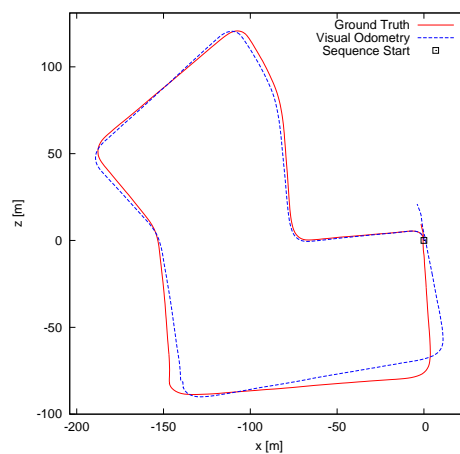
(b) Sequence 03.



(c) Sequence 04.



(d) Sequence 06.



(e) Sequence 07.

Fonte: Autoria própria.

Os testes realizados neste trabalho permitiram verificar que as taxas de fps do método ELAS atingiram entre 2.6 fps e 8.7 fps para as respectivas resoluções de 672x186 e 336x93.

Tabela 5.4 – Comparativo entre diferentes métodos de correspondência sobre a resolução de 672x186.

Método	Dataset	FPS	Tempo VO(ms)	Tempo Total(ms)	Ptos Frame	Total Ptos
sgm	1	3.10	290.326	291.627	34.644	21.279.548
bm	1	9.00	69.015	70.348	8.763	6.392.124
bmGPU	1	8.07	192.102	211.395	21.973	18.020.300
elas	1	2.59	616.849	618.046	30.027	30.659.206
sgm	3	2.83	225.742	228.198	66.908	20.728.694
bm	3	7.44	85.549	88.021	36.511	10.371.065
bmGPU	3	7.40	146.957	175.855	40.724	14.925.682
elas	3	2.29	587.060	588.404	55.260	31.664.134
sgm	4	3.20	59.733	61.085	30.049	6.014.096
bm	4	6.44	49.072	51.177	61.659	15.973.773
bmGPU	4	7.58	44.325	53.138	19.536	4.734.757
elas	4	2.51	154.789	155.931	22.730	8.765.338
sgm	6	2.91	299.684	302.643	49.699	24.849.403
bm	6	9.00	107.045	109.110	25.908	10.615.977
bmGPU	6	7.17	211.000	245.928	32.420	21.087.897
elas	6	2.58	590.500	591.768	36.579	35.025.037
sgm	7	2.91	293.698	295.703	63.468	26.364.301
bm	7	9.01	104.010	111.585	31.422	13.670.578
bmGPU	7	7.58	201.204	237.288	41.122	22.705.659
elas	7	2.65	584.188	585.269	45.503	47.697.220

Cabe salientar que para a exposição dos resultados adquiridos por (GEIGER; ZIEGLER; STILLER, 2011) foi utilizado um hardware melhor (core i7 CPU com 3.0 GHz). As melhores taxas de fps foram adquiridas pelo método BM, superando até mesmo a implementação de BM GPU. É necessário ressaltar que esses métodos apresentam implementações diferenciadas como descrito na Seção 4.6. Na Tabela 5.5 a quantidade de pontos tanto para o primeiro *frame* quanto o número total do algoritmo BM GPU é maior que as demais abordagens, isso porque o filtro para a nuvem pontos que remove informações ruidosas, foi removido para a realização dos testes, só assim foi possível gerar o mapa 3D, mesmo que com uma quantidade muito grande de ruído.

O método ELAS apresentou melhores resultados diante do método SGM para resoluções menores, o que não aconteceu com a utilização de resoluções maiores. Embora o método ELAS tenha sido proposto com a idéia de produzir resultados com resoluções maiores.

Tabela 5.5 – Comparativo entre diferentes métodos de correspondência sobre a resolução de 336x93.

Método	Dataset	FPS	Tempo VO(ms)	Tempo Total(ms)	Ptos Frame	Total Ptos
sgm	1	6.38	270.275	271.637	2.186	3.056.878
bm	1	9.35	179.344	180.824	648	624.597
bmGPU	1	7.08	13.777	15.968	9.705	904.564
elas	1	8.50	246.835	248.065	673	447.157
sgm	3	6.20	225.120	226.628	5.301	2.856.771
bm	3	8.88	185.694	186.980	2.651	1.174.102
bmGPU	3	8.03	168.946	170.926	9.705	8.108.099
elas	3	7.00	209.964	211.304	3.680	1.034.571
sgm	4	7.00	66.392	67.770	1.457	743.877
bm	4	8.93	310.131	46.698	195	45.371
bmGPU	4	7.43	174.215	171.918	9.705	8.108.099
elas	4	8.40	50.342	51.714	476	284.420
sgm	6	5.80	342.648	344.182	1.756	3.620.955
bm	6	9.25	227.644	229.219	513	1.292.602
bmGPU	6	7.50	181.422	184.830	10.025	10.774.075
elas	6	7.61	267.488	268.984	613	1.551.320
sgm	7	5.80	329.926	331.680	1.335	4.649.524
bm	7	9.43	238.924	240.412	210	2.034.869
bmGPU	7	8.15	166.452	168.518	9.705	8.108.099
elas	7	8.75	273.734	274.981	565	2.305.664

A Figura 5.7, exibe o mapeamento 3D da sequência 07 formado por uma nuvem de pontos, nesse são exibido os resultados dos algoritmos SGM, BM e ELAS. É possível notar que a abordagem SGM pode ser considerada como mais adequada para este cenário diante de uma melhor definição dos carros que compõem o ambiente.

O mapeamento 3D da sequência 01 é ilustrado na Figura 5.8, nesse são apresentados os resultados dos algoritmos SGM, ELAS, BM e BM GPU. É possível notar que nessa sequência de imagens a abordagem SGM e ELAS tiveram resultados muito similares. No entanto o método BM também apresenta resultados aceitáveis no sentido de possibilitar o entendimento da cena.

Figura 5.7 – Imagem tridimensionais da sequência 07.



(a) Imagem original.



(b) Mapeamento com SGM.

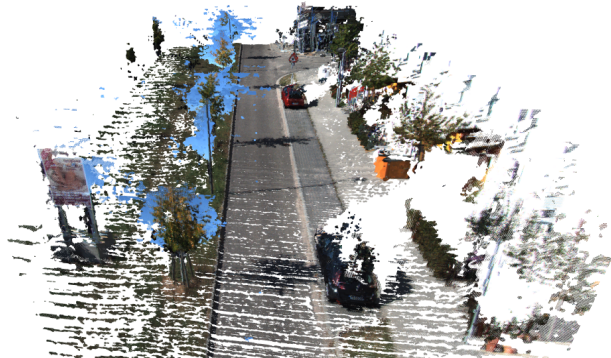


(c) Mapeamento com ELAS.

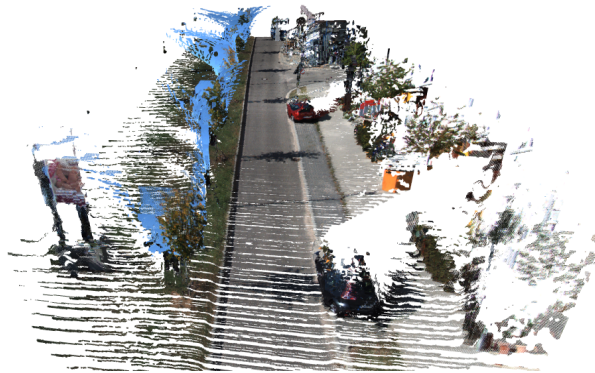


(d) Mapeamento com BM.

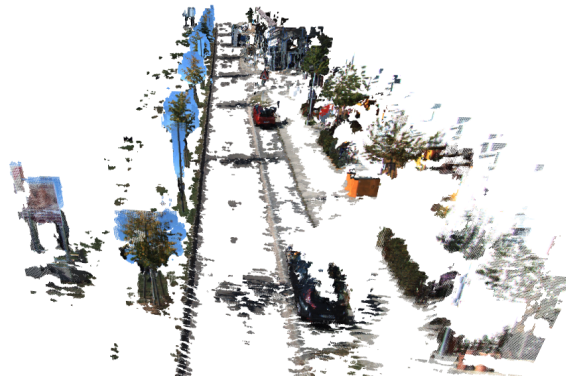
Figura 5.8 – Imagem tridimensionais da sequência 01.



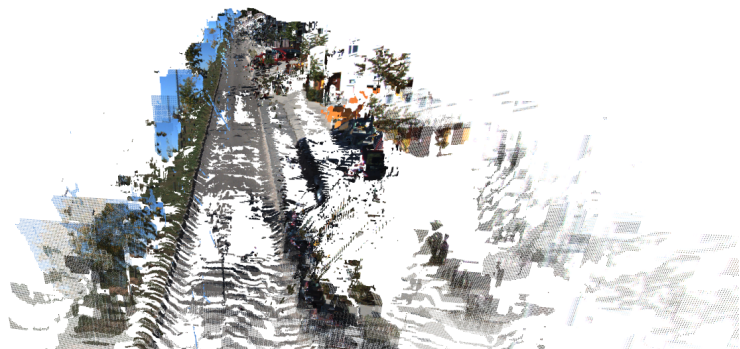
(a) Mapeamento com SGM.



(b) Mapeamento com ELAS.



(c) Mapeamento com BM.



(d) Mapeamento com BM GPU.

6 CONSIDERAÇÕES FINAIS

Neste trabalho foi desenvolvido uma arquitetura de software para realizar o mapeamento 3D de ambiente através de odometria visual e estereoscopia. Desse modo, para o problema de estereoscopia quatro diferentes algoritmos de grande relevância na literatura foram analisados considerando desempenho e robustez. Foi apresentado um algoritmo para o cálculo da odometria visual para atingir o objetivo de reconstrução de ambientes externos.

Testes com a utilização de *datasets* que abrangem vários ambientes urbanos foram realizados com a finalidade de avaliar de forma qualitativa e quantitativa. Mediante a análise qualitativa pode-se observar o modelo tridimensional sobre a aplicação das quatro técnicas de correspondência estéreo. A avaliação quantitativamente permitiu verificar a taxa de quadros por segundo obtida entre as diferentes técnicas de correspondência.

Os resultados obtidos diante dos testes, permitiram concluir que a abordagem de correspondência "BM" possui uma taxa de atualização de quadro por segundo (FPS) maior que os demais algoritmos analisados. Tais resultados foram melhores que os resultados da proposta original de mapeamento desenvolvida por (GEIGER; ZIEGLER; STILLER, 2011). Porém a qualidade da imagem tridimensional apresenta uma nuvem de pontos mais esparsa.

Portanto os objetivos inicialmente definidos foram alcançados, sendo esses a construção de uma arquitetura de *software* que pode ser utilizada como *framework*. Ainda foi atingido taxas de FPS melhores que a proposta definida em (GEIGER; ZIEGLER; STILLER, 2011). O *framework* esta disponível em um repositório *web*¹ e pode ser acessado e baixado.

6.1 Trabalho Futuros

Nesta seção serão apresentadas algumas alternativas de abordagens futuras para a proposta apresentada. Diante das possibilidades buscou-se organizar tendo em vista aperfeiçoamentos e também aplicabilidades.

6.1.1 Aperfeiçoamentos

O trabalho desenvolvido em questão apresentou alguns gráficos do cálculo da odometria visual com um dado erro aparente, resultado evidenciado diante de avaliações de banco de

¹ <https://bitbucket.org/cristianofs/svo3dmapping>

imagens com *ground truth* do mapa de posições. Para trabalhos futuros, é interessante prever o emprego de um algoritmo de otimização de gráfico de deslocamento, assim através da detecção de locais já visitados poderia ser reduzida a taxa de erros.

Melhorias na geração de mapas de disparidades principalmente no que se refere a desempenho, poderia ser uma boa aplicação para trabalhos a serem realizados, já que ficou provado que o mapa de disparidades interfere tanto no desempenho da aplicação como no resultado final do mapeamento 3D.

Fatores como ambientes externos com uma quantidade considerável de dinamismo também podem ser alvo de trabalhos futuros. Tais ambientes, se apresentam desafiadores tanto para a representação tridimensional da cena como para estimativa de movimento por algoritmos de odometria visual.

6.1.2 Aplicação

A arquitetura de software desenvolvida possibilita o mapeamento e localização 3D com uma taxa de FPS que tornam possível o mapeamento em tempo real mesmo que com um pequeno atraso. Diante disso, pode ser interessante para trabalhos futuros embarcar esta proposta em um dispositivo como um drone, dessa maneira seria possível realizar mapeamento de ambientes desconhecidos ou até em ambientes de difícil acesso. Essa aplicação poderia ser utilizada em missões militares ou até mesmo para mapeamento de áreas com fins arquitetônicos. Além disso, embarcar este sistema em um drone habilitaria a correção de erros provenientes do algoritmo de odometria visual, dados que os drones possuem em seu *hardware* dispositivos que fornecem informações de posicionamento como GPS, acelerômetro e magnetômetro, utilizados para o seu controle.

A aplicação na automação de dispositivos é outra possibilidade de trabalho, diante da facilidade de acessar as informações tridimensionais através das nuvens de pontos ou mapa de disparidades, seria possível detectar objetos próximos ao dispositivo prevendo possíveis obstáculos. Ainda diante do mapa 3D seria fácil encontrar um determinado caminho para alguma parte do cenário.

REFERÊNCIAS

- AHMED, M. F. **Development of a Stereo Vision system for Outdoor Mobile Robots**. 2006. Tese (Doutorado em Ciência da Computação) — University of Florida.
- BELLAVIA, F. et al. Robust selective stereo SLAM without loop closure and bundle adjustment. In: **Image Analysis and Processing–ICIAP 2013**. [S.l.]: Springer, 2013. p.462–471.
- BOUGUET, J.-Y. **Camera calibration toolbox for Matlab, example 5: stereo rectification algorithm @ONLINE**. 2015.
- BRADSKI, A. **Learning OpenCV, [Computer Vision with OpenCV Library ; software that sees]**. 1. ed..ed. [S.l.]: O'Reilly Media, 2008. Gary Bradski and Adrian Kaehler.
- CROWLEY, J. L.; BOBET, P.; SCHMID, C. Dynamic calibration of an active stereo head. In: **COMPUTER VISION, 1993. PROCEEDINGS., FOURTH INTERNATIONAL CONFERENCE ON. Anais...** [S.l.: s.n.], 1993. p.734–739.
- DANG, T.; HOFFMANN, C.; STILLER, C. Continuous Stereo Self-Calibration by Camera Parameter Tracking. **Image Processing, IEEE Transactions on**, [S.l.], v.18, n.7, p.1536–1550, July 2009.
- DUANE, C. B. Close-range camera calibration. **Photogrammetric Engineering**, [S.l.], v.37, n.8, p.855–866, 1971.
- ENDRES, F. et al. An evaluation of the RGB-D SLAM system. **Robotics and Automation (ICRA), 2012 IEEE International Conference**, [S.l.], v.1, n.c, p.1691–1696, 2012.
- ENDRES, F. et al. 3D Mapping with an RGB-D Camera. **IEEE Transactions on Robotics**, [S.l.], v.30, n.1, p.177–187, Feb 2014.
- ENGEL, J.; SCH, T.; CREMERS, D. LSD-SLAM: large-scale direct monocular slam. **Computer Vision–ECCV**, [S.l.], p.1–16, 2014.
- ENGEL, J.; STUCKLER, J.; CREMERS, D. Large-scale direct slam with stereo cameras. In: **INTELLIGENT ROBOTS AND SYSTEMS (IROS), 2015 IEEE/RSJ INTERNATIONAL CONFERENCE ON. Anais...** [S.l.: s.n.], 2015. p.1935–1942.

ENGEL, J.; STURM, J.; CREMERS, D. Semi-dense visual odometry for a monocular camera. **Proceedings of the IEEE International Conference on Computer Vision**, [S.l.], p.1449–1456, 2013.

GEIGER, A. et al. Vision meets Robotics: the kitti dataset. **International Journal of Robotics Research (IJRR)**, [S.l.], 2013.

GEIGER, A.; LENZ, P.; URTASUN, R. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In: CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). **Anais...** [S.l.: s.n.], 2012.

GEIGER, A.; ROSER, M.; URTASUN, R. Efficient Large-Scale Stereo Matching. In: ASIAN CONFERENCE ON COMPUTER VISION (ACCV). **Anais...** [S.l.: s.n.], 2010.

GEIGER, A.; ZIEGLER, J.; STILLER, C. StereoScan: dense 3d reconstruction in real-time. **IEEE Intelligent Vehicles Symposium, Proceedings**, [S.l.], n.Iv, p.963–968, 2011.

GONZALEZ, R.; WOODS, R. **Processamento de imagens digitais**. 1nd.ed. [S.l.]: Edgard Blucher, 2000.

HIRSCHMULLER, H. Stereo Processing by Semiglobal Matching and Mutual Information. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, [S.l.], v.30, n.2, p.328–341, Feb 2008.

KONOLIGE, K. Small vision systems: hardware and implementation. In: **Robotics Research**. [S.l.]: Springer, 1998. p.203–212.

LABBE, M.; MICHAUD, F. Online global loop closure detection for large-scale multi-session graph-based SLAM. In: INTELLIGENT ROBOTS AND SYSTEMS (IROS 2014), 2014 IEEE/RSJ INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2014. p.2661–2666.

LACROIX, S. et al. Rover self localization in planetary-like environments. In: ARTIFICIAL INTELLIGENCE, ROBOTICS AND AUTOMATION IN SPACE. **Anais...** [S.l.: s.n.], 1999. v.440, p.433.

LEMAIRE, T. et al. Vision-based slam: stereo and monocular approaches. **International Journal of Computer Vision**, [S.l.], v.74, n.3, p.343–364, 2007.

LUCAS, B. D.; KANADE, T. et al. An iterative image registration technique with an application to stereo vision. In: IJCAI. **Anais...** [S.l.: s.n.], 1981. v.81, p.674–679.

MENZE, M.; GEIGER, A. Object scene flow for autonomous vehicles. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. **Proceedings...** [S.l.: s.n.], 2015. p.3061–3070.

NEGRE, P.; BONIN-FONT, F.; OLIVER, G. Stereo graph SLAM for autonomous underwater vehicles. In: INTERNATIONAL CONFERENCE ON INTELLIGENT AUTONOMOUS SYSTEMS (IAS). **Proceedings...** Springer, 2014.

NISTER, D.; NARODITSKY, O.; BERGEN, J. Visual odometry. In: COMPUTER VISION AND PATTERN RECOGNITION, 2004. CVPR 2004. PROCEEDINGS OF THE 2004 IEEE COMPUTER SOCIETY CONFERENCE ON. **Anais...** [S.l.: s.n.], 2004. v.1, p.I-652–I-659 Vol.1.

OLSON, C. F. et al. Robust stereo ego-motion for long distance navigation. In: COMPUTER VISION AND PATTERN RECOGNITION, 2000. PROCEEDINGS. IEEE CONFERENCE ON. **Anais...** [S.l.: s.n.], 2000. v.2, p.453–458.

PI, S. et al. Stereo Visual SLAM System in Underwater Environment. In: OCEANS 2014 - TAIPEI. **Anais...** IEEE, 2014. p.1–5.

PRADEEP, V.; MEDIONI, G.; WEILAND, J. Robot vision for the visually impaired. In: COMPUTER VISION AND PATTERN RECOGNITION WORKSHOPS (CVPRW), 2010 IEEE COMPUTER SOCIETY CONFERENCE ON. **Anais...** [S.l.: s.n.], 2010. p.15–22.

RUSU, R. B.; COUSINS, S. 3d is here: point cloud library (pcl). In: ROBOTICS AND AUTOMATION (ICRA), 2011 IEEE INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2011. p.1–4.

SCARAMUZZA, D.; FRAUNDORFER, F. Visual Odometry [Tutorial]. **IEEE Robotics & Automation Magazine**, [S.l.], v.18, n.4, p.80–92, 2011.

SCARAMUZZA, D.; FRAUNDORFER, F. Tutorial: visual odometry. **IEEE Robotics and Automation Magazine**, [S.l.], v.18, n.4, p.80–92, 2011.

SCHARSTEIN, D.; SZELISKI, R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. **International journal of computer vision**, [S.l.], v.47, n.1-3, p.7–42, 2002.

STEINBRUCKER, F.; STURM, J.; CREMERS, D. Real-time visual odometry from dense RGB-D images. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION WORKSHOPS (ICCV WORKSHOPS), 2011. **Anais...** IEEE, 2011. p.719–722.

STIVANELLO, M. E.; STEMMER, M. R. Development of an Active Stereo Vision System for Robotics Applications. **Induscon**, [S.l.], v.1, 2012.

SZELISKI, R. **Computer vision: algorithms and applications**. [S.l.]: Springer Science & Business Media, 2010.

VARGAS, J. A. D. et al. **Localização e Navegação de Robô Autônomo Através de Odometria e Visão Estereoscópica**. 2012. Dissertação de Mestrado — Universidade Estadual de Campinas.

ZHANG, Z. A Flexible New Technique for Camera Calibration. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, [S.l.], v.22, n.11, p.1330–1334, Nov 2000.